Supporting Information Searching
in
Software Architecture Documents

by

Moon Ting Su

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)
in
Computer Science

The University of Auckland

2013

Abstract

Finding needed content in overwhelming amounts of documentation is a key problem of software documentation. This problem also applies to Software Architecture Documents (ADs). Most existing studies focused mainly on the production aspect of ADs or Architectural Knowledge (AK), to support information finding. Our research complements them by focusing on the consumption of ADs to gain insights to support information searching in these documents.

We proposed the notion of *chunk* to alleviate information searching problem. As a collection of related pieces of architectural information needed for a particular task, a chunk simplifies finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit. We identify chunks by finding “commonality” in the consumers' usages of the architectural information in AD when engaged with certain information-seeking task. The “commonality” serves as possible means to group architectural information into a chunk for the task. If such chunks exist and are made available, they serve as alternative approach to search for information in ADs.

We collected two types of consumers' usage data to identify chunks: annotation data (such as ratings, highlighted content, and so on) and interaction data (which capture users' interaction with the pages of a document and elements on pages). We developed KaitoroCap, an online prototype tool for creating ADs, and for capturing consumers' interaction data (manifested as consumers' exploration paths through the documents), and for collecting annotation data.

We found that chunks based on consumers' usage data (in particular annotation data) of ADs existed. We found that consumers' common preference of a document's sections, in the forms of their ratings of the sections aggregated based on frequency count or average, show potentials in producing chunks of average goodness for the respective task, when the contributing consumers have strong software architecture background and explored the documents in local electronic or printed environment. The goodness of a chunk was determined using criteria which trade-off the recall and precision measures of the chunk, calculated by benchmarking against the oracle set for the task. The goodness of a chunk approximates to its support for information searching in the document for similar information-seeking task.

We found some differences between industry practitioners and academic professionals in terms of chunks found and information needed. In addition to architectural information chunking, we investigated architectural information foraging to study the behaviours of architectural information foragers in terms of common forages, foraging sequences, and foraging styles.

As the conclusion, our research produced some early insights on usage-based chunking of architectural information and architectural information foraging. However, much still need to be done to gain better insights into supporting information searching in ADs based on their actual consumption by consumers, with many exciting challenges-cum-opportunities waiting to be addressed.

---- Blank ----

Dedication

Although unworthy but to HIM, for the gift of life, without which nothing is possible.

---- Blank ----

Acknowledgements

I would like to thank my supervisors, Professor John Hosking and Professor John Grundy for their guidance, support, and help throughout this nearly five years of my study. Without their unfailing support and understanding, I could not have persevered. I am also in debt to Associate Professor Ewan Tempero, who joined the supervisory team from the fourth year of my study. I appreciate all the guidance, support and help that he had provided me during the final stage of my study.

I want to acknowledge the sponsors which made the completion of my PhD possible: University of Malaya; Ministry of Higher Education, Malaysia; PReSS, University of Auckland; and FRST Software Process and Product Improvement project.

I thank Antony Tang (Swinburne University of Technology), David Maplesden and Thiam Kian Chiew for their constructive feedback on my work. I am grateful to those who had helped me in one way or another in recruiting the participants for my studies. I also thank all the participants for their valuable time and effort. I appreciate my acquaintances from ECSA and WICSA conferences, for stimulating interesting ideas in my work.

I would like to thank my friends, whose friendship and support had beautified and enriched my life in Auckland: Josephine, Ann, Peter, Rose, Yu-Cheng, Habib, Ema, Reyhaneh, Mehwish and Muhammad, other lab-mates and friends at the CS Department, members of John & John's Research Group, Annie, Angela, Helyna, Gulshad, John, Robert, Shixiao, Allison, Aishikin, Megumi, Grace, Ally, Albert Yeap, Sam, Afshin, church members, and prayer group members at North Shore. I also thank Suraya, Sri Devi, Por, Liew, Ling, Ang, Phang, Lee, Ow, and many more from University of Malaya; Wilson; and my old-time friends, Yieng, Lisa, Poh, Mak and Gee.

Last but not least, I thank my family and relatives, for their continuous love, support, and prayers, which had sustained me all this while.

Thank you all, without whom I would not have the privilege to write this acknowledgment!

---- Blank ----

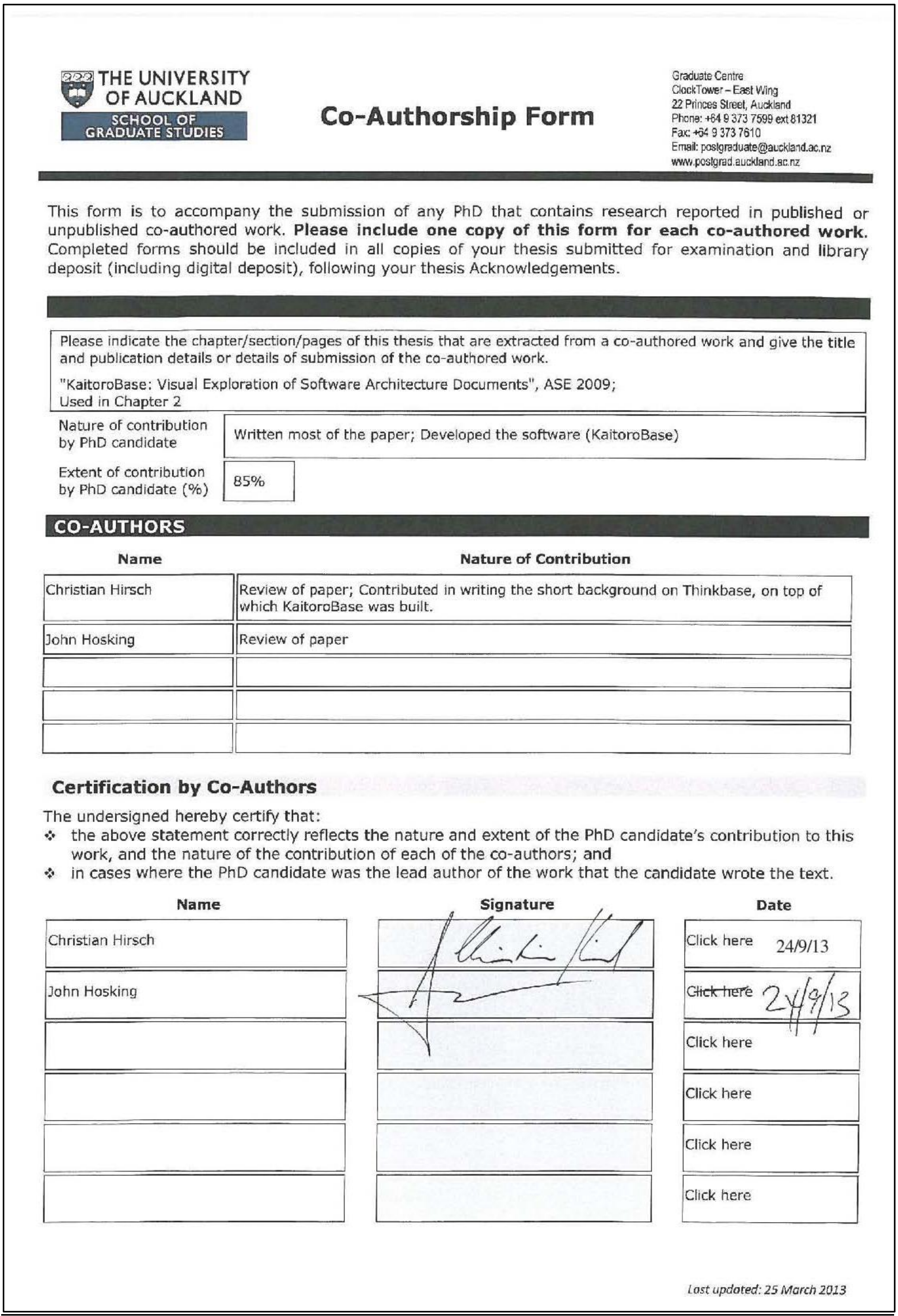
Table of Contents

Lists of Figures

---- Blank ----

Lists of Tables

Co-Authorship Forms



THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains research reported in published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"KaitoroBase: Visual Exploration of Software Architecture Documents", ASE 2009;
Used in Chapter 2

Nature of contribution by PhD candidate	Written most of the paper; Developed the software (KaitoroBase)
Extent of contribution by PhD candidate (%)	85%

CO-AUTHORS

Name	Nature of Contribution
Christian Hirsch	Review of paper; Contributed in writing the short background on Thinkbase, on top of which KaitoroBase was built.
John Hosking	Review of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- in cases where the PhD candidate was the lead author of the work that the candidate wrote the text.

Name	Signature	Date
Christian Hirsch		Click here 24/9/13
John Hosking		~~Click here~~ 24/9/13
		Click here
		Click here
		Click here
		Click here

Last updated: 25 March 2013

---- Blank ----

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains research reported in published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"Capturing architecture documentation navigation trails for content chunking and sharing", WICSA 2011;
Used for Chapter 4

Nature of contribution by PhD candidate: Written most of the paper; Developed the software

Extent of contribution by PhD candidate (%): 90%

CO-AUTHORS

Name	Nature of Contribution
John Hosking	Review of paper
John Grundy	Review of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- in cases where the PhD candidate was the lead author of the work that the candidate wrote the text.

Name	Signature	Date
John Hosking		~~Click here~~ 3/5/13
John Grundy		Click here 29/4/2013
		Click here
		Click here
		Click here
		Click here

---- Blank ----

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains research reported in published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"KaitoroCap : a document navigation capture and visualisation tool", WICSA 2011;
Used for Chapter 4

Nature of contribution by PhD candidate	Written most of the paper; Developed the software
Extent of contribution by PhD candidate (%)	90%

CO-AUTHORS

Name	Nature of Contribution
John Hosking	Review of paper
John Grundy	Review of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- in cases where the PhD candidate was the lead author of the work that the candidate wrote the text.

Name	Signature	Date
John Hosking		~~Click here~~ 3/5/13
John Grundy		Click here 29/4/2013
		Click here
		Click here
		Click here
		Click here

Last updated: 25 March 2013

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains research reported in published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"A Study of Architectural Information Foraging in Software Architecture Documents ", WICSA 2012;
Used for Chapter 16

Nature of contribution by PhD candidate	Written most of the paper
Extent of contribution by PhD candidate (%)	85%

CO-AUTHORS

Name	Nature of Contribution
Ewan Tempero	Review of paper
John Hosking	Review of paper
John Grundy	Review of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- in cases where the PhD candidate was the lead author of the work that the candidate wrote the text.

Name	Signature	Date
Ewan Tempero		Click here 1/5/2013
John Hosking		~~Click here~~ 3/5/13
John Grundy		Click here 30/4/2013
		Click here
		Click here
		Click here

Last updated: 25 March 2013

---- Blank ----

1 Introduction

This chapter sets the background for this research. It explains the problem of finding relevant information in Software Architecture Documents (ADs) and our proposal to alleviate this by chunking the architectural information in the documents based on consumers' usage data. The chapter continues to explain the key research questions, which focus on the chunking of architectural information in ADs. The chapter gives an overview of the methodology adopted to carry out this research. It concludes by outlining the contributions of this research, and the organisation of the remaining chapters of this thesis.

1.1 Motivation

1.1.1 Software Architecture Documents (ADs) and Architectural Knowledge (AK)

A Software Architecture Document (AD) is the tangible product of software architecture documentation which is the activity of writing down a software architecture (Clements et al., 2003). It can be formal or informal (Clements et al., 2003), may or may not contain models (Clements et al., 2003), and can be recorded using any format or media (ISO/IEC/IEEE, 2011).

A Software Architecture Document is also known as an Architectural Description (Greefhorst, Koning, & Vliet, 2006). Architectural Description is defined as "collection of products to document an architecture" in International Standard ISO/IEC 42010:2007 on recommended practice for architectural description of software-intensive systems (ISO/IEC, 2007). Rozanski and Woods took this definition further to specify a good Architectural Description, which is "it is a set of products that documents an architecture in a way its stakeholders can understand and demonstrates that the architecture has met their concerns" (Rozanski & Woods, 2005). In ISO/IEC/IEEE 42010:2011, the latest revision of ISO/IEC 42010:2007, Architecture Description is defined as "work product used to express an architecture" (ISO/IEC/IEEE, 2011).

The importance of ADs cannot be underestimated. As Kruchten wrote in his foreword to the first edition of "Documenting software architectures: views and beyond", architectural elements and architectural decisions not captured in an AD are regarded as if they did not exist (Clements et al., 2003). This is because most architectural constructs are abstract and do not exist directly at the programming level (Bass & Kazman, 1999). Furthermore documentation is regarded as the medium of design and no design decisions are considered made until their incorporation into the documents (Parnas & Clements, 1986). In addition, 'documenting the architecture is the crowning step to crafting it' (Bass, Clements, & Kazman, 2003), where the documenting step is tightly-woven into the architecting process (Bass & Kazman, 1999) (Parnas & Clements, 1986).

"A well-documented architecture facilitates a shorter development interval". A well-documented architecture also reduces the needs for architects to be personally present during numerous meetings of the development process (Smolander & Paivarinta, 2002b). It is also perceived that the person who produces proper documents would have performed the analysis needed to ensure the high quality of the product (Parnas, 2005).

The importance of ADs is further accentuated by the increasing emphasis by the software architecture (SA) community on the sharing of Architectural Knowledge (AK) (Lago, Avgeriou, Capilla, & Kruchten, 2008). AK consists of architecture design and the design decisions (Kruchten, Lago, & van Vliet, 2006). AK existed mainly as documented knowledge (R. C. de Boer & van Vliet, 2011) in the form of an Architecture Description or Architecture Documentation (Jansen, Avgeriou, & van der Ven, 2009). As a result, the absence of architecture documentation causes the loss of AK.

1.1.2 The Problem of Finding Architectural Information

Finding useful content in overwhelming amounts of documentation is not easy. It is a key problem of software documentation aside from the perennial problems of often out of date (but sometimes still useful), poorly written and untrustworthy documents that have a high creation cost (Lethbridge, Singer, & Forward, 2003). "Finding useful content in documentation can be so challenging that people might not try to do so" (Lethbridge et al., 2003), thus rendering the painstakingly-written documents useless.

This difficulty of finding the needed information also applies to ADs (Koning & van Vliet, 2006). This is further affirmed by the findings of a recent survey focused specifically on SA documentation from the perspective of developers (Rost, Naab, Lima, & von Flach Chavez, 2013). Three of the five main findings of the survey are related to the problem of finding the needed information in SA documentation (Rost et al., 2013): 1) The inability of "one-size-fits-all" architecture documentation to provide the right information for specific stakeholders and their tasks. Despite the warning that architecture documentation should not be a "one-size-fits-all" case (Bass et al., 2003), the survey shows that cases like this are still common; 2) Lack of sufficient navigation support to easily find the right information; 3) Scattered documentation across different artefacts making it difficult for finding relevant information. The other two main findings are architecture documentation is often not up to date (a problem that still persists after a decade long from the survey done by Lethbridge et al.) and lacks utility, and inconsistent.

In terms of AK, an AD holds many benefits for AK sharing but as documentation increases with the size and complexity of the system, there are many challenges awaiting the current SA documentation approaches (Jansen et al., 2009). One of these challenges is locating the relevant AK (Jansen et al., 2009), (Avgeriou, Kruchten, Lago, Grisham, & Perry, 2007) either across multiple documents or within these documents (Jansen et al., 2009). This challenge is exemplified by the findings of a case study that software product auditors expressed their dire needs for a 'reading guide' to help them in finding relevant AK in documents (R. C. de Boer, 2006).

It is also important that the access to and the delivery of the right AK be made available to the right person in a Just-in-Time manner (Farenhorst & Van Vliet, 2009). This requires effective, lightweight AK search, including the ability to find relevant information in documents (Farenhorst & Van Vliet, 2009). However, knowledge retrieval features in existing AK management tools are typically simple and reactive (Tang, Avgeriou, Jansen, Capilla, & Babar, 2009). Current search tools use free text (unstructured; difficult to identify relevancy) and first-order predicate logic (limitation in data scalability). There was also a suggestion of deriving ontologies to encode AK so that more accurate search queries can be performed and knowledge bases from different organisations can be combined (Avgeriou et al., 2007).

It is undeniable that despite the importance of ADs and the wealth of architectural information or knowledge that they contain, the documents might not be used because of the difficulty of finding the needed information from them.

1.1.3 Different Needs of AD Stakeholders

The problem of finding the needed information from ADs is further aggravated by stakeholders' partial interest in the content of the documents (Koning & van Vliet, 2006). Many stakeholders' concerns are addressed by a small fraction (sometimes less than 25%) of the AD. Consequently, the readers of ADs complain of too much information.

The partial interest of the stakeholders of an AD can be attributed to the fact that different stakeholders of the document have different roles that require them to use it for different purposes (Bass et al., 2003) (Smolander & Paivarinta, 2002b). In terms of AK, different AK stakeholders also requires different types of AK (Avgeriou et al., 2007). We outline some of the studies that shed some light on these in Section "2.1.1 The Uses of Architecture Documentation and the Needs of Architectural Stakeholders" and Section 2.2.1 "The Uses of AK and the Needs of AK Stakeholders" in Chapter 2.

1.1.4 Forager of Architectural Information

In addition to the limitations of ADs and of the searching capabilities built into them, as well as the partial interest of AD stakeholders, the process of finding information from ADs can also be affected by the behaviour of the information seekers. Humans are *informavores* (Miller, 1983), and so try to maximize the value of knowledge gained per unit cost of interaction (Pirolli, 2007). This is similar to the way animals forage in patchy environment for food and try to maximize the rate of energy gained per effort expended (Stephens & Krebs, 1987). This maximization tendency of the forager for architectural information could be due to the limited resources (in terms of time and budget) the forager can spend on finding the information. Given unlimited resources, the stakeholders of an AD can eventually find the architectural information that they need from the document even if it is poorly written or organised, unless if it was not documented at all or was ambiguously documented making it hard for the stakeholders to discern the information that they need.

1.1.5 The Chunking of Architectural Information

The terms *chunking* and *chunk* appeared in a number of research areas. In areas such as the study of chess, perception, cognition, learning and structured writing, *chunking* generally refers to the grouping of related items into a single unit or *chunk*. Refer to Section "2.4.1 Chunking and Chunk" in Chapter 2 for further details. We adopt similar notions for these terms in our research: ***chunking*** refers to the ***grouping of related pieces of information*** and a ***chunk*** is ***a collection of related pieces of architectural information***. We posit that chunks simplify finding of information, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

In the field of SA, there is no general consensus on the chunks of architectural information Software Architecture Documents or Architectural Descriptions should comprise (Greefhorst et al., 2006). Existing works can be grouped into three main categories:

1) Chunking supported by architecture documentation constructs such as architecture framework, view, view packet, and templates.
   These constructs provide guidance on grouping of architectural information. Architecture frameworks such as Zachman's Framework (Zachman, 1987), The Open Group's Architecture Framework (TOGAF) (The Open Group, 2013) among others, provide guidance on what the chunks should be (Greefhorst et al., 2006); A view is a representation of a coherent set of architectural elements and the relations among them (Bass et al., 2003); View packets organise view information in digestible chunks (Clements et al., 2003); Documentation templates such as interface template (Bass et al., 2003) and architecture decision template (Tyree & Akerman, 2005) assist the documentation of interface and decision by providing guidance on what should be documented for interface and decision and the organisation of their constituents. Using templates such as these in documentation put pieces of information which are related together, by following the standardised groupings suggested by the templates. In addition, the use of perspective (Rozanski & Woods, 2005) on top of viewpoint helps to organise or group certain types of information (i.e. how a quality property is achieved) across views (de Graaf, Tang, Liang, & Van Vliet, 2012).

2) Chunking supported by searching facilities.
   A search using the searching facilities of the documentation environment returns pieces of information which are related in certain ways. In keyword-based searching, items retrieved are related because they contain the same or similar terms as the searched terms. In query-initiated discovery of semantic structure of documents or texts based on words used in the documents or texts (R. C. de Boer & van Vliet, 2008) (R. C. de Boer, 2006), the documents or the units of texts retrieved are related because of their semantic structures. In the retrieval of architectural information chained by underlying models (such as ontologies, meta-models, and domain models), architectural elements or knowledge instances retrieved are related because of the pre-defined relations in the underlying meta-models.

3) Chunking supported by automatic generation of stakeholder-specific ADs.
   Sections or knowledge instances in the dedicated documents are related either in terms of the semantic information in the sections' profiles (Nicoletti, Diaz-Pace, & Schiaffino, 2012) (Diaz-Pace, Nicoletti, Schiaffino, Villavicencio, & Sanchez, 2013), or meta-models used to capture the knowledge (Eloranta, Hylli, Vepsalainen, & Koskimies, 2012) (Rost, 2012).

The onus of identifying chunks has always been on the producers of ADs instead of their consumers. The producers apply the architecture documentation constructs, or AD and knowledge meta-models, in grouping architectural information. The role of the consumers in this aspect is mostly confined to the choice of the terms they provided to the searching facilities of the documentation environment which returns a set of results (or chunk) based on the terms the consumers supplied.

Most of the studies except for the study by Nicoletti et al. (Nicoletti et al., 2012) (Diaz-Pace et al., 2013) have ignored the role of the consumers in chunking of architectural information. In particular, to the best of our knowledge there is no previous study on chunking based on the actual consumption of ADs, to support information finding. Our work is novel as we investigate if chunk(s) of architectural information exist based on consumers' usage of ADs. Insight from this can be used in AD production to enable a closer match between authors' intentions and readers' expectations, which is an important factor in determining the effectiveness of the documentation (R. C. de Boer & van Vliet, 2009).

Our basic idea in identifying chunk is by finding "commonality" in the consumers' usages of the architectural information in the ADs when engaged with certain information-seeking task. The "commonality" serves as possible means to group architectural information into a chunk for the task. If such chunks exist and are made available, they serve as alternative approach in searching for information in ADs. We posit that chunks simplify finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

To help us in finding chunks using consumers' usage data, we refine our concept of *chunk*. We define a chunk as:

A chunk for an information-seeking task performed on a software architecture document is a collection of related pieces of architectural information needed for the task by the majority of a group of users. Specifically it comprises section(s) of a software architecture document needed for the task by the majority of a group of users.

A section can be either paragraph(s) of text, table(s), image(s) or hyperlink(s) or combinations of these. For example a chunk might consist of the following sections of an AD: section 'Logical Components' which provides textual explanation on the logical components and Section "Logical Components Diagram" which contains an image of this and both of these sections contain information required by the majority of a group for a particular information-seeking task.

To start with, we experiment with 'document section' as the level of granularity for chunk elements. This approach is similar to existing work that studied the relevance of the elements of ADs to perceived stakeholders and their concerns (Koning & van Vliet, 2006).

1.2 Research Questions

The purpose of this research is to study the chunking (grouping) of architectural information in Software Architecture Documents (ADs) based on how the consumers use ADs, to support finding relevant information for information-seeking tasks. The above overarching purpose is broken down into three main research questions (RQs):

RQ1: Do chunks exist?

If yes,

> **RQ2: Do chunks contain information (or sections of document) compulsory for the information-seeking task?**
>
> **RQ3: Do chunks contain information (or sections of document) not needed for the information-seeking task?**

The first and foremost question is whether chunks based on consumers' usage data exist. If yes, we assess the chunks to see if they contain information (or sections of document) compulsory for the particular information-seeking task, and if the chunks contain information (or sections of document) not needed for the task. This sheds some light on whether chunks found will be useful for the task.

We collected two types of consumers' usage data for the identification of chunks. The first type is interaction data, represented as consumers' exploration (or navigation) paths through document. We use the terms 'navigation' and 'exploration' interchangeably in this thesis. The second type of usage data is annotation data. This includes ratings, tags, comments, specifications of from which sections answer was found, and the presence of highlighted information in sections.

1.3 Research Methodology

Figure 1.1 shows the research methodology we employed in this research. It shows the steps taken and the studies conducted.

We conducted an extensive literature review on SA documentation and issues related to the understanding of SA of software systems. This led to the identification of a research gap in the area – how to support the location of needed information in ADs. We interviewed practicing software architects to gain industry insight to strengthen our literature discovery in these areas. Subsequently, we identified the key research questions of this research. These questions focus on the chunking of architectural information in ADs based on how their consumers used them, to support finding relevant information in the documents.

This research puts forward the argument that the architectural information in an AD needs to be structured into or presented as chunks, to support finding relevant information in these documents. As collections of related pieces of architectural information, chunks enable related architectural information, which otherwise may be dispersed in an AD, to be retrieved collectively as a unit thus simplifying search by other users. The ability of finding the right architectural information is the first step to the (re-)use as well as the sharing of the architectural information or knowledge within the same software project or across similar projects.

Our main approach to identify chunks was by analysing the data of consumers' usage of software architecture documents when they performed certain information-seeking tasks. We looked for "commonality" in the usage data. The "commonality" serves as possible means to group architectural information into a chunk for the task.

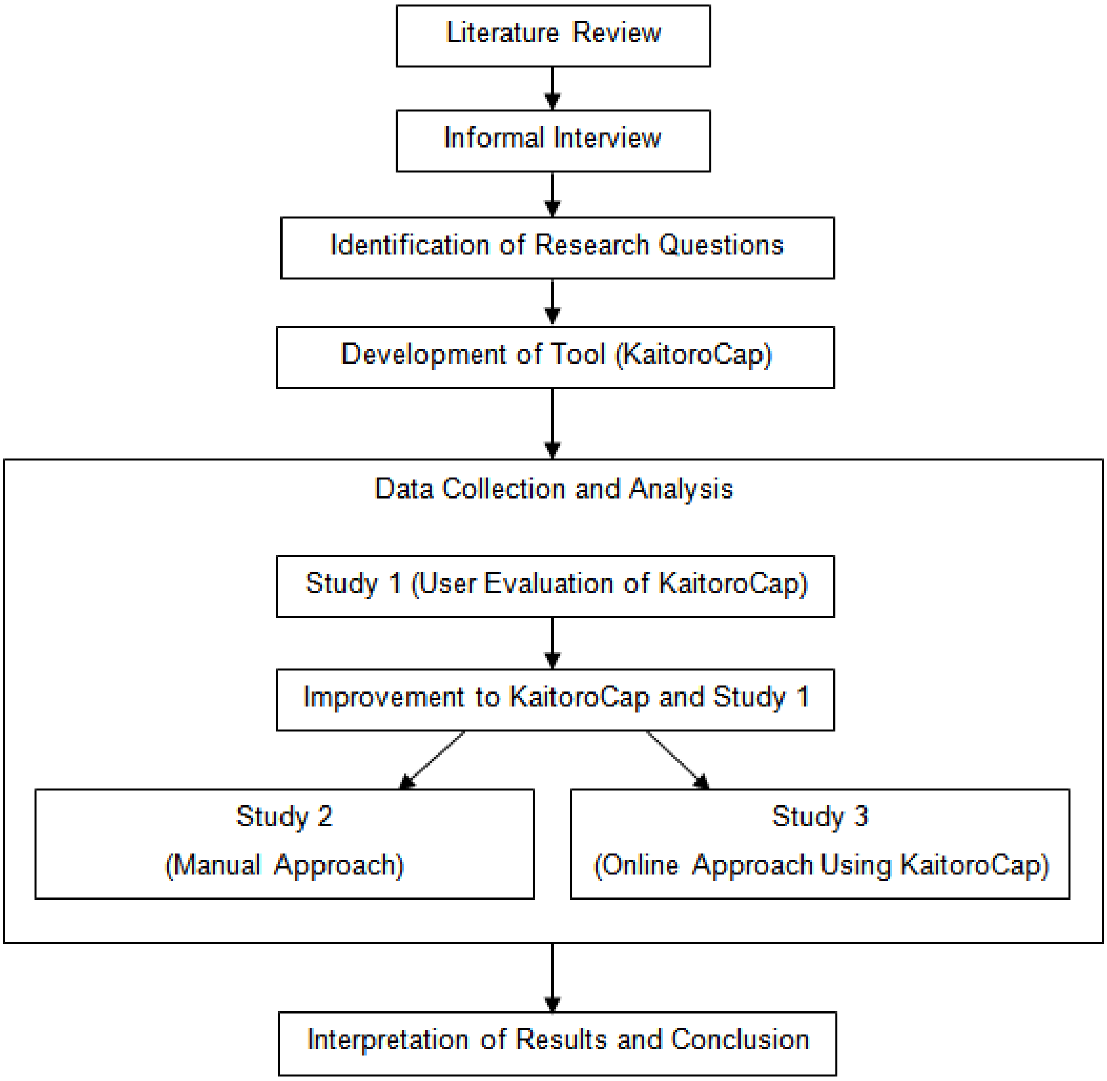

Literature Review
Informal Interview
Identification of Research Questions
Development of Tool (KaitoroCap)
Data Collection and Analysis
Study 1 (User Evaluation of KaitoroCap)
Improvement to KaitoroCap and Study 1
Study 2
(Manual Approach)
Study 3
(Online Approach Using KaitoroCap)
Interpretation of Results and Conclusion

Figure 1.1: Research Methodology

We conducted three studies. Study 1 was a user evaluation study conducted to gain feedback on the features of KaitoroCap. KaitoroCap is the online prototype tool we developed for creating software architecture documents as wiki pages in Atlassian Confluence Enterprise Wiki (Atlassian, 2013a), for capturing and visualising consumers' exploration paths through the documents, and for collecting annotation data provided by the consumers. Feedback from Study 1 was used to improve KaitoroCap. The improved version of KaitoroCap was used in Study 3 (Online Approach Study) to collect usage data. We also conducted Study 2

(Manual Approach Study) to collect usage data (in particular annotation data) manually without the use of KaitoroCap,

We analysed the usage data collected in Studies 2 and 3 quantitatively to identify chunks. In doing that, we also investigated the difference between industry practitioners and academic professionals in terms of chunks found and information needed. We also investigated the issues of architectural information foraging by analysing some usage data from Study 2 quantitatively and qualitatively. We interpreted the results from the studies with reference to the research questions. We presented our key findings, main contributions and possible future work. Further details on our research methodology can be found in Chapter 3.

1.4 Research Contributions

The main contributions of this research are:

a. Novelty of chunking based on usage of ADs

We proposed the notion of *chunk* to alleviate information searching problem. We identified chunks by finding “commonality” in the consumers’ usages of the architectural information in the ADs when engaged with certain information-seeking task. The idea of finding chunks using the “commonality” in the usage data of a group of consumers performing similar task, raises the sense of ‘community of users’ in the otherwise lonely quest of information finding. This brings a number of other research areas into supporting information finding in architecture documentation. This ranges from computational wear, social navigation to collaborative filtering.

We captured two types of usage data: interaction data (manifested as exploration paths in KaitoroCap) and annotation data (ratings, tags, comments, and so on). Exploration paths serve as usage ‘wear’ left by previous consumers. Visible usage data (exploration paths, ratings, comments and tags) serves as information traces supporting some forms of social navigation. Chunks serve as collaborated filters for information needed for specific tasks.

Exploration paths support the transfer of procedural knowledge (Nickols, 2000). This special type of tacit knowledge (i.e. the knowledge on how people explore existing AK in ADs) had often been ignored by the SA community. Chunks serve as guides for exploring architectural information in ADs when performing certain information-seeking tasks. In addition, these usage-based chunks provide alternative suggestions to the documents producer on how architectural information in ADs could be chunked or grouped for easier finding of related pieces of information.

b. Development of KaitoroCap

We developed KaitoroCap, a fully-functional prototype tool that supports document creation and dynamic re-structuring; annotation; exploration path capturing, visualisation and searching. KaitoroCap can be used for all types of documents, and not limited to ADs.

c. Reusability of research instruments

The instruments developed in our research (such as studies designs, data collection instruments, template for the construction of oracle set, and KaitoroCap) can be adapted to study usage-based chunking of information in other types of documentations for examples requirement documents, user manuals, API documentations, among others.

Some of the works of this research had been published in international conferences:

1. Su, M. T., C. Hirsch and J. Hosking (2009). KaitoroBase: Visual Exploration of Software Architecture Documents. Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering, IEEE Computer Society: 657-659.
2. Su, M. T. (2010). Capturing exploration to improve software architecture documentation. Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. Copenhagen, Denmark, ACM: 17-21.
3. Su, M. T., J. Hosking and J. Grundy (2011). Capturing Architecture Documentation Navigation Trails for Content Chunking and Sharing. Proceedings of the 2011 Ninth Working IEEE/IFIP Conference on Software Architecture, IEEE Computer Society: 256-259.
4. Su, M. T., J. Hosking and J. Grundy (2011). KaitoroCap: A Document Navigation Capture and Visualisation Tool. Proceedings of the 2011 Ninth Working IEEE/IFIP Conference on Software Architecture, IEEE Computer Society: 359-362.
5. Su, M. T., E. Tempero, J. Hosking and J. Grundy (2012). A Study of Architectural Information Foraging in Software Architecture Documents. Proceedings of the 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, IEEE Computer Society: 141-150.

1.5 Thesis Organisation

The rest of the chapters of this thesis are organised as below:

Chapter 2: Literature Review

This chapter presents the literature related to our research: documentation of SA, AK management, different needs of stakeholders, the problem of finding architectural information, chunking and chunk, and leveraging usage data. The background reviews on Information Retrieval and Information Foraging are provided in Chapters 5 and 14 respectively.

Chapter 3: Research Methodology

This chapter describes the methodology we have adopted to carry out this research. It describes the steps taken and the set of studies conducted; the different kinds of research methods used for each study; and the techniques and rationale for these choices.

Chapter 4: KaitoroCap and User Evaluation Study (Study 1)

This chapter presents the design, implementation and usage examples of KaitoroCap; the user evaluation study conducted to gain feedback on KaitoroCap's features; the improvements made to KaitoroCap; and a comparison of KaitoroCap with other AKM tools reviewed in Chapter 2.

Chapter 5: Study 2 - Exploration of Software Architecture Documents (Manual Approach Study)

This chapter explains the way we designed and carried out Study 2. It also lays down the foundation for usage-based chunking of architectural information in our research. Chapter 5 is a very important chapter for Studies 2 and 3, and should be read prior to reading the subsequent chapters.

Chapter 6 – 11: Chunking of Architectural Information

This set of chapters present the chunking of architectural information for the six information-seeking tasks in Study 2. Chapter 5 should be read prior to reading these chapters.

Chapter 12: Study 3 - Exploration of Software Architecture Documents (Online Approach Study)

This chapter describes Study 3, the 'online' version of Study 2 (Manual Approach Study). Chapter 12 contains two main parts: 1) Study 3 itself and 2) Chunking of architectural information for the six information-seeking tasks in Study 3. Chapter 5 should be read prior to reading Chapter 12.

Chapter 13: Summary and Discussion of Chunking Results

This chapter presents the overall results of our investigation into the chunking of architectural information. It presents the answering of our research questions, and all relevant results. This chapter also presents our own assessment of our approach, the threats to validity of our findings in all three studies, comparison with the related work in the literature, and the advantages and limitations of our approach.

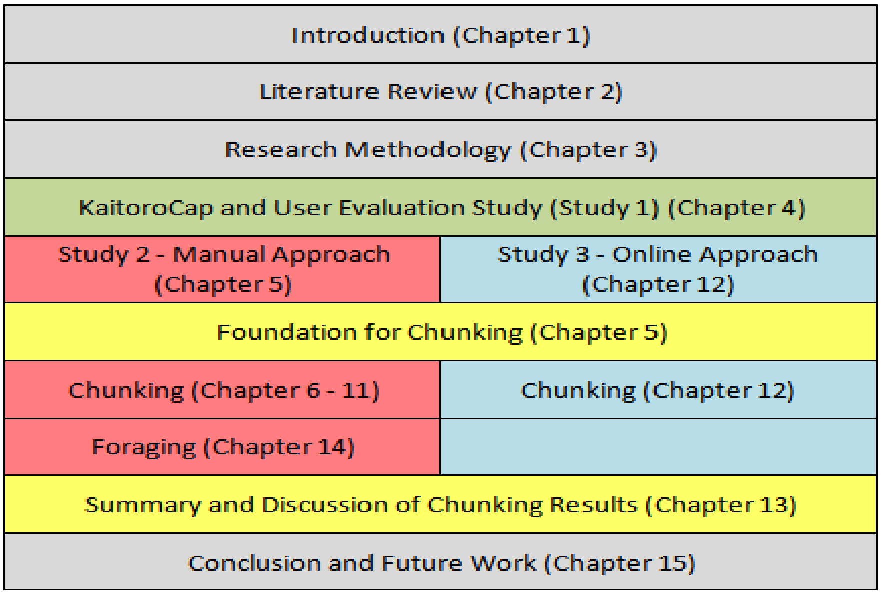
Chapter 14: A Study of Architectural Information Foraging

This chapter presents our investigation into - forages, foraging sequence and styles - related to architectural information foraging in ADs.

Chapter 15: Conclusion and Future Work

This chapter concludes the thesis by summarising our research, highlighting the key findings, main contributions, and possible future work.

Table 1.1 shows the mapping of the thesis content to the respective chapters. The content can be organised into five groups: KaitoroCap (green cells), Study 2 (red cells), Study 3 (blue cells), both Studies 2 and 3 (yellow cells), and whole research (grey cells).



Introduction (Chapter 1)	
Literature Review (Chapter 2)	
Research Methodology (Chapter 3)	
KaitoroCap and User Evaluation Study (Study 1) (Chapter 4)	
Study 2 - Manual Approach (Chapter 5)	Study 3 - Online Approach (Chapter 12)
Foundation for Chunking (Chapter 5)	
Chunking (Chapter 6 - 11)	Chunking (Chapter 12)
Foraging (Chapter 14)	
Summary and Discussion of Chunking Results (Chapter 13)	
Conclusion and Future Work (Chapter 15)	

Table 1.1: Thesis Content Mapping

---- Blank ----

2 Literature Review

This chapter presents the literature related to our research. This includes the background information and related works in the following areas: documentation of Software Architecture (SA), Architectural Knowledge (AK) management, different needs of stakeholders, the problem of finding architectural information, chunking and chunk, and leveraging usage data. The background reviews on Information Retrieval and Information Foraging are provided in Chapters 5 and 14 respectively.

2.1 Documentation of Software Architecture (SA)

The architecture of a system comprises “what is essential about that system considered with reference to its environment” (ISO/IEC/IEEE, 2011). The architecture of a system is abstract (ISO/IEC/IEEE, 2011) and needs to be described or expressed or documented, for stakeholders to understand it.

The explanation on what a Software Architecture Document (AD) is, and the importance of documenting SA can be found in Section “1.1.1 Software Architecture Documents (ADs) and Architectural Knowledge (AK)” in Chapter 1.

The SA community’s recognition of the importance of documenting SA can be further seen especially through the establishment of the International Standard on Architecture (or Architectural) Description IEEE Std 1471-2000 (IEEE., 2000), and its two revisions namely ISO/IEC 42010:2007 (ISO/IEC, 2007) and ISO/IEC/IEEE 42010:2011 (ISO/IEC/IEEE, 2011). This set of standards proposes the uses of Architecture Descriptions to address the creation, analysis and sustainment of architectures of systems. To support that, the standard defines a conceptual model of Architecture Description and specifies the contents of an Architecture Description. The latest revision of the standard also specifies the desired properties of architecture viewpoints, architecture frameworks and Architecture Description Languages (ADLs) (ISO/IEC/IEEE, 2011), all of which are important for the creation of Architecture Descriptions.

2.1.1 The Uses of Architecture Documentation and the Needs of Architectural Stakeholders

There are many uses of architecture documentation by a variety of stakeholders throughout a system’s life cycle (ISO/IEC/IEEE, 2011) (Clements et al., 2003), For some stakeholders, architecture documentation prescribes what should be true by placing constraints on decision to be made (Bass et al., 2003) (Clements et al., 2003). For other stakeholders, architecture documentation describes what is true by recounting design decisions already made (Bass et al., 2003) (Clements et al., 2003).

A number of works present the uses of and need for architecture documentation, and which stakeholders have an interest in it. Some of the works are: stakeholders of software system and the uses of architectural descriptions (IEEE., 2000; ISO/IEC, 2007) (ISO/IEC/IEEE, 2011); key stakeholder roles and their relationship to architecture, emerging problems and reasons to produce or use architectural descriptions (Smolander & Paivarinta, 2002b); architectural stakeholders and their communication needs (Bass et al., 2003), (Clements et al., 2003).

A discussion on architecture documentation is incomplete without addressing what is SA of system. The perception of what constitutes SA of system largely determines what would be documented and how the ‘what’ should be documented. Both aspects affect the uses of architecture documentation by architectural stakeholders. The needs of architectural stakeholders in turn determine the ‘what’ and ‘how’ of documentation.

Software architecture (SA) is a subtle subject which warrants many definitions in the literature, with a long list of these being maintained by the Software Engineering Institute (Software Engineering Institute, 2012). Instead of delving into these definitions, we focus on general perceptions of practitioners on SA and architecture descriptions by highlighting the works of Smolander and his colleagues. Their studies on the practice of architecture design and description using grounded theory showed that SA and architecture description mean differently to different stakeholders (Smolander, 2002) (Smolander, Rossi, & Purao, 2005). Four metaphors for architecture were discovered: blueprint, literature, language and decision (Smolander, 2002). The metaphors overlap and most of the stakeholders use all of them but with different emphasis. The blueprint metaphor views architecture as residing in the working implementation of the system, and architecture description as high-level implementation of the system and it prescribes the future system. In the literature metaphor, architecture is seen as the solutions made in the past and its description as documentation of the past for future readers. This metaphor is related to the documentation of technical structures and to capturing, managing and sharing of AK overtime (for examples reference architecture, architecture framework and product line architecture). The language metaphor sees architecture as providing the common understanding of the high-level structures of the system between different stakeholders at present, and architecture description serves as the communication between different stakeholders. The decision metaphor views architecture as the decisions about the structure of the system, and architecture description serves as rational decision-making, prescribing the future solution.

The studies reviewed in this section are not exhaustive but enough to show the existence of different needs of architectural stakeholders, which drive their different expectations and uses of architecture documentation.

2.2 Architecture Knowledge (AK)

Architecting a software system is a knowledge-intensive process, involving the consumption and production of large and rich knowledge (Lago et al., 2008) (Dingsøyr & Vliet, 2009). To avoid the problem of knowledge vaporization (Bosch, 2004), the knowledge needs to be represented and managed. The realization of this motivated a significant number of recent works in Architecture (or Architectural) Knowledge (AK).

AK consists of architecture design and the design decisions (Kruchten et al., 2006) (Dingsøyr & Vliet, 2009). The different types of AK are shown in Figure 2.1 (Farenhorst & de Boer, 2009): tacit versus explicit, and application-generic versus application-specific (Lago & Avgeriou, 2006). Tacit knowledge is built up from experience and expertise, whereas explicit AK is externalised in artefacts such as Architecture Descriptions. Application-generic knowledge refers to AK applicable to a number of applications irrespective of their domains. Examples are existing architectural patterns, tactics, and reference architectures, among others. Application-specific knowledge refers to design decisions and architecture designs related to just one particular system.

The combinations of the four types of AK resultes in four main categories of AK (Farenhorst & de Boer, 2009): 1) Application-generic tacit AK (design knowledge gained from experience); 2) Application-specific tacit AK (contextual domain knowledge affecting architectural solution, such as business goals and stakeholder concerns); 3) Application-generic explicit knowledge (design knowledge codified in discussion, books, standards and so on, such as patterns, tactics, reference architecture); 4) Application-specific explicit AK (all explicit knowledge of a system, such as architectural views and models, codified design decisions and rationale). The last category is the most tangible type of AK.

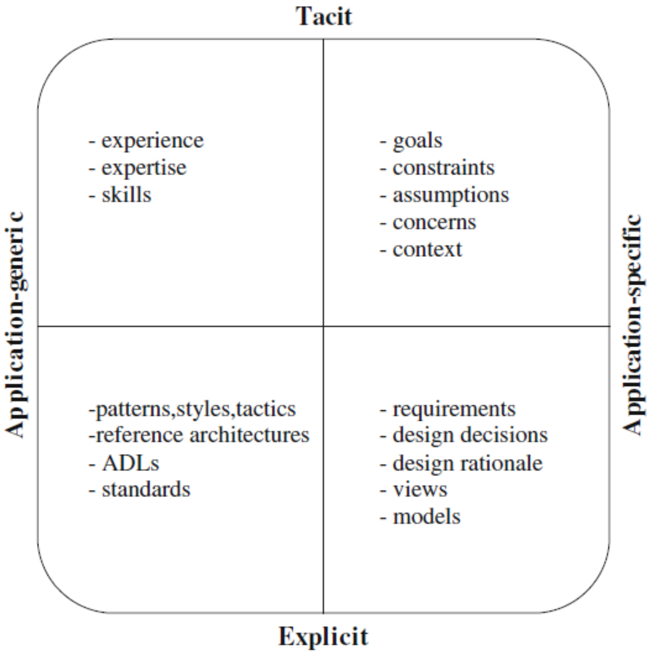


Figure 2.1: AK Categories
(Farenhorst & de Boer, 2009)

We would like to argue that application-specific tacit AK in top-right quadrant of the AK Categories is not tacit knowledge because it can be made explicit if wanted to. Many in the field of AK (Dingsøyr & Vliet, 2009) and Knowledge Management in general (Nickols, 2000) see tacit knowledge as defined by Polanyi who coined the term, which is tacit knowledge cannot be articulated (Polanyi, 1997). A better classification of application-specific tacit AK would be application-specific implicit knowledge. Implicit knowledge refers to knowledge that can be articulated but has not been attempted (Nickols, 2000). Implicit knowledge can be extracted from a competent performer by a task analyst, knowledge engineer or someone skilful in this, to be made explicit knowledge.

2.2.1 The Uses of AK and the Needs of AK Stakeholders

Different AK stakeholders require different types of AK (Avgeriou et al., 2007). Some of the studies that shed light on this are: stakeholders having AK-related concerns (Avgeriou et al., 2007); use-case model that describes the actors (roles) and use cases (what the roles do) of AK repository (Kruchten et al., 2006); The AK needed by software architects and how to support the needs (Farenhorst, Izaks, Lago, & van Vliet, 2008)

(Farenhorst & Van Vliet, 2009). It was found that software architects especially needed 'Just-In-Time (JIT)' architectural knowledge (Farenhorst et al., 2008) (Farenhorst & Van Vliet, 2009); Use case model for architectural decisions which defines the requirements for an architectural decisions management system envisioned to be a knowledge grid (van der Ven, Jansen, Avgeriou, & Hammer, 2006). Knowledge Grid is "an intelligent and sustainable interconnection environment that enables people and machines to effectively capture, publish, share and manage knowledge resources" (Hai, 2004).

The studies reviewed in this section are not exhaustive but enough to make the point that, different AK stakeholders have different needs and uses of AK.

2.2.2 Architecture Documentation and AK

Recent development in architecture documentation and AK has witnessed some convergence of these two areas. At the core, architecture documentation of a system seeks to capture architecture decisions and rationales alongside architecture design (ISO/IEC/IEEE, 2011), which are all recognised as essential components of AK of a system (Kruchten et al., 2006). However, along the dimension of tacit and explicit knowledge, documentation of SA refers only to the explicit aspect of AK (Kruchten, 2009).

Explicit knowledge is differentiated into documented knowledge or formal knowledge (Liang & Avgeriou, 2009) (Jansen et al., 2009), based on the level of formality of its representation regardless of whether the knowledge is application-generic or application-specific. Documented knowledge is expressed in natural language or images in documents. Formal knowledge is codified using a formal language or model with defined semantics (for example ADL, domain models, ontologies and so on). In other words, documented knowledge is unstructured whereas formal knowledge is structured (R. C. de Boer, 2011).

AK existed mainly as documented knowledge (R. C. de Boer, 2011) in the form of an Architecture Description or Architecture Documentation (Jansen et al., 2009). As a result, the absence of architecture documentation causes the loss of AK. The possible consequence of the loss of AK is the misunderstanding of the system by different stakeholders (Buchgeher & Weinreich, 2008). Consequently, it is not surprising that architectural description was proposed as a central element in software architecting (IEEE., 2000) (ISO/IEC, 2007) (ISO/IEC/IEEE, 2011), or even in the whole development process (Buchgeher & Weinreich, 2008).

2.2.3 Architecture Knowledge Management (AKM)

Architecture Knowledge Management (AKM) is defined as "an approach to improving the SA process outcomes by introducing various processes and practices for identifying, capturing AK and expertise and making it available for transfer and reuse across projects in an organisation" (Babar, 2009). To support AKM, strategies to manage knowledge; and suggestions to implement the strategies from the 'schools' for knowledge management (Earl, 2001) (Dingsøyr & Vliet, 2009) (Dingsøyr, 2009) were adopted from the field of knowledge management.

The knowledge management strategies are explained next, followed by two models which look at AKM from the production and consumption aspects of AK. These serve as background for the discussion of our work in subsequent chapters.

2.2.3.1 Knowledge Management Strategies

The following knowledge management strategies were explored for AKM (Babar, 2009): codification which centres on computer and personalisation which centres on people (Hansen, Nohria, & Tierney, 1999). In codification strategy, knowledge is extracted from the people who have it, carefully codified and stored in databases for easy access and use by company's staff (Hansen et al., 1999). Contrarily, personalisation strategy focuses on the identification of the person "who knows" and communication of knowledge through direct person-to-person contacts (Hansen et al., 1999). In personalisation strategy, computers mainly assist people to communicate knowledge and not store it (Hansen et al., 1999). Some researchers recommended hybrid strategy to manage knowledge in global software development efforts (Desouza, Awazu, & Baloh, 2006) and in AKM (Farenhorst et al., 2008) (Farenhorst & Van Vliet, 2009). Hybrid strategy leverages codification and personalisation strategies, which complement each other. The use of codification strategy is commonly reported by published works from research and practise, however many companies adopted personalisation strategy unintentionally (Babar, de Boer, Dingsoyr, & Farenhorst, 2007). Refer to Section "2.2.4 AKM Tools" for the strategy supported by a number of current AKM tools.

2.2.3.2 Models for AKM

Tang et al. proposed a producer-consumer model to manage AK within the life-cycle of software architecting process (Tang et al., 2009). AK is mainly created during the first three stages (i.e. analysis, synthesis, evaluation), and used during the two later stages of the process (i.e. implementation and maintenance). To understand what types of AK may be created or used (or both) at each stage of the process, AK was grouped into four general categories: 1) context (related to problem space); 2) general (existing designing knowledge such as architecture styles, patterns, and tactics); 3) reasoning (reasoning information about a design such as design decisions, rationale, design alternatives, and trade-offs); and 4) design knowledge (designs of a system such as components and architectural models).

The model also shows AK activities of producer and consumer of AK (Figure 2.2). The AK activities were mapped onto an evaluation framework to create a set of criteria to compare AKM tools. AK activities of a producer include: architect the solutions thereby creating reasoning knowledge, integrate context knowledge into AK, share AK, create necessary traces between the four categories of AK, synthesise design knowledge, distil design as general knowledge, and apply general knowledge. AK activities of consumers include: learn AK, search or retrieve AK, evaluate reasoning and design knowledge, and trace between the different categories of AK. Most stakeholders frequently switch roles from a producer to a consumer of AK and vice versa during the architecting process (Tang et al., 2009). For example, during the architectural synthesis stage, an architect produces reasoning knowledge and at the same time uses general knowledge.

Liang and Avgeriou proposed a use-case model (Figure 2.3) for AKM Tools (Liang & Avgeriou, 2009). Five actors were defined: users and four specialised users (namely architects, reviewers, requirement engineers and maintainers). Twenty-four use cases were identified and grouped into two main categories of Consuming and Producing AK. The other two categories are Knowledge Management which is related to low-level functionality to manage AK data and Intelligent Support which is related to automating more complex AKM tasks.

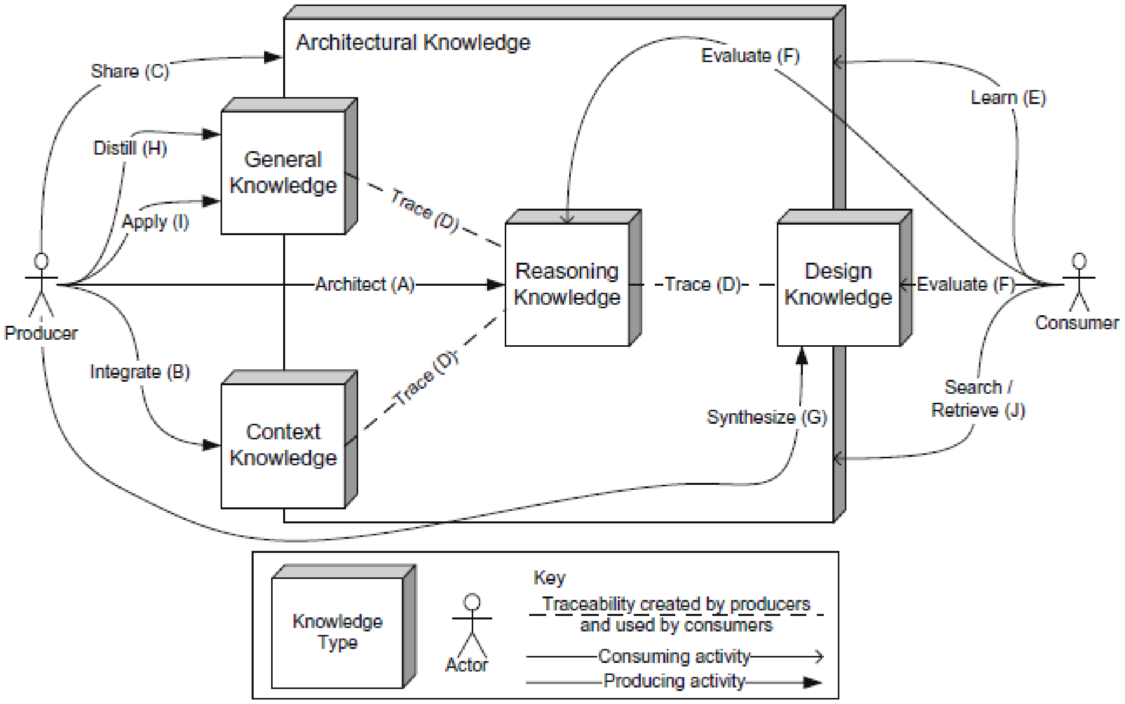


Figure 2.2: AK Activities (Tang et al., 2009)

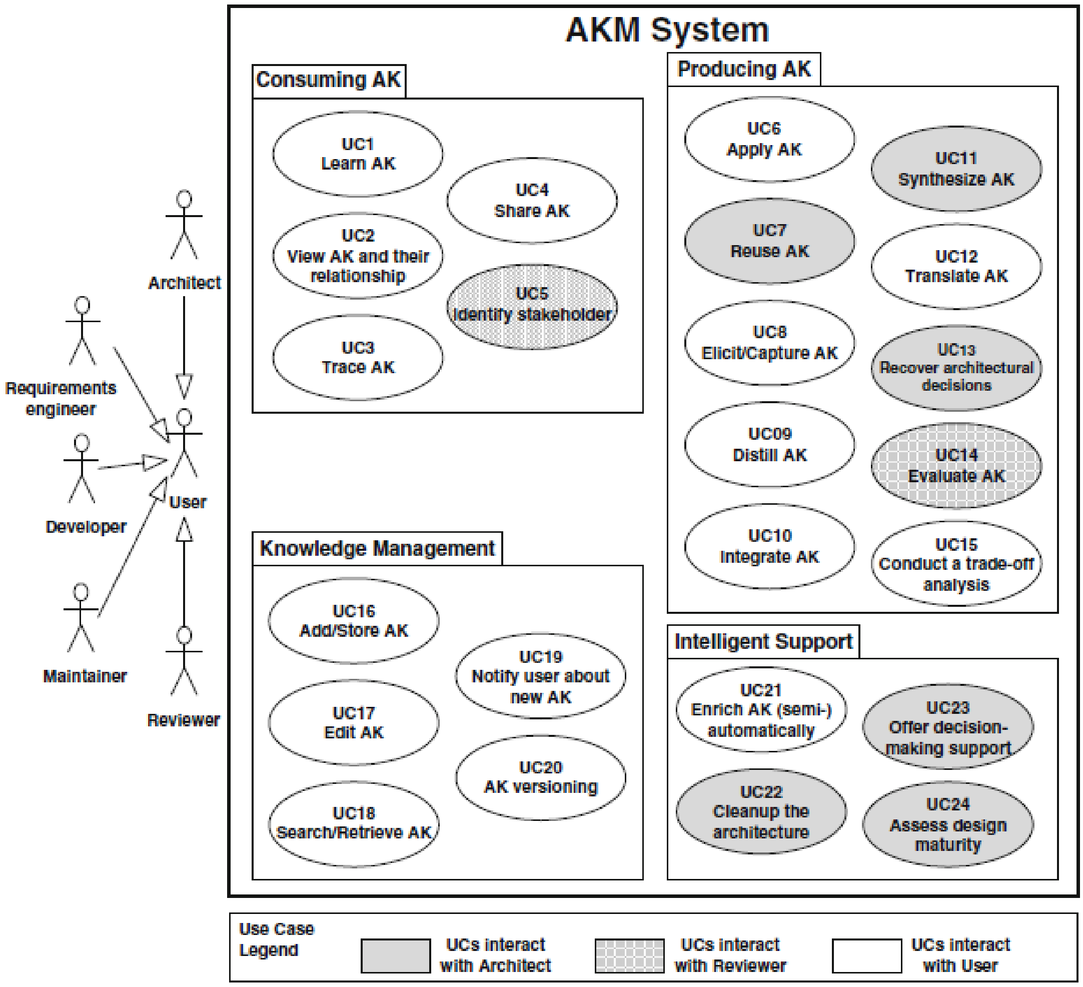


Figure 2.3: AKM Use-Case Model (Liang & Avgeriou, 2009)

Some of the use-cases (integrate, share, synthesise, learn, evaluate, search or retrieve, distil and apply) were derived from the producer-consumer model explained earlier (Tang et al., 2009). Nevertheless some of these use-cases were classified under a different category (for example share, evaluate and search or retrieve use-cases).

2.2.4 AKM Tools

A number of tools had been built for AK management. Examples are ARCHIUM (Jansen, Van der Ven, Avgeriou, & Hammer, 2007) (Jansen & Bosch, 2005), ADDSS (Capilla, Nava, Pérez, & Dueñas, 2006), PAKME (Babar et al., 2007; Babar & Gorton, 2007; Babar, Gorton, & Jeffery, 2005), Expertise Site (Farenhorst, Lago, & van Vliet, 2007a), Knowledge Repository (Farenhorst et al., 2007a) / Best Practice Repository (Farenhorst et al., 2008), Knowledge Maps system (Farenhorst et al., 2007a), AK Sharing Platform (Farenhorst et al., 2007a) / JIT AK Sharing Portal (Farenhorst et al., 2008) /EAGLE (Farenhorst, Lago, & van Vliet, 2007b). Many of these tools support codification strategy (ARCHIUM, ADDSS, Expertise Site, Knowledge Repository / Best Practice Repository), EAGLE and PAKME support hybrid strategy, whereas Knowledge Maps System supports personalisation strategy.

Some of the above tools and other tools such as AREL (Tang, Jin, & Han, 2007), Knowledge Architect (Jansen et al., 2009), SEURAT (Burge & Brown, 2008), and a general wiki used for documenting SA (Bachmann & Merson, 2005) and a semantic wiki (ADkwik) used for documenting formal AK (Schuster, Zimmermann, & Pautasso, 2007) (Zimmermann, Kopp, & Pappe, 2009), were reviewed in other works (Tang, Avgeriou et al. 2009, Farenhorst, Lago et al. 2007), and (Liang & Avgeriou, 2009). All the additional tools (AREL, Knowledge Architect, SEURAT and the two wikis) support codification strategy (Liang & Avgeriou, 2009).

In terms of the type of knowledge supported (Liang & Avgeriou, 2009), the first wiki (Bachmann & Merson, 2005) also known as SEI-ADWiki in (Liang & Avgeriou, 2009) supports documented AK, ADkwik and ADDSS support both documented and formal AK, and ARCHIUM, AREL, Knowledge Architect, SEURAT, EAGLE and PAKME support formal AK.

2.3 The Problem of Finding Architectural Information

This section describes existing approaches that support the finding of architectural information or AK in architecture documentation and their limitations. The description of the problem of finding architectural information or AK is provided in Section “1.1.2 The Problem of Finding Architectural Information” of Chapter 1.

2.3.1 Existing Approaches and Their Limitations

Clements’ Views & Beyond approach suggests the use of documentation roadmap, and view template to help new stakeholders to find the information that they need in an AD (Bass et al., 2003) (Clements et al., 2003). Documentation roadmaps introduce readers to the organisation of an AD. As a standard organisation for a view, a view template helps readers to quickly find sections of interest. Views & Beyond approach also suggests a view catalogue (containing introductory information about all the documented views such as the name of a view and its style, view's element and relation types and properties, description of a view's purpose, and so on) for the same reason. However, there is no report of benefit based on actual uses of each of the suggestions.

Concept maps were used to show the main concepts of ADs to improve their readability (Koning & van Vliet, 2006). The responses to the idea of concept maps were favourable but their benefits based on the actual usage were not reported.

Some studies have investigated ways to support finding relevant AK in ADs. Latent Semantic Analysis (LSA) was used to construct reading suggestions to guide software product auditors through documentation to find required AK (R. C. de Boer & van Vliet, 2008) (R. C. de Boer, 2006) (R. C. de Boer & van Vliet, 2007a). LSA uses implicit higher-order (or latent) structure in the association of terms with documents ("semantic structure") to detect relevant documents based on searched terms in queries (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). The semantic structure of the information in a set of documents discovered using LSA was further analysed to construct a reading guide which provides guidance on where to start reading, which documents to consult for more details on a topic and progressing through the documentation (R. C. de Boer, 2006). Though the results show that the reading guide is sensible to the auditors, the approach is dependent on the selection of the initial terms to explore the documentation. Choosing different terms would likely produce a different set of documents. In addition, significant human interpretation is needed to select suggested documents to read and to determine sequence to read (R. C. de Boer & van Vliet, 2008).

Some studies made use of formal models to capture AK as a prelude to supporting finding of AK. These include: 1) deriving ontologies to encode AK so that more accurate search queries can be performed and knowledge bases from different organisations can be combined (Avgeriou et al., 2007); 2) using semantic wiki for documentation (Su, Hirsch, & Hosking, 2009) (R. C. de Boer, 2011); and 3) augmenting ADs with AK through semantic annotations based on underlying meta-models (Jansen et al., 2009) (Tang, Liang, & van Vliet) (de Graaf et al., 2012). We elaborate on some of these studies below.

We developed KaitoroBase (Su et al., 2009), a SA documentation tool built within the Thinkbase Visual Wiki (Hirsch et al., 2009) (Thinkbase, 2009) powered by Freebase semantic wiki (Freebase), as part of this research. The meta-model for AD is based on the Attribute-Driven Design (ADD) method (Bass et al., 2003). KaitoroBase provides graph-based interactive visualisation of the structure of the entire AD, allows for exploratory search, non-linear navigation, and at the same time connects to low-level details of the document. The dynamic expansion and collapsing of each element of the document and graphically-grouped elements of the same type, reduce the cognitive load on the user. One limitation is the automatic adjustment of the positions of some neighbouring nodes during the expansion and collapsing of nodes, might cause some users to inspect again some parts of the graph which had already been explored.

Jansen et al. looked into enriching ADs by annotating them with formal AK based on domain models (Jansen et al., 2009) to support AK activities which includes locating of AK. The main idea was to extract AK from an AD and store this knowledge in a formal model in a knowledge repository. A tool suite named Knowledge Architect (Jansen et al., 2009) (Liang, Jansen, & Avgeriou, 2009) (Liang, Jansen, & Avgeriou, 2010) was developed to support this approach. The tool suite comprises: 1) Document Knowledge Client, a Microsoft Word 2003 plug-in to annotate Word documents with formal AK; 2) Excel and Python Plug-ins for quantitative architectural analysis; 3) Knowledge Explorer to search, inspect, and trace AK; 4) Knowledge Translator to translate formal AK based on one domain model to AK based on another; and 5) Knowledge Repository which holds all AK and provides storing and retrieving interfaces for various tools in the suite.

Tang et al. proposed ontology-based documentation to tackle the difficulty in retrieving knowledge in file-based documents (Tang et al.) (de Graaf et al., 2012). The approach focused on indexing software documents (including ADs) with a software engineering ontology (Tang et al.) (de Graaf et al., 2012). The basic idea was a user could select some text from a document and associate the text with a concept from the ontology. A knowledge instance of the concept would be created and indexed in the knowledge base. A user could also create link between indexed knowledge instances which are related, to create knowledge triple. A prototype system ArchiMind (de Graaf et al., 2012) based on OntoWiki semantic wiki (AKSW, 2013) was developed to support knowledge indexing and retrieval from documents written in wiki pages. ClioPatria (VU University of Amsterdam, 2010), a semantic search and reasoning tool, was used for advanced knowledge retrieval. Experiment results showed that ontology-based approach was better than file-based approach in AK retrieval. Both the enriching (Jansen et al., 2009) and indexing (Tang et al.) (de Graaf et al., 2012) ADs approaches described above, transform documented knowledge into formal knowledge based on underlying domain models or ontologies.

Approaches that use formal models to capture AK assist knowledge retrieval by enabling automated reasoning and querying. Querying retrieves network of knowledge instances, made possible by the pre-defined relationships between concepts in the meta-model, domain model or ontology. The retrieved knowledge could then be inspected by following the trails of related knowledge (Tang et al.). The knowledge retrieval advantages of model-based approaches come at the cost of rigidity (R. C. de Boer, 2011). The producers need to follow the underlying meta-models and be consistent in labelling the knowledge instances (de Graaf et al., 2012). In addition, the learning curve is steeper and there is less support for unstructured or semi-structured knowledge (R. C. de Boer, 2011).

Recently, a number of studies focused on automatically generating stakeholder-specific ADs (Nicoletti et al., 2012) (Diaz-Pace et al., 2013) (Eloranta et al., 2012) (Rost, 2012). These studies are described next.

Nicoletti et al. worked on personalising the content of ADs to suit the information needs of stakeholders based on their profiles (Nicoletti et al., 2012) (Diaz-Pace et al., 2013). In particular, stakeholders are linked to sections of an AD based on their profiles. The study used a user profiling tool that applies text mining techniques to build stakeholders' profiles and the profiles for each section of an AD. Stakeholders' profiles are bootstrapped with stakeholders' general characteristics proposed by V & B approach (Clements et al., 2010). The profiles are then updated with topics (or concepts) of interest mined from the textual comments provided by the stakeholders regarding the AD. A sections' profile is constructed from concepts mined from the content of the section. Sections of a document are predefined by templates for AD, from the V & B approach (Clements et al., 2010). Similarity measures between a stakeholder's profile and each section's profile are calculated and ranked. A certain percentage of the top-ranked sections are chosen as relevant for the particular stakeholder. These sections are recommended to the documenter, who then decides which documentation tasks to prioritise for the next revision of the document. One disadvantage of the approach of Nicoletti et al. is the lack of flexibility since the producers need to conform to the predefined templates of AD.

TopDocs (Eloranta et al., 2012) dynamically generates topical ADs using information in an Architectural Knowledge Base (AKB). The information in AKB is organised based on a meta-model. A topical document is an information package tailored to a specific task or concern at hand. TopDocs allows stakeholder-specific documents to be produced for several stakeholder groups: quality-centred document for architect, component-centred document for developer and scenario-centred document for maintainer. The approach involves looking at the meta-model from the viewpoint of a particular stakeholder, identifying elements in the meta-model that serve as key elements that identify a major concern for the stakeholder, and identifying all elements connected to the key elements to be included in the topical document. Based on the concern-key-contents mapping, script can be written to traverse AKB to retrieve all relevant items that might satisfy the stakeholder's needs. TopDocs can also generate non-parameterized topical documents for specific purposes (for example for architectural evaluation using ATAM method), and also generate the comprehensive AD.

Rost et al. proposed to generate a task-specific architecture documentation for each individual developer, automatically from the general documentation (Rost, 2012). To do that, a software architect needs to create a specification of a developer's task. The task specification includes the architectural elements a developer is responsible for. Using the task specification, the documentation generator produces the architecture documentation based on pre-defined identification model and representation model. Identification model contains rules that specify which architectural elements can be classified as relevant or irrelevant. The relevant ones will be included in the generated documentation. Representation Model specifies how the relevant elements will be represented in the generated documentation. This study focused on architecture documentation in the form of architecture models. The study is still at an early stage. Nevertheless, one of the main challenges of the approach will be the construction of the identification model.

The work on generating topical documents (Eloranta et al., 2012) and task-specific documents (Rost, 2012) are fundamentally model-based. Model-based approaches as we mentioned earlier suffer in terms of (R. C. de Boer, 2011): rigidity, steeper learning curve and less support for unstructured or semi-structured knowledge.

2.4 Chunks to Support Finding of Information

To support finding relevant information in an AD, we argue that architectural information in it needs to be structured into or presented as *chunks*. Chunks as mentioned in Chapter 1 are basically *collections of related pieces of architectural information*. We posit that chunks simplify finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit. In this section, we review the concepts of *chunking* and *chunk* by drawing upon other research areas.

2.4.1 Chunking and Chunk

The terms *chunking* and *chunk* appeared in a number of research areas. In areas such as the study of chess, perception, cognition, learning and structured writing, *chunking* generally refers to the grouping of related items into a single unit or *chunk*. We adopt similar notions for these terms in our research. Recall from Chapter 1 that in our research, *chunking* refers to the *grouping of related pieces of information* and a *chunk* is basically *a collection of related pieces of architectural information*. In the following, we elaborate on the notions of these terms in the aforementioned research areas.

Miller, a cognitive psychologist observed that our short-term (or working) memory has limited holding capacity of “seven plus-or-minus two” items (Miller, 1956). The limited amount of information (i.e. between five to nine items) that we can hold in our working memory hampers our information processing and recall abilities. However, the capacity of the short term mental workspace can be increased through a *chunking* process, where items with similar or related attributes are bound conceptually to form a single unit or *chunk* (Curtis, 1984) (Miller, 1956). Since Miller’s work in 1956, works in cognitive science have established chunking as one of the key mechanisms of human cognition (Gobet et al., 2001).

In the field of human learning, a *chunk* is defined as “meaningful unit of information built from smaller pieces of information”, and *chunking* is “the process of creating a new chunk” (Gobet & Lane, 2012). These notions of the two terms are also applicable to the study of expertise, acquisition of language and education, all of which are related to learning. Chunking can be goal-oriented, involving a deliberate conscious process (Gobet & Lane, 2012). An example is Miller’s re-coding of the information (Gobet & Lane, 2012) as fewer chunks with more bits per chunk (Miller, 1956). For example, the 9-digit binary number 111001110 can be re-coded as a 3-digit decimal number 716, which is easier to process and remember. Another type of chunking is perceptual chunking which is more of an automatic and continuous process that occurs during perception (Gobet & Lane, 2012). Perceptual chunking was used to explain the ability of chess experts to recall briefly presented positions with high precision.

In structured writing (Horn, 1997), chunking refers to grouping of information into manageable units, called information blocks and information maps. An information block is the basic unit of a subject matter and an information map is a collection of information blocks. The benefits of information maps are twofold: helping the writer in the organisation of large amounts of information and helping the reader in understanding the structure of the subject matter and the document. The use of information maps can be found in many written communication where structured writing had been adopted. These include instructional materials, documentation, reports, memos, proposals, online and hypertext applications.

The existing approaches in chunking of architectural information in the field of SA are provided in Section “1.1.5 The Chunking of Architectural Information” in Chapter 1. The next section reviews existing works which make use of consumers’ usage data, from other research areas.

2.5 Leveraging Usage Data

This section presents background concepts and related works involved in using data of previous consumers’ usage of some artefacts (such as documents, source code and so on) to assist information searching by other consumers. These include computational wear, social navigation and information filtering. There are two types of usage data: interaction data and annotation data.

2.5.1 Computational Wear

In the domain of document processing, computational wear (Hill, Hollan, Wroblewski, & McCandless, 1992) refers to recording users’ interactions with computational objects (such as documents, menus, emails, and so on), and making the interactions history available as parts of the objects, thereby creating history-enriched objects (Hill & Hollan, 1994). Some examples of computational wear are edit wear and read wear. Edit wear involves capturing of authors, and read wear involves capturing of readers interactions histories with computational objects and making that available on the objects. Wears (usage marks) provides additional information (for example most unstable portions of a document where editing are frequent and most read portions of a document) that helps in sense-making of the materials.

One example of an implementation of computational wear is: Mapping of wears onto document scroll bars where editing and reading occurs, in Zmacs editor (Hill et al., 1992). Of particular interest to us is the read wear, which graphically displays a document's readership history based on read time accumulated for each visible line.

2.5.2 Social Navigation

Social navigation is one type of information navigation mechanism (Dourish & Chalmers, 1994), where decisions of movement from one item to another in the information space are informed by the behaviour of other people (Dieberger, Dourish, Höök, Resnick, & Wexelblat, 2000). The behaviour can be visible such as the movement of users from one online chat room to another, or aggregated and hidden in the interaction history of an object. Examples of social navigation are navigation towards a cluster of people, and selecting items because others have examined them (Dourish & Chalmers, 1994).

Ideas in social navigation often originate from the physical world (Dieberger et al., 2000). For example the inclination to go to a crowded restaurant as this might suggests good food, or the tendency to choose to read a worn and dog-eared book as these signs suggest many people have read it. Corresponding example of social navigation in an online world is joining a chat room based on the current active users and discussion topic, or the tendency to follow information traces (Dieberger et al., 2000) left by previous users. The traces such as comments, reviews, ratings and so on, can be used as short-cuts to information needed. Following information traces is much like following the "footprints in the snow" (Munro, Hook, & Benyon, 1999). Users leave "footprints" (Munro et al., 1999), known as computational wear (Hill et al., 1992) that others can use to find their ways through the same information space more quickly.

Key properties of social navigation are dynamism and personalisation (Dieberger et al., 2000). Social navigation traces are not pre-planned but grown dynamically. Therefore, social navigation reflects what people actually do than what designers expect people should be doing. In addition, the navigational advice is personalised to the user. Other researchers regard the visibility of the aggregated behaviour of the community as an important aspect of social navigation systems (Schafer, Frankowski, Herlocker, & Sen, 2007).

Some examples of social navigation systems or tools are (Dieberger et al., 2000): Amazon.com; EPOL for online food shopping (Svensson, Höök, Laaksolahti, & Waern, 2001); CoWEB (Dieberger & Guzdial, 2003) annotates links on web pages with access and edits information of the hyperlinked pages; IBM's WBI (Web Intermediaries) toolkit (Maglio & Barrett, 2000); Footprints (Wexelblat & Maes, 1999); Edit wear and read wear in Zmacs editor (Hill et al., 1992), among others.

In the following, we provide further descriptions on IBM's WBI and Footprints as these tools also record navigation paths. This is similar to our KaitoroCap.

IBM's WBI is a development kit and server software to add intermediaries to the Web (Maglio & Barrett, 2000). Intermediaries in this context are programs that transform information either through customisation, filtering, annotation, aggregation, transcoding or caching. A number of intermediaries that personalise information were developed as plug-ins for WBI. Of interest here are: Personal History plug-in which records the sequence of pages visited by a user together with the content of each page (Maglio & Barrett, 2000). The personal history of the user can then be inspected by using 1) keyword queries to retrieve pages this user accessed that contain the searched keywords, or 2) path browsing to see paths taken through a page; Short Cuts plug-in creates short-cuts to recurring paths by adding links to pages a user regularly visits within some radius from the current page.

Footprints is built based on physical-world navigation metaphor to study history-rich navigation in information spaces (Wexelblat & Maes, 1999). This work was inspired by Hill and colleagues' work on computational wear (Hill et al., 1992). Footprints consists of a set of tools which provides different visualisations of the interaction history information: 1) Map - showing traffic through pages in a web site in the form of hyperbolic geometry where nodes represent pages visited and links represent transitions between pages. Popularity of documents is encoded in shades of red; 2) Paths - showing paths taken by other users that are relevant to the current page, in hierarchical top-down view. Common starting points are merged to create forks in the path. Thickness of line which represents a path, increases if the path is more frequently used; 3) Annotations - the percentage of users that have followed the link are inserted next to each link on a web page; and 4) Signposts - the path view allows users to provide comments on pages and paths. The user study conducted showed that the use of Footprints enabled users to finish the same work with significantly less effort.

2.5.3 Information Filtering

Information filtering refers to a range of processes involved in the delivery of information to people who need it (Belkin & Croft, 1992). On an abstract level, there is little difference between information filtering and information retrieval (Loeb & Terry, 1992) (Belkin & Croft, 1992). Both have the same goal of getting information to people who need it. However, differences existed (Belkin & Croft, 1992).

The common characteristics of the process of information filtering are (Belkin & Croft, 1992): it involves unstructured, and semi-structured (such as email messages with well-defined header fields but unstructured text body) data; deals mainly with textual information (which is unstructured data), and media data (such as images, voice, video); involves huge amounts of data; usually involves streams of incoming data; filtering is based on user profiles (i.e. descriptions of individual or group information preference) which typically represent users long-term interests or periodic goals or desires; often implies removal of data from rather than finding data in an incoming stream.

The next three sub-sections explain three different types of information filtering. We are particularly interested in collaborative filtering but also explain the other two to show their differences in order to avoid confusion.

2.5.3.1 Content-based Filtering

In content-based filtering, objective features of items are extracted and composed into items profiles (Schafer et al., 2007). Then a user profile comprising the features of the items the user has rated is constructed. The user profile is compared to item profiles to find items of possible interests to the user. For example if a user gave a high rating to a web page with the words “software architecture”, he or she will be presented with other web pages containing the same words.

Content-based filtering such as keyword-based filtering and latent semantic indexing (Deerwester et al., 1990) have the following limitations (Shardanand & Maes, 1995): 1) Items must be in the format that can be parsed by machine, but automatic analysis of media such as images, voice, video and so on is difficult. Alternatively, attributes should have been manually assigned to the items. However, manual analysis is not practical; 2) Suggests only what a user has liked before (i.e. no serendipitous finds); 3) Cannot filter based on assessment of more abstract attributes such as quality. For example cannot differentiate between well-written and badly-written text if both used the same terms or words (Schafer et al., 2007).

2.5.3.2 Collaborative Filtering

The limitations of content-based filtering can be addressed by collaborative filtering (Goldberg, Nichols, Oki, & Terry, 1992), which is also known as social information filtering (Shardanand & Maes, 1995). However, some regards content-based filtering and collaborative filtering as complementary (Schafer et al., 2007). Collaborative filtering is “the process of filtering or evaluating items using the opinions of other people” (Schafer et al., 2007). In collaborative filtering, people collaborate to help one another to perform filtering of information by recording their reactions (in the form of annotations) to documents they read (Goldberg et al., 1992). The annotations can be used by others’ *filters* to sieve the information to receive. For example, as in the case of Tapestry system in supporting moderated newsgroups (Goldberg et al., 1992), a user can filter for articles annotated by, or replied to by a particular user.

Collaborative filtering systems typically use users’ ratings to filter information. The ratings can be scalar, ordinal, binary or unary (Schafer et al., 2007). The ratings can be collected through explicit means (where a user is asked to give opinions) or implicit means, or both (Schafer et al., 2007). Implicit means refers to inferring from a user’s actions. For example a user who visited a product page has some interest in the product, but a user who bought the product has a much stronger interest. Other annotations or social feedback such as users’ tags, recommendations, and comments, can also be used to filter information. For example, Tapestry’s users can also perform annotation queries such as finding items that are being tagged as “excellent” by a particular user (Schafer et al., 2007).

2.5.3.3 Automated Collaborative Filtering

Automated collaborative filtering (Schafer et al., 2007) automates “Word of Mouth” recommendations (Shardanand & Maes, 1995). Instead of a user setting the filters, items are recommended to the user based on the social annotations given by other users with similar taste. A user’s interest profile is matched to those of other users to find users of similar taste. New items that users with similar taste liked are suggested to the user. Items-of-interest predicted by collaborative filtering can be positioned prominently to *personalise* content to suit individual user's needs (Schafer et al., 2007).

Examples of automated collaborative filtering systems are: Ringo a personalised music recommender system (Shardanand & Maes, 1995) which was renamed Firefly (Bruckman, 1998) and bought over by Microsoft; GroupLens which filters Usenet news (Konstan et al., 1997); MovieLens which recommends movies (Konstan, Riedl, Borchers, & Herlocker, 1998); Amazon (Lerman, 2007); Netflix (Lerman, 2007), among others.

2.5.4 The Relation Between Social Navigation and Collaborative Filtering

There is a close relation between social navigation and collaborative filtering. Both make use of other people interaction or annotation data. In social navigation this data is used by a user as guides in navigating in an information space. In collaborative filtering this data is used to filter information delivered to a user.

2.5.5 Social Filtering

Previous works on collaborative filtering and social navigation focus on "implicit" collaboration (Schafer et al., 2007). In "implicit" collaboration the users from whom recommendations were based remain anonymous. In "explicit" collaboration the identifying information of the members of a community are more exposed. This enables users to choose people with similar tastes or interests, and then refine the trust he or she has on these users based on other users' opinions of them. The users-rating-users (Schafer et al., 2007) which is related to the notion of reputation (Dieberger et al., 2000), creates explicit social networks which users could navigate to find items-of-interest. This brings to **social filtering** (also known as social recommendation or social information processing) (Lerman, 2007).

Social filtering involves social media sites which include blogs, wikis, media sharing sites such as Flickr, del.icio.us and Digg (Lerman, 2007). The characteristics of social media sites are: users actively creating or contributing to the content, annotating content with tags, evaluating content by voting or using it, creating social networks by making those with similar interests 'contacts' or 'friends'.

Social filtering extends collaborative filtering to include social networks as a mechanism to filter information. Users can designate other users having similar tastes to be their 'contacts' or 'friends', and use them to find and filter information. Social filtering refines social navigation to centre on explicitly-created personal social networks.

Examples of social filtering systems are social news aggregators such as Digg (Lerman, 2007) and Reddit (Reddit Inc., 2013), and photos sharing site Flickr (Yahoo! Inc., 2013).

2.5.6 Wear-based Filtering

Wear-based filtering (DeLine, Khella, Czerwinski, & Robertson, 2005) refers to a combination of computational wear (Hill et al., 1992) and collaborative filtering (Goldberg et al., 1992) (Shardanand & Maes, 1995). It makes use of data collected on previous users' interactions with certain computational objects to filter information presented to future users.

Wear-based filtering was proposed to improve the interface of programming development environment to aid programmers in navigating and understanding unfamiliar source code in order to make changes to it (DeLine, Khella, et al., 2005). This involves tracking programmers' code traversal information by using a logger and using this interaction data to guide other programmers to portions of code that are most related to the current code under inspection. Three conceptual visualisations using wear-based filtering were proposed: 1) Frequently Accessed Next (FAN) List which shows the code elements (i.e. methods and fields) that previous programmers visited the most after visiting the selected code element; 2) Code Favourites, a browser for exploring projects, types, members and classes defined in those projects, with the tree view representation filtered to show only those most frequently-accessed (or "hot spots" in the code) and hiding the rest under "More members" node which is expandable if required; 3) Wear for degree-of-interest highlights, where instead of superimposing interaction data as a heat map on diagrams (such as UML class diagrams) which gives an overview of the system components and their relationships, interaction data can be used to filter the information appearing on such diagrams to address the scalability problem of such diagrams for large systems.

Two of the proposed visualisations were implemented in Team Tracks (DeLine, Czerwinski, & Robertson, 2005): 1) Class View Favourites or Code Favourites (DeLine, Khella, et al., 2005), and 2) Related Items which is similar to Frequently Accessed Next List (DeLine, Khella, et al., 2005) but also includes most frequently-visited element just before the selected element. The user studies conducted showed that navigation data sharing can improve program comprehension and was welcomed by users.

2.5.7 Other Form of Filtering

All the background and related works on leveraging usage data presented prior to this section, focus mainly on using the usage data of other people (be it individual or group) to assist a user in finding the information needed. One exception is content-based filtering. In this section, we describe a different form of filtering, one that makes use of a user's own usage history.

To reduce information overload during development, the information presented by a development environment can be filtered and ranked by using a degree-of-interest (DOI) model constructed based on a programmer's interaction history with software program artefacts (Kersten & Murphy, 2005) (Mik & Gail, 2006). This approach involves constructing a task context using a programmer's interaction history with software program artifacts when working on a task (Mik & Gail, 2006) or tasks (Kersten & Murphy, 2005), and from the structural relationships of the program artifacts. A weighting for each element in the task context is calculated based on the frequency of interactions with the element and how recent the interactions are. The weighting represents the programmer's degree of interest (DOI) in the element when performing a specific task, or his general degree of interest in the element when engaged with multiple tasks. DOI of an element translates to how relevant an element is to the current task. The elements that the programmer interacted with the most will be the elements of highest relevance.

DOI was implemented in Mylar (Mik & Gail, 2006) or Mylyn (The Eclipse Foundation, 2013) for the Eclipse development environment. On the activation of a task, Mylar monitors a user interaction with code, and uses that to capture in a DOI model, the relevance of code elements to the task (Mik & Gail, 2006). The DOI model is used to filter the Eclipse interface to show only essential program elements related to the task-at-hand. The productivity of programmers who used Mylar showed significant improvement.

In later work, DOI was combined with degree-of-authorship (DOA) to form degree-of-knowledge (DOK) (Thomas, Jingwen, Gail, & Emerson, 2010) to find who knows what about the code. A developer's DOK value for each source code element was calculated by combining the developer's authorship and interaction information for that element. The DOK value serves as an indicator of a developer's knowledge of that code element. The overall DOK model for a developer portrays the developer's knowledge of the entire source code. Case studies results show that DOK models can be used for expert finding, knowledge transfer and identifying changes of interest. In comparison to Mylyn, DOI in DOK is based on all the interactions of a single developer with code elements without splitting the interactions according to the task for which the interactions occurred.

Tasktop (Elves, 2010) or Tasktop Dev (Tasktop Technologies Inc., 2013) extends Mylyn's tracking of interactions with code to include interactions with web pages and desktop documents. The DOI model built from these is used to filter Eclipse Interface to include only the code, URLs of web pages and documents that are needed for a task. Tasktop extracts anchors (or hyperlinks) on web pages and presents them as sub-nodes of the web page node in its Navigator view.

2.6 Summary

This chapter reviews the literature related to our research. It provides important background information for the discussion of our work in later chapters (in particular, Section "4.6 Related Work" in Chapter 4 and Section "13.12 Related Work" in Chapter 13). The background reviews on Information Retrieval and Information Foraging are provided in Chapters 5 and 14 respectively.

---- Blank ----

3 Research Methodology

This chapter describes the methodology we have adopted to carry out this research. It describes the steps taken; the three studies conducted; the different kinds of research methods used for each study, the techniques and rationale for these choices; and the relationships between the studies.

3.1 Overview of Research Methodology

The research methodology we employed in this research is summarised in Figure 1.1 of Chapter 1. It shows the steps taken and the studies we conducted in this research. The software engineering research methods we used are mainly quasi-experiment, and survey (Easterbrook, Singer, Storey, & Damian, 2008). The survey technique we used is interviews (Wohlin et al., 2000).

The following sections describe the key steps involved in this research. They also provide an overview of each of the three studies undertaken. The detailed description of each study can be found in its respective chapter: Study 1 in Chapter 4, Study 2 in Chapter 5 and Study 3 in Chapter 12.

3.2 Literature Review

We conducted an extensive literature review on SA documentation and issues related to the understanding of SA of software systems. This led to the identification of a research gap in the area – how to support the location of needed information in ADs.

To support finding relevant information in an AD, architectural information in it needs to be structured into or presented as *chunks*, which (as mentioned in Chapter 1) are basically *collections of related pieces of architectural information*. We posit that chunks simplify finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

The onus of identifying chunks has always been on the producers of ADs instead of their consumers. Most of the studies except for the study by Nicoletti et al. (Nicoletti et al., 2012) (Diaz-Pace et al., 2013) have ignored the role of the consumers in chunking of architectural information. In particular, there is no study on how the problem of finding required information can be addressed by chunking the architectural information based on the consumers' actual usage of ADs.

3.3 Informal Interviews

Following on from this literature review, we conducted semi-structured informal interviews with two software architects from different organisations and domains. We wanted to gain some quick insights into representative industry practices relating to SA documentation and the issues of understanding the SA of complex, real-world software systems. These were the same issues that we reviewed in the literature.

The informality of the interviews allowed us to gain some quick industry insights on the aspects under study. Semi-structured interviews combine both specific and open-ended questions (Siw Elisabeth & Bente, 2005). The former are used to elicit specific information. The latter are used to discover unexpected types of information. The semi-structured nature of the interviews gave us the opportunity to refine our preliminary research ideas as well as to explore other possible ideas for our research with the industry practitioners. We chose an interview over a questionnaire survey as our main purpose was to have more ad-hoc discussion with the practitioners regarding our preliminary research ideas.

We believed the two interviews with two highly experienced software architects from different organisations and domains were adequate to provide us with a quick reality check for our preliminary research ideas formed from and backed up by the literature. The main outcomes of the interviews were the confirmation of the feasibility of the idea of capturing the exploration process involved in understanding SA of a system and making that available for later use and analysis. These interviews also motivated the choice of the Atlassian Confluence Enterprise Wiki (Atlassian, 2013a) environment for the prototype (KaitoroCap) we developed in this research to capture architecture documentation exploration paths.

3.4 Identification of Research Questions

Subsequently, we formalized our research idea into a set of related research questions. The questions focus on the chunking of architectural information in ADs based on how their consumers used them, to support finding relevant information in the documents.

We collected two types of consumers' usage data for the identification of chunks. The first type is interaction data, represented as consumers' exploration (or navigation) paths through document. The second type of usage data is annotation data. This includes ratings, tags, comments, specifications of from which sections answer was found, and the presence of highlighted information in sections.

Using this usage data, we explored a number of ways to find chunks. If chunks exist, we wish to analyse their compositions, to assess whether they contain information (or sections of document) compulsory for the information-seeking task, and whether they contain information (or sections of document) not needed for the task. This would shed some light on whether discovered chunks would be useful for the task.

3.5 Chunking Based on Usage of Software Architecture Documents

In order to answer our research questions we conducted three studies to investigate the chunking of architectural information based on the usage of ADs: Study 1 (User Evaluation Study), Study 2 (Manual Approach Study) and Study 3 (Online Approach Study). Study 1 is more of a user evaluation study of our prototype KaitoroCap (explained in next section).

In each of these studies, we recruited participants with SA background to explore ADs for certain information-seeking tasks and collected their usage data of the documents. Studies 1 and 3 made use of KaitoroCap to capture usage data including the exploration paths. Study 2 did not make use of KaitoroCap. In this case, the actual exploration path could not be captured.

As the three studies involved collecting the actual usage data of ADs, a survey method either in the form of interview or questionnaire (Wohlin et al., 2000) is inappropriate as survey focuses more on investigating retrospective events (Wohlin et al., 2000) (Dag, Tore, & Magne, 2007). A case study investigates the studied phenomenon within its real-life context (Dag et al., 2007). We did not use case study because in order to collect substantial usage data of ADs, we need to define a reasonable scope for the exploratory process involved in the consumers' usage of the documents. This involved selecting the specific ADs to be explored by the consumers and predefining the information-seeking tasks for which usage data would be collected. These requirements made the experiment research method more suitable for the three studies we conducted. Experiment research method deliberately separates the studied phenomenon from its context (Dag et al., 2007) to allow the control (Wohlin et al., 2000) that we needed.

We conducted Studies 2 and 3 as quasi-experiments instead of experiments. A quasi-experiment is a variant of the true experiment research method. (Easterbrook et al., 2008). It can be used where the latter is impossible. For example, when there is difficulty in assigning subjects randomly to the treatments. We tried as much as possible to assign the study participants (i.e. subjects) randomly to the two ADs and the information-seeking tasks (the treatments). However, we could not claim the allocations were fully randomized. It was difficult to recruit participants for our studies and we could not foresee the number of participants that would actually take part. Our strategy was to obtain our target number of participants (12) for one of the exemplar ADs first. Subsequent participants that we recruited we then allocated to use the second AD.

In Study 1, there was no problem in randomly assigning the subjects to the treatment as all participants performed all three information-seeking tasks and only one AD was used. As such, the research method used in Study 1 was more that of an experiment. In Studies 2 and 3, two out of three information-seeking tasks were given to the participants and two ADs were used.

We analysed the usage data collected in Studies 2 and 3 quantitatively to identify chunks. We also did a small investigation into the issues of architectural information foraging by analysing some usage data from Study 2 quantitatively and qualitatively.

3.5.1 Development of KaitoroCap

To study the usage of ADs, a straightforward approach would be to ask the ADs' consumers how they had made use of the documents. Questions on which sections of the documents they used, which sections were important, what order did they read the sections and so on could be asked to understand the usage of the documents. However, responses elicited this way might not reflect the real usage situation of the documents as the consumers sometime might not realize their actual ways of using them. To cater for this limitation, an unbiased way to study the usage of ADs is to capture the consumers' actual exploration paths through the documents when exploring architectural information in these documents.

To realize this approach, we developed KaitoroCap. KaitoroCap is the online prototype tool we developed for creating ADs as wiki pages in Atlassian Confluence Enterprise Wiki (Atlassian, 2013a), for capturing and visualising consumers' exploration paths through the documents, and for collecting annotation data provided by the consumers. Its main features are exploration paths capture, retrieval, analysis, hierarchical tree-view visualisation of paths, path searching, section rating, tagging, commenting, expanding or collapsing of document's sections, and page model generation to enable dynamic restructuring of the document. The design, implementation and usage examples of KaitoroCap can be found in Chapter 4.

3.5.2 Exploration Path Capture Using KaitoroCap (Studies 1 and 3)

KaitoroCap was used in Studies 1 and 3. In these studies, the participants explored ADs in the form of wiki pages in Atlassian Confluence Enterprise Wiki (Atlassian, 2013a) environment. Their exploration or navigation paths through the documents as well as annotation data (such as ratings, tags, comment, and so on) were captured using KaitoroCap.

3.5.2.1 Study 1 - User Evaluation Study of KaitoroCap

Study 1 served as a user evaluation of KaitoroCap where its features were assessed in terms of their usefulness, effectiveness and ease of use. It involved one AD and three information-seeking tasks. We observed how the participants used KaitoroCap and how they explored the AD while performing the information-seeking tasks. Prior to that, we briefly demonstrated the features of KaitoroCap to the participants.

Feedback gained from this study was used to improve KaitoroCap and the study design for subsequent collection of ADs' usage data in Studies 2 and 3.

Since it was a user evaluation study, the selection criteria used to recruit participants for Study 1 was less strict. Qualifying participants were only required to have some SA background, regardless of the number of years of experience in SA.

3.5.2.2 Study 3 - Exploration of Software Architecture Documents (Online Approach Study)

Study 3 made use of an improved version of KaitoroCap (Version 2.0) along with a refined version of the study design from Study 1. The improvements were based on the feedback from Study 1. Other improvements were also made to enable remote participation. We improved KaitoroCap in terms of its features, user interface, performance and help on usage (refer to Section "4.5 KaitoroCap Version 2" of Chapter 4). We also simplified and improved the study design. It was changed to involve only two information-seeking tasks instead of three. The scope of the study was expanded to include another AD and collection of some additional data. A comprehensive help file was made available instead of a demonstration of KaitoroCap's features to the participants. There was also no observation of the participants during their information seeking due to the physical distance. All the data in Study 3 were collected online, via KaitoroCap and online questionnaires.

We analysed the usage data from Study 3 quantitatively to find chunks. The participants in Study 3 consisted of both 'novices' and 'experts' in SA. Novices were those with less than two years and experts were those with at least two years of experience in SA. We could not recruit an adequate number of either novices or experts for Study 3 and had to use a mixture of the two different groups of participants for this study.

3.5.3 Study 2 - Exploration of Software Architecture Documents (Manual Approach Study)

The focus of Study 2 (Manual Approach Study) was to examine exploration of ADs without the extra requirement of using KaitoroCap for exploration path capture. The design of Study 2 is similar to Study 3 and therefore it is also an improvement over Study 1. Study 2 is effectively a 'manual' version of Study 3 (Online Approach Study) in the sense that the prototype was not used and the exploration path was not captured. Instead of wiki pages, participants explored ADs in Microsoft Word format or printed versions of these documents. Participants manually saved some of their responses such as tags, comments, ratings, suggested reading sequences in the Word documents. If they explored the printed ADs, they annotated this information on the printed documents. In Study 3, these types of response (except for the suggested reading sequence) were saved by KaitoroCap automatically as a participant navigated away from an annotated wiki page.

We analysed the usage data from Study 2 quantitatively to find chunks. We also analysed some of this data quantitatively and qualitatively to gain insights into the issues of architectural information foraging.

We used more stringent selection criteria to recruit participants for Study 2. Potential participants were required to have at least two years of industry experience related to SA or have taught a SA course or provided training on SA. This resulted in the recruitment of participants with stronger SA background (i.e. 'experts') for Study 2.

3.6 Discussion of Results and Conclusion

For each of the studies, we discussed our findings by drawing upon the related literature. We also described the limitations of each study. We consolidated the findings of our studies in an overall discussion. We also interpreted the results to answer the research questions. Finally, we concluded key findings from this research by presenting our main contributions, and key ideas for future research.

3.7 Summary

This chapter provides an overview of the research methodology adopted in this research. The detailed descriptions of each study are in their respective chapters. The next chapter focuses on KaitoroCap, the tool we developed to capture usage data.

4 KaitoroCap and User Evaluation Study (Study 1)

The first part of this chapter describes the design, implementation and usage examples of KaitoroCap. KaitoroCap is the online prototype tool we developed for creating ADs as wiki pages in Atlassian Confluence Enterprise Wiki (Atlassian, 2013a), for capturing and visualising users' exploration paths through the documents, and for collecting annotation data provided by the users. Its main features are exploration paths capture, retrieval, analysis, hierarchical tree-view visualisation of paths, path searching, section rating, tagging, commenting, expanding or collapsing of document's sections, and page model generation to enable dynamic re-structuring of document, making these paths available for future retracing and analysis.

The second part of this chapter presents the user evaluation study (Study 1) we conducted to gain feedback on the features of KaitoroCap. This includes the design of Study 1, user evaluation results and discussion, and improvements made to KaitoroCap. Feedback gained from Study 1 was used to improve KaitoroCap and the study design for subsequent collection of ADs' usage data in Studies 2 and 3. The last part of this chapter compares KaitoroCap with other AKM tools reviewed in Chapter 2.

4.1 Design of KaitoroCap

KaitoroCap is implemented as a plugin for Atlassian Confluence Enterprise Wiki (Atlassian, 2013a). Using a Wiki for collaboration and knowledge sharing is appealing for SA documentation (Louridas, 2006), (Farenhorst & van Vliet, 2008). Studies show wiki-based ADs support better document navigation (Bachmann & Merson, 2005).

KaitoroCap captures users' exploration paths through a document and make these paths available for future retracing and analysis. KaitoroCap saves exploration paths together with metadata to provide contextual information about the exploration. During the exploration, it allows ratings (R), tagging (T), commenting (C) of sections (abbreviated to RTC), and expanding or collapsing of sections. It also provides the ability to visualise exploration paths as hierarchical tree-views that are collapsible and expandable to provide both succinct and detailed views of the paths. The content of visited sections is extracted and embedded inside a tree-view visualisation of an exploration path, effectively a restructuring of the AD.

Figure 4.1 shows the high-level design of KaitoroCap. It consists of two main modules: Authoring and Exploration. The Authoring module comprises the Architecture Document (AD) Modeller, Template Maker and AD Authoring sub-modules. The AD Modeller is used to model the meta-model of ADs if it is required. The meta-model is saved and can be fed into the Template Maker to generate templates for ADs. The AD Authoring sub-module provides functions to create and edit the actual documents which are saved in the repository. It is also responsible for automatically creating a page model for each page created or edited. The page model improves performance by minimizing the amount of detail saved during the navigation path capture and for AD restructuring.

The Exploration module comprises DocViewer, Exploration Recorder, Exploration Analyser and Exploration Visualiser sub-modules. The DocViewer displays the pages of an AD and dynamically constructs the rating (R), tagging (T), commenting (C), expanding and collapsing features of the pages opened for viewing. The on-demand insertion of these features enables a clean separation of these features from the content of the pages. An exploration path consists of the elements (such as pages, sections and hyperlinks on pages, RTC fields) a user interacted with during his or her exploration session. The Exploration Recorder is responsible for capturing and saving exploration paths, metadata and annotation data (such as ratings, tags and comments). The metadata provides contextual information for an exploration, enabling it to be searched. The Exploration Recorder allows saved exploration and annotation data to be retrieved and displayed. The Exploration Analyser supports analysis (encoding and aggregation) of the raw exploration and annotation data. The Exploration Visualiser provides functionality to visualise analysed exploration paths as tree-views. It also provides search functionality to find saved exploration paths.

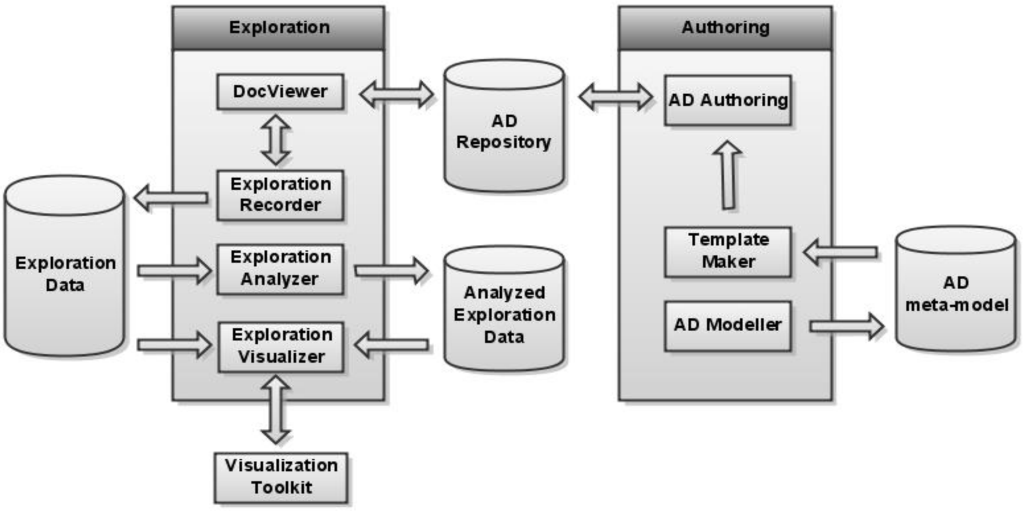


Figure 4.1: High-level design of KaitoroCap

4.2 Implementation of KaitoroCap

KaitoroCap was developed using several technologies including Atlassian plugin SDK (Atlassian, 2013b) , XWork Action for the controller (Atlassian, 2013c), Java beans for the model, Velocity (Atlassian, 2013d) for the view, and JQuery (The jQuery Foundation, 2013) for client-side scripting. Figure 4.2 shows the implementation of KaitoroCap.

The Architecture Document (AD) Authoring and DocViewer sub-modules are built on top of Confluence's functionality for creating and editing, and viewing of pages respectively. The former extends functionality to cater for the automatic creation of a page model for page created or edited. The relevant Confluence XWork actions are overridden and chained to a custom action to generate the page model. A page model contains information that uniquely identifies a page, the sections and hyperlinks on the page. It also comprises the details of these items (for example, the page title, the sections' titles and contents, hyperlinks' texts and urls, and so on).

The DocViewer sub-module extends the existing page viewing capability via JQuery script to dynamically insert the rating, tagging, commenting, expanding and collapsing features into each section of a page opened for viewing. A section starts with 'level 2 html heading (<H2>)' and comprises this heading and all the succeeding elements before the next 'level 2 html heading (<H2>)'. Each section is surrounded by a dynamic border, which is shown when the mouse pointer enters the section and hidden otherwise. The border provides visual cues regarding the section currently in focus for rating, tagging, commenting, expanding or collapsing.

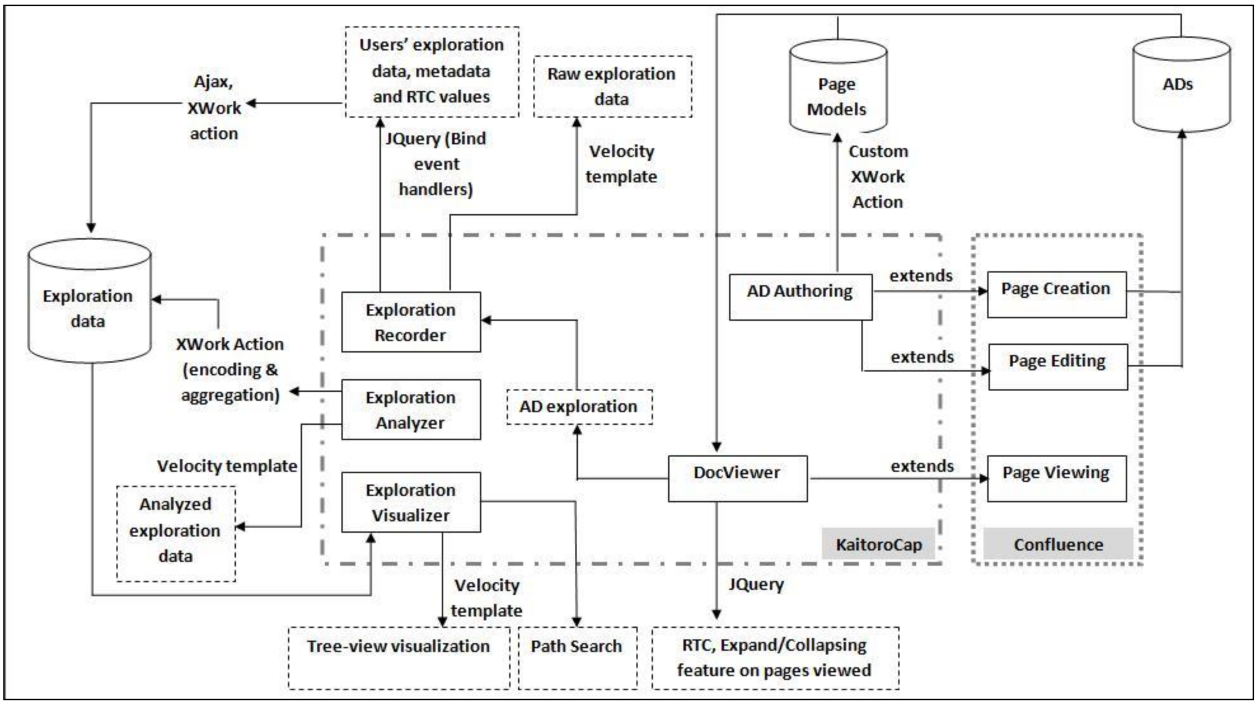


Figure 4.2: Implementation of KaitoroCap

The Exploration Recorder sub-module enables the user to capture exploration paths and save them together with contextual metadata (path name, keywords, role, reason, task). Other metadata such as start time, end time and path id is generated by the system and saved together with the paths. Annotation data (sections ratings, tags and comments) provided by users are also saved.

During an AD exploration session, event data are generated from the interaction of the user with the elements within and across the pages. An element in this context refers either to a star rating, tag, comment, expand/collapsing feature or a hyperlink. Consequently, the types of interactions captured include clicking on the star rating features to rate a section, entering tags and comments, clicking to expand or collapse a section as well as clicking on hyperlinks to navigate to other pages.

The series of event data items generated from the user interactions constitutes the exploration path data. A single event data item contains an event identifier (id), id and type of the affected element, information about the interaction (such as the type of interaction, timestamp and so on) and details of the page (the sequence of page within the exploration session and page model identifier).

The Exploration Recorder makes use of JQuery to dynamically bind event handlers to the respective events (mouse click, mouse over, 'enter' key press) of the elements on the pages of AD. All the event data are later passed via AJAX to an XWork action class to be parsed and saved to the database. The saved exploration path data is displayed by populating a Velocity template.

In the Exploration Analyser sub-module, exploration data is retrieved and analysed in an XWork Action class. This involves encoding and aggregation of the raw exploration data and the determination of the semantic events. Encoding and aggregation enables the abstraction of the low-level interaction events in the event data into higher-level semantic navigation events. Encoding is performed by comparing each event data item with its succeeding event data item in the series. This is used to determine the occurrence of changes (in terms of element id, type of interaction, values, time stamp, and so on) from one event item to the next. The encoded data is then analysed for the possibility of event aggregation. Event data in sequence, same element id and type of interaction are candidates for aggregation.

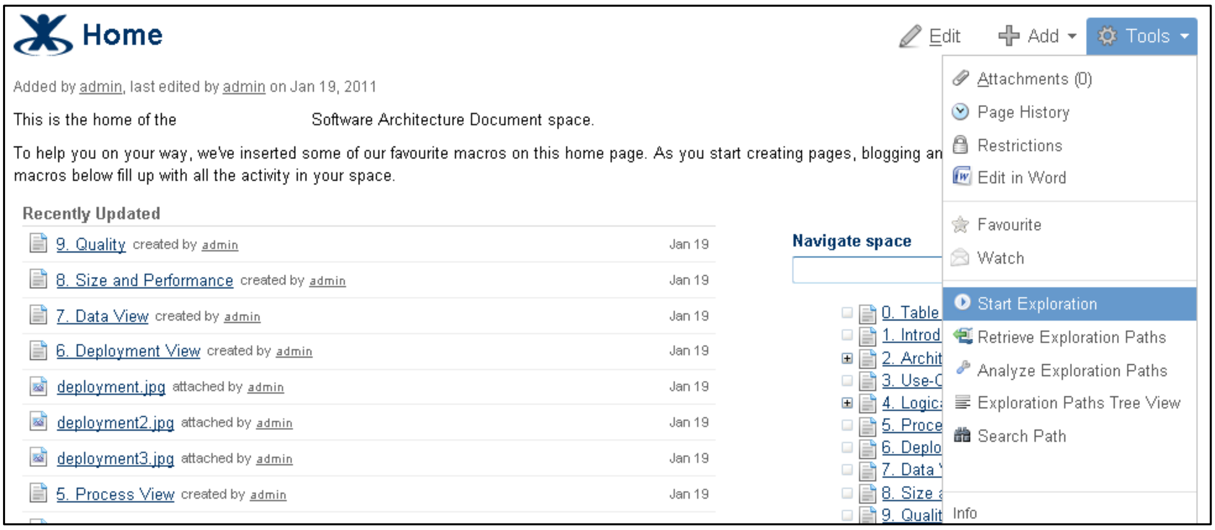
Aggregation combines similar events into a single description (Maybury, 1995). In this prototype, event data which are in a sequence and have the same element id and type of interaction are candidates for aggregation. Depending on the type of interaction, other fields of the event need to be checked to decide on aggregation. For example, if the type of interaction is a click on the star rating feature, then checking on the value of the rating needs to be done to determine change. If there is no change in the value, the two successive events are aggregated. The cumulative time difference is calculated and used to determine the overarching semantic navigation event for the aggregation. Currently, the semantic navigation events are limited to browsing, reading and other task. The raw, encoded and aggregated exploration data are displayed using velocity template.

The Exploration Visualiser sub-module displays the exploration paths in the form of hierarchical tree-views that are collapsible and expandable. The minutiae of an exploration path such as the details of the visited pages (for example page title), the details of the visited sections (section title and content) and hyperlinks (text and url) are extracted from the respective page models and embedded in the tree-view. The Exploration Visualiser parses exploration data to build internal tree structures which are fed to the graphical display of the tree-views created using JQuery. It also provides the function to search for the exploration paths based on the metadata captured at the start of each exploration.

4.3 Usage Example

In this section, we describe a detailed usage example to show the main features of KaitoroCap. The usage scenario is: a user is engaged with the information-seeking task of finding out how the system (described by the AD) was designed at the architectural level to achieve security. Figure 4.3 shows the main user interface of KaitoroCap, extending the ‘Tool’ menu in Atlassian Confluence Wiki (Atlassian, 2013a). The user initiates the capturing of the exploration path for the information-seeking task, by choosing the ‘Start Exploration’ menu item (Figure 4.3) and filling in suitable metadata (path name, keywords, role, reason, task) for the path (Figure 4.4). The values chosen for the metadata should be relevant to the information-seeking task as they serve as contextual information for an exploration and are used in the ‘Search’ feature to find suitable exploration paths.

By clicking on the ‘Save’ button, the metadata is saved and the exploration capturing session is started. The ‘Start Exploration’ menu item is changed to ‘Stop Exploration’ (Figure 4.5) which can be clicked anytime to terminate the capturing of the exploration session.



Home

Added by admin, last edited by admin on Jan 19, 2011

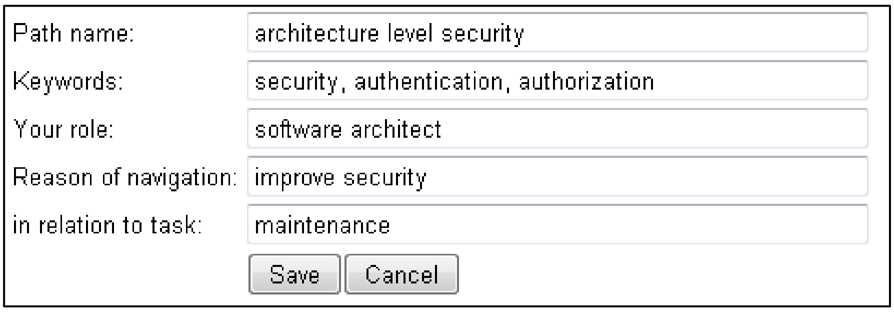
This is the home of the Software Architecture Document space.

To help you on your way, we've inserted some of our favourite macros on this home page. As you start creating pages, blogging an macros below fill up with all the activity in your space.

Recently Updated

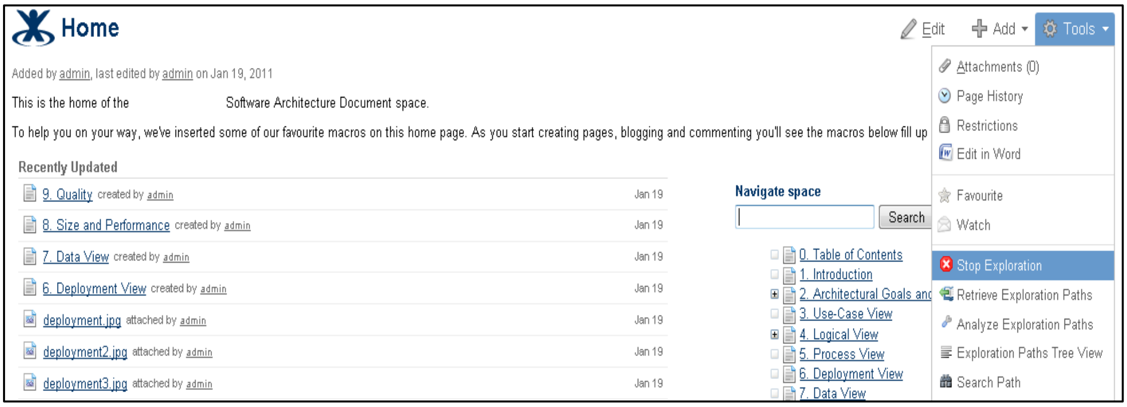
9. Quality created by admin	Jan 19
8. Size and Performance created by admin	Jan 19
7. Data View created by admin	Jan 19
6. Deployment View created by admin	Jan 19
deployment.jpg attached by admin	Jan 19
deployment2.jpg attached by admin	Jan 19
deployment3.jpg attached by admin	Jan 19
5. Process View created by admin	Jan 19

Figure 4.3: Main User Interface Of KaitoroCap



Path name:	architecture level security
Keywords:	security, authentication, authorization
Your role:	software architect
Reason of navigation:	improve security
in relation to task:	maintenance

Figure 4.4: Saving the Metadata of an Exploration Path



Home

Added by admin, last edited by admin on Jan 19, 2011

This is the home of the Software Architecture Document space.

To help you on your way, we've inserted some of our favourite macros on this home page. As you start creating pages, blogging and commenting you'll see the macros below fill up

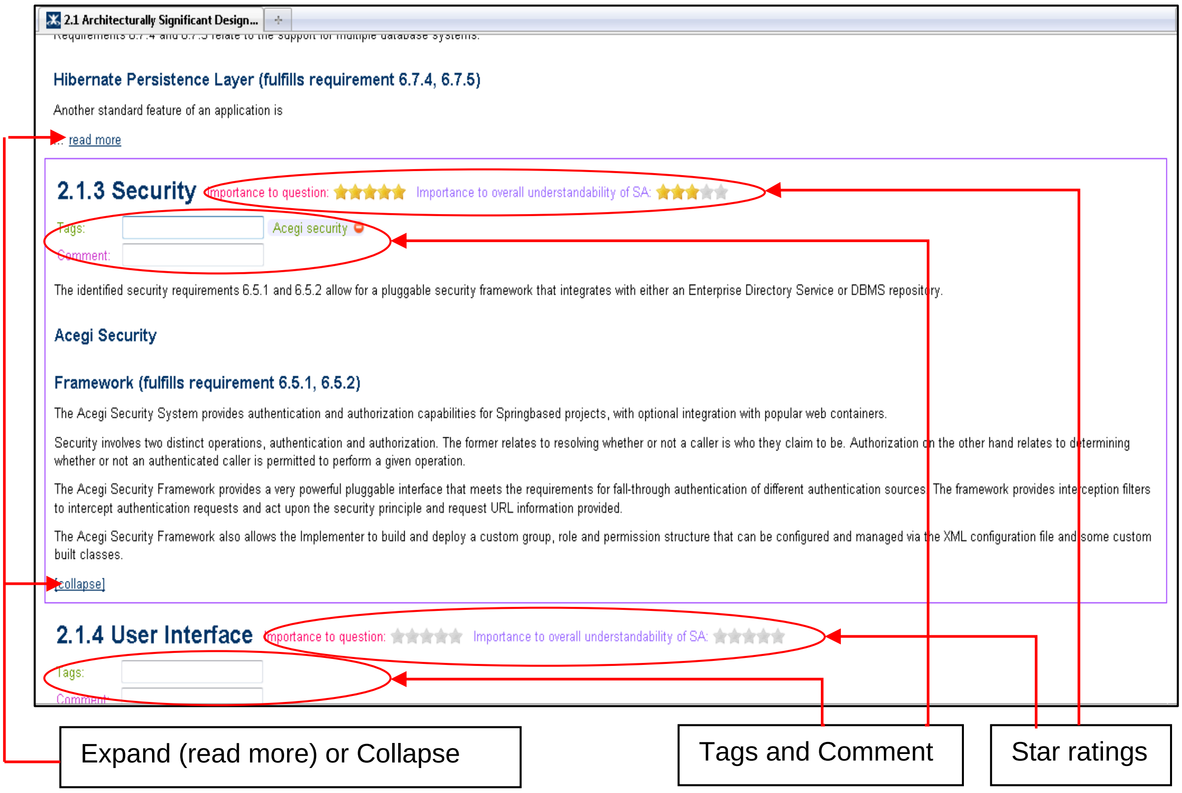
Recently Updated

9. Quality created by admin	Jan 19
8. Size and Performance created by admin	Jan 19
7. Data View created by admin	Jan 19
6. Deployment View created by admin	Jan 19
deployment.jpg attached by admin	Jan 19
deployment2.jpg attached by admin	Jan 19
deployment3.jpg attached by admin	Jan 19

Figure 4.5: Stopping an Exploration Capturing Session

Within an exploration session, the user can navigate to any page of an AD. KaitoroCap automatically inserts star ratings, tag and comment, expand (read more) and collapse features into each section of an opened page. Figure 4.6 shows the example of the user opening “2.1 Architecturally Significant Design Decisions” page for viewing. RTC, expand (read more) and collapse features are inserted into all the seven sections on this page. Figure 4.6 shows Section “2.1.3 Security” and parts of its preceding and succeeding sections.

As the user explores the pages of the AD, all the user interactions with the elements (ratings, tag, comment, expand and collapsing features, and hyperlinks) on the pages are captured in the background together with the sequence of navigation of the pages. All this data is saved whenever the user navigates away from a page.



2.1 Architecturally Significant Design...

Requirements 6.7.4 and 6.7.5 relate to the support for multiple database systems.

Hibernate Persistence Layer (fulfills requirement 6.7.4, 6.7.5)

Another standard feature of an application is

... read more

2.1.3 Security Importance to question: ★★★★★ Importance to overall understandability of SA: ★★★☆☆

Tags: Acegi security

Comment:

The identified security requirements 6.5.1 and 6.5.2 allow for a pluggable security framework that integrates with either an Enterprise Directory Service or DBMS repository.

Acegi Security

Framework (fulfills requirement 6.5.1, 6.5.2)

The Acegi Security System provides authentication and authorization capabilities for Springbased projects, with optional integration with popular web containers.

Security involves two distinct operations, authentication and authorization. The former relates to resolving whether or not a caller is who they claim to be. Authorization on the other hand relates to determining whether or not an authenticated caller is permitted to perform a given operation.

The Acegi Security Framework provides a very powerful pluggable interface that meets the requirements for fall-through authentication of different authentication sources. The framework provides interception filters to intercept authentication requests and act upon the security principle and request URL information provided.

The Acegi Security Framework also allows the Implementer to build and deploy a custom group, role and permission structure that can be configured and managed via the XML configuration file and some custom built classes.

[collapse]

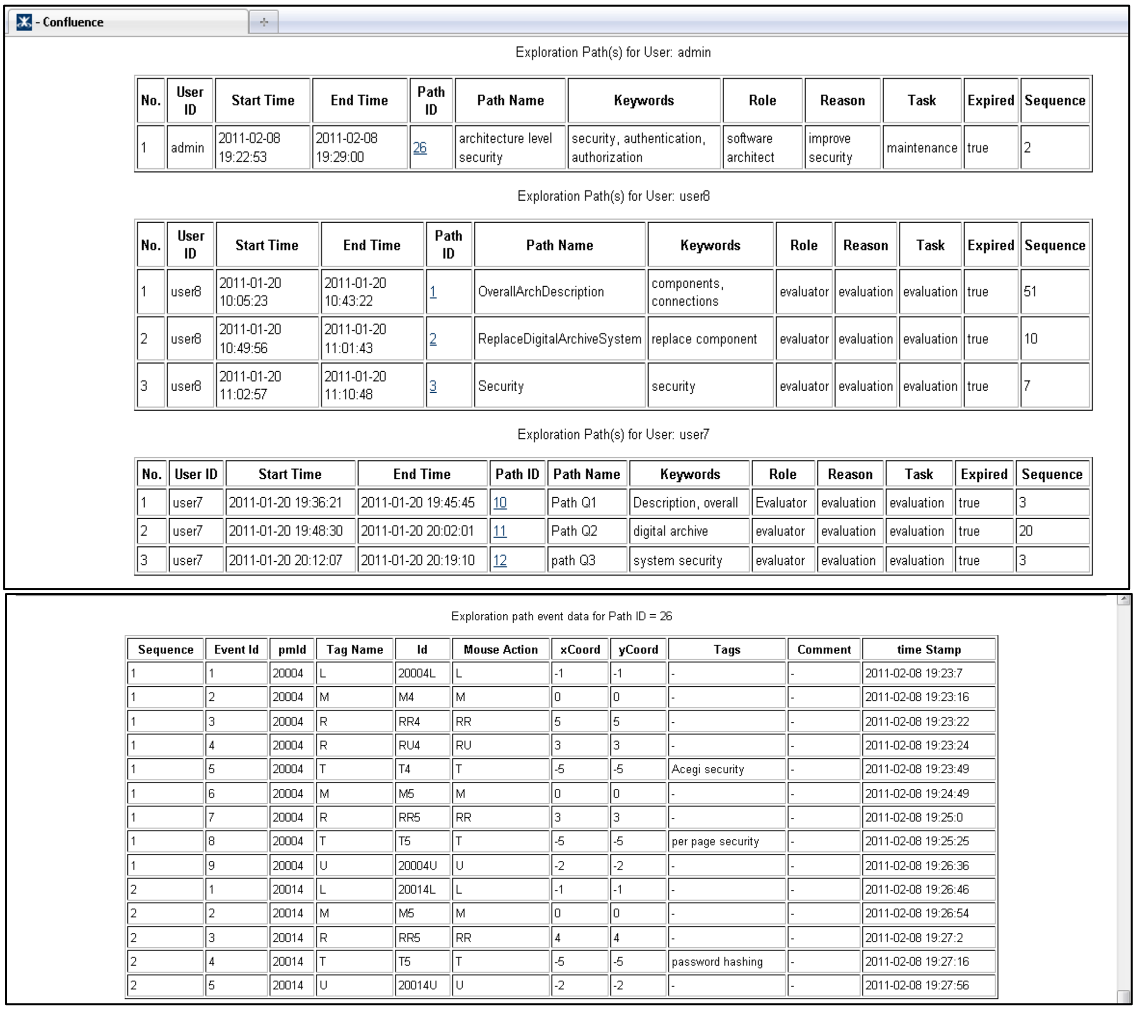
2.1.4 User Interface Importance to question: ☆☆☆☆☆ Importance to overall understandability of SA: ☆☆☆☆☆

Tags:

Comment:

Figure 4.6: Star ratings, Tagging, Commenting, Expanding or Collapsing Features of Sections

The user can retrieve the raw exploration data (Figure 4.7), the analysed exploration data (Figure 4.8) and the tree-views of the exploration paths (Figure 4.9, 4.10, 4.11), by selecting the ‘Retrieve Exploration Paths’, ‘Analyse Exploration Paths’ and ‘Exploration Paths Tree View’ menu item respectively, from the main user interface of KaitoroCap (Figure 4.3).



Exploration Path(s) for User: admin

No.	User ID	Start Time	End Time	Path ID	Path Name	Keywords	Role	Reason	Task	Expired	Sequence
1	admin	2011-02-08 19:22:53	2011-02-08 19:29:00	26	architecture level security	security, authentication, authorization	software architect	improve security	maintenance	true	2

Exploration Path(s) for User: user8

No.	User ID	Start Time	End Time	Path ID	Path Name	Keywords	Role	Reason	Task	Expired	Sequence
1	user8	2011-01-20 10:05:23	2011-01-20 10:43:22	1	OverallArchDescription	components, connections	evaluator	evaluation	evaluation	true	51
2	user8	2011-01-20 10:49:56	2011-01-20 11:01:43	2	ReplaceDigitalArchiveSystem	replace component	evaluator	evaluation	evaluation	true	10
3	user8	2011-01-20 11:02:57	2011-01-20 11:10:48	3	Security	security	evaluator	evaluation	evaluation	true	7

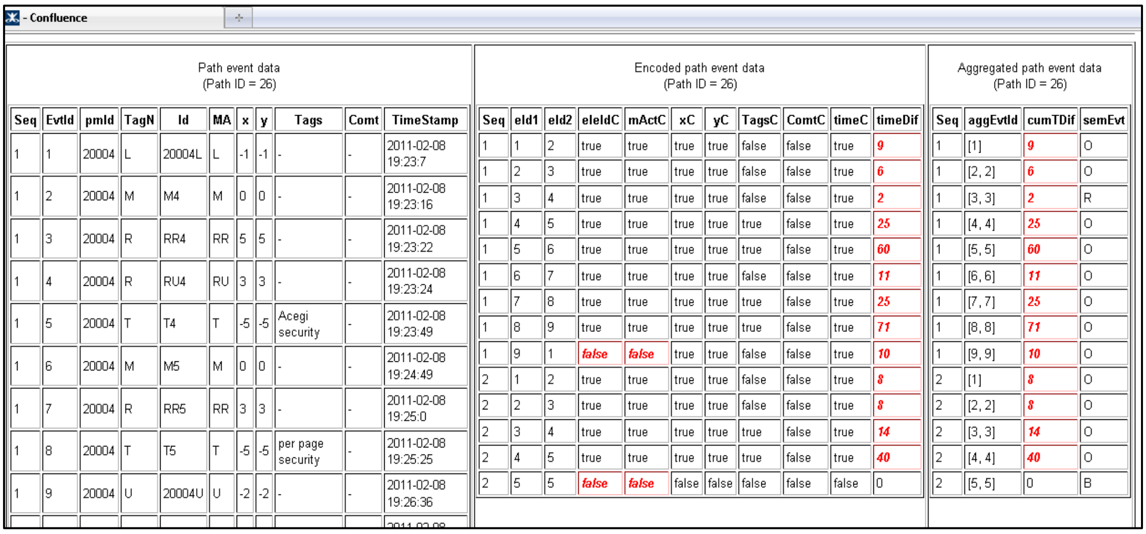
Exploration Path(s) for User: user7

No.	User ID	Start Time	End Time	Path ID	Path Name	Keywords	Role	Reason	Task	Expired	Sequence
1	user7	2011-01-20 19:36:21	2011-01-20 19:45:45	10	Path Q1	Description, overall	Evaluator	evaluation	evaluation	true	3
2	user7	2011-01-20 19:48:30	2011-01-20 20:02:01	11	Path Q2	digital archive	evaluator	evaluation	evaluation	true	20
3	user7	2011-01-20 20:12:07	2011-01-20 20:19:10	12	path Q3	system security	evaluator	evaluation	evaluation	true	3

Exploration path event data for Path ID = 26

Sequence	Event Id	pmId	Tag Name	Id	Mouse Action	xCoord	yCoord	Tags	Comment	time Stamp
1	1	20004	L	20004L	L	-1	-1	-	-	2011-02-08 19:23:7
1	2	20004	M	M4	M	0	0	-	-	2011-02-08 19:23:16
1	3	20004	R	RR4	RR	5	5	-	-	2011-02-08 19:23:22
1	4	20004	R	RU4	RU	3	3	-	-	2011-02-08 19:23:24
1	5	20004	T	T4	T	-5	-5	Acegi security	-	2011-02-08 19:23:49
1	6	20004	M	M5	M	0	0	-	-	2011-02-08 19:24:49
1	7	20004	R	RR5	RR	3	3	-	-	2011-02-08 19:25:0
1	8	20004	T	T5	T	-5	-5	per page security	-	2011-02-08 19:25:25
1	9	20004	U	20004U	U	-2	-2	-	-	2011-02-08 19:26:36
2	1	20014	L	20014L	L	-1	-1	-	-	2011-02-08 19:26:46
2	2	20014	M	M5	M	0	0	-	-	2011-02-08 19:26:54
2	3	20014	R	RR5	RR	4	4	-	-	2011-02-08 19:27:2
2	4	20014	T	T5	T	-5	-5	password hashing	-	2011-02-08 19:27:16
2	5	20014	U	20014U	U	-2	-2	-	-	2011-02-08 19:27:56

Figure 4.7: Raw Exploration Data



Path event data
(Path ID = 26)

Seq	EvtId	pmId	TagN	Id	MA	x	y	Tags	Comt	TimeStamp
1	1	20004	L	20004L	L	-1	-1	-	-	2011-02-08 19:23:7
1	2	20004	M	M4	M	0	0	-	-	2011-02-08 19:23:16
1	3	20004	R	RR4	RR	5	5	-	-	2011-02-08 19:23:22
1	4	20004	R	RU4	RU	3	3	-	-	2011-02-08 19:23:24
1	5	20004	T	T4	T	-5	-5	Acegi security	-	2011-02-08 19:23:49
1	6	20004	M	M5	M	0	0	-	-	2011-02-08 19:24:49
1	7	20004	R	RR5	RR	3	3	-	-	2011-02-08 19:25:0
1	8	20004	T	T5	T	-5	-5	per page security	-	2011-02-08 19:25:25
1	9	20004	U	20004U	U	-2	-2	-	-	2011-02-08 19:26:36

Encoded path event data
(Path ID = 26)

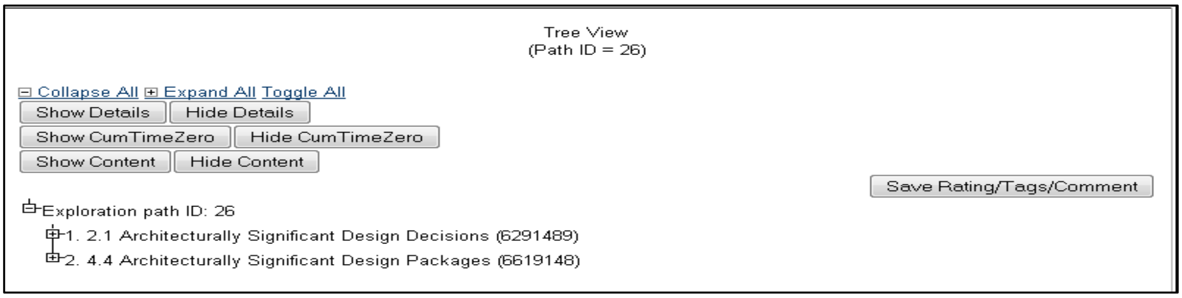
Seq	eId1	eId2	eleIdC	mActC	xC	yC	TagsC	ComtC	timeC	timeDif
1	1	2	true	true	true	true	false	false	true	9
1	2	3	true	true	true	true	false	false	true	6
1	3	4	true	true	true	true	false	false	true	2
1	4	5	true	true	true	true	true	false	true	25
1	5	6	true	true	true	true	true	false	true	60
1	6	7	true	true	true	true	false	false	true	11
1	7	8	true	true	true	true	true	false	true	25
1	8	9	true	true	true	true	true	false	true	71
1	9	1	false	false	true	true	false	false	true	10
2	1	2	true	true	true	true	false	false	true	8
2	2	3	true	true	true	true	false	false	true	8
2	3	4	true	true	true	true	true	false	true	14
2	4	5	true	true	true	true	true	false	true	40
2	5	5	false	false	false	false	false	false	false	0

Aggregated path event data
(Path ID = 26)

Seq	aggEvtId	cumTDif	semEvt
1	[1]	9	O
1	[2, 2]	6	O
1	[3, 3]	2	R
1	[4, 4]	25	O
1	[5, 5]	60	O
1	[6, 6]	11	O
1	[7, 7]	25	O
1	[8, 8]	71	O
1	[9, 9]	10	O
2	[1]	8	O
2	[2, 2]	8	O
2	[3, 3]	14	O
2	[4, 4]	40	O
2	[5, 5]	0	B

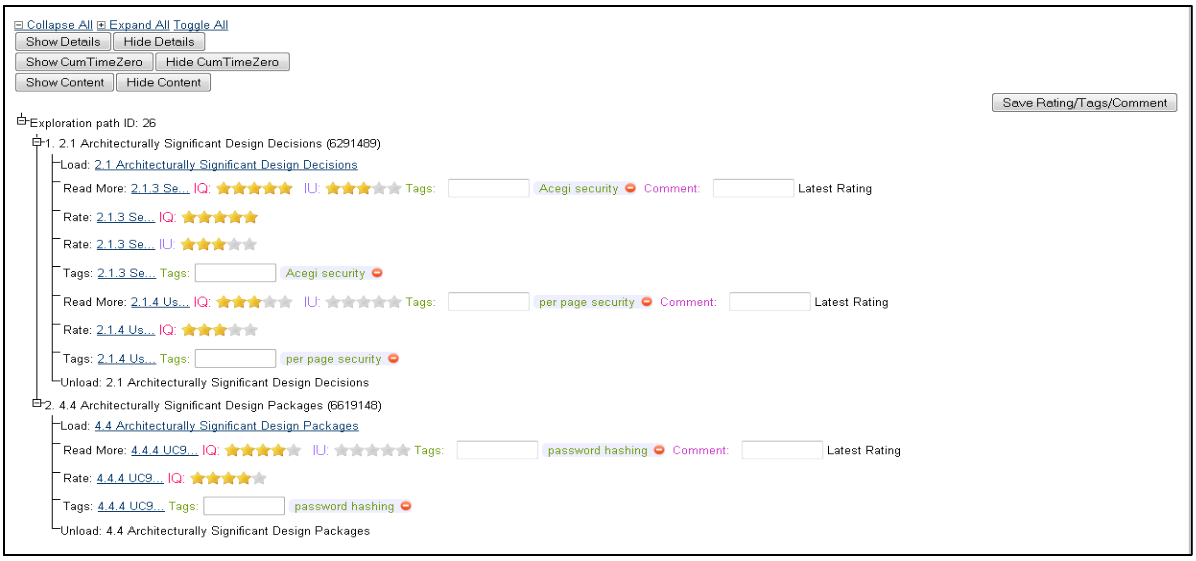
Figure 4.8: Analysed Exploration data

The hierarchical tree-view can be collapsed (Figure 4.9) and expanded (Figure 4.10) to provide both succinct and detailed views of the paths. It can also be toggled to reverse between the two views. Besides, details of event data and the content of the visited sections in the tree-view can also be shown (Figure 4.11) or hidden. Embedding the content of the visited sections in the tree-view visualisation of an exploration path (Figure 4.11) effectively makes the path a restructuring of the AD.



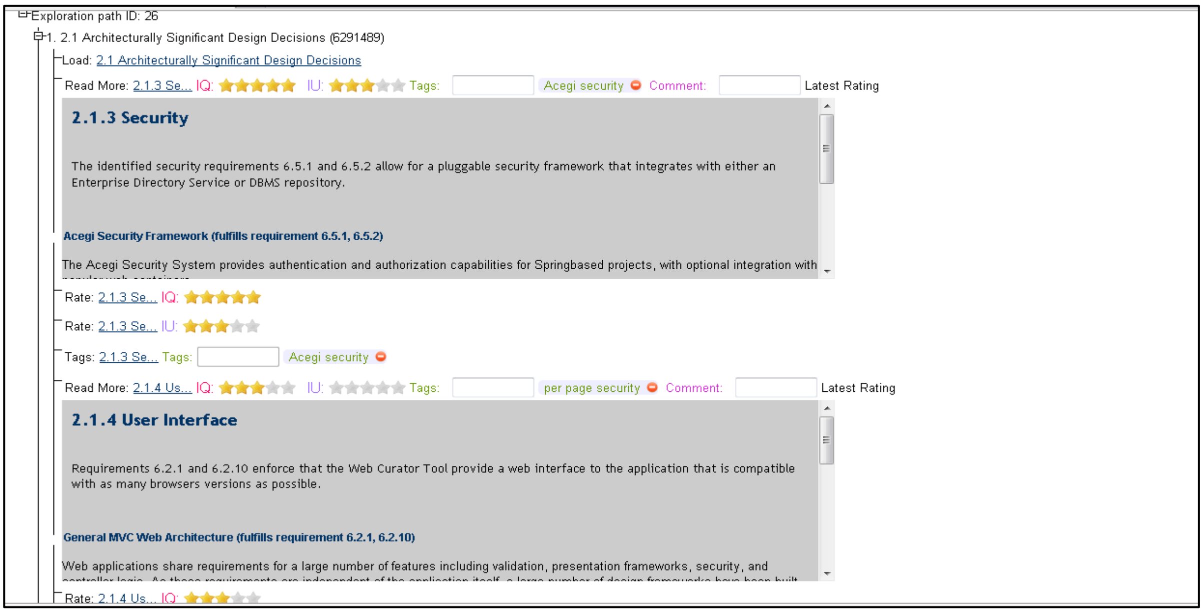
Exploration path ID: 26
1. 2.1 Architecturally Significant Design Decisions (6291489)
2. 4.4 Architecturally Significant Design Packages (6619148)

Figure 4.9: Collapsed Tree-view



Exploration path ID: 26
1. 2.1 Architecturally Significant Design Decisions (6291489)
Load: 2.1 Architecturally Significant Design Decisions
Read More: 2.1.3 Se... IQ: ★★★★★ IU: ★★★☆☆ Tags: Acegi security Comment: Latest Rating
Rate: 2.1.3 Se... IQ: ★★★★★
Rate: 2.1.3 Se... IU: ★★★☆☆
Tags: 2.1.3 Se... Tags: Acegi security
Read More: 2.1.4 Us... IQ: ★★★☆☆ IU: ☆☆☆☆☆ Tags: per page security Comment: Latest Rating
Rate: 2.1.4 Us... IQ: ★★★☆☆
Tags: 2.1.4 Us... Tags: per page security
Unload: 2.1 Architecturally Significant Design Decisions
2. 4.4 Architecturally Significant Design Packages (6619148)
Load: 4.4 Architecturally Significant Design Packages
Read More: 4.4.4 UC9... IQ: ★★★★☆ IU: ☆☆☆☆☆ Tags: password hashing Comment: Latest Rating
Rate: 4.4.4 UC9... IQ: ★★★★☆
Tags: 4.4.4 UC9... Tags: password hashing
Unload: 4.4 Architecturally Significant Design Packages

Figure 4.10: Expanded Tree-view



Exploration path ID: 26
1. 2.1 Architecturally Significant Design Decisions (6291489)
Load: 2.1 Architecturally Significant Design Decisions
Read More: 2.1.3 Se... IQ: ★★★★★ IU: ★★★☆☆ Tags: Acegi security Comment: Latest Rating

2.1.3 Security

The identified security requirements 6.5.1 and 6.5.2 allow for a pluggable security framework that integrates with either an Enterprise Directory Service or DBMS repository.

Acegi Security Framework (fulfills requirement 6.5.1, 6.5.2)

The Acegi Security System provides authentication and authorization capabilities for Springbased projects, with optional integration with

Rate: 2.1.3 Se... IQ: ★★★★★
Rate: 2.1.3 Se... IU: ★★★☆☆
Tags: 2.1.3 Se... Tags: Acegi security
Read More: 2.1.4 Us... IQ: ★★★☆☆ IU: ☆☆☆☆☆ Tags: per page security Comment: Latest Rating

2.1.4 User Interface

Requirements 6.2.1 and 6.2.10 enforce that the Web Curator Tool provide a web interface to the application that is compatible with as many browsers versions as possible.

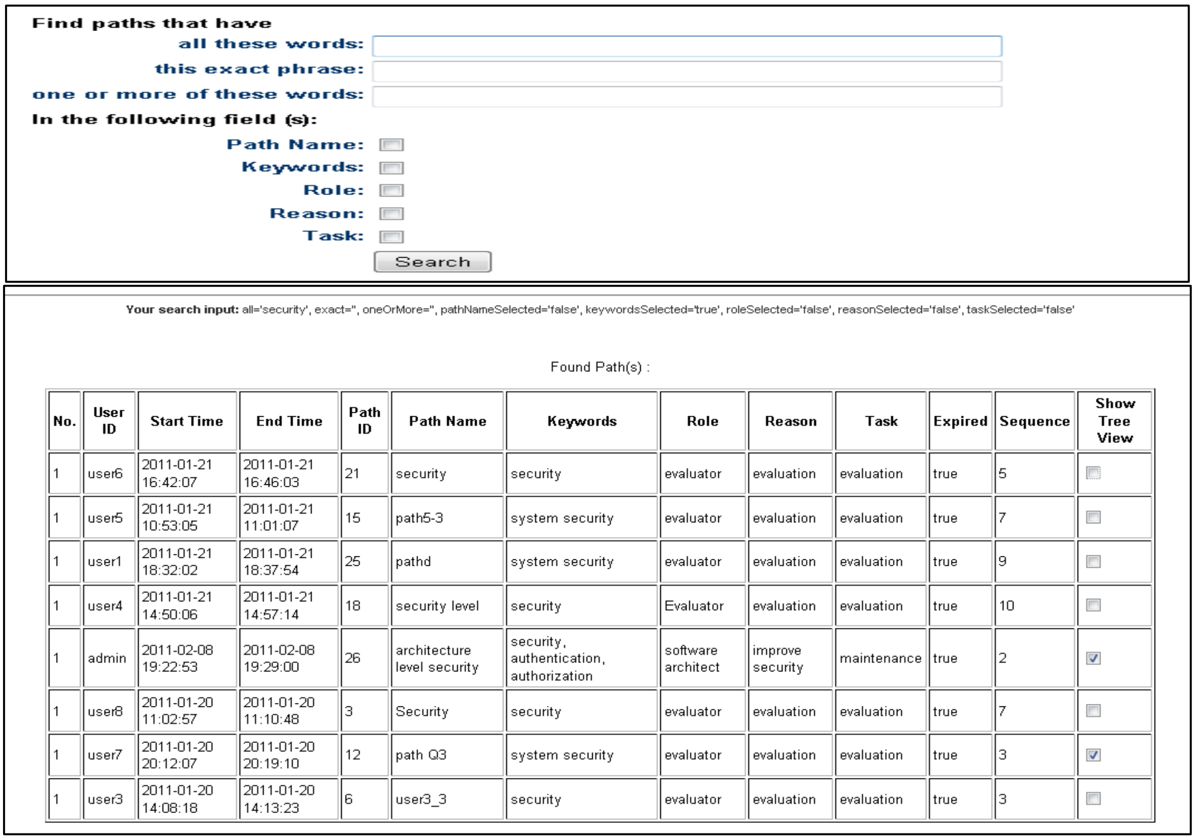
General MVC Web Architecture (fulfills requirement 6.2.1, 6.2.10)

Web applications share requirements for a large number of features including validation, presentation frameworks, security, and

Rate: 2.1.4 Us... IQ: ★★★☆☆

Figure 4.11: Expanded Tree-view with Embedded Content

We describe an exemplar usage scenario of the captured exploration paths: a new user wants to find out how the system was designed to support security. Instead of navigating through the AD in search of the information related to security, with KaitoroCap, s/he has the option of retrieving and following previous navigation paths which were driven by similar information-seeking task, which is searching for ‘security’, assuming that these paths were captured. S/he searches for existing exploration paths related to security by using the search feature (Figure 4.12) by selecting the ‘Search Path’ menu item from the main user interface of KaitoroCap (Figure 4.3). The user then searches by providing the search terms and choosing the metadata to be searched. The resulting paths are shown in a table and the tree-views of the paths can be selected to be displayed. This allows the user to retrace others’ navigation paths. Besides the founded navigation paths can be displayed side-by-side (Figure 4.13) for manual comparison.



Find paths that have

all these words:

this exact phrase:

one or more of these words:

In the following field (s):

Path Name: □

Keywords: □

Role: □

Reason: □

Task: □

Your search input: all='security', exact='', oneOrMore='', pathNameSelected='false', keywordsSelected='true', roleSelected='false', reasonSelected='false', taskSelected='false'

Found Path(s) :

No.	User ID	Start Time	End Time	Path ID	Path Name	Keywords	Role	Reason	Task	Expired	Sequence	Show Tree View
1	user6	2011-01-21 16:42:07	2011-01-21 16:46:03	21	security	security	evaluator	evaluation	evaluation	true	5	□
1	user5	2011-01-21 10:53:05	2011-01-21 11:01:07	15	path5-3	system security	evaluator	evaluation	evaluation	true	7	□
1	user1	2011-01-21 18:32:02	2011-01-21 18:37:54	25	pathd	system security	evaluator	evaluation	evaluation	true	9	□
1	user4	2011-01-21 14:50:06	2011-01-21 14:57:14	18	security level	security	Evaluator	evaluation	evaluation	true	10	□
1	admin	2011-02-08 19:22:53	2011-02-08 19:29:00	26	architecture level security	security, authentication, authorization	software architect	improve security	maintenance	true	2	☑
1	user8	2011-01-20 11:02:57	2011-01-20 11:10:48	3	Security	security	evaluator	evaluation	evaluation	true	7	□
1	user7	2011-01-20 20:12:07	2011-01-20 20:19:10	12	path Q3	system security	evaluator	evaluation	evaluation	true	3	☑
1	user3	2011-01-20 14:08:18	2011-01-20 14:13:23	6	user3_3	security	evaluator	evaluation	evaluation	true	3	□

Figure 4.12: Searching of exploration paths

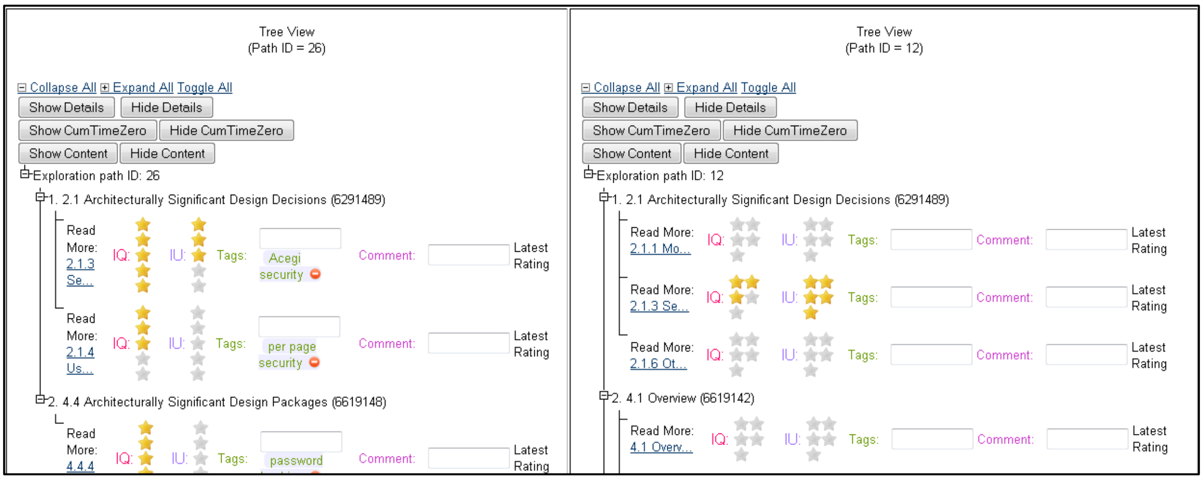
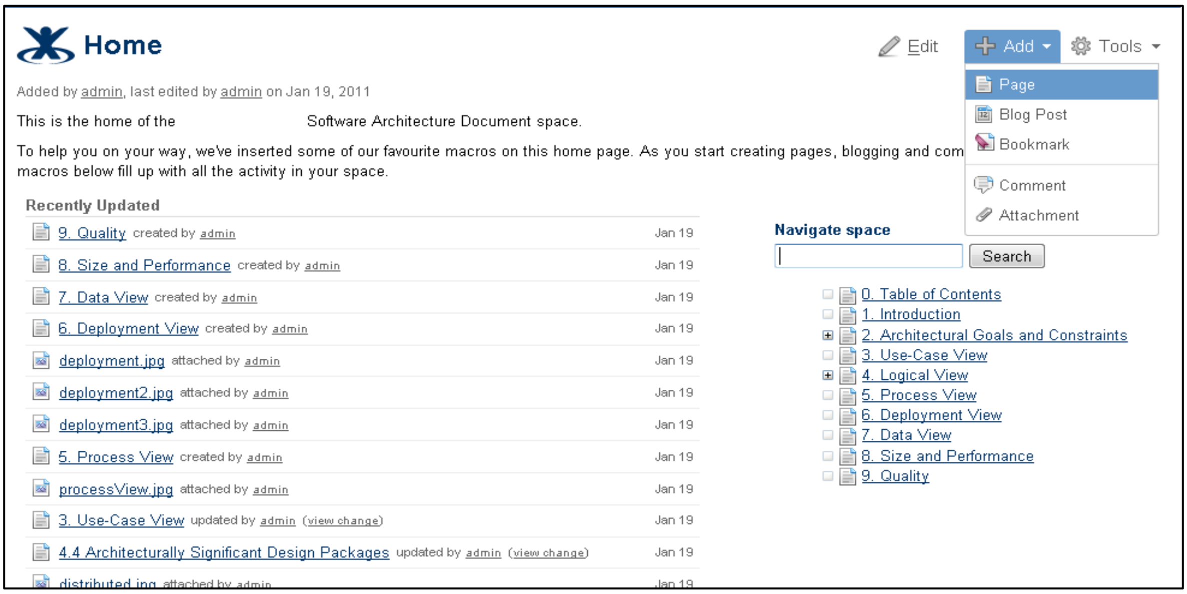


Figure 4.13: Displaying tree-views of exploration paths side-by-side

The last feature is the creation of a new AD page through the 'Page' menu item under 'Add' menu provided by Atlassian Confluence (Figure 4.14). After filling in the title and the content of the page, the page is saved. Upon saving the page, a page model for the page is generated automatically. The page model is invisible to the user.



Added by admin, last edited by admin on Jan 19, 2011

This is the home of the Software Architecture Document space.

To help you on your way, we've inserted some of our favourite macros on this home page. As you start creating pages, blogging and com macros below fill up with all the activity in your space.

Recently Updated

- 9. Quality created by admin Jan 19
- 8. Size and Performance created by admin Jan 19
- 7. Data View created by admin Jan 19
- 6. Deployment View created by admin Jan 19
- deployment.jpg attached by admin Jan 19
- deployment2.jpg attached by admin Jan 19
- deployment3.jpg attached by admin Jan 19
- 5. Process View created by admin Jan 19
- processView.jpg attached by admin Jan 19
- 3. Use-Case View updated by admin (view change) Jan 19
- 4.4 Architecturally Significant Design Packages updated by admin (view change) Jan 19
- distributed.jpg attached by admin Jan 19

- 0. Table of Contents
- 1. Introduction
- ⊞ 2. Architectural Goals and Constraints
- 3. Use-Case View
- ⊞ 4. Logical View
- 5. Process View
- 6. Deployment View
- 7. Data View
- 8. Size and Performance
- 9. Quality

Figure 4.14: Creation of A New AD Page

4.4 Study 1 - User Evaluation Study Of KaitoroCap

This section describes the user evaluation study (Study 1) we conducted to gain feedback on the features of KaitoroCap. Feedback gained from Study 1 was used to improve KaitoroCap and the study design for subsequent collection of ADs' usage data in Studies 2 and 3.

4.4.1 Study Design

To assess the features of KaitoroCap, we recruited participants to explore an AD (AD) to find information (or answer) for three information-seeking tasks (or questions).

The AD used was a 24-page document defining the architecture of a web curator system (WCT) that manages and harvests digital web content for future preservation (National Library of New Zealand, 2006). This WCT document was also used in Studies 2 and 3 later. The reasons of choosing this AD are given in Section "5.2.1 Choice of Documents" in Chapter 5 (which describes Study 2). The WCT document was created as wiki pages in Atlassian Confluence Enterprise Wiki (Atlassian, 2013a), where KaitoroCap could be used to capture the users' exploration through the AD and their annotation (such as ratings, tags and comment) on the sections of the AD. Refer to Section "12.2.3 Preparation of Documents" in Chapter 12 (which describes Study 3) for further details on the creation of AD pages in Atlassian Confluence.

The three information-seeking tasks were related to finding: the overall SA of WCT; how to change a specific part of WCT and access which parts will be affected; and how WCT was designed to achieve security. These tasks were refined for Study 2. For the rationale of specifying these tasks, refer to Section "5.2.2 Determination of Information-seeking Tasks and Roles" in Chapter 5 (Study 2).

For each task, a participant captured his exploration path using KaitoroCap's path-capturing feature. The participant was asked to state the time the task was started (start time), record his or her answer, and finishing time (stop time) of the task. The participant was asked to rate AD sections he or she visited in terms of their importance to answering the question and the importance of the sections to his or her overall understanding of the SA of the system. He or she was also encouraged to tag or comment on visited sections. A tag is a user-defined keyword reflecting a section's content. A comment is a participant's more elaborated opinion on a section.

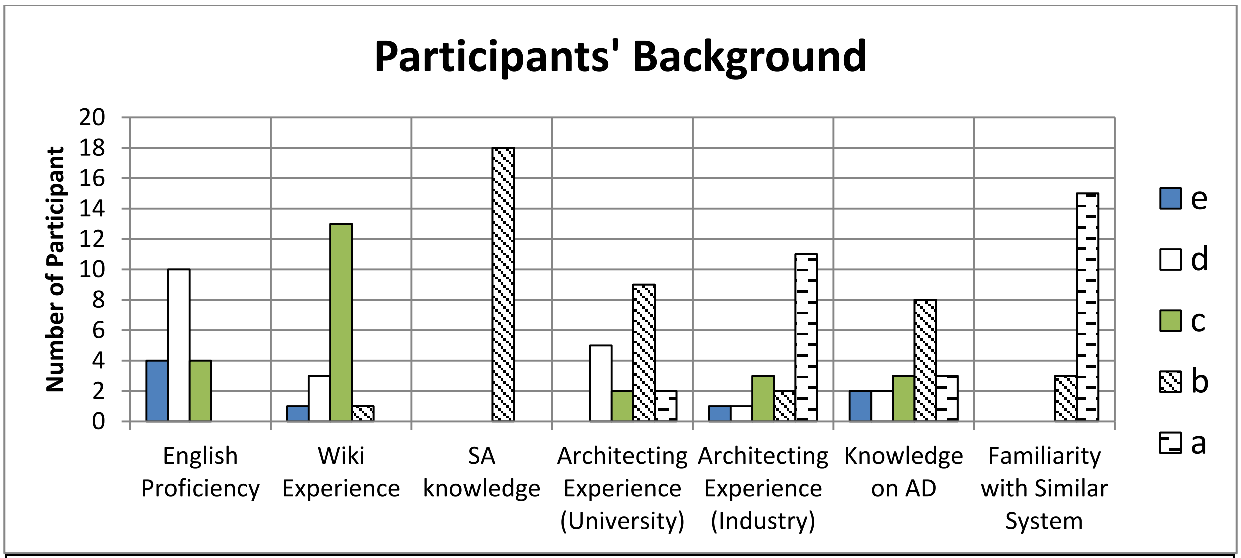
Prior to the exploration of the AD, participants completed a pre-questionnaire on their background: level of education, English language proficiency, Software Architecture Knowledge and experience, Wiki and AD experience, and knowledge of system similar to WCT. After completing the three tasks, participants completed a post-questionnaire on their perceptions of KaitoroCap's features, the AD, their navigation characteristics, and suggestions for improvement.

A set of data collection instruments (Appendix 4.1) was given to those who consented to take part in the study. We explained the tasks to be completed, clarified any question and briefly demonstrated the features of KaitoroCap. We observed how the participants used KaitoroCap and how they explored the AD while performing the information-seeking tasks.

Since it was a user evaluation study, the selection criteria used to recruit participants for Study 1 was less strict. Qualifying participants were only required to have some SA background, regardless of the number of years of experience in SA. We invited potential participants from three different institutions, namely University of Auckland (New Zealand), University of Malaya (Malaysia) and Swinburne University of Technology (Australia). We did so by emailing invitations using our professional contacts and their referrals, disseminating the recruitment advertisement with the help of our local acquaintances, and giving brief talks in lectures attended by students with SA background.

4.4.2 User Evaluation Results and Discussion

We recruited 21 participants (comprising 12 PhDs, 2 Master students, 2 Undergraduates, 3 Postdoctoral scholars, 1 academic staff and 1 researcher) from three different institutions. One of them did not complete the questionnaires; two did not have SA background. These three participants were excluded from the results presented here. Figure 4.15 shows the background of the eighteen participants.



Legends:
English Proficiency: Very Poor (a), Poor (b); Average (c); Good (d); Very Good (e);

Wiki Experience: No (a), Read and navigated on several occasion (b), on many occasions (c), several days per week (d), almost every day (e)

SA (Software Architecture) knowledge: No prior knowledge (a), Taken SA-related course (b)

Architecting Experience (University): No experience (a), Designed SA of a simple system (b), of a sizable system (c), of a few different types of sizable systems (d), of a few different types of complex systems (e)

Architecting Experience (Industry): No experience (a), Less than or 1 year of experience (b), Less than or 3 years of experience (c), Less than or 5 years of experience (d), More than 5 years of experience (e)

Knowledge on AD: No prior exposure (a), Learnt generally in course (b), Read ADs (c), Read and made use of ADs in tasks (d); Written ADs (e)

Familiarity with the type of system described by AD: No knowledge or experience (a), Read (b), Documented (c), Architected (d), Maintained (e) similar type of system

Figure 4.15: Participants' Background

Participants were confident with their proficiency in English (with most of them rated good or very good, and four rated average). All participants had some experience reading and navigating in wiki environment. All the participants had taken course(s) related to SA. Many of them did not have experience in architecting software systems in the industry. Only two participants had no experience in architecting software system during university study. These two also did not have industry experience on this. In terms of knowledge on AD, three have had no prior exposure, and the rest had either learnt in course, read, read and made use, or written AD(s).

In short, generally the participants had the necessary background (good command of English, wiki experience, exposure to SA and AD, some experience with architecting software system) needed for the user evaluation study. Although most of the participants did not have prior knowledge or experience with the type of system described by the AD, this was not a concern for this study. On prior experience with tool that captures and provides visualisation of exploration paths, only one participant indicated having used a tool that captures users' movement. This in a way shows the novelty of KaitoroCap in these aspects.

Figure 4.16 shows the user evaluation of KaitoroCap's features for exploration path capture, tree-view's suitability to visually represent paths, path search and KaitoroCap's overall usability. Participants agreed or strongly agreed that the prototype is useful, effective, and easy to use (with two undecided) for path capture. The tree-view visualisation was rated only slightly little less positively with a small number undecided on its usefulness, representativeness, and ease of interaction. One disagreed with the ease of interaction with tree-view paths. The search path feature had more spread in perception with one each undecided and disagreed, but results were still very positive. Overall usability of KaitoroCap across all features was also assessed. All participants agreed or strongly agreed with the characteristics, with a small of them undecided.

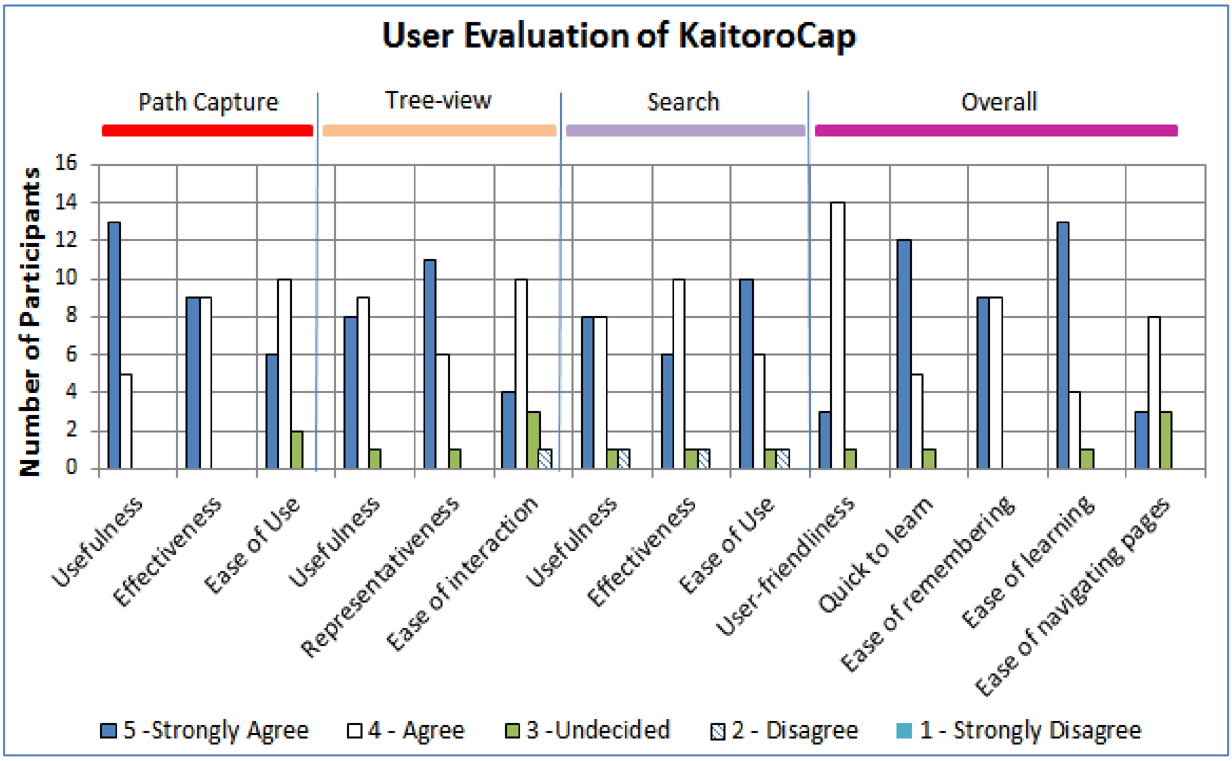


Figure 4.16: User Evaluation of KaitoroCap

Figure 4.17 shows the participants perceptions of AD and the usefulness of navigation paths. Participants agreed or strongly agreed with the ease of understanding the language the AD is written in, and had no problem in understanding the domain concepts in the AD. Only three of the eighteen participants were either undecided or disagreed on these aspects. This showed the suitability of the chosen AD and the recruited participants for this study.

Participants were positive with the usefulness of capturing own navigation paths, with only one of the seventeen who responded on this disagreed. All eighteen participants (including the one who did not respond on whether he or she agreed with the usefulness of capturing own path) gave the reasons for their perceptions on the usefulness of capturing own navigation paths. The reasons include reusing of paths for easier searching of information (retrace own exploration steps to see only the parts that had been viewed) saves time, remembering path taken, capturing history and reason of arriving at the useful information, for review purposes, insight on own use of documentation (such as most attractive information) for self-improvement, and foreseeing users with same searching pattern. Another one indicated that if he or she became familiar with the document, he or she would proceed directly to the information that is most relevant.

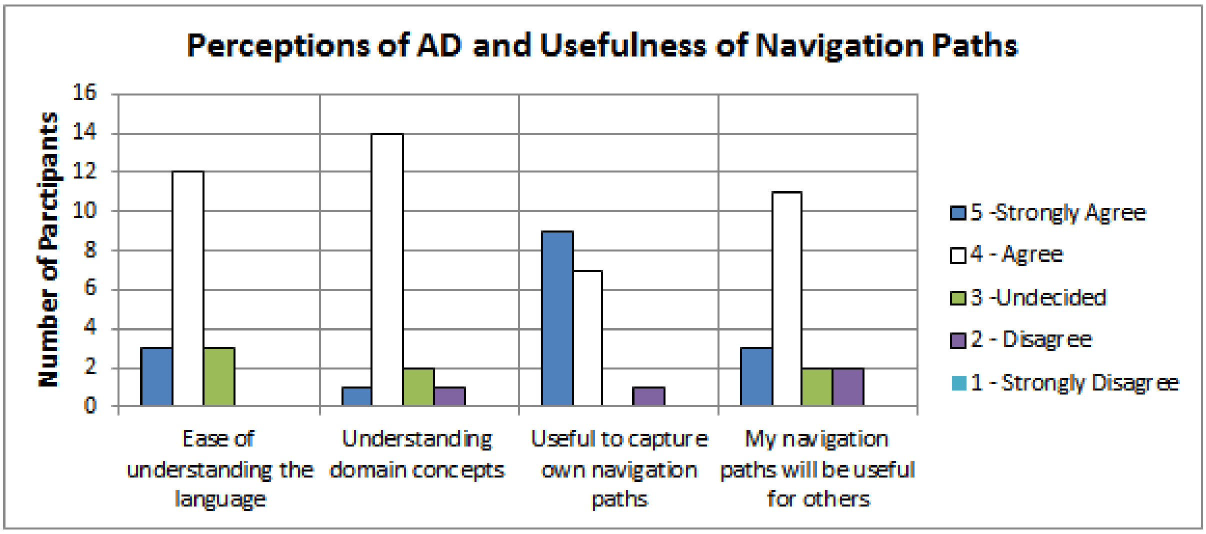


Figure 4.17: Perceptions on AD and Usefulness of Navigation or Exploration Paths

Most of the participants thought their navigation paths would be useful for others, with two each undecided and disagreed. When probed for the reasons, four of the undecided or disagreed did not respond. The reasons of most of the other fourteen participants were related to reusing of paths for easier searching of information (see only the parts that had been viewed) saves time. Other reasons quoted were: ratings showed importance of page for finding information needed, sharing with colleagues, other could learn the path, suggesting reading paths and document rewriting, common problems could be identified and documentation for next project could be improved, for comparison purpose, paths could be combined for identifying patterns of exploration.

In short, the participants were positive with the usefulness of paths regardless of own paths or others' paths. The main benefits they anticipated were paths could be retraced for easier searching of information, and accessing only the information that had been viewed in paths saves time. While we think that *chunk* found from ‘commonality’ in usage data (such as navigation paths or annotation data) could assist in finding needed information, the participants were enthusiastic with even the basic level, which is the paths in supporting information finding.

Participants were also queried on improvements. On suggestion for improving capturing of navigation paths, four participants stated the current capturing features were good, four indicated that they have no suggestion, seven did not respond, one each suggested rating path, capturing time spent on pages, and capturing search queries and text copied and pasted. On suggestion for improving the display of navigation paths, two participants stated the tree-view representations of paths were good, two indicated they had no suggestion and five did not respond. Main suggestions by remaining participants included minimizing details on paths, showing time spent on path and use more colour. On suggestion to improve the ‘search path’ feature, one stated the search feature was good, two stated they had no suggestion, five did not respond. Other participants suggested widening the scope of search to include tags and comments, keyword assistant for search, and improvement on search fields (for example where to put them and so on).

Overall, results show significant promise. Comments mainly suggested enhancements rather than questioning the core rationale of KaitoroCap.

The following were the main things we found from our observation of the participants during their explorations: Users tendency to forget to press ‘Enter’ key for 'tags' and 'comments' to be inserted; A few participants used Atlassian Confluence’s built-in search function located at the ‘home’ or start page of ‘WCT Software Architecture Document’ space, to search for certain terms when working on the second or third information-seeking task, or both tasks. This could be due to the participants became confident with searching for specific terms to find the information they needed for the second and the third task, after having explored the document during their attempt of the first information-seeking task.

We also noticed some limitations of KaitoroCap: The loading of tree-view of paths were slow as more paths were captured in database and retrieved during search; Pressing ‘enter’ key did not insert comments provided for the last section of a page.

Our interactions with the participants revealed one main issue: some of the participants viewed the 'role' or the ‘perspective’ undertaken by them (for example as an evaluator or developer, and so on) very critical in driving their finding of the information needed. Based on this, we refined the specification of the information-seeking tasks for our subsequent studies.

4.4.3 Threats to Validity

Most of the threats that might have affected the validity of the findings of Study 1 are also applicable to Studies 2 and 3. To avoid repetition, we detail these threats under “Section 13.10 Threats to Validity” in Chapter 13.

4.5 KaitoroCap Version 2

This section describes the improvements made to KaitoroCap following the feedback from the user evaluation study, and our aim to gather more exploration data by recruiting remote participants. The following are the main improvement made to KaitoroCap:

i. Improved path capturing: auto-saving exploration data occurred on a page after certain timeframe of inactivity.

ii. Improved tree-view representation of paths: hide events types, more colours (Figure 4.18); more compact tree-view showing only pages visited and sections expanded via 'read more' feature (Figure 4.19).

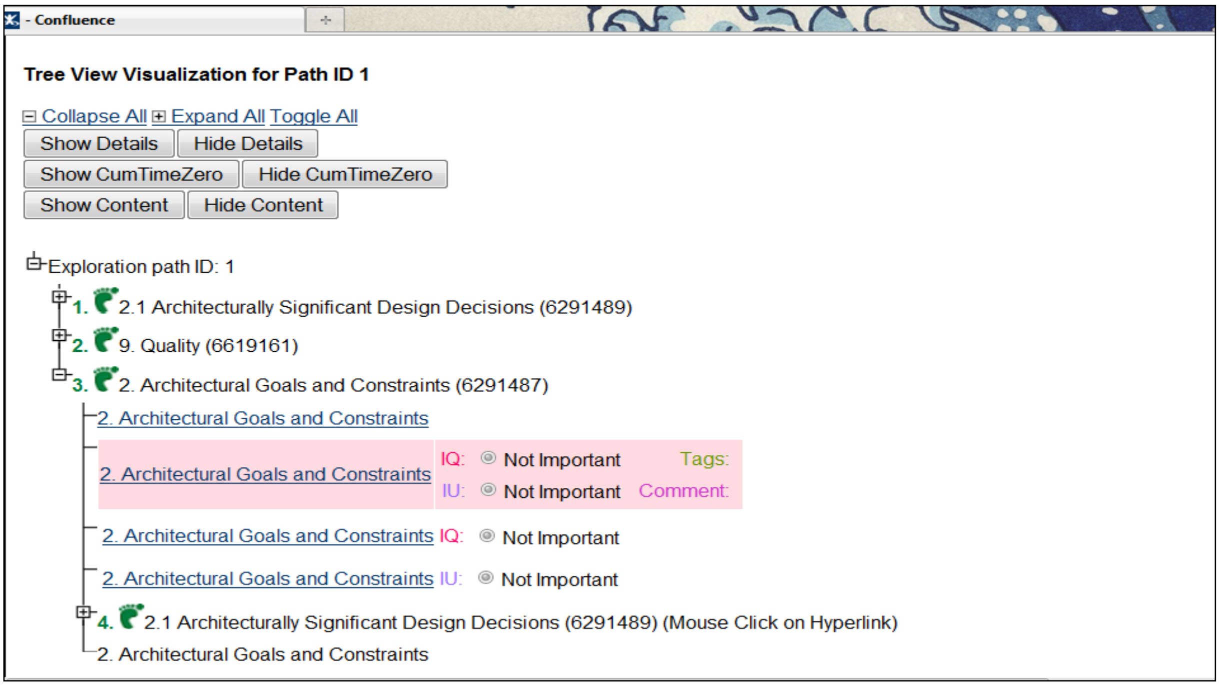
iii. Improved search features: provide searching for paths based on tags, comments, users, average path rating, average answer rating; provide assistance for searching; option to show all paths.

iv. More user-friendly user interface: main interface only exposing path capturing, tree view and search features; inline instruction, explanation and examples to guide user; saving tags and comments when 'lost focus' event occurred instead of having to press 'enter' key; improved ratings fields (Figure 4.20); current active section highlighted with light green background in addition to surrounding border (Figure 4.20).

v. Extra features: enter and save answer or information found any time; answer and path ratings by owner or other users; average answer and path ratings; latest ratings; user's self-assessment of amount of information found for task, own satisfaction level with exploration, expertise in role assumed; capturing entering and leaving sections.

vi. Improved performance in displaying tree-view and searching paths.

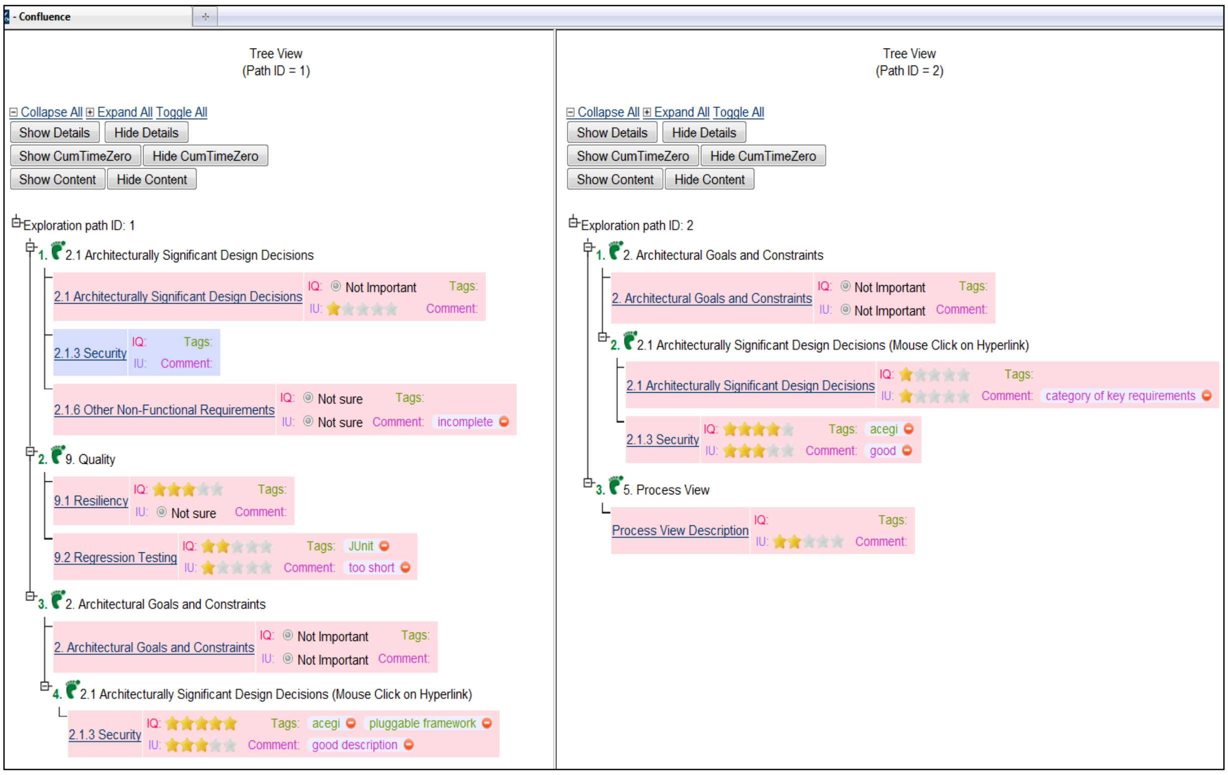
vii. Rectified bugs.



Tree View Visualization for Path ID 1

Exploration path ID: 1
1. 2.1 Architecturally Significant Design Decisions (6291489)
2. 9. Quality (6619161)
3. 2. Architectural Goals and Constraints (6291487)
2. Architectural Goals and Constraints
2. Architectural Goals and Constraints IQ: ○ Not Important Tags:
IU: ○ Not Important Comment:
2. Architectural Goals and Constraints IQ: ○ Not Important
2. Architectural Goals and Constraints IU: ○ Not Important
4. 2.1 Architecturally Significant Design Decisions (6291489) (Mouse Click on Hyperlink)
2. Architectural Goals and Constraints

Figure 4.18: Tree View of An Exploration Path



Tree View
(Path ID = 1)

Exploration path ID: 1

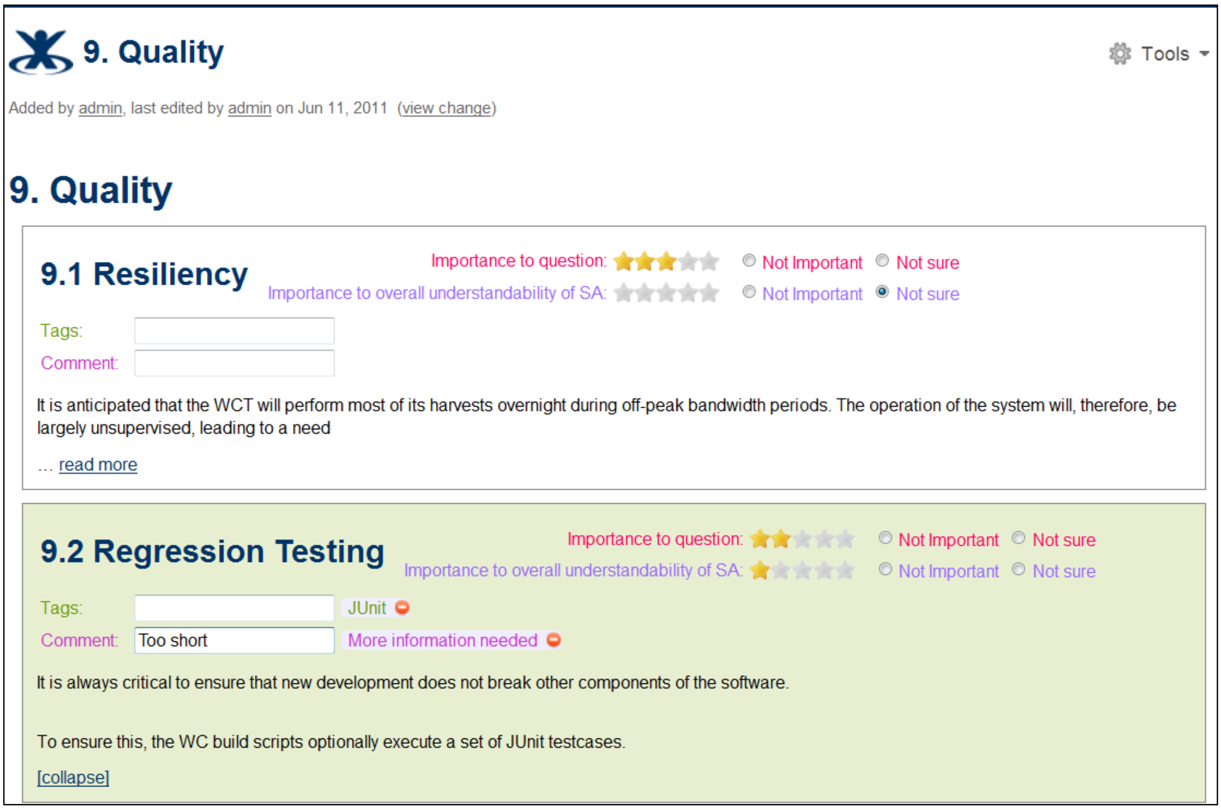
1. 2.1 Architecturally Significant Design Decisions
   - 2.1 Architecturally Significant Design Decisions IQ: ○ Not Important Tags: IU: ★ Comment:
   - 2.1.3 Security IQ: Tags: IU: Comment:
   - 2.1.6 Other Non-Functional Requirements IQ: ○ Not sure Tags: IU: ○ Not sure Comment: incomplete
2. 9. Quality
   - 9.1 Resiliency IQ: ★★★ Tags: IU: ○ Not sure Comment:
   - 9.2 Regression Testing IQ: ★★ Tags: JUnit IU: ★ Comment: too short
3. 2. Architectural Goals and Constraints
   - 2. Architectural Goals and Constraints IQ: ○ Not Important Tags: IU: ○ Not Important Comment:
4. 2.1 Architecturally Significant Design Decisions (Mouse Click on Hyperlink)
   - 2.1.3 Security IQ: ★★★★★ Tags: acegi pluggable framework IU: ★★★ Comment: good description

Tree View
(Path ID = 2)

Exploration path ID: 2

1. 2. Architectural Goals and Constraints
   - 2. Architectural Goals and Constraints IQ: ○ Not Important Tags: IU: ○ Not Important Comment:
2. 2.1 Architecturally Significant Design Decisions (Mouse Click on Hyperlink)
   - 2.1 Architecturally Significant Design Decisions IQ: ★ Tags: IU: ★ Comment: category of key requirements
   - 2.1.3 Security IQ: ★★★★ Tags: acegi IU: ★★★ Comment: good
3. 5. Process View
   - Process View Description IQ: Tags: IU: ★★ Comment:

Figure 4.19: Compact Tree-View of Expanded Exploration Paths



9. Quality

Added by admin, last edited by admin on Jun 11, 2011 (view change)

9. Quality

9.1 Resiliency

Importance to question: ★★★ ○ Not Important ○ Not sure
Importance to overall understandability of SA: ○ Not Important ● Not sure

Tags:
Comment:

It is anticipated that the WCT will perform most of its harvests overnight during off-peak bandwidth periods. The operation of the system will, therefore, be largely unsupervised, leading to a need

… read more

9.2 Regression Testing

Importance to question: ★★ ○ Not Important ○ Not sure
Importance to overall understandability of SA: ★ ○ Not Important ○ Not sure

Tags: JUnit
Comment: Too short More information needed

It is always critical to ensure that new development does not break other components of the software.

To ensure this, the WC build scripts optionally execute a set of JUnit testcases.

[collapse]

Figure 4.20: Improved Ratings Fields, and Current Active Section

Refer to ‘Help file’ on how to use KaitoroCap Version 2 in Appendix “12.1 Data Collection Instruments (Study 3)” for further details on KaitoroCap Version 2.

4.6 **Related Work**

In this section we compare KaitoroCap with other AKM tools reviewed in Chapter 2.

KaitoroCap supports the management of ADs, the main medium where AK resides (Jansen et al., 2009). KaitoroCap supports the management of explicit AK in the form of documented AK expressed in natural language or images in ADs.

KaitoroCap supports codification of AK. In addition, the exploration paths captured by KaitoroCap supports the transfer of a special type of tacit knowledge, namely the knowledge on how people explore existing AK in ADs. This knowledge can be regarded as procedural knowledge (Nickols, 2000), which is explained in the next paragraph.

Cognitive psychologists classified knowledge mainly into declarative and procedural (Nickols, 2000). Declarative knowledge comprises descriptions of facts and things or of methods and procedures. It is knowledge that can be and has been articulated; an instance of explicit knowledge. There are different views on Procedural knowledge. In knowing-is-in-the doing view (Anderson, 1976, 1993, 1995), procedural knowledge manifests itself in the doing of something, and is reflected in motor or manual skills and in cognitive or mental skills (Nickols, 2000). For examples, playing piano, riding bicycle, reading customers’ faces and moods, thinking, reasoning and making decisions. Another view of procedural knowledge is it is knowledge about how to do something (Nickols, 2000). This view accepts a description of the steps of a task or procedure as procedural knowledge. This makes procedural knowledge no difference from declarative knowledge of tasks or procedures. Nevertheless, “a description of an act is not the act just as the map is not the territory” (Nickols, 2000). Therefore declarative knowledge should be reserved to “describing” and procedural knowledge reserved for “doing”. Following that, all declarative knowledge is explicit and all procedural knowledge is tacit. Being tacit, procedural knowledge cannot be articulated but can be communicated or transferred. As an example, the ability to recognize a face cannot be verbalized but can be developed through exposure to pictures of the face.

We employed a number of technologies deemed valuable for building AKM tools (Liang & Avgeriou, 2009) in the development of KaitoroCap: wiki, voting and ranking, plug-ins and Web 2.0. KaitoroCap is a plug-in for Atlassian Confluence Enterprise Wiki (Atlassian, 2013a). Most of text-based architecture documentation and communication are in wiki (Liang et al., 2009). Wiki environment is also a more suitable lightweight platform than web portal for sharing AK (Farenhorst & Van Vliet, 2009). In terms of voting and ranking, instead of soliciting community’s voting or ranking of members’ personal information to create more reliable personalisation strategy (Liang & Avgeriou, 2009), KaitoroCap enables users to rate each section of a document in terms of their importance for certain information-seeking task and for understanding the described SA. KaitoroCap also provides the means to tag (related to Web 2.0) each section of a document, and the functionality to search exploration paths comprising document’s sections tagged with the searched terms.

With reference to the AKM Use-Case Model (Liang & Avgeriou, 2009) in Section "2.2.3.3 Models for AKM), the use-cases supported by KaitoroCap is similar to those of SEI-ADWiki (Liang & Avgeriou, 2009), another general wiki used for documenting SA (Bachmann & Merson, 2005). In particular, KaitoroCap supports the following use-cases: view AK and their relationships (for example mapping between architecture views as stated in the wiki pages of an AD), trace AK (for example trace between documented AK in wiki pages through hyperlinked structures of wiki pages), share AK (share the content of ADs captured in wiki pages), add or store AK (in terms of architectural artefacts in wiki pages), edit AK (by changing or removing the content of AD in wiki pages), search or retrieve (documented AK using keywords). KaitoroCap also supports the creation of traceability between different categories of AK (Liang & Avgeriou, 2009) via hyperlinks.

On top of those use-cases, KaitoroCap supports a new type of sharing activity, in particular the sharing of the ways consumers explore architectural information in ADs (as captured by their exploration paths) and the sharing of consumers' perceptions of sections of the documents (given in the annotation form of tags, comments and ratings). By making available consumers' exploration paths and annotations, KaitoroCap provide alternative support to other consumers (especially novices) in searching or retrieving related AK. Chunks found in our research can also be made available in KaitoroCap for the same purpose.

None of the existing AKM tools captures users' exploration of architectural information in ADs as KaitoroCap does. AKM tools having features that show some similar ideas are: 1) ADDSS tool which allows users to replay design decision tree (Capilla et al., 2006) (Tang et al., 2009), whereas KaitoroCap allows the retracing of users' exploration paths through different sections of a document. 2) Knowledge Architect a tool suite to support collaborative architecting process (Liang et al., 2010), (Liang et al., 2009), (Jansen et al., 2009) (Tang et al., 2009) has a tool called Knowledge Explorer which supports searching and visualisation of AK instances and their relationships, and navigation to related AK instances via traceability links. KaitoroCap supports the searching and visualisation of exploration paths. A exploration path shows the sequence of sections visitation during an exploration session. The contents of visited sections are dynamically embedded into a path making the path a re-structuring of the AD. A path supports future navigation to the same collection of sections.

4.7 Summary

This chapter presents the design, implementation and usage examples of KaitoroCap; the user evaluation study (Study 1) conducted to gain feedback on KaitoroCap's features; the improvements made to KaitoroCap; and a comparison of KaitoroCap with other AKM tools reviewed in Chapter 2. The user evaluation of KaitoroCap showed promising results in terms of capturing user's exploration paths, the tree-view visualisation and searching of exploration paths.

The next chapter presents Study 2 (Manual Approach Study).

---- Blank ----

5 Study 2 - Exploration of Software Architecture Documents (Manual Approach Study)

This chapter describes Study 2 which we conducted to investigate the chunking of architectural information based on the ADs' usage data (including ratings) by people with a strong SA background.

This chapter presents the main differences between Study 2, and Studies 1 and 3; the study design; participant recruitment; the administration of the study; and grouping of participants' responses. This chapter also provides the foundation for the concept of 'chunking' of architectural information. This includes the definition of a 'chunk', detailed descriptions of the factors explored for chunk identification, the benchmarking of discovered chunks against oracle sets as a way to evaluate the chunks and the chunk-identification factors based on standard recall and precision measures; and all the relevant terms that we used in our discussion of the results of chunks identification. In addition, this chapter explains the criteria used to assess the 'goodness' of chunks found in our studies, how best chunk and the best chunk-identification factor are determined, interpretation of chunks, and determination of commonly-missed oracle set's sections and common false sections, and the process of the construction of oracle set.

This is a very important chapter for Studies 2 and 3. It provides the foundation to understand all the subsequent chapters, and should be read prior to reading those chapters. A discussion of threats to validity of findings can be found in Chapter 13, where the overall results for Studies 2 and 3 are summarised.

5.1 Overview of Study 2

The main differences between Study 2, and Studies 1 and 3 are:

a) The KaitoroCap on-line tool was not used and the questionnaire was administered off-line in Study 2. We wanted the participants to be more focused on exploring the ADs without the extra requirement of learning and using a new tool for navigation path capture. To make the approach more consistent for the participants, and to reduce the burden of accessing the questionnaire separately online, we bundled the questionnaire together with the rest of the data collection instruments. Consequently, in Study 2, the participants explored ADs in Microsoft Word format or printed versions of these documents, instead of wiki pages in KaitoroCap. No navigation path was captured. Participants manually saved their responses on the data collection instruments given to them as Word documents or annotated them on the printed versions of these instruments. Participants returned their completed data collection instruments by uploading to an online dropbox setup especially for this study or by sending them to the researcher's mailbox. There is only one questionnaire in Study 2 which is a combination of the pre- and post-questionnaires of Study 3, but without the questions related to educational background, wiki, tool similar to KaitoroCap, KaitoroCap's features, navigation paths and patterns. Study 2 is effectively the *manual* version of Study 3 (the *online* version) in the sense that the study design is similar to Study 3 but the collection of data was off-line. In Study 3, participants explored the ADs in the form of wiki pages in KaitoroCap. KaitoroCap was used to capture the navigation paths as well as all the participants' responses. The questionnaires were also administered online using SurveyMonkey (SurveyMonkey, 2012), an online survey tool.

b) Use of more stringent selection criteria to recruit the participants. Apart from the willingness and ability to commit the required time (approximately 1 hour 15 minutes) and effort to participate in the study, the potential participants were required to either (i) have had at least 2 years of industry experience related to SA; (ii) to have taught a SA course, or to have provided training on SA. This allowed us to collect usage data of ADs from industry and academic professionals who have a stronger SA background than in Studies 1 and 3. Collectively the participants recruited for Study 2 were considered to be more 'expert' in SA than those recruited in Study 1 and 3 as a whole. For Study 1 the participants were only required to have some SA background. This could be having taken a course related to SA, instead of having industry or teaching experience in SA. As a result, collectively the participants recruited for Study 1 were considered to be predominantly 'novices' in SA. The participants who took part in Study 3 were a mixture of 'novices' and 'experts' in SA as we could not recruit an adequate number of either group for this study.

This study was conducted as a quasi-experiment (Easterbrook et al., 2008). The reason was that we did not have full control in assigning the study participants (i.e. subjects) randomly to the ADs and the information-seeking tasks (the treatments). The recruitment of highly specialized participants was difficult and we could not predict the actual number of participants that would participate. To ensure that we obtained the target number of participants we wanted for at least one of the documents we used, we allocated the participants we recruited to one document first. Subsequent participants that we recruited were then assigned to the second document.

5.2 Study Design

To collect a substantial amount of actual usage data for the sample ADs we need to define a reasonable scope for the exploratory process involved in the consumers' usage of the documents. This involved selecting specific ADs to be explored by the consumers and predefining the information-seeking tasks for which usage data would be collected.

This section presents the design of our study. It explains our choices of the ADs, the information-seeking tasks and roles required of participants, and our rationale behind these choices. In addition, it describes the data that we collected, the instruments used for this data collection and the preparation of the documents.

5.2.1 Choice of Documents

We used two ADs in this study. This allowed us to make some comparisons of the findings from using the two documents.

We considered the following characteristics of a document when deciding whether to use it in this study: length, complexity, quality, availability and, mixture of textual and graphical representations. These should suit reasonably well the exploratory information-seeking tasks that were involved in this study. A suitable document length is critical for this study. If the document is too short the participants may just read the whole document instead of exercising some selective exploration of the document. If the document is too long the participants may not be able to find the needed information within the estimated timeframe. The chosen documents were of reasonable technical complexity and details. We chose documents that described systems that are not too specialized, to cater for a diverse background of the target participants. In addition, the chosen documents should have reasonable quality in terms of language, organisation and legibility. The chosen documents were available on the Internet and we had obtained permissions to use them in this research. They also have a mixture of text and graphics.

To ensure the object for exploration was realistic, we chose existing ADs describing real systems in use. The first AD we used is a 24-page document defining the architecture of a web curator system (WCT) that manages and harvests digital web content for future preservation (National Library of New Zealand, 2006). The second AD is 21 pages and describes the SA of the Aperi Storage Manager (ASM), an open source storage management platform (Slupesky & Singleton, 2006).

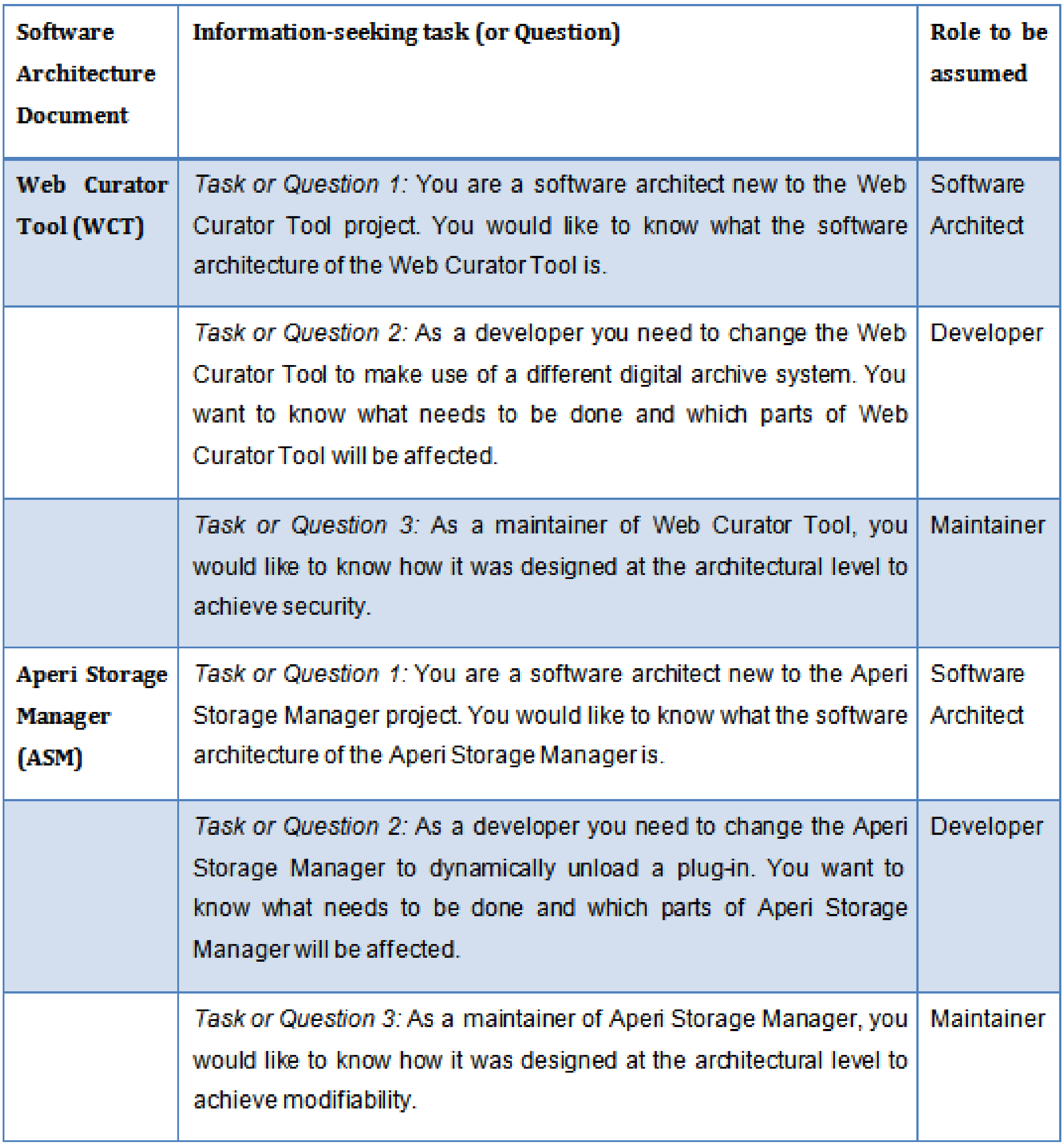
5.2.2 Determination of Information-seeking Tasks and Roles

To guide the document exploration process, one thing was imposed on the participants: the information-seeking tasks (or 'questions' for which the participants sought the answers), which drive the exploration processes. The information-seeking tasks for the study participants were phrased in terms of a scenario together with the role to be assumed. For example for the first information-seeking task: "You are a software architect new to the X project. You would like to know what the software architecture of X is". Taking on the same role helps participants to view a task with a similar mindset. The tasks were designed to suit the corresponding roles. The roles were chosen because they are related to the main stakeholders of ADs or Architectural Knowledge (Kruchten et al., 2006).

There were three information-seeking tasks for each document and they were similar for the two documents. Table 5.1 contains the details of these tasks.The first task was to find out what the SA of the system described by the given document is by assuming the role of a software architect new to the software project of the described system. The participants' usage data of the documents gathered from here were used to identify chunks of architectural information related to the overall software architectures of the described systems. We are particularly interested to see the common set of architectural information needed to obtain an overview of the SA of a system.

The second task was to find out how to change a certain part of the system and in doing so which parts of the system will be affected. The role to be assumed was a developer. The identification of chunks from here allowed us to study the common set of architectural information needed for specific tasks (such as making a change to the software system). We wanted to see in general what AD users look for to perform this type of task and to assess the impact of the change. As such we did not state specifically what 'parts' and 'affected' mean. 'Parts' can refer to anything of the system that might be affected: system; subsystems; configuration; no change, etc. "Affected" could mean code changes, quality or other things.

The third task was to find out how the system was designed at an architectural level to achieve a certain quality attribute (such as security and modifiability), by assuming the role of a system maintainer. The chunks of architectural information found here gave some insights into the common set of information needed to address cross-cutting concerns which span the entire system.



Software Architecture Document	Information-seeking task (or Question)	Role to be assumed
Web Curator Tool (WCT)	*Task or Question 1:* You are a software architect new to the Web Curator Tool project. You would like to know what the software architecture of the Web Curator Tool is.	Software Architect
	Task or Question 2: As a developer you need to change the Web Curator Tool to make use of a different digital archive system. You want to know what needs to be done and which parts of Web Curator Tool will be affected.	Developer
	Task or Question 3: As a maintainer of Web Curator Tool, you would like to know how it was designed at the architectural level to achieve security.	Maintainer
Aperi Storage Manager (ASM)	*Task or Question 1:* You are a software architect new to the Aperi Storage Manager project. You would like to know what the software architecture of the Aperi Storage Manager is.	Software Architect
	Task or Question 2: As a developer you need to change the Aperi Storage Manager to dynamically unload a plug-in. You want to know what needs to be done and which parts of Aperi Storage Manager will be affected.	Developer
	Task or Question 3: As a maintainer of Aperi Storage Manager, you would like to know how it was designed at the architectural level to achieve modifiability.	Maintainer

Table 5.1: The Software Architecture Documents and the Information-seeking tasks

5.2.3 Data Collected

For each task assigned, a participant explored the given AD to find his or her required answers or the information he or she needed for the task. The participant was asked to state the time the task was started (start time) and the keywords or terms searched for in relation to the task, at the beginning of the task. As the participant looked for an answer, he or she was asked to highlight information in the document relevant to the assigned task. For each section visited for the task, the participant was asked to provide two ratings (in terms of its importance to the task and in terms of its importance to his or her overall understandability of the described SA). The participant also tagged and commented visited sections, although this was not mandatory. A tag is a user-defined keyword reflecting a section's content. A comment is a participant's more elaborated opinion on a section.

Each participant was asked to provide his or her answer to the task in bullet-point form, as comprehensively as possible. He or she was asked to also state in which section of the document each bullet-point answer was found, and whether it was found by looking at the section's title, reading the section or it was not obvious from the document but came from his or her past experience and knowledge. After completing the answer, the participant recorded his or her finishing time. Then a participant suggested his or her view of the "best" sequence of reading for those sections relevant to his or her assigned task. This sequence, in his or her opinion, would support a better understanding of the described SA. The participant did this by assigning the section that he or she thought should be read first with number '1', the second section to be read with number '2' and so on.

After completing all the tasks, the participant completed a questionnaire. The main content of the questionnaire is explained in the next section. The estimated time of participation for each participant was 1 hour and 15 minutes. Each participant was given one of the ADs and two of the three tasks for the given document. Two tasks were given instead of three as in Study 1 to allow the participants better focus on a smaller number of tasks. The two tasks were given in a different sequence for each alternate participant. This resulted in 6 sets of tasks (or 6 sets of questions) for each document: Set A (Task 1, Task 2), Set B (Task 2, Task 1), Set C (Task 1, Task 3), Set D (Task 3, Task 1), Set E (Task 2, Task 3), Set F (Task 3, Task 2). The rationale was to balance-off the influence of the familiarity with the document acquired during the first task, on the second task.

5.2.4 Data Collection Instruments

The data collection instruments (Appendix 5.1) comprised of:

- Participant Information Sheets (PISs) that detailed the terms and condition of participation in the study, and Consent Forms (CFs) for participants to indicate consent to take part in the study.
- The instruction on the tasks to be completed for the study, the information-seeking tasks (or questions) to be attempted, and how to provide ratings, tags, comments and suggested reading sequence as well as highlight relevant information on the given AD.
- A form for each task for the participants to provide the start time, stop time, keywords, answer in bullet-point form, where and how each of the answer point was found.
- Example on how to fill in the answer form.
- One of the ADs for the participants to explore and provide ratings, tags, comments, suggested reading sequences and highlight relevant information. These documents are instrumented with annotation fields (refer to next section) to capture this data.
- A questionnaire that contained multiple-choice, structured Likert-scale and un-structured free-text questions. The questionnaire included questions on occupational background related to SA, industry experience related to SA, experience with ADs, exposure to the system described by the document,

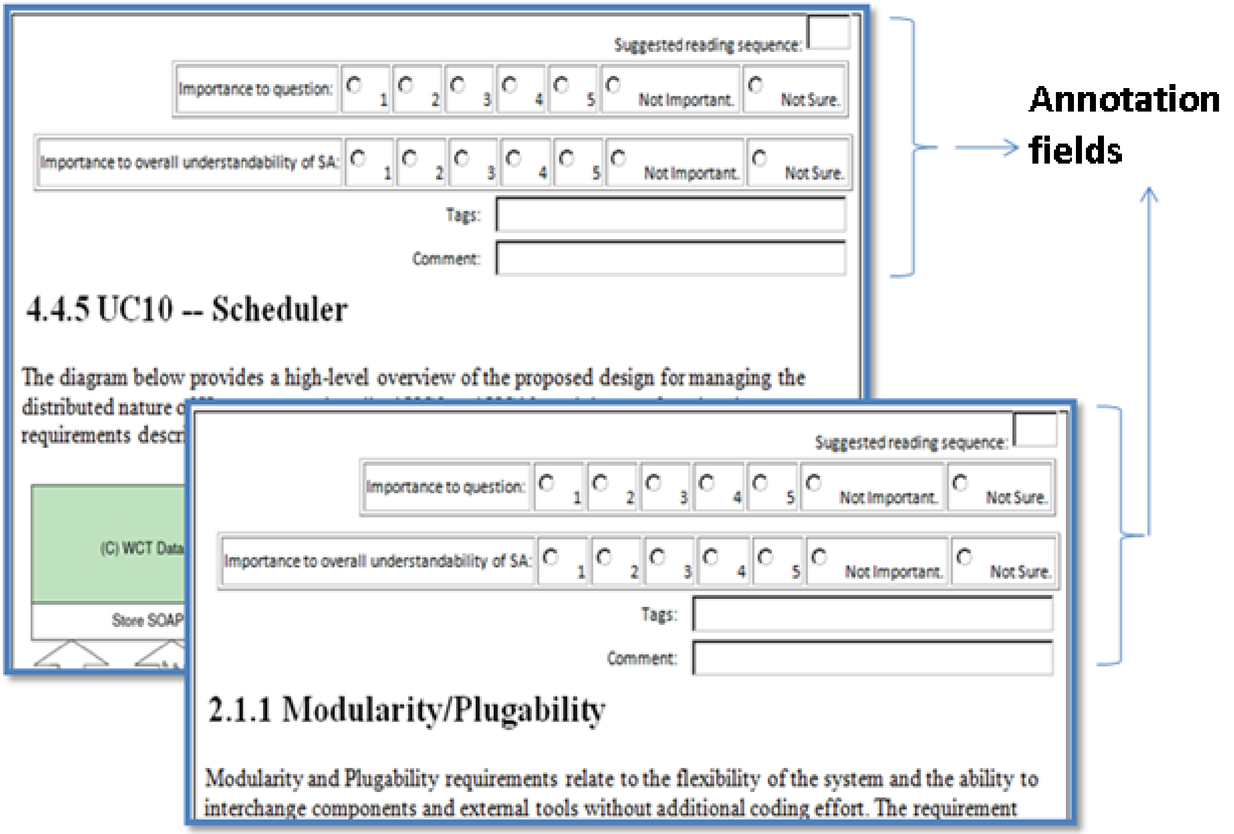
perception of the content of the given document, ways the given document supported and hindered the understanding of the described SA, exploration characteristics, and perception of the usage of textual descriptions and diagrams as well as their usefulness in supporting understanding.

The data collection instruments in this study were adapted from Study 1.

5.2.5 Preparation of Documents

We wanted to gather document consumers' perceptions of the content of the ADs we used in this study. For this, we inserted annotation fields to capture tags, comments, ratings and suggested reading sequence, to the beginning of the sections of the documents following their original organisation. Each section that could be annotated was also enclosed by a border to visually distinguish it from others. Figure 5.1 shows examples of two annotatable sections of one of the documents.

The annotation fields and border were inserted automatically using a script. Sub-sections that contained a substantial amount of information or contained distinct information by themselves were also instrumented with the annotation fields and the border (but with no change to order or organisation).To gather perceptions of different types of representation of architectural information, diagrams were instrumented with the annotation fields separately from surrounding text. The first document (WCT) contained 47 annotatable sections, with 6 containing diagrams (one containing two closely-related diagrams). The second document (ASM) had 62 such sections, 7 containing diagrams.



Suggested reading sequence: □

Importance to question: ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ Not Important. ○ Not Sure.

Importance to overall understandability of SA: ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ Not Important. ○ Not Sure.

Tags:

Comment:

4.4.5 UC10 -- Scheduler

The diagram below provides a high-level overview of the proposed design for managing the distributed nature o
requirements descr

Suggested reading sequence: □

Importance to question: ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ Not Important. ○ Not Sure.

Importance to overall understandability of SA: ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ Not Important. ○ Not Sure.

Tags:

Comment:

2.1.1 Modularity/Plugability

Modularity and Plugability requirements relate to the flexibility of the system and the ability to interchange components and external tools without additional coding effort. The requirement

Figure 5.1: Examples of Annotatable Sections of One of the Documents

5.3 Participant Recruitment

This study sought participants with stronger SA backgrounds when compared to Study 1. However, the exploratory nature of the information-seeking tasks still entailed a considerable amount of time and effort (1 hour 15 minutes) to complete the tasks. Consequently, to be realistic in the recruitment of the participants, the following selection criteria were used: one of (i) have had at least 2 years of industry experience related to SA, (ii) to have taught a SA course or to have provided training on SA; and willingness and ability to commit the required time and effort. We believe that these criteria are reasonable for this study, since the tasks of seeking information by reading specialized documents are common tasks that can be performed reasonably well with the right background and the availability of time.

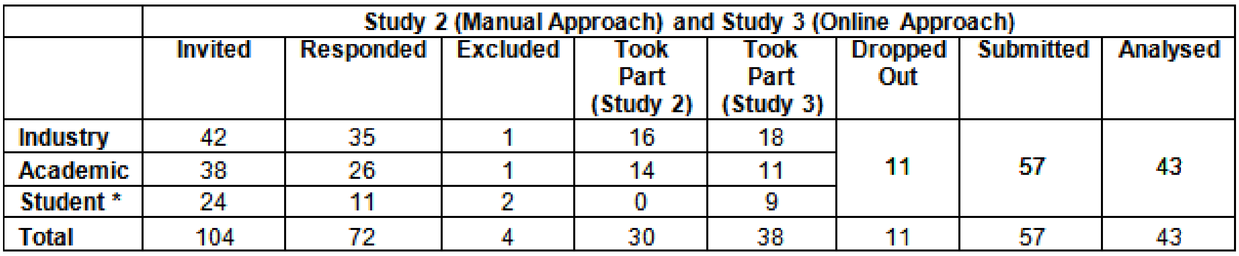
The selection criteria discouraged the use of random sampling as the targeted groups often do not respond to invitations from unfamiliar sources. Therefore non-probabilistic sampling techniques i.e. convenience and snowball sampling were used to invite potential participants. The former involves recruiting participants who meet the selection criteria and are available and willing to participate in the study (Barbara & Shari Lawrence, 2002). The latter refers to asking participants of the study to recommend other potential respondents (Barbara & Shari Lawrence, 2002).

We conducted Study 2 (Manual Approach Study) and Study 3 (Online Approach Study) at about the same time. For those we knew having stronger SA background (with at least 2 years of industry, teaching or training experience related to SA), we invited these ‘experts’ to take part in Study 2. For those we knew having less than 2 years of industry, teaching or training experience related to SA, we invited to take part in Study 3. For those we did not have this information, we asked the interested candidates to respond with their SA background to decide whether to assign Study 2 or Study 3 to them.

In our following explanation of the recruitment of participants for both studies (including the explanation in Chapter 12), we classify students who had industry experience in SA as industry participants and students with teaching experience in SA as academic participants. For Study 3, we also targeted students who had taken or were taking course related to SA, and those having research experience related to SA. We classify this group of students as student participants.

The recruitment of the participants for Studies 2 and 3 was undertaken by sending email invitations to potential participants using our personal and professional contacts, and their referrals. We also requested help from our acquaintances to post recruitment advertisement on related Yahoo and FaceBook groups. Refer to Chapter 12 (that describes Study 3) for our other means of recruiting students.

Table 5.2 shows the figures of our recruitment of participants for Studies 2 and 3. We invited 42 industry practitioners for the two studies. Out of the 35 who responded 1 were excluded for not having SA background, 16 of them were recruited to take part in Study 2 and the other 18 were recruited for Study 3. We invited 38 academics for the two studies. Out of the 26 who responded, 1 could not proceed further. Fourteen of the respondents were recruited to take part in Study 2 and the other 11 were recruited for Study 3. We invited 24 students (mostly PhD students) for Study 3. Eleven students responded out of which 2 were excluded as they did not have SA background, and 9 took part in the Study 3.



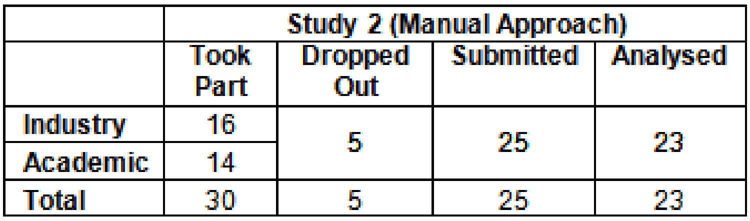
	Study 2 (Manual Approach) and Study 3 (Online Approach)							
	Invited	Responded	Excluded	Took Part (Study 2)	Took Part (Study 3)	Dropped Out	Submitted	Analysed
Industry	42	35	1	16	18	11	57	43
Academic	38	26	1	14	11			
Student *	24	11	2	0	9			
Total	104	72	4	30	38	11	57	43

* Refer to students who had taken or were taking course related to software architecture, and those doing research related to software architecture, recruited for Study 3.

Table 5.2: Recruitment of Participants (Studies 2 and 3)

As a summary, we invited 80 potential industry and academic participants for Studies 2 and 3, and 24 students for Study 3. Seventy-two responded, out of which 4 were excluded, 68 took part in either one of the studies, 11 dropped out half-way through, 57 submitted their responses out of which 43 were included in our analysis.

For Study 2 (Table 5.3), in total 30 participants (16 industry practitioners and 14 academics) took part with 5 drop-outs, and 25 submitted responses out of which 23 were analysed. One respondent was excluded from the analysis as the responses given were too vague to make any useful interpretation. Another respondent had worked as a Software Engineer for 2 years but further inspection of the participant’s response indicated that he or she did not fulfil our selection criteria of at least 2 years of experience related to SA. Therefore this respondent was also excluded from the analysis. Although we did not specify the minimum number of years of SA teaching experience, all the academic participants who took part in Study 2 have at least 2 years of experience in this aspect. The details of the participants for Study 3, is given in Chapter 12.



	Study 2 (Manual Approach)			
	Took Part	Dropped Out	Submitted	Analysed
Industry	16	5	25	23
Academic	14			
Total	30	5	25	23

Table 5.3: Participants of Study 2 (Manual Approach Study)

5.4 Administration of the Study

A set of data collection instruments was given to those who consented to take part in the study. For physically accessible participants, a face-to-face meeting was conducted with them to explain the tasks to be completed and to clarify any question. For more distant participants, clarification was either done through email or online chat discussion. Participants returned their responses by uploading to an online dropbox or by sending them to the researcher’s mailbox.

5.5 Industry Practitioners Versus Academic Professionals

Studies have shown considerable differences between industry and academics in their perception of software architecture and reusable assets (Bosch, 1999). Consequently, for our investigation into the chunking of architectural information in ADs, the participants' responses were segregated into either an industry practitioner (I) group or an academic (A) group. Analyses were then performed on the I and A Groups, in addition to analysing the participants' responses as the whole combined (C) group. This enables us to investigate whether there is any different emphasis on architectural information in ADs between the SA academics and industry practitioners. Any differences between the two groups can serve as motivation for further study to reduce the gaps between academic research and industrial practice in terms of SA documentation.

The assignment of group was made based on whether the participants' SA experiences were from the industry, or from the academic teaching or training. For those who had experience from both, the length of experience decided to which group their responses were assigned.

5.6 Chunking of Architectural Information

This section reiterates the meaning of a 'chunk' as used in this research. It also describes the various factors we explored to identify possible chunks based on the participants' usage of the ADs while performing the information-seeking tasks.

5.6.1 Definition of Chunk

Recall that in Chapter 1, a chunk is defined in this research as below:

A chunk for an information-seeking task performed on a software architecture document is a collection of related pieces of architectural information needed for the task by the majority of a group of users. Specifically it comprises section(s) of a software architecture document needed for the task by the majority of a group of users.

A section can be either paragraph(s) of text, table(s), image(s) or hyperlink(s) or combinations of these. For example a chunk might consist of the following sections of an AD: section 'Logical Components' which provides textual explanation on the logical components and Section "Logical Components Diagram" which contains an image of this and both of these sections contain information required by the majority of a group for a particular information-seeking task.

To start with, we experimented with 'document section' as the level of granularity for chunk elements. This approach is similar to existing work that studied the relevance of the elements of ADs to perceived stakeholders and their concerns (Koning & van Vliet, 2006).

5.6.2 Factors Used to Identify Chunks

We developed and explored several factors to identify possible chunks from participants' usage of an AD during their information-seeking tasks. These factors made use of the participants' responses on where the needed information (answers) can be found, their highlighting of information in the document they think relevant to the task and their ratings of sections visited in terms of importance to the assigned task. These three aspects of usage data were used for the identification of chunks as they carried the participants' explicit indications of which sections of the document they needed for the task. Other data (such as the other rating) was not included in our analysis to find chunks.

Identification of chunks was performed on each group's collective responses. The groups are the industry practitioner (I) group, the academic (A) group and the total combined group (C). The rationale for having the two sub-groups (I and A) for chunk identification is given in Section 6.5.

There are 3 types of chunk-identification factors: basic (Factor A, H and R3), composite (Factor A|H|R3) and average rating (Factor AveR3 and AveR3F). The rest of this section explains the factors in detail together with specific examples.

5.6.2.1 Section Inclusion Formula

We used the preference of the majority of the participants for a section to decide whether to include the section in the chunk found using a factor. The preference of the majority of the participants for a section is determined by using frequency count as the mechanism to aggregate the participants' responses in the two categories of indications ('needed' or 'not needed' for the task). We take the minimum stance on what we mean by majority, which is as long as it is more than half of the total number of participants involved. The formula we used to determine the minimum number of participants in order to be considered as majority is as below:

minimum majority (minM) = floor (T / 2) + 1

where T is the number of participants involved and floor(y) = [y] is the largest integer not greater than y. Depending on which factor, the term 'involved' can refer either to all the participants in a group or only those in the group who provided the responses.

Consequently, if the 'needed' category received a minimum majority count, then the section is considered 'needed' by the group for the task and the section is included in the respective chunk.

For each factor, we defined section inclusion criteria to decide whether to include a section of the document in the chunk found using the factor. Using the above notion of preference of the majority, we defined the **section inclusion formula:** a document's section is included in a chunk for a factor if more than or equal to m% of the participants involved met the section inclusion criteria for that factor, where m is ((minM / T) * 100).

This formula ensures that a section is included in a chunk found using a particular factor only if more than half the participants involved met the section inclusion criteria for that factor. For example, for an I group with 4 participants involved, a section is included in a chunk for the corresponding factor if more than or equal to 75% (i.e. (((floor (4 / 2) + 1) / 4) * 100)) of the industry participants meet the section inclusion criteria for that factor (i.e. at least 3 out of 4). For the combined group with 8 participants involved, a section is included in the chunk for this group if more than or equal to 62.5% (i.e. at least 5 out of 8) of the participants meet the section inclusion criteria.

In our explanation of the chunk-identification factors from this point onward, we could have made it simpler by saying ‘if more than half of’ instead of ‘if more than or equal to m% of’ the participants involved met the section inclusion criteria for a factor. The reason that we did not is the latter allows us to change the minimum majority formula in future work, without changing much of our explanation on the chunk-identification factors.

We also used a different aggregation mechanism for the ratings data (Factor AveR3 and AveR3F) to arrive at the preference of a group. This is done by averaging the ratings. The section inclusion formula is not applicable to Factor AveR3. We explain Factor AveR3 and AveR3F in detail in Section “6.6.2.4 Average Rating Factors” of this chapter.

5.6.2.2 Basic Factors

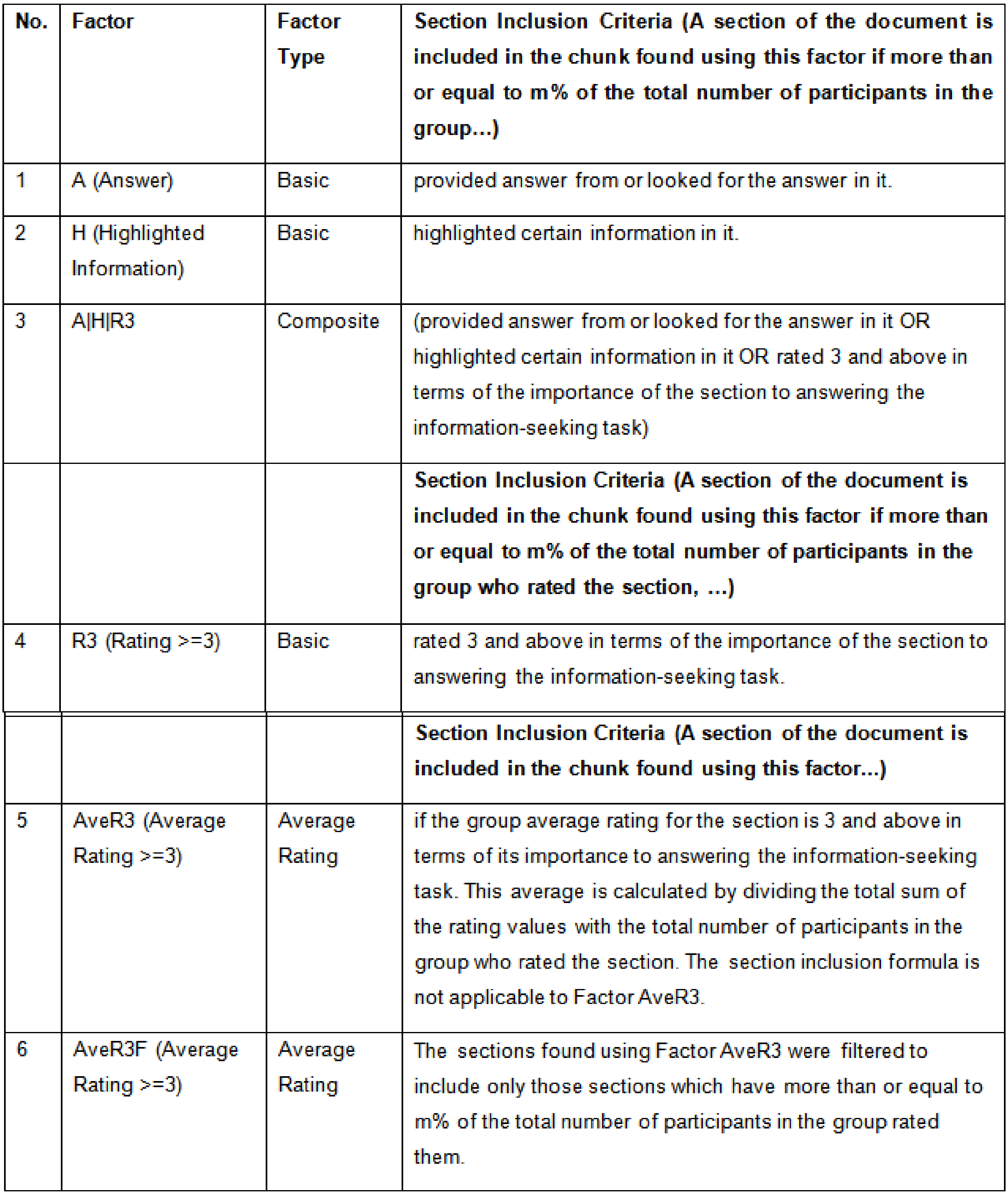
Table 5.4 summarizes all the factors we used to find chunks. Each participant was asked to provide his or her answer to the task in bullet-point form, as comprehensively as possible. He or she was asked to also state in which section of the document each bullet-point answer was found. Finding chunks using the participants’ answer (Factor A) for the task is based on the participants’ indications of the sections of the document where each of their bullet-point answers was provided or searched from. These indications revealed which sections of the document are ‘needed’ by the participants for the task.

Participants were instructed to give their answers as comprehensively as possible. Therefore, we interpreted any section not specified by a participant as ‘not needed’ by the participant for the task. Consequently, the notion of majority used in **Factor A** (Answer) is based on the total number of participants in the group and not the total number of those who provided the indications. In other words, the value T in the minimum majority formula used to determine the minimum number of participants in order to be considered as majority is referring to the total number of participants in the group.

For chunks based on Factor A, a section of the document is included in a chunk if more than or equal to m% of the total number of participants in the group provided an answer from or looked for the answer in it. Recall that m is ((minM / T) * 100). It is important to note that the chunks discovered based on Factor A did not take into consideration the correctness of the answers.

Example: Assume that 6 out of 8 participants in a group indicated that one or more of their bullet-point answers was or were found in section X.Y. This is interpreted as section X.Y is ‘needed’ by them for the task. For the other 2 participants where no such indication was given, this is interpreted as section X.Y is ‘not needed’ by them. Since the minimum number of participant that is needed in order to be considered as majority is 5 (62.5% of the participants), this section is ‘needed’ based on the common preference of this group of participants and therefore included in the chunk found for this group using this factor.

For chunks based on **Factor H** (highlighted information), a section of the document is included if more than or equal to m% of the total number of participants in the group highlighted certain information in it. As the participants looked for the answer to the information-seeking task, they were asked to highlight information in the document relevant to the assigned task. Following that, highlighting part of the content or the whole content of a section is another form of participants' indication of their needs of a section to the task. The absence of highlighted information in a section is interpreted as the section is 'not needed' by a participant for the assigned task. Consequently, when using the presence or absence of highlighted information in a section to find common preference of the participants for that section, the notion of majority is also based on the total number of participants in the group and not just the number of those who highlighted information in a section.



No.	Factor	Factor Type	Section Inclusion Criteria (A section of the document is included in the chunk found using this factor if more than or equal to m% of the total number of participants in the group…)
1	A (Answer)	Basic	provided answer from or looked for the answer in it.
2	H (Highlighted Information)	Basic	highlighted certain information in it.
3	A|H|R3	Composite	(provided answer from or looked for the answer in it OR highlighted certain information in it OR rated 3 and above in terms of the importance of the section to answering the information-seeking task)
			Section Inclusion Criteria (A section of the document is included in the chunk found using this factor if more than or equal to m% of the total number of participants in the group who rated the section, …)
4	R3 (Rating >=3)	Basic	rated 3 and above in terms of the importance of the section to answering the information-seeking task.
			Section Inclusion Criteria (A section of the document is included in the chunk found using this factor…)
5	AveR3 (Average Rating >=3)	Average Rating	if the group average rating for the section is 3 and above in terms of its importance to answering the information-seeking task. This average is calculated by dividing the total sum of the rating values with the total number of participants in the group who rated the section. The section inclusion formula is not applicable to Factor AveR3.
6	AveR3F (Average Rating >=3)	Average Rating	The sections found using Factor AveR3 were filtered to include only those sections which have more than or equal to m% of the total number of participants in the group rated them.

Table 5.4: Factors to Identify Chunks and the Section Inclusion Criteria

Example: An example of determining whether a section is to be included in the chunk found using Factor H would be similar to the example given for Factor A.

The third form of participants' indication of whether a section is 'needed' for the assigned task is their ratings of the section in terms of its importance to the task. These are more explicit because the participants specifically indicated the level of importance of a section to the assigned task.

The participant's rating of the visited section is captured on a Likert-scale ranging from 1 (lowest) to 5 (highest), but with two other options, "Not Important" and "Not Sure". "Not Important" is assigned zero whereas "Not Sure" option is not assigned any value. If the latter is chosen, it is treated as if the participant did not rate the section.

Several possible assumptions could be made of the situation whereby a participant did not rate a section. Firstly, an unrated section could mean it is totally unrelated or unimportant to the task and was skipped by a participant skimming for relevant sections. Consequently an unrated section could be regarded as unimportant and be assigned a zero rating. However, there is an explicit "Not Important" option. Therefore, if a participant did not rate a section it is unconvincing to deduce that the section is unimportant and assign it a rating value of zero. Secondly, an unrated section could also mean that the participant was not sure and therefore reluctant to indicate his or her choice. However, there is an option of "Not Sure" and therefore this speculation is unconvincing as well. Thirdly, the participant could have forgotten to rate it or deliberately did not rate it as this required extra effort. In this situation, it is difficult to speculate the importance of that section to the participant with regards to the assigned task.

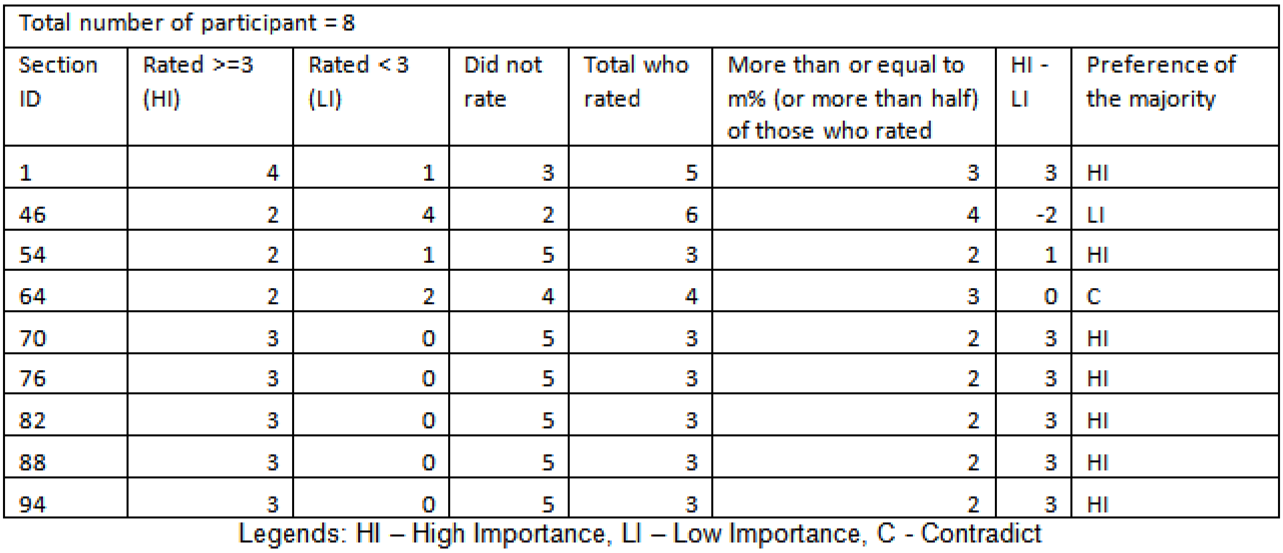
Since there are several possible reasons of why a participant did not rate a section and none of them leads to convincing speculation, we make no assumption about the value that should be assigned to an unrated section. We made use of only the explicit rating values given by those who rated the section, for identification of chunks based on ratings. In other words, the T in the minimum majority formula now refers to the number of participants in the group who provided ratings for the section and not the total number of participants in the group.

We wanted to find chunks that include sections that are of high importance for the task. A rating value of 3 and above is interpreted as the section is of high importance (HI) and therefore 'needed' by the participant for the task, whereas a value below 3 is interpreted as of low importance (LI) and therefore 'not needed'. As a result, chunks based on ratings include only those sections rated 3 and above, by more than or equal to m% of the participants who rated the sections (**Factor R3**).

There is no doubt that chunks found using Factor R3 which is based on only those who rated, could be susceptible to the bias of certain participants. For example, if only one participant rated a section and gave it a rating of 5 (i.e. 'needed'), this would be considered as the preference of the majority for the section and the section would be included in the chunk. However, we argue that by focusing on the explicit ratings, the discovery of any common preference is based on clear indications from the participants on the importance of the section to the assigned task and not affected by any speculation that we could have made. We believe that focusing on the explicit ratings is reasonable for our preliminary analysis to find common preference in the ratings data.

Example: We describe an example of determining whether a section is to be included in the chunk found using Factor R3 by using some data from WCT Task 1. For each information-seeking task, not all the participants who completed the task provided the ratings for each section of the document. For WCT Task 1, there were 8 participants who completed this task. However, the number of participants who rated each section varied. Table 5.5 shows part of the respective data for WCT Task 1. Five, 6 and 3 out of 8 participants rated section with ID 1, 46 and 54, respectively, and so on.

Since the ratings either fall into the category of high importance (HI) or low importance (LI), and minimum stance was taken on the meaning of majority, if more than half of those who rated, rated the section as of high importance, then the section is commonly regarded as of high importance and included in the chunk, because the remaining at most (half - 1) participants actually rated it as of low importance. If less than half of those who rated, rated the section as of high importance, than the section is commonly regarded as of low importance and excluded from the chunk, because the remaining at least (half + 1) participants actually rated it as of low importance. If exactly half the number of those who rated, rated it as of high importance, then there are two contradictory opinions on the importance of the section to the task. This is because the other half of those who rated, rated it as of low importance. The section with split opinion among the participants who rated is not included in the chunk.



Total number of participant = 8							
Section ID	Rated >=3 (HI)	Rated < 3 (LI)	Did not rate	Total who rated	More than or equal to m% (or more than half) of those who rated	HI - LI	Preference of the majority
1	4	1	3	5	3	3	HI
46	2	4	2	6	4	-2	LI
54	2	1	5	3	2	1	HI
64	2	2	4	4	3	0	C
70	3	0	5	3	2	3	HI
76	3	0	5	3	2	3	HI
82	3	0	5	3	2	3	HI
88	3	0	5	3	2	3	HI
94	3	0	5	3	2	3	HI

Legends: HI – High Importance, LI – Low Importance, C - Contradict

Table 5.5: Example of Chunk Identification Using Factor R3

In other words, if the ‘high importance’ category receives a higher frequency count for a section, the section is included in the chunk. Otherwise, it is not included. For example, the number of those who rated the section with ID 1 as of high importance (4) is more than those who rated it as of low importance (1). This section is included in the chunk. For the section with ID 46, the number of those who rated it as of high importance (2) is less than those who rated it as of low importance (4). This section is excluded from the chunk. For the section with ID 64, the number of those who rated it as of high importance (2) is the same as those who rated it as of low importance (2). This section is excluded from the chunk.

This example does not show the 3rd possibility that causes a section to be excluded from a chunk: the section was not rated at all by any of the participant of the group.

5.6.2.3 Composite Factor

As explained in the previous sub-section (”6.7.2.2 Basic Factors”), a participant provided 3 different forms of indication which we used to determine whether a section of the document is needed by him or her for the assigned task. The 3 different forms of indications are: 1) by specifying that the section was where one of his or her bullet-point answers was found or searched from, 2) by highlighting relevant information in the section, and 3) by rating the section in terms of its importance to the task. A rating of 3 and above is interpreted as the section is needed by the participant.

A participant might provide either one of the 3 forms of indications or a combination of two or all of them. The lack of the first two forms of indications could be due to the section supporting the understanding of other sections but a sought answer is not directly found in it. As for the lack of rating, this could be due to the participant forgetting to rate it after giving either one (or both) of the first two forms of indications or deliberately did not rate it as this required extra effort.

To cater for alternative indication by a participant, we also explored composite factor (**Factor A|H|R3**) (Table 5.4) which takes into consideration the 3 different forms of indications in a ‘OR’ (‘|’) relationship for chunks identification. For this factor, T in the minimum majority formula refers to the total number of participants in the group. Though our stand was to exclude those who did not provide ratings for a section, if the section is indeed needed by the participant who did not rate it, he or she could have specified that in either or both of the other two forms of indications.

We used the 3 forms of indication separately to find chunks as in Factor A, H and R3 because we wanted to know which factor(s) could give us better results in identification of chunks.

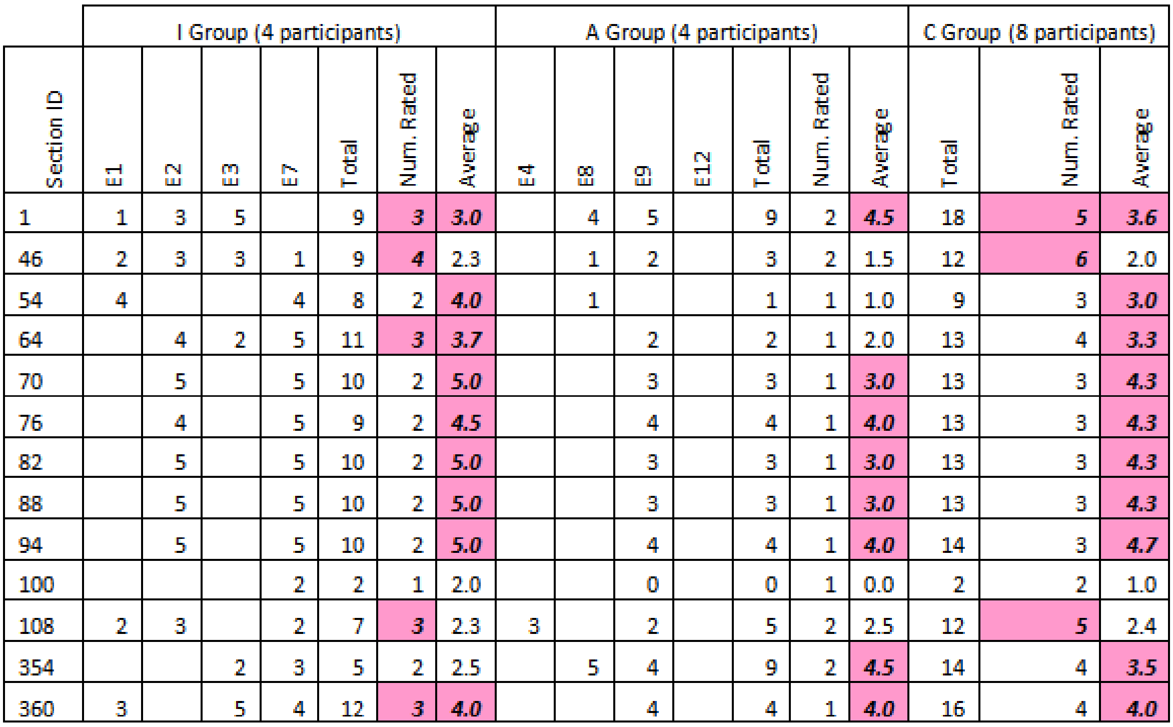
Example: An example of determining whether a section is to be included in the chunk found using Factor A|H|R3 would be similar to the example given for Factor A and therefore is skipped.

5.6.2.4 Average Rating Factors

Another type of factor (i.e. average rating factor) was also explored for the identification of chunks (Table 5.4). This type of factor uses a different way to aggregate the collective opinions of the group for a section, which is by averaging the ratings of the importance of the section to answering the information-seeking task. We then took average 3 and above as indication that a section is ‘needed’ and included it in a chunk, and an average below 3 as indication that a section is ‘not needed’ and excluded it from a chunk found based on average rating.

The average rating of a section is calculated by dividing the total sum of its rating values (including the zeroes for “Not Important”), by the number of participants who rated it (**Factor AveR3**). Participants who did not rate the section are not included in the divisor. This is done for the same reason as for Factor R3 where we make no assumption about the value that should be assigned to an unrated section by a participant. Chunks based on Factor AveR3 could also be susceptible to the bias of certain participants. For example, if only one participant rated a section and gave it a rating of 5, the average rating for this section would be 5 and it would be included in the chunk. Consequently, the initial chunks are filtered to arrive at new chunks containing only those sections rated by more than or equal to m% (or by more than half) of the total number of participants in the group and with average ratings at least 3 (**Factor AveR3F**). In other words, those sections which have a small number of participants rating them are eliminated.

Example: We describe an example of determining whether a section is to be included in the chunk found using Factor AveR3 and AveR3F by using some of the data from WCT Task 1 (Table 5.6). An empty cell in Table 5.6 means the participant did not provide rating value for the section. For example, participant E7 did not provide rating for section with ID 1. A zero rating value means that the participant selected the “Not Important” option for the section. For example, participant E9 rated section with ID 100 as not important to the task. Participant E1, E2 and E3 in the industry (I) group gave section with ID 1 the rating values of 1, 3 and 5, respectively. The average rating for this section is 3 (9 divided by 3). This met the section inclusion criteria and the section is included in the chunk found for I Group using Factor AveR3. Section with ID 46 was rated by 4 participants in I Group and its average (9 divided by 4) is 2.3. This is below 3 and the section inclusion criteria was not met. The section is not included in the chunk found for I Group using Factor AveR3.



	I Group (4 participants)							A Group (4 participants)							C Group (8 participants)		
Section ID	E1	E2	E3	E7	Total	Num. Rated	Average	E4	E8	E9	E12	Total	Num. Rated	Average	Total	Num. Rated	Average
1	1	3	5		9	***3***	***3.0***		4	5		9	2	***4.5***	18	***5***	***3.6***
46	2	3	3	1	9	***4***	2.3		1	2		3	2	1.5	12	***6***	2.0
54	4			4	8	2	***4.0***		1			1	1	1.0	9	3	***3.0***
64		4	2	5	11	***3***	***3.7***			2		2	1	2.0	13	4	***3.3***
70		5		5	10	2	***5.0***			3		3	1	***3.0***	13	3	***4.3***
76		4		5	9	2	***4.5***			4		4	1	***4.0***	13	3	***4.3***
82		5		5	10	2	***5.0***			3		3	1	***3.0***	13	3	***4.3***
88		5		5	10	2	***5.0***			3		3	1	***3.0***	13	3	***4.3***
94		5		5	10	2	***5.0***			4		4	1	***4.0***	14	3	***4.7***
100				2	2	1	2.0			0		0	1	0.0	2	2	1.0
108	2	3		2	7	***3***	2.3	3		2		5	2	2.5	12	***5***	2.4
354			2	3	5	2	2.5		5	4		9	2	***4.5***	14	4	***3.5***
360	3		5	4	12	***3***	***4.0***			4		4	1	***4.0***	16	4	***4.0***

Table 5.6: Example of Chunk Identification Using Factor AveR3 and AveR3F

The average for the combined (C) group is calculated by adding up all the rating values across both I and A (Academic) group and dividing it with the grand total number of participants who rated the section. For example, 2 participants from I Group and 1 participant from A Group rated the section with ID 54. The average rating is 3 (9 divided by 3). This met the section inclusion criteria and this section is included in the chunk found for C Group using Factor AveR3.

All the sections with average rating 3 and above are included in the chunk found for a group using Factor AveR3. Looking at the partial data in Table 5.6, the chunk for I Group includes section with ID 1, 54, 64, 70, 76, 82, 88, 94, and 360. It does not include section with ID 46, 100, 108 and 354. When this chunk is filtered to include only those sections which have more than half of the total number of participants in the group who rated them (Factor AveR3F), the chunk now only includes sections with ID 1, 64 and 360.

5.6.2.5 Other Factors Considered

Initially, we also considered other factors such as Factor R1, A|H|R1, AveR3T and AveR3TF. We did not pursue any of the factors further because of the reasons explained below.

Factor R1 is similar to Factor R3 whereas Factor A|H|R1 is similar to A|H|R3. The difference is that Factor R1 and A|H|R1 involved sections rated 1 and above, instead of sections rated 3 and above as in Factor R3 and A|H|R3. However, Factor R1 and A|H|R1 made it easy to indiscriminately include many sections with low ratings (which may not be needed for the task) in a chunk.

Factor AveR3T is similar to Factor AveR3 whereas Factor AveR3TF is similar to Factor AveR3F. The difference is that for Factor AveR3T and AveR3TF, the average rating of a section is calculated by dividing the total sum of rating values with the total number of participants in a group regardless of whether they rated the section. The assumption was an unrated section is unimportant to a participant and therefore assigned a zero rating value. However, we have argued that this assumption is not convincing because there is an explicit ‘Not Important’ rating option. The participant most likely could have forgotten to rate it or deliberately did not rate it as this required extra effort. In this situation, it is difficult to speculate the importance of that section to the participant with regards to the assigned task.

5.7 Benchmarking of Chunks against an Oracle Set

This section gives the definition of oracle set, recall and precision measures. It also explains the terms (i.e. size of chunk, the most basic chunk, false section, the best chunk and best chunk-identification factor) that we used in our discussion of the benchmarking results, the chunk discovery results and the overall results for Study 2 and 3. This section also explains the criteria that we used to assess the ‘goodness’ of chunks found in our studies.

5.7.1 Definition of the Oracle Set, Recall and Precision Measures

To evaluate the chunks and the chunk identification factors, we borrowed some basic ideas from information retrieval system evaluation: recall and precision measures based on a set of relevance judgments of either relevant or non-relevant for each query-document pair, (Manning, Raghavan, & Schütze, 2008). We took the idea of the set of relevance judgments further in two aspects, to create what we termed the oracle set. Firstly, the set of relevance judgments in our study contains the judgments of not merely the relevance of each section of a document to the information-seeking task (the query), but also whether the sections are mandatorily, optionally or not required. Secondly, in making these judgments, consideration needs to be taken on whether together these sections provide the collective information needed for the task.

An oracle set is defined in this research as below:

The oracle set for an information-seeking task is the set that contains all the sections of the document which are judged through the use of a vigorous process as compulsory for the task, and that together these sections provide the collective information needed for the task.

As the person who defined the information-seeking task, the researcher is the most reasonable person to judge what is relevant to the task, in other words the sections that are 'needed' for the task. This is in tandem with the use of the person who formulated the query to choose the relevant documents in the aspect of relevance judgments in information retrieval (Wallis & Thom, 1996). However, a person would make the relevance judgments based on his or her information need (Wallis & Thom, 1996). Consequently, to reduce the impact of the researcher's information need on the relevance judgments set constructed for a task, other professionals with SA background were invited to build separate sets of relevance judgments independently. The sets of relevance judgments were then reconciled to build the oracle set for the task. **Only those sections agreed by all the judges (including the researcher) as mandatory to be included in the oracle set made it into the oracle set.**

The construction of the oracle set for an information-seeking task follows a vigorous process as detailed in Section "6.8 Construction of Oracle Sets" of this chapter. In comparison to the chunks discovered from the participants' responses data, the oracle set is better because it was constructed using a vigorous process.

An oracle set was constructed for each of the information-seeking tasks for the two ADs. Then, we calculated the standard measures of recall and precision (Manning et al., 2008) for each chunk against the respective oracle set. The two measures are defined as:

Recall (R) = number of needed sections retrieved / Total needed sections

Precision (P) = number of needed sections retrieved / Total sections retrieved

The benchmarking of the chunks found for each information-seeking task against its oracle set and the discussion of the chunk discovery results can be found in the respective chapter for each task (Chapter 6 to 11). Chapter 13 compares the results for the 3 information-seeking tasks for the 2 ADs, and presents the summary of the overall results for Studies 2 and 3.

5.7.2 Size of Chunk, The Most Basic Chunk, and False Section

The size of a chunk is measured in terms of the number of the document's sections that the chunk comprises. This measure can be used to find out the proportion of the number of the document's sections needed by a group of participants.

The most basic chunk found for a group of participants for the particular task is the group's chunk that contains the smallest number of sections of the AD. A false section of a chunk is a section of the AD, which is not in the oracle set for the task.

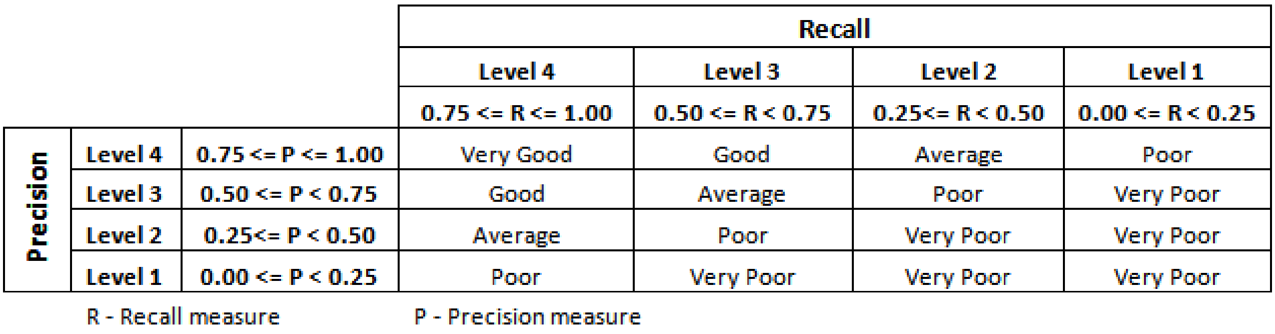
5.7.3 Criteria to Assess the "Goodness" of a Chunk

In many situations, high recall i.e. finding a fairly good amount of all the relevant information ('needed' information in our case) rather than high precision are needed (Wallis & Thom, 1996). Examples are when finding materials to lodge a new patent or searching for precedence cases in legal work. The information-seeking tasks in our study also call for high recall. The first task is related to the SA of the described system. It is desirable to be able to find a high amount of all the needed information in order to get an overview of the SA of the system. The second task is to find out how to change a certain part of a system and in doing so which parts of the system will be affected. It is important not to miss out too much of the needed information, when assessing where and how the changes need to be made, and the possible change impacts. The third task is to find out how a system was designed at an architectural level to achieve a certain quality attribute (such as security and modifiability). For these cross-cutting concerns which span the entire system, a high recall is also required.

Though the information-seeking tasks in our study require high recall, an acceptable chunk for each task should at the same time not containing too many 'not needed' sections. A chunk with higher recall and higher precision measures is considered relatively better or more favourable for the particular task than a chunk with lower recall and lower precision measures. The chunk with the highest recall and highest precision measures is the most favourable or the best chunk found for a group of participants for the respective task. Nevertheless, recall and precision measures are inversely related (Rajaraman, Leskovec, & Ullman, 2013) or they trade-off against each other (Manning et al., 2008). A chunk could have a higher recall measure but lower precision measure, whereas another chunk for the same group could have lower recall measure but higher precision measure.

We defined criteria in Table 5.7 which takes into consideration the trade-off between recall and precision measures, to assess how 'good' the chunks are for the particular task. The boundary values of the 4 levels of recall and precision measures shown in Table 5.7, were arbitrarily chosen. Despite that, we tried to choose values that are of minimal sensitivity to the categorization of the 'goodness' of the chunks in our studies. In other words, if we were to make some small adjustments to these boundary values, the number of the chunks that fell under each category of 'goodness' would not be much affected. This is discussed further in Section "14.7.1 Boundary Values Used in Assessment of Goodness of Chunks" of Chapter 14.

As mentioned earlier, the information-seeking tasks in our study require chunks with high recall measures but acceptable chunks for each task should at the same time do not have too low precision measures (i.e. contain too many 'not needed' sections). Therefore, when assessing the 'goodness' of chunk using recall and precision measures, we emphasized recall but at the same time wanted certain level of precision. The goodness measure (either 'Very Good', 'Good', 'Average', 'Poor' or 'Very Poor') that we assigned to each combination of the recall and precision level is based on our intuition of how 'good' a chunk is for the respective task, as a results of the trade-off nature between recall and precision.



			Recall			
			Level 4	Level 3	Level 2	Level 1
			0.75 <= R <= 1.00	0.50 <= R < 0.75	0.25<= R < 0.50	0.00 <= R < 0.25
Precision	Level 4	0.75 <= P <= 1.00	Very Good	Good	Average	Poor
	Level 3	0.50 <= P < 0.75	Good	Average	Poor	Very Poor
	Level 2	0.25<= P < 0.50	Average	Poor	Very Poor	Very Poor
	Level 1	0.00 <= P < 0.25	Poor	Very Poor	Very Poor	Very Poor

R - Recall measure P - Precision measure

Table 5.7: Criteria to Assess the Goodness of Chunks

We think that the criteria that we defined to assess the 'goodness' of chunk are reasonable for our purpose, which is mainly to help us to make consistent assessment of the chunks found across the different information-seeking tasks in our studies. Further explanation on the criteria is given next.

A chunk with recall measure equal to or above 0.75 (Level 4) contains many (i.e. at least 75%) of the sections compulsory (based on our oracle set) for the respective task. If at the same time more than or equal to 75% of its sections are compulsory for the task (Level 4 of precision measure), we regard this chunk as 'Very Good' for the respective task comparing to reading the whole document. With the high recall of 0.75 or above, we regard the chunk as 'Good' for the respective task even if only 50% to less than 75% of its sections are compulsory for the task (Level 3 of precision measure). The chunk is considered 'Average' for the task if only 25% to less than half of its sections are compulsory for the task (Level 2 of precision measure). It is considered 'Poor' for the task if less than 25% of its sections are compulsory for the task (Level 1 of precision measure).

With the decrease of recall measure to less than 0.75 but more than or equal to 0.50 (Level 3), we lower the level of goodness measure assigned to a chunk. A chunk at this level of recall measure contains 50% to less than 75% of the number of sections compulsory for the task. It excludes a considerable number of sections compulsory for the task. Nevertheless, it is considered 'Good' for the task if at least 75% of its sections are compulsory for the task. The chunk is considered 'Average' for the task if 50% to less than 75% of its sections are compulsory for the task. It is considered 'Poor' for the task if only 25% to less than half of its sections are compulsory for the task. It is considered 'Very Poor' for the task if less than 25% of its sections are compulsory for the task.

With further decrease of recall measure to less than 0.50 but more than or equal to 0.25 (Level 2), we further lower the level of goodness measure assigned to a chunk. A chunk at this level of recall measure does not contain 50% of the number of sections compulsory for the task. Thus, the chunk is considered 'Average' for the respective task even though more than or equal to 75% of its sections are compulsory for the task. It is considered ‘Poor’ for the task if 50% to less than 75% of its sections are compulsory for the task. It is considered ‘Very Poor’ for the task if less than half of its sections are compulsory for the task (Level 2 and 1 of precision measure).

A chunk with recall measure below 0.25 (Level 1) contains at most 25% of the sections compulsory for the task. A chunk with this low level of recall measure excludes many sections compulsory for the task. Therefore, the chunk is regarded as ‘Poor’ for the task even if more than or equal to 75% of its sections are compulsory for the task. It is regarded as ‘Very Poor’ for the task if less than 75% of its sections are compulsory for the task.

5.7.4 The “Best” Chunk and The Best Chunk-Identification Factor

We used the goodness measure criteria explained in the previous section to determine how ‘good’ a chunk is for a particular information-seeking task. For each group of participants, we compared the composition of the chunks that are at the best level of goodness measure, to decide the best (or the most favourable) chunk for the group. We did so by assessing the criticality of the inclusion and the omission of the different sections in the chunks, to the task.

The factor that produces the best chunk for a group is the best chunk-identification factor for the group, with regards to the respective task.

5.7.5 Interpretation of Chunks Found Using Factor R3, AveR3 and AveR3F

For a chunk found using Factor R3, we cannot immediately regard sections which are excluded from the chunk as ‘not needed’ by the majority of the group who rated them, for the respective task. The reason is there are 3 possibilities when a section did not make it into a chunk found using Factor R3: 1) It was indeed rated lowly (i.e. below 3) by the majority of the group who rated it and therefore interpreted as ‘not needed’ for the task. 2) There was a split opinion among the participants who rated the section (i.e. there was equal number of participants who rated it lowly and highly). In this case, the criteria of having the majority of the group who rated the sections, rated it highly (3 and above) was not fulfilled and therefore the section is interpreted as ‘not needed’ for the task. 3) It was not rated at all by any participant of the group. Since we made no assumption about the value that should be assigned to an unrated section, we do not interpret this section as ‘not needed’ for the task.

Similar situation applies to a chunk found using Factor AveR3 or AveR3F. A section could be excluded from Factor AveR3’s or AveR3F’s chunk due to low average rating (i.e. average rating below 3 and interpreted as ‘not needed’ for the task) or because it was not rated at all by any participant of the group (of which we do not interpret as ‘not needed’ for the task).

To distinguish that a section is excluded from chunks found using Factor R3, AveR3 and AveR3F due to not being rated by any of the participants in a group, from the other reason(s), we need to inspect the raw data. This seemingly drawback of our chunks is acceptable since our main interest is on the sections ‘needed’ for a task, which our chunks revealed. We only look at sections ‘not needed’ for a task when it is necessary.

The only situation we are concerned with a section being excluded from a Factor R3's, AveR3's and AveR3F's chunk due to not being rated at all, is during our determination of the common oracle set's sections missed by a group of participants. This is explained in the next section.

5.7.6 Determination of the Commonly-Missed Oracle Set's Sections and Common False Sections

In identifying the common oracle set's sections missed by a group of participants, we only consider oracle set's sections missed by more than half of the chunk-identification factors. This criteria ensures that an identified oracle set's section is indeed commonly missed by the group since it is excluded by most of the factors. We interpret a commonly-missed oracle set's section as generally 'not needed' by the respective group of participants.

Oracle set's sections not rated at all by any participant in a group do not qualify as commonly-missed oracle set's sections for the group. The reason is we made no assumption about the value to assign to a section not rated by a participant and do not interpret such a section as 'not needed' by the participant for the task.

Our critieria of having more than half of the chunk-identification factors excluding an oracle set's section from chunks, in order for the section to be qualified as a commonly-missed section, prevents an oracle set's section which was not rated at all from being identified as a commonly-missed section. A totally unrated section would be excluded from chunks found using Factors R3, AveR3 and AveR3F, which are based only on ratings data. Therefore, even if this totally unrated section is included by all the other 3 factors (Factors A, H and A|H|R3), this section would not be able to fulfil the criteria of commonly-missed section.

We use similar criteria in identifying the common false sections (i.e. sections not from the oracle set) in chunks found for a group, which is we only consider false sections included by more than half of the chunk-identification factors. This criteria ensures that an identified false section is indeed a common false section for a particular group of participants since it is included by most of the chunk-identification factors. We interpret a common false section as generally 'needed' by the group of participants.

5.8 Information Needed

We also investigated the information needed by each group of participants. For Study 2, this is based on the groups' chunks found using Factor A|H|R3. Our reason is Factor A|H|R3 caters for the participants' different preferences in indicating the relevance of a section of the document to them for the assigned task. In addition, this factor is based on the majority of the participants in the group and not just the majority of those who gave the responses.

5.9 The Construction of Oracle Sets

To ensure the construction of oracle sets is done consistently across different tasks, we defined a template for relevance judgments (Appendix 5.2). The template was designed to solicit the following information for each section of the document with regards to the specific information-seeking task: type of information found in it, in what way the information in the section is relevant to the task, whether the section should be included (mandatory, optional or no) in the oracle set for the task and the rationale behind it, how important is the information in the section to the task by giving a rating value and any extra information needed from elsewhere (such as the Internet or other documents), and a comprehensive answer for the task with indications of from which sections of the document each important aspect of the answer was found.

The process involved in the construction of an oracle set for an information-seeking task is depicted in Figure 5.2. The researcher performed an in-depth analysis of the AD to construct the set of relevance judgments of the document's sections with reference to the task by filling in the relevance judgment template. This was done with careful consideration of the information-seeking task itself, any concept (such as 'software architecture', 'security', 'modifiability' and so on) involved in the task and the role to be assumed.

A third party judge was selected to construct a separate set of relevance judgments for the task. The main criteria in the selection of this judge were: he or she is meticulous in nature, is a professional with SA background, and is willing to commit the time and effort required by the whole process to construct the final oracle set.

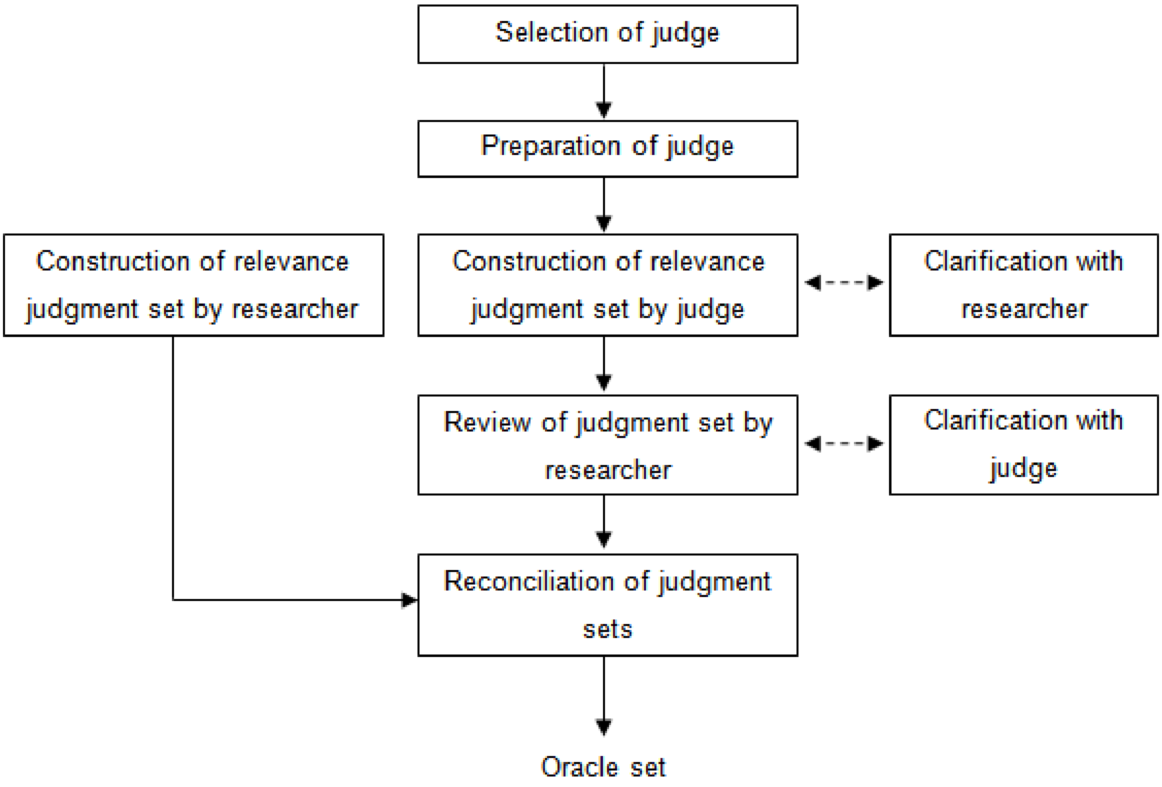

Selection of judge
Preparation of judge
Construction of relevance judgment set by researcher
Construction of relevance judgment set by judge
Clarification with researcher
Review of judgment set by researcher
Clarification with judge
Reconciliation of judgment sets
Oracle set

Figure 5.2: The Construction Process of an Oracle Set

Prior to working on the relevance judgment set, the judge undertook a preparation session with the researcher. The aims of this session were to help the judge to understand the purpose of building the relevance judgement set, what needed to be done and the proper way of filling in the relevance judgment template. The judge was also given the AD and was asked to seek any needed clarification. The judge set a time with the researcher on when he or she would be working on the relevance judgment set. The researcher stood-by during that time to give any immediate clarification should the need arise.

The judge submitted the completed template to the researcher. The researcher reviewed it and sought clarifications from the judge on any ambiguous responses. The researcher and the judge exchanged their sets of relevance judgments and set a time to discuss any discrepancy in the two. The purpose of this discussion was to reconcile the two sets into one final oracle set.

The process of the oracle set construction was rigorous: the careful inspection of the elements involved (such as the AD, the task and so on), the use of the template, the judge preparation session, on-site clarification during relevance judgment set construction, review with follow-up clarification, and the reconciliation session. There was no time constraint imposed when constructing the relevance judgment set since “the accuracy of human judgments decreases under time pressure”(Edland & Svenson, 1993). The main concern of each judge was to build ‘the oracle’. All these gave us the confidence in the accuracy of the final oracle set for each task.

For the first AD (WCT), the researcher and another professional were involved in the construction of the oracle sets for the 3 tasks. The same two people constructed the oracle sets for all the 3 tasks. The construction of the oracle sets for the 3 tasks for ASM document also followed the same way but involving the researcher and a different professional.

All the professionals (including the researcher) involved in the construction of the oracle sets in this study are academics who have taught SA. The average number of SA teaching experience is 9.67 years (minimum 4 years, maximum 20 years). We had difficulty finding industry professionals who could devote the significant amount of time and effort required for the oracle set construction.

The oracle set for each task can be found in the respective chapter (Chapter 6 to Chapter 11). Any concept (such as ‘software architecture’, ‘modifiability’ and so on) involved in the respective task is also presented in those chapters.

5.10 **Summary**

This chapter explains the way we designed and carried out Study 2. It also lays down the foundation for the chunking of architectural information by defining the notion of a ‘chunk’ and describes the factors we used to identify chunks. It also presents the benchmarking of the discovered chunks against oracle sets as a way to evaluate the chunks and chunk-identification factors; all the relevant terms that we used in our discussion of the results of chunks identification; the criteria used to assess the ‘goodness’ of chunks found; how best chunk and the best chunk-identification factor are determined; interpretation of chunks; determination of commonly-missed oracle set’s sections and common false sections, and the process of oracle set construction. Chapter 5 is a very important chapter for Studies 2 and 3, and therefore should be read prior to reading the subsequent chapters. The following 6 chapters present the chunking of architectural information for the six information-seeking tasks specified for the Web Curator Tool (WCT) and the Aperi Storage Manager (ASM) software architecture documents, in Study 2. Chapter 13 presents the summary and discussion of the overall results of Studies 2 and 3.

6 Chunking of Architectural Information (WCT Task 1)

This chapter presents the chunking of architectural information for the first information-seeking task for the Web Curator Tool (WCT Task 1) software architecture document, for Study 2. The task was:

> "You are a software architect new to the Web Curator Tool project. You would like to know what the software architecture of the Web Curator Tool is".

The results presented in this chapter are based on the responses of 8 participants who completed WCT Task 1. Four were industry participants and four were academic participants.

This chapter describes the identification and interpretation of chunks for WCT Task 1; benchmarking of discovered chunks against the respective oracle set; the concept involved in the task; the oracle set for the task; the results of benchmarking the chunks, and the discussion of chunks discovery results. Chapter 7 to 11 which are on the other 5 information-seeking tasks for Study 2, adopt the same structure as this chapter.

Chapter 5 provides the foundation to understand this chapter and should be read prior to reading this chapter.

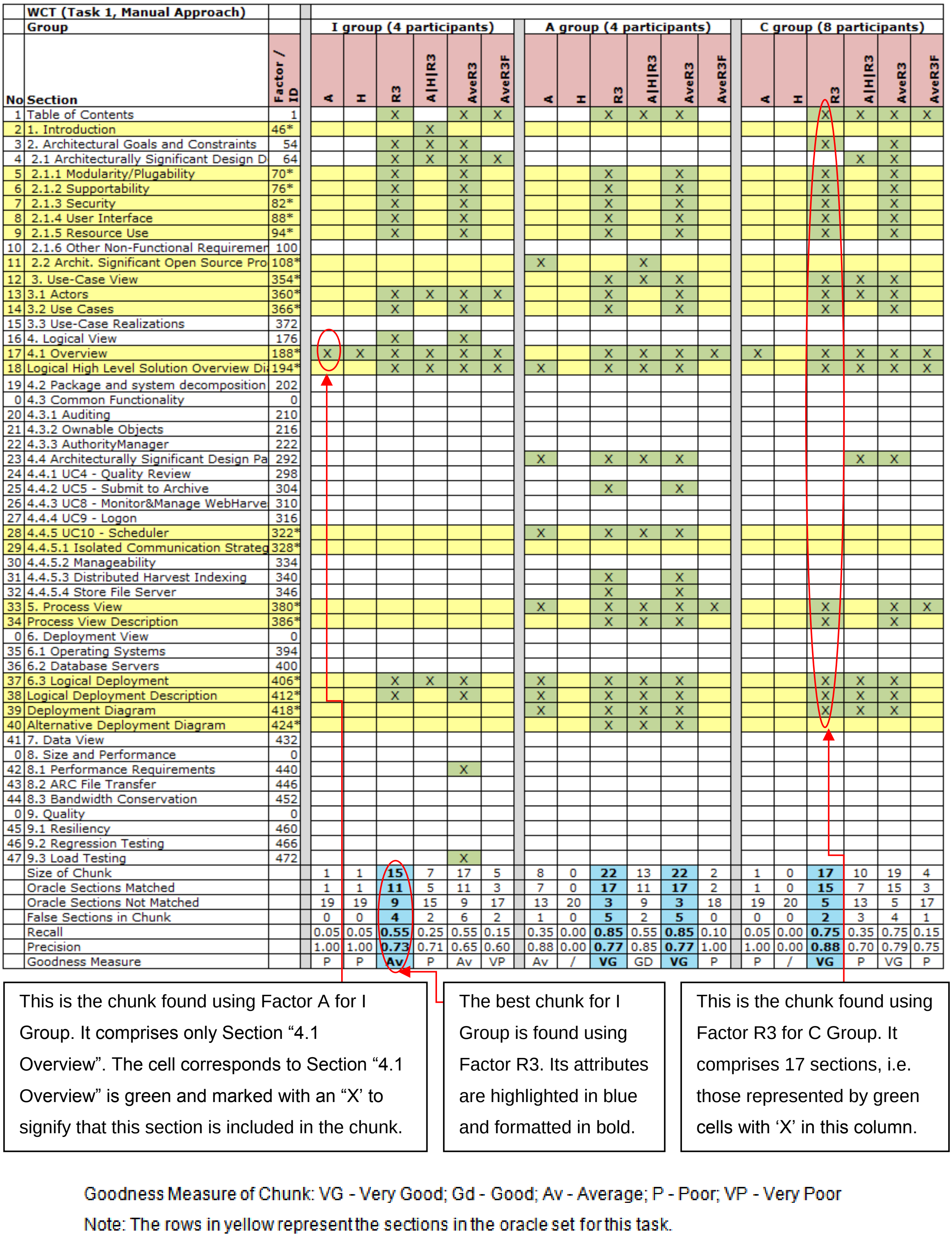
6.1 Identification and Interpretation of Chunks

We used the 6 chunk-identification factors to identify the chunks for this task. This was done for the 3 groups of participants: the industry practitioner (I) group, the academic (A) group, and the combined (C) group of the previous two.

Table 6.1 shows the chunks discovered for WCT Task 1, for each group of participants. The sizes of the chunks are near the bottom of the table. Recall that the size of a chunk is measured in terms of the number of the document's sections that the chunk comprises. It can be used to find out the proportion of the number of the document's sections needed by a group of participants.

The chunk found using a factor consists of those sections represented by green cells with 'X' in the column of that factor. For example, in the column of Factor A for Group I, the cell corresponding to Section "4.1 Overview" is green and marked with an "X". This should be interpreted as meaning that Section "4.1 Overview" is included in the chunk found for Group I using Factor A. There is only one section in this chunk and its size is 1. The rows in yellow represent the sections in the oracle set for this task. Other attributes of chunk (namely the numbers of oracle set's sections matched and not matched by chunk, the number of false sections in chunk, recall, precision and goodness measures of chunk) are related to benchmarking of chunk against the respective oracle set and discussed under Section "6.2.3 Benchmarking Results" of this chapter. The best chunk for a group of participants has its attributes highlighted in blue and formatted in bold.

Factor A: The chunks discovered using Factor A for C and I Groups contain only Section "4.1 Overview" of the document. The chunk for A Group is larger with 8 sections.



	WCT (Task 1, Manual Approach)																			
	Group		I group (4 participants)						A group (4 participants)						C group (8 participants)					
No	Section	Factor / ID	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
1	Table of Contents	1			X		X	X			X	X	X				X	X	X	X
2	1. Introduction	46*				X														
3	2. Architectural Goals and Constraints	54			X	X	X										X		X	
4	2.1 Architecturally Significant Design D	64			X	X	X	X										X	X	
5	2.1.1 Modularity/Plugability	70*			X		X				X		X				X		X	
6	2.1.2 Supportability	76*			X		X				X		X				X		X	
7	2.1.3 Security	82*			X		X				X		X				X		X	
8	2.1.4 User Interface	88*			X		X				X		X				X		X	
9	2.1.5 Resource Use	94*			X		X				X		X				X		X	
10	2.1.6 Other Non-Functional Requiremer	100																		
11	2.2 Archit. Significant Open Source Pro	108*							X			X								
12	3. Use-Case View	354*									X	X	X				X	X	X	
13	3.1 Actors	360*			X	X	X	X			X		X				X	X	X	
14	3.2 Use Cases	366*			X		X				X		X				X		X	
15	3.3 Use-Case Realizations	372																		
16	4. Logical View	176			X		X													
17	4.1 Overview	188*	X	X	X	X	X	X			X	X	X	X	X		X	X	X	X
18	Logical High Level Solution Overview Di	194*			X	X	X	X	X		X	X	X				X	X	X	X
19	4.2 Package and system decomposition	202																		
0	4.3 Common Functionality	0																		
20	4.3.1 Auditing	210																		
21	4.3.2 Ownable Objects	216																		
22	4.3.3 AuthorityManager	222																		
23	4.4 Architecturally Significant Design Pa	292							X		X	X	X					X	X	
24	4.4.1 UC4 - Quality Review	298																		
25	4.4.2 UC5 - Submit to Archive	304									X		X							
26	4.4.3 UC8 - Monitor&Manage WebHarve	310																		
27	4.4.4 UC9 - Logon	316																		
28	4.4.5 UC10 - Scheduler	322*							X		X	X	X							
29	4.4.5.1 Isolated Communication Strateg	328*																		
30	4.4.5.2 Manageability	334																		
31	4.4.5.3 Distributed Harvest Indexing	340									X		X							
32	4.4.5.4 Store File Server	346									X		X							
33	5. Process View	380*							X		X	X	X	X			X		X	X
34	Process View Description	386*									X	X	X				X		X	
0	6. Deployment View	0																		
35	6.1 Operating Systems	394																		
36	6.2 Database Servers	400																		
37	6.3 Logical Deployment	406*			X	X	X		X		X	X	X				X	X	X	
38	Logical Deployment Description	412*			X		X		X		X	X	X				X	X	X	
39	Deployment Diagram	418*							X		X	X	X				X	X	X	
40	Alternative Deployment Diagram	424*									X	X	X							
41	7. Data View	432																		
0	8. Size and Performance	0																		
42	8.1 Performance Requirements	440					X													
43	8.2 ARC File Transfer	446																		
44	8.3 Bandwidth Conservation	452																		
0	9. Quality	0																		
45	9.1 Resiliency	460																		
46	9.2 Regression Testing	466																		
47	9.3 Load Testing	472					X													
	Size of Chunk		1	1	**15**	7	17	5	8	0	**22**	13	**22**	2	1	0	**17**	10	19	4
	Oracle Sections Matched		1	1	**11**	5	11	3	7	0	**17**	11	**17**	2	1	0	**15**	7	15	3
	Oracle Sections Not Matched		19	19	**9**	15	9	17	13	20	**3**	9	**3**	18	19	20	**5**	13	5	17
	False Sections in Chunk		0	0	**4**	2	6	2	1	0	**5**	2	**5**	0	0	0	**2**	3	4	1
	Recall		0.05	0.05	**0.55**	0.25	0.55	0.15	0.35	0.00	**0.85**	0.55	**0.85**	0.10	0.05	0.00	**0.75**	0.35	0.75	0.15
	Precision		1.00	1.00	**0.73**	0.71	0.65	0.60	0.88	0.00	**0.77**	0.85	**0.77**	1.00	1.00	0.00	**0.88**	0.70	0.79	0.75
	Goodness Measure		P	P	**Av**	P	Av	VP	Av	/	**VG**	GD	**VG**	P	P	/	**VG**	P	VG	P

This is the chunk found using Factor A for I Group. It comprises only Section “4.1 Overview”. The cell corresponds to Section “4.1 Overview” is green and marked with an “X’ to signify that this section is included in the chunk.

The best chunk for I Group is found using Factor R3. Its attributes are highlighted in blue and formatted in bold.

This is the chunk found using Factor R3 for C Group. It comprises 17 sections, i.e. those represented by green cells with ‘X’ in this column.

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

Note: The rows in yellow represent the sections in the oracle set for this task.

Table 6.1: The Chunks for WCT Task 1 Identified Using Different Factors

Interpretation: Recall that Factor A is based on the participants’ indications of the sections of the document where each of their bullet-point answers was provided or searched from. Indications of this type show very clearly the sections needed by a participant for the task.

The chunk for C Group shows that the majority of the group needed Section “4.1 Overview” for WCT Task 1. The fact that it appears as the only section in the chunk shows the importance of this section to the majority of the group when attempting this task.

Section “4.1 Overview” describes the main components, end-user access and functionalities of the system, as well as the interface layer to all external systems and tools required by WCT.

The chunks found for the two sub-groups of the participants are interesting in comparison: I Group’s chunk shows that the majority of the industry participants agreed only on Section “4.1 Overview” as the section needed for the task. On the other hand, A Group’s chunk shows that there were more sections (8) that the majority (i.e. at least 3 out of 4) of the academic participants needed for WCT Task 1. These sections are mainly related to open source products used (Section “2.2”), high-level logical components diagram (Section “Logical High Level Solution Overview Diagram”), brief introduction to architecturally significant use-cases of WCT (Section “4.4 Architecturally Significant Design Packages”), the diagram on managing the distributed nature of Harvesters (Section “4.4.5 UC10 – Scheduler”), process view diagram (Section “5. Process View”), logical mapping of components to nodes (diagram and its textual description) (Section “6.3 Logical Deployment” and Section “Logical Deployment Description”), and deploying the deployment packages to application server instance (first of the two diagrams) (Section “Deployment Diagram”). **Interestingly Section “4.1 Overview” (textual description of main logical components) which is the only section in I Group’s chunk is absented from A Group’s chunk. In its place is Section “Logical High Level Solution Overview Diagram” which is the graphical counterpart of Section “4.1 Overview”.**

Factor H: No chunk is discovered for C Group nor A Group using this factor. The same chunk consisting of Section “4.1 Overview” is found for I Group.

Interpretation: Recall that Factor H is based on the presence of highlighted information in sections as indications that these sections contain relevant information to the task and therefore were ‘needed’ by the participants when they attempted the respective task. For A Group, one possible reason no chunk is found is that the majority of the academic participants provided the first form of indication which is, the section(s) of the document where each of their bullet-point answers was provided or searched from (as used in Factor A) instead of highlighting relevant information in the sections. This speculation is supported by the finding of A Group’s chunk using Factor A. Since half of the combined group are these academics, the speculation that the first form of indication is preferred could also be the possible reason of the absence of a chunk for C Group when Factor H is used. **The discovery of the same one-element chunk for I Group using Factor A and Factor H accentuates the importance of the textual description of main logical components to the majority of I Group for WCT Task 1.**

Chunk is found for A Group using Factor A but not Factor H. This is the same as WCT Task 2. One possible reason is that the majority of A Group preferred to specify the sections of the document where each of their bullet-point answers was provided or searched from, instead of highlighting the relevant information in the sections.

Factor R3: C Group’s chunk contains 17 sections: “Table of Contents”, “2. Architectural Goals and Constraints”, 5 sub-sections of Section 2.1 (“2.1.1 Modularity/Plugability”, “2.1.2 Supportability”, “2.1.3 Security“, “2.1.4 User Interface” and “2.1.5 Resource Use”), 3 of the 4 sections related to use-case view (“3. Use-Case View”, “3.1 Actors”, “3.2 Use Cases”), 2 of the 3 sections related to logical view (“4.1 Overview” and ““Logical High Level Solution Overview Diagram”), both the sections related to process view (“5. Process View” and “Process View Description”) and 3 of 6 sections related to deployment (“6.3 Logical Deployment”, “Logical Deployment Description” and “Deployment Diagram”).

I Group’s chunk contains 15 sections. It is very similar to C Group’s chunk with the addition of 2 sections (Section “2.1 Architecturally Significant Design Decisions”, the remaining section related to logical view i.e. Section “4. Logical View”), and the exclusion of 4 sections (i.e. Section “3. Use-Case View”, the two sections on process view, and Section “Deployment Diagram”).

The chunk found for A Group using Factor R3 contains 22 sections. In comparison to C Group’s chunk, it includes 6 additional sections: “4.4 Architecturally Significant Design Packages”, “4.4.2 UC5 - Submit to Archive”, “4.4.5 UC10 – Scheduler”, “4.4.5.3 Distributed Harvest Indexing”, “4.4.5.4 Store File Server” and “Alternative Deployment Diagram”. It excludes Section “2. Architectural Goals and Constraints” which contains only links to its two sub-sections.

This chunk contains most of the 15 sections in I Group’s chunk, with the exclusion of Sections “2. Architectural Goals and Constraints”, “2.1 Architecturally Significant Design Decisions” and “4. Logical View”. In addition, it contains Sections “3. Use-Case View”, “4.4 Architecturally Significant Design Packages”, “4.4.2 UC5 - Submit to Archive”, “4.4.5 UC10 – Scheduler”, “4.4.5.3 Distributed Harvest Indexing”, “4.4.5.4 Store File Server”, “5. Process View”, “Process View Description”, “Deployment Diagram” and “Alternative Deployment Diagram”.

Interpretation: Recall that a section is included in a chunk found using Factor R3 if the majority of the participants who rated the importance of the section to the task, gave it a high rating value of 3 and above. The section is interpreted as of high importance to the task and was ‘needed’ by the majority of the group who rated the section, when they attempted the task.

C Group’s chunk found using Factor R3 comprises section related to Table of Contents, Section “2. Architectural Goals and Constraints” that contains the links to its sub-sections, sections related to how each of the key requirements (quality requirements) are achieved including the overview of the existing designs and technologies used to achieve these requirements, most of the sections related to use-case view (use-case diagram, description of actors and use-cases), most of the sections related to logical view (diagram and textual description of main logical components), all the sections related to process view and half of the sections related to deployment. **This shows that for a task related to the overview of the SA of WCT, a significant number of sections related to key requirements (quality requirements), use-case view, logical view, process view and deployment view were needed by the majority of the participants who rated them.**

Looking specifically at the industry participants performing WCT Task 1, in addition to sections on key quality requirements, the section on the overview of the categories of these key requirements (Section “2.1 Architecturally Significant Design Decisions”) was also needed by the majority of those who rated this section.

In terms of the logical view, all the sections on this topic were needed by the majority of the industry participants who rated them.

Interestingly, though sections related to use-case view (excluding section on use-case realizations) and logical deployment were needed, those related to use-case and deployment diagrams were not needed by the majority of I Group who rated them. Another observation is that all the sections on the process view were not needed by the majority of I Group who rated them. Verification with the raw data shows that these ‘not needed’ sections are excluded from the chunk due to not receiving high ratings from the majority of I Group who rated them. They are excluded not because of not being rated at all by any participant of I Group. Otherwise, we could not interpret that they were ‘not needed’ since we did not make assumption on the value that should be assigned to an unrated section.

One of the 4 academic participants did not provide rating for any section, when attempting WCT Task 1. This could have affected A Group’s chunk found using Factor AveR3F but not the chunks found using Factors R3 and AveR3. This is because Factor AveR3F is based on the majority of a group, and Factors R3 and AveR3 are based on the majority of those who rated the respective sections.

For the academic participants, in addition to sections on key quality requirements, use-case view, logical view, process view and deployment view, sections related to the brief introduction to architecturally significant use-cases of WCT (Section “4.4 Architecturally Significant Design Packages”) and detailed description of some use-cases were also needed by the majority of those who rated them.

Comparing to I Group’s chunk, A Group’s chunk is larger and most of the additional sections are sub-sections under Section “4.4 Architecturally Significant Design Packages” related to the detailed description of use-cases, sections on process view, deployment diagrams and use-case diagram. **This shows that the majority of A Group who rated the respective sections also needed sections that provide detailed description of use-cases, process view, deployment diagrams and use-case diagram, whereas (as verified with raw data) the majority of I Group who rated the respective sections did not require these sections, when they attempted WCT Task 1.**

The chunk found for a group using Factor R3 is usually superset of the group’s chunks found using Factors A, H, A|H|R3 and AveR3F, with the exclusion of one or two sections. The chunk found for a group using Factor R3 is very similar to the group’s chunk found using Factor AveR3. Recall that Factors A, H, A|H|R3 and AveR3F are based on the majority of a group. On the other hand, Factors R3 and AveR3 are based on only those who provided responses (in terms of ratings for these two factors). This could explain the situation where generally chunks found using Factors A, H, A|H|R3 and AveR3F are smaller compared to chunks found using Factors R3 and AveR3.

An obvious observation of the chunks (regardless of the group of participants) found using Factors R3 and AveR3 is that they contain sections which provide the description of the key quality requirements that influence the way the system is designed and how these requirements are achieved, including the overview of the existing designs and technologies used to achieve these requirements. These sections are Sections “2.1.1 Modularity/Plugability”, “2.1.2 Supportability”, “2.1.3 Security”, “2.1.4 User Interface” and “2.1.5 Resource Use”. They also contain Section “3.2 Use Cases”. These sections do not appear in the chunks discovered using Factors A, H, A|H|R3 and AveR3F.

Factor A|H|R3: C Group’s chunk comprises 10 sections: “Table of Contents”, “2.1 Architecturally Significant Design Decisions”, “3. Use-Case View”, “3.1 Actors”, “4.1 Overview”, “Logical High Level Solution Overview Diagram”, “4.4 Architecturally Significant Design Packages”, “6.3 Logical Deployment”, “Logical Deployment Description” and “Deployment Diagram”.

I group’s chunk contains only 7 sections: “1. Introduction”, “2. Architectural Goals and Constraints”, “2.1 Architecturally Significant Design Decisions”, Section “3.1 Actors”, “4.1 Overview”, “Logical High Level Solution Overview Diagram” and “6.3 Logical Deployment”.

The chunk found for A Group is almost double the size of I Group’s. Its sections are mostly different from those of I Group’s chunk. The chunk found for A Group includes Sections “Table of Contents”, “2.2 Architecturally Significant Open Source Products”, “3. Use-Case View”, “4.1 Overview”, “Logical High Level Solution Overview Diagram”, “4.4 Architecturally Significant Design Packages”, “4.4.5 UC10 – Scheduler”, the two sections related to process view, and 4 sections related to logical deployment. The only overlapping sections between the two chunks are Sections “4.1 Overview”, “Logical High Level Solution Overview Diagram” and “6.3 Logical Deployment”.

Interpretation: Recall that for composite Factor A|H|R3, if a participant provided one of the three forms of indications for a section (i.e. the answer is provided from or searched for in the section, or some information in the section is highlighted, or the section is given an importance rating of three and above), the section is regarded as ‘needed’ by the participant. If there is a minimum majority of participants in the group providing one form of indication or any combination of the three forms of indications, the section is included in a chunk found using Factor A|H|R3 and interpreted as ‘needed’ by the majority of the group.

Factor A|H|R3’s chunks are derived from more comprehensive responses of participants which involve the 3 different forms of indications as to whether a section was ‘needed’ during the attempt of the task. Unsurprisingly, a chunk found using Factor A|H|R3 covers more sections than a chunk found using Factor A or Factor H for the same group of participants.

C Group’s chunk found using Factor A|H|R3 shows that sections on the overview of the document, categories of key quality requirements, use-case diagram, description of actors, main logical components (textual description and diagram), brief introduction to architecturally significant use-cases of WCT, logical deployment diagram and its description, and deployment diagram were needed by the majority of the participants, to find out what is the SA of WCT.

The three overlapping sections between the chunks found for I and A Groups show that sections related to the main logical components (textual description and diagram) and the logical mapping of components to nodes (diagram) were needed by the majority of both the industry practitioner and academic groups. **These sections are mainly on logical components.**

The sections which are different in these chunks indicated that in addition to logical components, the majority of I and A Groups needed different information with regards WCT Task 1. The majority of I Group also needed the categories of key quality requirements and textual description of actors, whereas the majority of A Group also needed Table of Contents, open source products used, use-case diagram, brief introduction to architecturally significant use-cases of WCT, the management of the distributed nature of Harvesters, process view diagram and its textual description, logical deployment description, and deployment diagrams.

Factor AveR3: Chunks found using this factor have relatively larger sizes (17, 22, 19 for Groups I, A, C respectively) compared to the groups' chunks found using other factors (except for chunks found using Factor R3).

The chunk found using Factor AveR3 for each group of participants is very similar to the respective group's chunk found using Factor R3. I Group's chunk found using Factor AveR3 is superset of the group's chunk found using Factor R3, with Factor AveR3's chunk containing the additional sections of "8.1 Performance Requirements" and "9.3 Load Testing". C Group's chunk found using Factor AveR3 is also superset of the group's chunk found using Factor R3, with Factor AveR3's chunk containing additional sections of "2.1 Architecturally Significant Design Decisions" and "4.4 Architecturally Significant Design Packages". The chunk found using Factor AveR3 for A Group is identical to the group's chunk found using Factor R3.

Interpretation: Recall that a section with an average importance ratings of 3 and above (based on those who rated the section) made it into a chunk found using Factor AveR3. I Group's chunk found using Factor AveR3 shows that in addition to those sections in the group's chunk found using Factor R3, sections related to performance requirements and load testing were also needed by some of the industry participants who rated these sections. C Group's chunk found using Factor AveR3 shows that introductory sections on the categories of key quality requirements and architecturally significant use-cases were also needed by some of the participants in the combined group who rated these sections.

The chunk found for A Group using Factor AveR3 shows that sections on process view, most of the sections on key quality requirements, use-case view, logical view, and deployment view, section on brief introduction to architecturally significant use-cases of WCT (Section "4.4 Architecturally Significant Design Packages") and sections on detailed descriptions of some use-cases, were needed by some of the academic participants who rated these sections. In fact, these sections were needed by the majority of the academic participants who rated them, as shown by A Group's chunk found using Factor R3.

Factor AveR3F: This factor produces small chunk for each group. C Group's chunk contains Sections "Table of Contents", "4.1 Overview", "Logical High Level Solution Overview Diagram" and "5. Process View". I Group's chunk now includes only Section "Table of Contents", "2.1 Architecturally Significant Design Decisions", "3.1 Actors", "4.1 Overview" and "Logical High Level Solution Overview Diagram". The chunk for A Group comprises Sections "4.1 Overview" and "5. Process View".

Interpretation: Recall that for Factor AveR3F, chunk found using Factor AveR3 is filtered to arrive at new chunk containing only those sections rated by more than half of the participants in the group and with the average ratings of at least 3. The purpose is to remove those sections which have a small number of participants who rated them.

C Group’s chunk found using Factor AveR3F show that based on the average ratings contributed by the majority of the participants, organisation of the document, both the textual description and the diagram of the main logical components, and process view diagram, were needed for WCT Task 1. I and A Groups’ chunks found using Factor AveR3F show that organisation of the document, categories of key quality requirements, description of actors, both the textual description and the diagram of the main logical components were needed by I Group, and textual description of main logical components and process view diagram were needed by A Group, for the task.

6.2 Benchmarking Chunks against An Oracle Set

This section starts with a brief overview of the concept involved in WCT Task 1, which is SA. This is followed by the details of the oracle set for WCT Task 1 and the results of benchmarking the discovered chunks against this oracle set.

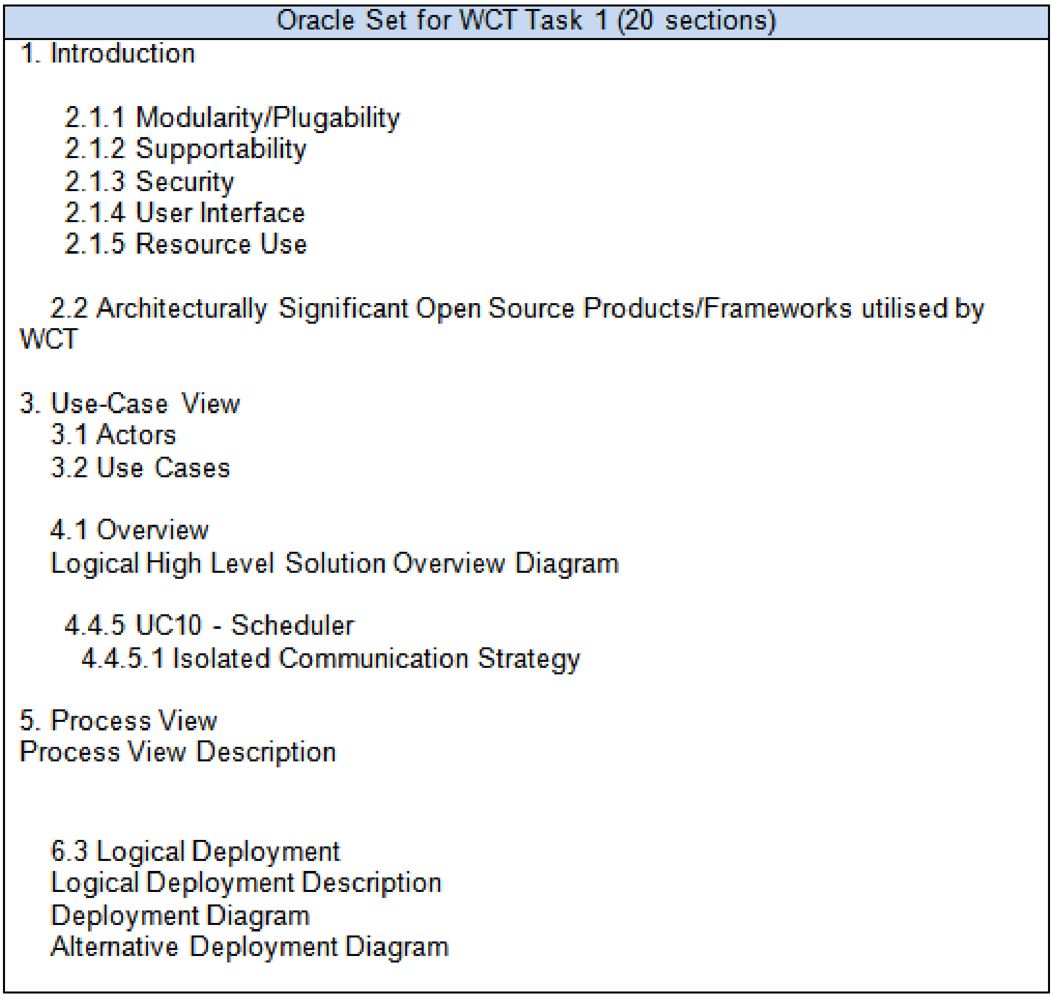
6.2.1 What is Software Architecture (SA)?

There are numerous definitions of the term SA. The Software Engineering Institute’s website contains a long list of these (Software Engineering Institute, 2012). Earlier work (Perry & Wolf, 1992) (Shaw & Garlan, 1996) (Garlan & Perry, 1995) (Bass et al., 2003) include these entities among others in describing a SA: elements, interactions or relationships among elements and its environment, structure(s) or form or organisation, style or pattern of composition and constraints, rationale, principles and guidelines, externally visible properties of those elements (behaviour) and interfaces. There are 3 main groups of architectural structures: module, component-and-connector and allocation (Bass et al., 2003). Other work includes architecture design decisions as a prominent element of SA (Grady, Rumbaugh, & Jacobson, 1999) (Bosch, 2004) (Babar et al., 2005) (Tang & Han, 2005) (Kruchten et al., 2006) (Avgeriou et al., 2007) (Farenhorst & van Vliet, 2008). In fact, some researchers take a more extreme view, seeing SA as just a set of architectural design decisions (Bosch, 2004) (Taylor, Medvidovic, & Dashofy, 2008).

These serve as guidelines in our construction of the oracle set for WCT Task 1 and ASM Task 1. We also included in the oracle set, sections that are helpful for understanding what the system is as that subsequently helps to understand its SA. This includes sections related to system functionalities which help in understanding what the system is.

6.2.2 The Oracle Set for WCT Task 1

The oracle set for WCT Task 1 comprises the sections of the WCT document agreed by both the researcher and a separate judge, as mandatory to obtain an overall picture of the SA of WCT. It comprises 20 of the 47 sections of the document, as shown in Table 6.2. In Table 6.1, these sections are denoted by yellow rows and their section IDs are marked with asterisks. This is to make it easier to compare the chunks found against the oracle set. The relevance of these sections to WCT Task 1 is explained next.



Oracle Set for WCT Task 1 (20 sections)
1. Introduction
2.1.1 Modularity/Plugability
2.1.2 Supportability
2.1.3 Security
2.1.4 User Interface
2.1.5 Resource Use
2.2 Architecturally Significant Open Source Products/Frameworks utilised by WCT
3. Use-Case View
3.1 Actors
3.2 Use Cases
4.1 Overview
Logical High Level Solution Overview Diagram
4.4.5 UC10 - Scheduler
4.4.5.1 Isolated Communication Strategy
5. Process View
Process View Description
6.3 Logical Deployment
Logical Deployment Description
Deployment Diagram
Alternative Deployment Diagram

Table 6.2: The Oracle Set for WCT Task 1

Section "1. Introduction" provides the purpose and scope of the document, and explains what Web Curator Tool is. It also includes definitions, acronyms and abbreviations used throughout the document. Knowing what is the system being described by the AD is important to understanding its SA. As a result, we included this section in the oracle set.

Section "3. Use-Case View" contains use-case diagram which shows the high-level use-cases and actors interactions. **Section "3.1 Actors"** contains the description of the actors of WCT system. **Section "3.2 Use Cases"** provides the explanation on the use-cases found in the use-case diagram. These three sections of the WCT software architecture document are related to the functionalities of WCT which are useful for understanding what the system is and subsequently its SA.

The following 5 sections describe the key quality requirements and the design decisions made to achieve them: **Section "2.1.1 Modularity/Plugability"** explains the use of interface, adapter design pattern, dependency injection framework (Spring framework) to support modularity; **Section "2.1.2 Supportability"** explains the support of multiple database systems by using Hibernate with SQL-92 standard and HQL (to meet database independence); **Section "2.1.3 Security"** explains the use of Acegi Security Framework (authentication and authorization) for security; **Section "2.1.4 User Interface"** explains the use of MVC (Model View Controller) pattern provided by Spring Framework and **Section "2.1.5 Resource Use"** explains that separate machine deployment of web harvesting subsystem (to distribute the load) will be supported using SOAP RPC for communication with remote devices. These quality requirements and the related design decisions affected the architectural design of WCT and their respective sections in the document are needed for WCT Task 1.

Section "2.2 Architecturally Significant Open Source Products/Frameworks utilised by WCT" provides general description of all the open source products or frameworks used by WCT. They are Hibernate, Spring Application Framework, Acegi Security System, Apache Axis, Apache commons logging, Quartz and Heritrix. These frameworks affected the architecture design of WCT. This section is needed for WCT Task 1 because it contains all the external products or frameworks needed by WCT. Some of these (Apache commons logging and Quartz) cannot be found in other sections of the document.

Section "4.1 Overview" gives an overview of the logical view of the SA of WCT. It explains the main logical components, end-user access and functionalities of the system, and the interface layer to all external systems and tools required by WCT. **Section "Logical High Level Solution Overview Diagram"** contains the high-level logical components diagram. This diagram shows the sub-components within the main logical components and the relationships between the components. The information in this section complements the information in Section "4.1 Overview". Both of these sections contain information about the main logical components of the system and their relationships, which is an important aspect of the architectural design of WCT. Therefore, both sections are needed for WCT Task 1.

Section "4.4.5 UC10 – Scheduler" contains the diagram related to the management of the distributed nature of Harvesters. This is important to understand the scheduling of harvests in a distributed environment. **Section "4.4.5.1 Isolated Communication Strategy"** presents the design decision and the rationales to have Harvester Manager and Agent components isolating all the distributed communication strategy. This section also explains the use of SOAP over HTTP for the distributed communication and its benefits. Section "4.4.5 UC10 – Scheduler" and "4.4.5.1 Isolated Communication Strategy" are needed for WCT Task 1 because they are related to the architecture design to support distributed Harvester, a critical element in WCT.

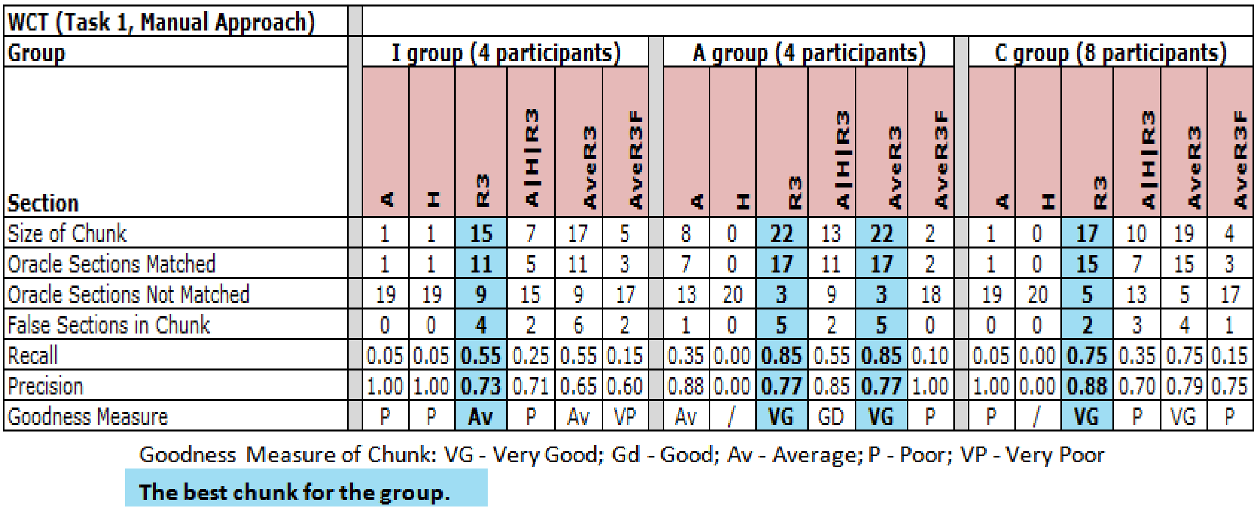
Section "5. Process View" contains the process view diagram. This diagram depicts the main processes and threads in WCT and their relationships. **Section "Process View Description"** provides the description of the elements in the process view diagram. Both the sections are important to understand the component-and-connector structure (or runtime architectural structure) of WCT, and are required for WCT Task 1.

The next 4 sections are parts of the deployment view. They are **Section "6.3 Logical Deployment"** which contains a diagram that shows the logical mapping of components to the nodes for WCT; **Section "Logical Deployment Description"** that explains that the logical grouping in the previous section would result in 3 deployment packages (components) delivered as standalone WAR file; **Section "Deployment Diagram"** that contains a deployment diagram that shows the deployment of the components to a single application server; **Section "Alternative Deployment Diagram"** that contains a deployment diagram that shows the alternative deployment of the components to separate application servers. These sections are important to understand the allocation structure of WCT and therefore required for WCT Task 1.

6.2.3 Benchmarking Results

Refer to Section “5.7 Benchmarking of Chunks against An Oracle Set” of Chapter 5 for all the concepts and terms needed to understand this section.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for WCT Task 1 are given in Table 6.3.



WCT (Task 1, Manual Approach)																		
Group	I group (4 participants)						A group (4 participants)						C group (8 participants)					
Section	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
Size of Chunk	1	1	**15**	7	17	5	8	0	**22**	13	**22**	2	1	0	**17**	10	19	4
Oracle Sections Matched	1	1	**11**	5	11	3	7	0	**17**	11	**17**	2	1	0	**15**	7	15	3
Oracle Sections Not Matched	19	19	**9**	15	9	17	13	20	**3**	9	**3**	18	19	20	**5**	13	5	17
False Sections in Chunk	0	0	**4**	2	6	2	1	0	**5**	2	**5**	0	0	0	**2**	3	4	1
Recall	0.05	0.05	**0.55**	0.25	0.55	0.15	0.35	0.00	**0.85**	0.55	**0.85**	0.10	0.05	0.00	**0.75**	0.35	0.75	0.15
Precision	1.00	1.00	**0.73**	0.71	0.65	0.60	0.88	0.00	**0.77**	0.85	**0.77**	1.00	1.00	0.00	**0.88**	0.70	0.79	0.75
Goodness Measure	P	P	**Av**	P	Av	VP	Av	/	**VG**	GD	**VG**	P	P	/	**VG**	P	VG	P

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 6.3: Attributes of Chunks (WCT Task 1)

For C Group, the goodness of chunks found using Factors R3 and AveR3 are ‘Very Good’. The chunks found using Factors A, A|H|R3 and AveR3F are ‘Poor’ and the chunk found using Factor H is ‘Very Poor’. We assessed the composition of the two ‘Very Good’ chunks to decide on the best chunk for C Group. The composition of Factor R3’s chunk (recall = 0.75, precision = 0.88, size = 17, goodness = ‘Very Good’) and Factor AveR3’s chunk (recall = 0.75, precision = 0.79, size = 19, goodness = ‘Very Good’) are very similar. They contain the same fifteen of the twenty sections in the oracle set. They miss five oracle set’s sections: “1.Introduction”, “2.2 Architecturally Significant Open Source Product”, “4.4.5 UC10 – Scheduler”, “4.4.5.1 Isolated Communication Strategy” and “Alternative Deployment Diagram”. Both chunks include these two false sections: “Table of Contents” and “2. Architectural Goals and Constraints”. Factor AveR3’s chunk has two additional false sections (“2.1 Architecturally Significant Design Decisions” and “4.4 Architecturally Significant Design Packages”). With almost the same composition, equal recall and higher precision, Factor R3’s chunk is the best chunk for C Group.

The most basic chunk for C Group (recall = 0.05, precision = 1.00, size = 1, goodness = ‘Poor’) is found using Factor A. It comprises only Section “4.1 Overview” that details the main logical components of WCT. This chunk misses a significant number of sections from the oracle set and therefore is not really useful for WCT Task 1.

The goodness of chunks found for I Group are: ‘Average’ for chunks found using Factors R3 and AveR3, ‘Poor’ for chunks found using Factors A, H and A|H|R3, and ‘Very Poor’ for chunk found using Factor AveR3F. The composition of the two ‘Average’ chunks are very similar, containing the same 11 sections from the oracle set and excluding the same 9 sections (“1. Introduction”, “2.2 Architecturally Significant Open Source Products”, “3. Use-Case View”, “4.4.5 UC10 – Scheduler”, “4.4.5.1 Isolated Communication Strategy”, “5. Process View”, “Process View Description”, “Deployment Diagram”, and “Alternative Deployment Diagram”). Factor R3’s chunk includes four false sections (“Table of Contents”, “2. Architectural Goals and Constraints”, “2.1 Architecturally Significant Design Decisions”, and “4. Logical View”). Factor AveR3’s chunk includes two other false sections (“8.1 Performance Requirements” and “9.3 Load Testing”). With almost the same composition, equal recall but higher precision, Factor R3’s chunk (recall = 0.55, precision = 0.73, size = 15, goodness = ‘Average’) is better than Factor AveR3’s chunk (recall = 0.55, precision = 0.65, size = 17, goodness = ‘Average’) and therefore the best chunk for I Group. The most basic chunk for I Group is identical to the most basic chunk for C Group. Both are found using Factor A.

For A Group, chunks found using Factors R3 and AveR3 are identical (recall = 0.85, precision = 0.77, size = 22, goodness = ‘Very Good’). With the highest goodness measure, this chunk is the best chunk for A Group. It contains a significant number (17) of and misses only three of the sections in the oracle set. It includes five false sections (“Table of Contents”, “4.4 Architecturally Significant Design Packages”, “4.4.2 UC5 – Submit to Archive”, “4.4.5.3 Distributed Harvest Indexing” and “4.4.5.4 Store File Server”). Chunks found using Factors A|H|R3, A and AveR3F for A Group are ‘Good’, ‘Average’, ‘Poor’, respectively. When Factor H is used, no chunk is found. The most basic chunk for A Group is found using Factor AveR3F. Though it comprises two important sections (“4.1 Overview” and “5. Process View”) for WCT Task 1, it misses many other sections in the oracle set.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C and I Groups is Factor R3. For A Group the best factor is Factor R3 or Factor AveR3.

Commonly-missed Oracle Set’s Sections and Common False Sections: Refer to Section “5.7.6 Determination of the Commonly-Missed Oracle Set’s Sections and Common False Sections“ of Chapter 5 on how common oracle set’s sections missed by a group of participants, and common false sections for a group of participants are determined.

For each group of participants, we try to give possible reason for commonly-missed oracle set’s sections and common false sections. If no reason is given, it means that we cannot think of any plausible reason other than a commonly-missed oracle set’s section was generally ‘not needed’, and a commonly-included false section was generally ‘needed’, by the respective group of participants when they attempted the task.

For WCT Task 1, all the sections in the oracle set were rated by at least one participant in each group.

The oracle set’s sections commonly missed by C Group are: “1.1 Introduction”, “2.2 Architecturally Significant Open Source Products”, “4.4.5 UC10 – Scheduler”, “4.4.5.1 Isolated Communication Strategy” and “Alternative Deployment Diagram” (missed by all 6 chunk-identification factors); Section 2.1.1 to 2.1.5 on quality requirements, Sections “3.2 Use Cases” and “Process View Description” (missed by 4 factors).

The oracle set’s sections commonly missed by I Group are: “2.2 Architecturally Significant Open Source Products”, “3. Use-Case View”, “4.4.5 UC10 – Scheduler”, “4.4.5.1 Isolated Communication Strategy”, “5. Process View”, “Process View Description”, “Deployment Diagram” and “Alternative Deployment Diagram” (missed by all 6 factors); Section “1. Introduction” (missed by 5 factors); Section 2.1.1 to 2.1.5 on quality requirements, Sections “3.2 Use Cases” and “Logical Deployment Description” (missed by 4 factors).

The oracle set’s sections commonly missed by A Group are: “1. Introduction” and “4.4.5.1 Isolated Communication Strategy” (missed by all 6 factors); Section 2.1.1 to 2.1.5 on quality requirements, Sections “2.2 Architecturally Significant Open Source Products”, “3.1 Actors” and “3.2 Use Cases” (missed by 4 factors).

Section “1.1 Introduction” gives an overview of the document and WCT. The common exclusion of this section by C and A Groups could be due to this section does not specifically contain information on the SA aspect of WCT. The common exclusion of Sections “4.4.5 UC10 – Scheduler” (by C and I Groups) and “4.4.5.1 Isolated Communication Strategy” (by all three groups) could be due to these sections focus mainly on just one element of WCT, which is the Harvester. In fact none of the chunks contains Section “4.4.5.1 Isolated Communication Strategy” which is related to the isolation of all the distributed communication strategy by the Harvester Manager and Agent components. This shows that this information was generally not needed by the participants when they attempted WCT Task 1.

As for the common exclusion of Section “2.2 Architecturally Significant Open Source Products” and Section 2.1.1 to 2.1.5 on quality requirements (by all the groups of participants), we could not think of any plausible reason other than they were generally not needed by these groups of participants when they attempted WCT Task 1.

There is only one common false section for each group: “Table of Contents” for C Group, “2.1 Architecturally Significant Design Decision” for I Group and “4.4 Architecturally Significant Design Packages” for A Group. Section “Table of Contents” is related to the organisation of the document and does not contain specific information for WCT Task 1. It was needed by C Group probably for exploring the document to find the required information. Section “2.1 Architecturally Significant Design Decisions” lists the categories of key quality requirements and Section “4.4 Architecturally Significant Design Packages” briefly introduces architecturally significant use-cases of WCT. These introductory sections do not contain specific information for WCT Task 1. They were needed by the respective group of participants probably for getting an overview of the subsequent materials.

6.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for WCT Task 1 for each group of participants. Generally, there are two clusters of chunks found for each group of participants. Cluster 1 consists of the smaller chunks found using Factors A, H, A|H|R3 and AveR3F. These factors are based on the preference of the majority of the participants in a group. The most basic chunks (i.e. chunk with the smallest number of section) for C Group (found using Factor A) and I Group (found using Factors A and H) are identical. It comprises only Section “4.1 Overview” which describes the main logical components. The most basic chunk for A Group is found using Factor AveR3F, comprises Section “5. Process View” in addition to Section “4.1 Overview”.

Cluster 2 consists of the larger chunks found using Factors R3 and AveR3. These factors are based on the preference of the majority of those who rated. These larger chunks with the exclusion of one or two sections are usually supersets of the chunks found using Factors A, H, A|H|R3 and AveR3F for the same group. The chunks found using Factors R3 and AveR3 for a group of participants, are very similar if not identical to each other (very similar for C and I Groups, and identical for A Group).

Part 1 of Table 6.4 summarises the main results of benchmarking of chunks. The number of participants in I and Groups are the same, which is 4 each. All the participants in each group provided some responses in terms of where the needed information (answers) can be found and highlighting of information they think relevant to the task. One academic participant did not rate sections visited in terms of their importance to the assigned task.

In terms of how 'good' the chunks are for WCT Task 1, C Group has 2 chunks with 'Very Good', 3 with 'Poor' and 1 with 'Very Poor' goodness measures. I Group has 2 chunks with 'Average', 3 with 'Poor' and 1 with 'Very Poor' goodness measures. A Group has 2 chunks with 'Very Good' goodness measures, 1 each of 'Good', 'Average', 'Poor' and 'Very Poor' goodness measures.

The best chunks for C and I Groups are found using Factor R3 (with goodness measure of 'Very Good' for Group C's chunk and 'Average' for Group I's chunk). The best chunk for A Group is found using Factor R3 or Factor AveR3. The factor that produces the best chunk for a group is the best chunk-identification factor for the group.

Part 2 of Table 6.4 summarises the information or sections needed by each group of participants. This is based on the groups' chunks found using Factor A|H|R3. Refer to Section "5.8 Information Needed" in Chapter 5 for our reason. C and I Groups' chunks found using Factor A|H|R3 are 'Poor', and A Group's chunk is 'Good'.

As shown in Table 6.4, both I and A Groups needed information on main logical components (both the textual description and diagram) and logical deployment diagram, when they attempted WCT Task 1. I Group also needed overview of the document and WCT (Section "1. Introduction"), organisation of the document (Section "2. Architectural Goals and Constraints"), categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision") and description of actors. A Group also needed organisation of the document (Section "Table of Contents"), open source products used, use-case diagram (Section "3. Use-Case View"), brief introduction to architecturally significant use-cases of WCT (Section "4.4 Architecturally Significant Design Package"), management of distributed nature of Harvesters (Section "4.4.5 UC10 - Scheduler"), process view diagram and its textual description, logical deployment description, and both deployment diagrams.

For both the industry practitioners and academics, the most notable information needed was on the logical components (Section "4.1 Overview", "Logical High Level Solution Overview Diagram" and "6.3 Logical Deployment"). The academics covered slightly more variation of topics (in particular more diversified architectural views which include the process view). The academics also required more comprehensive (or broader coverage of) information on deployment view (in particular the textual description of logical deployment and both deployment diagrams). They also required certain more detailed information (in particular use-case related to Scheduler), in supporting their understanding of the SA of WCT.

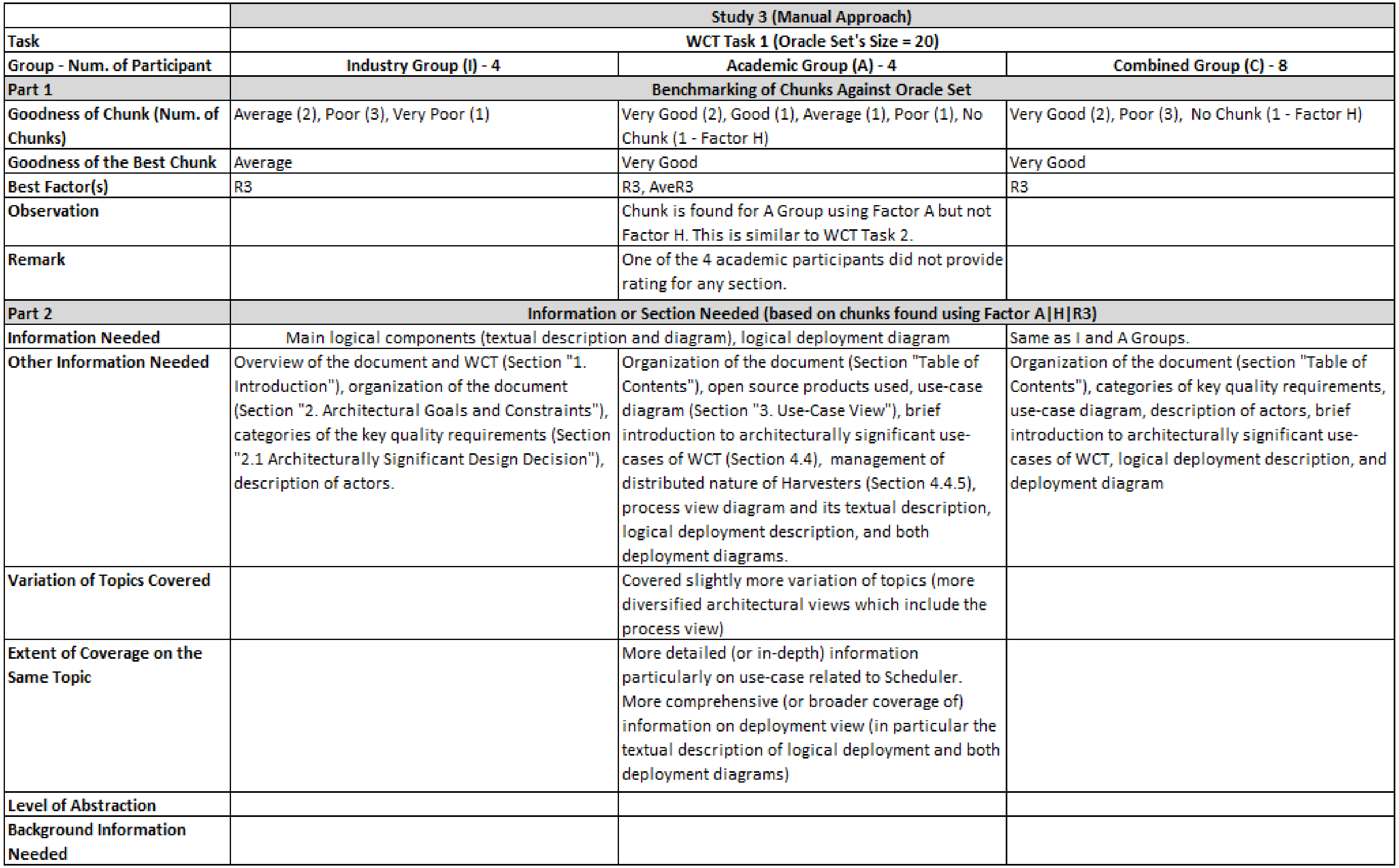
When both I and A Groups are combined into C Group, the following information was needed for WCT Task 1: organisation of the document (Section "Table of Contents"), categories of key quality requirements, use-case diagram, description of actors, brief introduction to architecturally significant use-cases of WCT, logical deployment description, and deployment diagram, in addition to main logical components (both the textual description and diagram) and logical deployment diagram.

C Group’s chunk shows that for a task related to getting an overview of the SA of WCT, in addition to Table of Contents and some introductory sections (Section "2.1 Architecturally Significant Design Decision" and Section “4.4 Architecturally Significant Design Package”), a number of sections on use-case view, logical view and logical deployment were needed.

6.4 Summary

This chapter describes the chunking of architectural information for WCT Task 1. As with ASM Task 1 (i.e. a similar task performed on the second AD), the results show the discovery of usage-based chunks for the task of getting an overview of the SA of a system.

Chapter 13 contains the comparison of the results for WCT Task 1 and ASM Task 1, and the other tasks; the summary of the overall results for Studies 2 and 3; and threats to validity of findings. Chapter 9 presents the chunking of architectural information for ASM Task 1. The next chapter details the chunking of architectural information for WCT Task 2.



	Study 3 (Manual Approach)		
Task	WCT Task 1 (Oracle Set's Size = 20)		
Group - Num. of Participant	Industry Group (I) - 4	Academic Group (A) - 4	Combined Group (C) - 8
Part 1	Benchmarking of Chunks Against Oracle Set		
Goodness of Chunk (Num. of Chunks)	Average (2), Poor (3), Very Poor (1)	Very Good (2), Good (1), Average (1), Poor (1), No Chunk (1 - Factor H)	Very Good (2), Poor (3), No Chunk (1 - Factor H)
Goodness of the Best Chunk	Average	Very Good	Very Good
Best Factor(s)	R3	R3, AveR3	R3
Observation		Chunk is found for A Group using Factor A but not Factor H. This is similar to WCT Task 2.	
Remark		One of the 4 academic participants did not provide rating for any section.	
Part 2	Information or Section Needed (based on chunks found using Factor A|H|R3)		
Information Needed	Main logical components (textual description and diagram), logical deployment diagram		Same as I and A Groups.
Other Information Needed	Overview of the document and WCT (Section "1. Introduction"), organization of the document (Section "2. Architectural Goals and Constraints"), categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision"), description of actors.	Organization of the document (Section "Table of Contents"), open source products used, use-case diagram (Section "3. Use-Case View"), brief introduction to architecturally significant use-cases of WCT (Section 4.4), management of distributed nature of Harvesters (Section 4.4.5), process view diagram and its textual description, logical deployment description, and both deployment diagrams.	Organization of the document (section "Table of Contents"), categories of key quality requirements, use-case diagram, description of actors, brief introduction to architecturally significant use-cases of WCT, logical deployment description, and deployment diagram
Variation of Topics Covered		Covered slightly more variation of topics (more diversified architectural views which include the process view)	
Extent of Coverage on the Same Topic		More detailed (or in-depth) information particularly on use-case related to Scheduler. More comprehensive (or broader coverage of) information on deployment view (in particular the textual description of logical deployment and both deployment diagrams)	
Level of Abstraction			
Background Information Needed			

Table 6.4: Summary of Results for WCT Task 1

7 Chunking of Architectural Information (WCT Task 2)

This chapter presents the chunking of architectural information for the second information-seeking task for the Web Curator Tool (WCT Task 2) software architecture document, for Study 2. The task was:

> "As a developer you need to change the Web Curator Tool to make use of a different digital archive system. You want to know what needs to be done and which parts of Web Curator Tool will be affected."

The results presented in this chapter are based on the responses of 8 participants who completed WCT Task 2. Five were classified as industry participants and three were classified as academic participants. We recruited 4 industry practitioners and 4 academics for this task just as what we did for the other tasks. Further inspection of the participants' data showed that one of the academics has more years of SA industry experience than teaching experience. This academic was therefore classified as industry participant.

This chapter adopts the same structure as Chapter 6. Chapter 5 provides the foundation to understand this chapter and should be read prior to reading this chapter.

7.1 Identification and Interpretation of Chunks

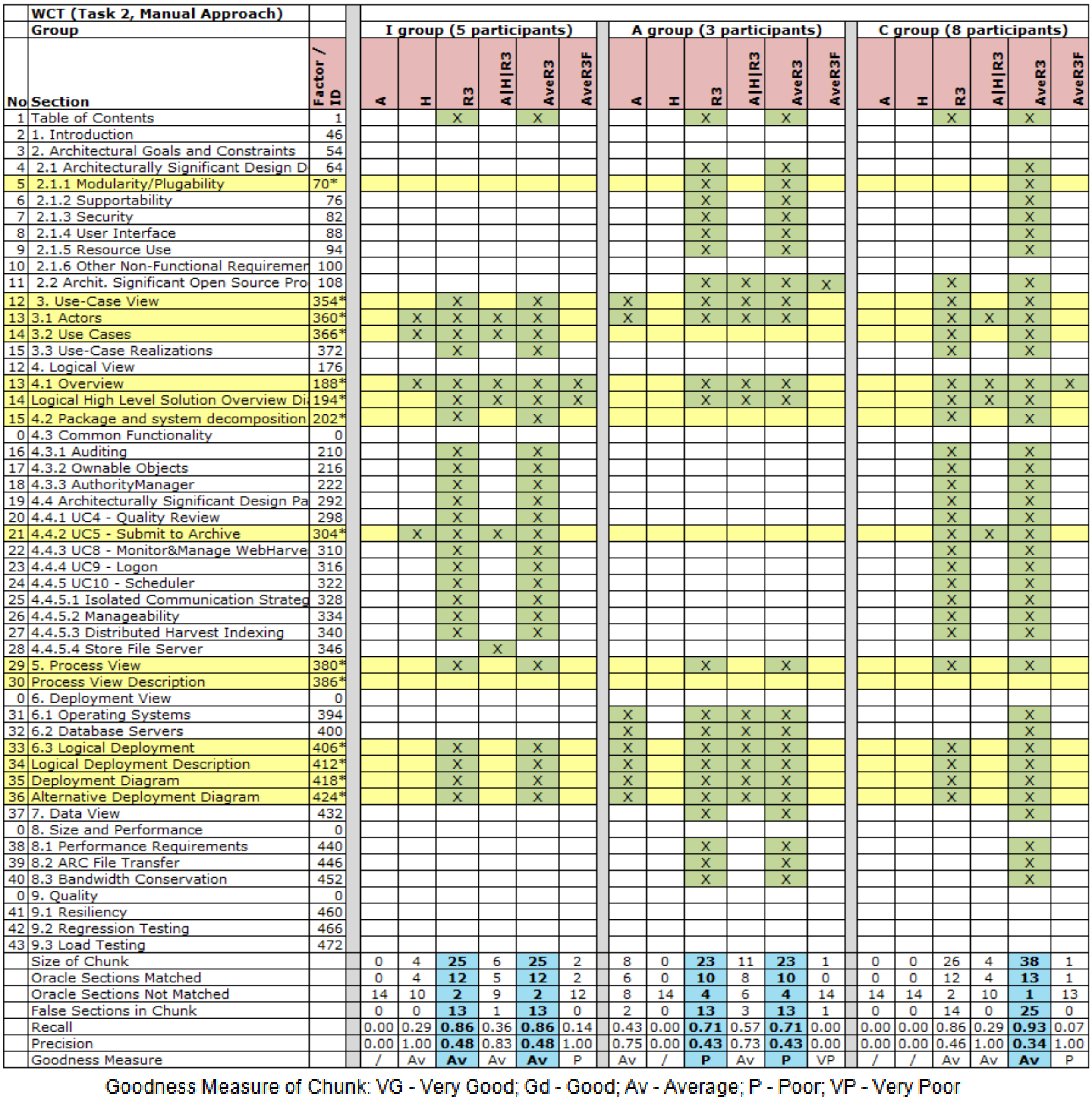
We used the 6 chunk-identification factors to identify the chunks for this task. This was done for the 3 groups of participants: the industry practitioner (I) group, the academic (A) group, and the combined (C) group of the previous two.

Table 7.1 shows the chunks discovered for WCT Task 2, for each group of participants. Refer to Section "6.1 Identification and Interpretation of Chunks" in Chapter 6 on how to interpret the chunks in the table. Recall that the size of a chunk is the number of the document's sections that the chunk comprises.

Factor A: No chunk is discovered for C and I Groups when Factor A is used. A Group's chunk contains 8 sections: "3. Use-Case View", "3.1 Actors" and all the 6 sub-sections under Section "6. Deployment View".

Interpretation: One possible reason no chunk is found for I Group is that the majority of this group highlighted relevant information in the sections instead of specifying the sections of the document where each of their bullet-point answers was provided or searched from (which is used in Factor A). This speculation is supported by the finding of I Group's chunk using Factor H. Since industry participants constituted 62.5% of C Group, the speculation that highlighting relevant information is preferred could also be the reason of the no chunk situation for C Group.

The chunk found for A Group using Factor A shows that the majority of the academic participants needed use-case diagram, description of actors and deployment view in their search for the answer to WCT Task 2.



	WCT (Task 2, Manual Approach)																				
	Group		I group (5 participants)						A group (3 participants)						C group (8 participants)					
No	Section	Factor / ID	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
1	Table of Contents	1			X		X				X		X				X		X	
2	1. Introduction	46																		
3	2. Architectural Goals and Constraints	54																		
4	2.1 Architecturally Significant Design D	64									X		X						X	
5	2.1.1 Modularity/Plugability	70*									X		X						X	
6	2.1.2 Supportability	76									X		X						X	
7	2.1.3 Security	82									X		X						X	
8	2.1.4 User Interface	88									X		X						X	
9	2.1.5 Resource Use	94									X		X						X	
10	2.1.6 Other Non-Functional Requiremen	100																		
11	2.2 Archit. Significant Open Source Pro	108									X	X	X	X			X		X	
12	3. Use-Case View	354*			X		X		X		X	X	X				X		X	
13	3.1 Actors	360*		X	X	X	X		X		X	X	X				X	X	X	
14	3.2 Use Cases	366*		X	X	X	X										X		X	
15	3.3 Use-Case Realizations	372			X		X										X		X	
12	4. Logical View	176																		
13	4.1 Overview	188*		X	X	X	X	X			X	X	X				X	X	X	X
14	Logical High Level Solution Overview Di	194*			X	X	X	X			X	X	X				X	X	X	
15	4.2 Package and system decomposition	202*			X		X										X		X	
0	4.3 Common Functionality	0																		
16	4.3.1 Auditing	210			X		X										X		X	
17	4.3.2 Ownable Objects	216			X		X										X		X	
18	4.3.3 AuthorityManager	222			X		X										X		X	
19	4.4 Architecturally Significant Design Pa	292			X		X										X		X	
20	4.4.1 UC4 - Quality Review	298			X		X										X		X	
21	4.4.2 UC5 - Submit to Archive	304*		X	X	X	X										X	X	X	
22	4.4.3 UC8 - Monitor&Manage WebHarve	310			X		X										X		X	
23	4.4.4 UC9 - Logon	316			X		X										X		X	
24	4.4.5 UC10 - Scheduler	322			X		X										X		X	
25	4.4.5.1 Isolated Communication Strateg	328			X		X										X		X	
26	4.4.5.2 Manageability	334			X		X										X		X	
27	4.4.5.3 Distributed Harvest Indexing	340			X		X										X		X	
28	4.4.5.4 Store File Server	346				X														
29	5. Process View	380*			X		X				X		X				X		X	
30	Process View Description	386*																		
0	6. Deployment View	0																		
31	6.1 Operating Systems	394							X		X	X	X						X	
32	6.2 Database Servers	400							X		X	X	X						X	
33	6.3 Logical Deployment	406*			X		X		X		X	X	X				X		X	
34	Logical Deployment Description	412*			X		X		X		X	X	X				X		X	
35	Deployment Diagram	418*			X		X		X		X	X	X				X		X	
36	Alternative Deployment Diagram	424*			X		X		X		X	X	X				X		X	
37	7. Data View	432									X		X						X	
0	8. Size and Performance	0																		
38	8.1 Performance Requirements	440									X		X						X	
39	8.2 ARC File Transfer	446									X		X						X	
40	8.3 Bandwidth Conservation	452									X		X						X	
0	9. Quality	0																		
41	9.1 Resiliency	460																		
42	9.2 Regression Testing	466																		
43	9.3 Load Testing	472																		
	Size of Chunk		0	4	**25**	6	**25**	2	8	0	**23**	11	**23**	1	0	0	26	4	**38**	1
	Oracle Sections Matched		0	4	**12**	5	**12**	2	6	0	**10**	8	**10**	0	0	0	12	4	**13**	1
	Oracle Sections Not Matched		14	10	**2**	9	**2**	12	8	14	**4**	6	**4**	14	14	14	2	10	**1**	13
	False Sections in Chunk		0	0	**13**	1	**13**	0	2	0	**13**	3	**13**	1	0	0	14	0	**25**	0
	Recall		0.00	0.29	**0.86**	0.36	**0.86**	0.14	0.43	0.00	**0.71**	0.57	**0.71**	0.00	0.00	0.00	0.86	0.29	**0.93**	0.07
	Precision		0.00	1.00	**0.48**	0.83	**0.48**	1.00	0.75	0.00	**0.43**	0.73	**0.43**	0.00	0.00	0.00	0.46	1.00	**0.34**	1.00
	Goodness Measure		/	Av	**Av**	Av	**Av**	P	Av	/	**P**	Av	**P**	VP	/	/	Av	Av	**Av**	P

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

Table 7.1: The Chunks for WCT Task 2 Identified Using Different Factors

Factor H: No chunk is discovered for C Group nor A Group when Factor H (which is based on highlighted information) is used. The chunk found for I Group comprises 4 sections: “3.1 Actors”, “3.2 Use Cases”, “4.1 Overview” and “4.4.2 UC5 – Submit to Archive”.

Interpretation: Inspection on the raw data shows that no chunk is found for A Group because more than half (2 out of 3) of the academic participants did not highlight information in the document and therefore none of the section could meet the inclusion criteria. This could be the reason no chunk is found for C Group.

I Group’s chunk shows that the majority of the industry participants needed the following information when attempting WCT Task 2: descriptions of actors and use-cases, textual description of main logical components and detailed description of the ‘Submit to Archive’ use-case.

Chunk is found for A Group using Factor A but not Factor H. This is the same as WCT Task 1. The opposite happens for I Group, which is, chunk is found using Factor H but not Factor A. One possible reason is that the majority of A Group preferred to specify the sections of the document where each of their bullet-point answers was provided or searched from, instead of highlighting the relevant information in the sections, and vice versa for I Group. Nevertheless, for I Group involved in ASM Task 1, chunk is found using Factor A but not Factor H.

When I Group's chunk found using Factor H is compared to A Group's chunk found using Factor A, the only thing they overlapped is on descriptions of actors. This shows the two groups also needed other information when dealing with WCT Task 2.

Factor R3: The chunks discovered using Factor R3 are relatively larger for all the groups (comprising 26, 25, 23 sections for C, I, A Groups respectively) when compared to the chunks found using the rest of the factors (with the exception of Factor AveR3). C Group's chunk includes Section "Table of Contents", Section "2.2 Architecturally Significant Open Source Products", Section "3. Use-Case View" and all its 3 sub-sections, all the 16 sub-sections (except "4.4.5.4 Store File Server") of Section "4. Logical View", Section "5. Process View", Section "6.3 Logical Deployment" and all its 3 sub-sections. I Group's chunk is very similar to C Group's chunk with the exclusion of Section "2.2 Architecturally Significant Open Source Products".

Almost half of the sections in chunks discovered for I and A Groups overlapped. These overlapping sections are: "Table of Contents", "3. Use-Case View", "3.1 Actors", the section on main logical components (textual description and diagram), "5. Process View", "6.3 Logical Deployment" and all its 3 sub-sections. More than half of the sections in these chunks are exclusive to each other. Those that appeared in I Group's chunk but not A Group's chunk are those sections related to description of use-cases, use-case realizations, packages of Java code, all the 3 sections on common functionalities, and all but one of the sections of "4.4 Architecturally Significant Design Packages". Those that appear in A Group's chunk but do not appear in I Group's chunk are: sections related to key quality requirements namely Section "2.1 Architecturally Significant Design Decisions" and all of its sub-sections (except "2.1.6 Other Non-Functional Requirements"), Sections "2.2 Architecturally Significant Open Source Product", "6.1 Operating Systems", "6.2 Database Servers", "7. Data View" and all the sections related to size and performance (Section 8).

Interpretation: Recall that for Factor R3, if a section obtained high rating values of 3 and above from the majority of the group who rated the section, it is interpreted as the section is of high importance to the task and therefore 'needed' by the majority of the group who rated the section, when attempting the task.

One of the 3 academic participants did not provide rating for any section, when attempting WCT Task 2. This could have affected A Group's chunk found using Factor AveR3F but not the chunks found using Factor R3 and AveR3. This is because Factor AveR3F is based on the majority of a group, and Factors R3 and AveR3 are based on the majority of those who rated the respective sections.

C Group's chunk found using Factor R3 shows that for WCT Task 2, sections related to the Table of Contents, open source products used, use-case view, logical view, packages of Java code, common functionalities, detailed descriptions of all use-cases, process view diagram and logical deployment (including deployment diagrams) were needed by the majority of the participants who rated them.

The overlapping sections in the I and A Groups' chunks show that Table of Contents, use-case diagram, description of actors, textual description of main logical components and the corresponding diagram, process view diagram and logical deployment (including deployment diagrams) were needed by the majority of each of the 2 groups who rated these sections.

The main sections which are different between the two chunks are both related to the requirements of WCT. Nevertheless, **the extra sections for I Group contain more specific detailed information on how requirements are met (common functionality and detailed description on all use-cases, i.e. all the sub-sections of 4.3, Section 4.4 and almost all of its sub-sections). On the other hand, the extra sections for A Group contain high-level overview of how key quality requirements are met including the overview of the existing designs (such as pattern) and technologies used to achieve them (Section 2.1 and all its sub-sections). Another obvious difference is A Group's chunk also includes all the sub-sections related to size and performance, whereas I Group's chunk does not.**

Factor A|H|R3: Using this factor, chunk with size 4, 6, 11 are found for group C, I and A, respectively. C Group's chunk contains Sections "3.1 Actors", "4.1 Overview", "Logical High Level Solution Overview Diagram" and "4.4.2 UC5 – Submit to Archive". I Group's chunk found using Factor A|H|R3 is very similar to the group's chunk found using Factor H, with the extra sections of "Logical High Level Solution Overview Diagram" and "4.4.5.4 Store File Server". A Group's chunk found using Factor A|H|R3 closely resembled the group's chunk found using Factor A, with the addition of Sections "2.2 Architecturally Significant Open Source Product", "4.1 Overview" and "Logical High Level Solution Overview Diagram".

Interpretation: C Group's chunk shows that the majority of the participants needed information related to description of actors, main logical components (textual description and diagram) and detailed description of 'Submit to Archive' use-case, when they attempted WCT Task 2. Looking specifically at I Group's chunk reveals that the majority of the industry participants needed information related to the description of actors and use-cases, main logical components (textual description and diagram), detailed description of 'Submit to Archive' use-case and store file server. A Group's chunk shows that the majority of the academic participants needed information related to open source products used, use-case diagram, description of actors, main logical components (textual description and diagram) and deployment view (all sections) for WCT Task 2.

Factor AveR3: C Group's chunk found using Factor AveR3 is made up of those sections from I and A Groups' chunks found using the same factor. It is the largest chunk with the size of 38 sections. I Group's chunk found using Factor AveR3 is identical to the group's chunk found using Factor R3. A Group's chunks found using Factor AveR3 and R3 are also identical.

Interpretation: C Group's chunk covers nearly 90% of the number of sections in the document. It excludes Section "1. Introduction" (overview of the document and WCT), Section "2. Architectural Goals and Constraints" (which contains the titles of its sub-sections), Section "2.1.6 Other Non-Functional Requirements" (which contains little information), Section "4. Logical View" (which gives an overview of what are contained in Section 4 of the document), sections on store file server, process view description, and sections on resiliency and testing. In other words, the rest of the sections of the document as included by C Group's chunk found using Factor AveR3 were needed by some of the participants, for WCT Task 2.

The interpretations for I and A Groups' chunks found using Factor R3 are also applicable here, since those chunks are identical to the ones found here.

Factor AveR3F: When the chunks discovered using Factor AveR3 are filtered to include only sections with more than half of the participants rated them and their average ratings are 3 and above, chunks with very small sizes were found. C Group's chunk contains only Section "4.1 Overview". I Group's chunk includes Sections "4.1 Overview" and "Logical High Level Solution Overview Diagram". A Group's chunk comprises only Section "2.2 Architecturally Significant Open Source Products".

Interpretation: Factor AveR3F's chunk shows that based on the average ratings contributed by the majority of the participants of a group, textual description of main logical components was needed by C Group, both the textual description of main logical components and the corresponding diagram were needed by I Group, and open source products used in WCT was needed by A Group, when they attempted WCT Task 2.

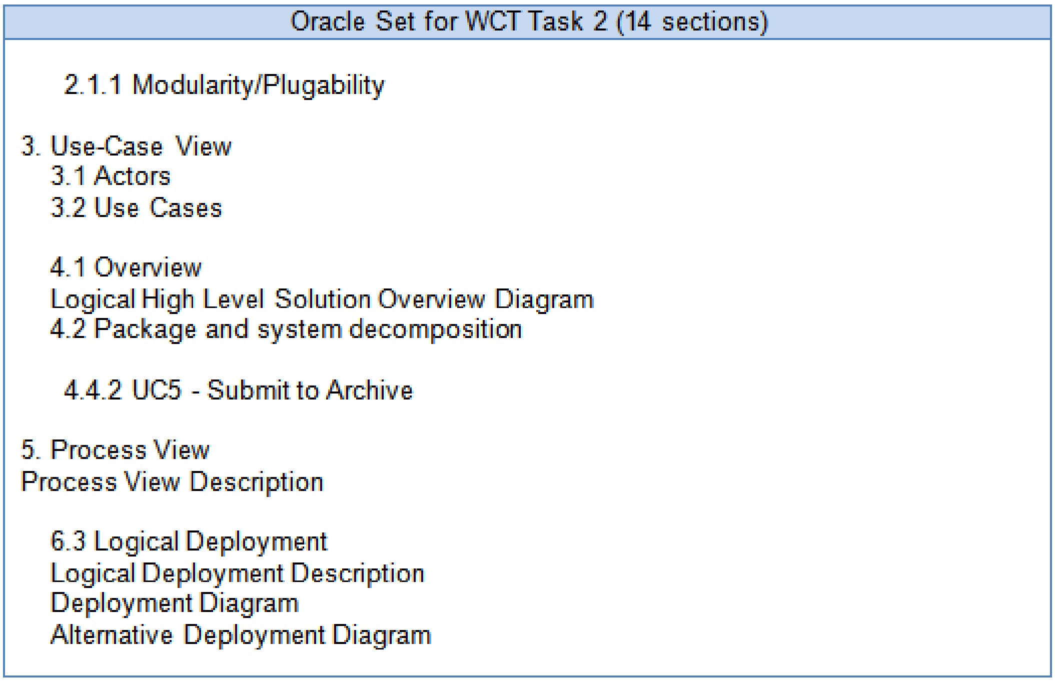
7.2 Benchmarking Chunks against An Oracle Set

This section describes the oracle set for WCT Task 2 and the results of benchmarking the discovered chunks against this oracle set.

7.2.1 The Oracle Set for WCT Task 2

The oracle set for WCT Task 2 comprises 14 of the 47 sections of the WCT software architecture document (Table 7.2). These sections were agreed by both the researcher and a separate judge, as mandatory for WCT Task 2. In Table 7.1, these sections are denoted by yellow rows and their section IDs are marked with asterisks. This is to make it easier to compare the chunks found against the oracle set. The relevance of these sections to WCT Task 2 is explained next. It is important to bear in mind that WCT Task 2 is a very specific task composing of two parts: 1) what needs to be done to change WCT to use a different digital archive system, and 2) identifying the parts of WCT that will be affected, both from the perspective of a developer.

Section "2.1.1 Modularity/Plugability" contains the specific information needed for WCT Task 2. This section states that to use a different digital archive system, a new adapter that implements the defined interface needs to be written and a change to the Spring configuration files need to be made. It also states that no changes to the rest of WCT would be required. **Section "4.4.2 UC5 – Submit to Archive"** describes that the archive submission feature needs to cater for different archival systems. It also explains that this is achieved through the use of adapter pattern, where an interface is defined by WCT and an adapter class that implements the interface will be written for each digital archive system. So, this section also contains some specific information on what to do to use a different digital archive system. However, it does not mention the change to the Spring configuration files.



Oracle Set for WCT Task 2 (14 sections)
2.1.1 Modularity/Plugability
3. Use-Case View
3.1 Actors
3.2 Use Cases
4.1 Overview
Logical High Level Solution Overview Diagram
4.2 Package and system decomposition
4.4.2 UC5 - Submit to Archive
5. Process View
Process View Description
6.3 Logical Deployment
Logical Deployment Description
Deployment Diagram
Alternative Deployment Diagram

Table 7.2: The Oracle Set for WCT Task 2

Without knowing what needs to be done to make the change, it is difficult to assess which parts of the system will be affected. In other words, without knowing the answer to Part 1 of the task, it is difficult to answer its Part 2. Consequently both Section “2.1.1 Modularity/Plugability” and “4.4.2 UC5 – Submit to Archive” are very important sections for WCT Task 2, especially for Part 1 of the task. Section “2.1.1 Modularity/Plugability” is considered the most important section for WCT Task 2 as it contains the most complete information for Part 1 of the task.

In order to change the digital archive system and to assess the change impact it is important to understand how digital archive system fits into WCT. There are 5 groups of sections that contain information on how digital archive system fits into WCT. Each group has different aspect or perspective of the information and are therefore needed for WCT Task 2.

The first group of sections that contain information on how digital archive system fits into WCT is related to use-case view. **Section “3. Use-Case View”** contains the use-case diagram for WCT. It shows “UC5: Submit to Archive” as one of the use cases and ‘Digital Archive System’ as an actor of the system. The description of ‘Digital Archive System’ actor in **Section “3.1 Actors”** contains information about the use of an interface to interact with this actor. Without the knowledge of adapter pattern, this information is insufficient for WCT Task 2. **Section “3.2 Use Cases”** contains some information with regards to “UC5 - Submit to Archive”. It explains that the submission to the digital archive system will convert the harvest result and metadata into a common Submission Information Package format that suits most digital archive systems.

The second group of sections that contain information on how digital archive system fits into WCT is related to logical view. **Section “4.1 Overview”** states that the digital archive system is one of the external systems and the access is provided through 'External Interface Tools' layer. The diagram in **Section “Logical High Level Solution Overview Diagram”** shows similar information with the addition that the access is provided over

SOAP HTTP. **Section "4.2 Package and system decomposition"** contains information about the packages of Java code that communicate with external archive system. For the developer role assumed for WCT Task 2, this section contains code that needs to be considered to be changed to support different digital archive systems.

The third group of sections that contain information on how digital archive system fits into WCT is related to process view. **Section "5. Process View"** contains a diagram that shows the digital archive system interface and the process related to it. The textual description of the process diagram is found in **Section "Process View Description"**. It contains information about one of the processes which is an implementation of the digital archive interface for exporting to external digital archive system. The textual description and diagram complement each other.

The fourth group of sections that contain information on how digital archive system fits into WCT is related to logical deployment. The diagram found in **Section "6.3 Logical Deployment"** shows the external interface and archive adapter for the digital archive system from deployment perspective. Those participants with knowledge on adapter design pattern could probably infer that a new adapter that implements the interface need to be constructed. The diagram also shows that the archive adapter is located in the WCT Digital Asset Store node. Changing WCT to use a different digital archive system will affect WCT Digital Asset Store since a new adapter needs to be implemented in this component. Section "6.3 Logical Deployment" is important for Part 2 of the task. The **3 sub-sections of Section "6.3 Logical Deployment"** contain information about the deployment of Digital Asset Store component which holds the archive adapter. This could reveal other parts of WCT that might be affected by the change to Digital Asset Store.

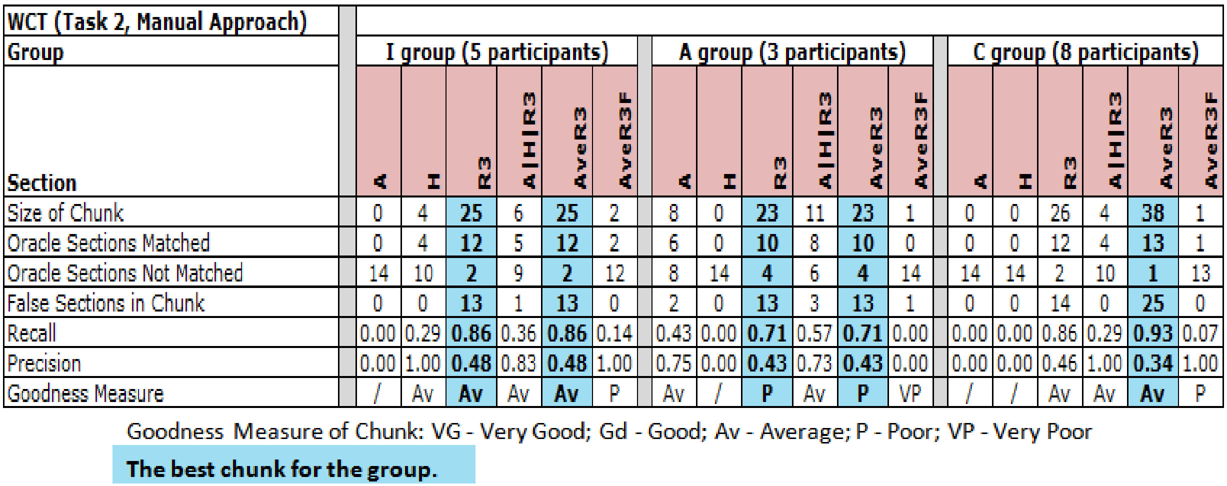
Based solely on what is written in the document, we think that no other parts of WCT will be affected other than the Digital Asset Store, when changing to use a different digital archive system. To arrive at this conclusion, all the 4 groups of sections that contain information on how digital archive system fits into WCT are needed.

7.2.2 Benchmarking Results

Refer to Section "5.7 Benchmarking of Chunks against An Oracle Set" of Chapter 5 for all the concepts and terms needed to understand this section. The attributes (size, numbers of oracle set's section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for WCT Task 2 are given in Table 7.3.

WCT Task 2 composes two parts. We have explained in Section "7.2.1 The Oracle Set for WCT Task 2" of this chapter that without knowing the answer to Part 1 of the task, it is difficult to answer its Part 2. As a result, chunks that do not contain the information needed for Part 1 of the task are regarded as *insufficient* for the task and are not discussed here. They are those found using Factor AveR3F for all three groups of participants.

For C Group, the goodness of chunks found using Factors R3, A|H|R3 and AveR3 are 'Average'. The chunk found using Factor AveR3F is 'Poor' and no chunk is found when Factors A and H are used. We assessed the composition of the three 'Average' chunks to decide on the best chunk for C Group. *C Group's chunk found* *using Factor AveR3 (recall = 0.93, precision = 0.34, size = 38, goodness = 'Average')* is large. With 38 out of 47 sections, it constitutes about 90% of the sections of the AD. The chunk's recall is very high, covering 13 of the 14 sections in the oracle set. It contains the two important sections (Sections "2.1.1 Modularity/Plugability" and "4.4.2 UC5 – Submit to Archive") for Part 1 of WCT Task 2. It also has a good amount of information to assess the change impact for Part 2 of the task. It includes Section "6.3 Logical Deployment" which is important for Part 2 of the task. It missed only one section from the oracle set, namely the section on process view description. Consequently, Factor AveR3's chunk is the best chunk for C Group though it has 25 false sections.



WCT (Task 2, Manual Approach)																		
Group	I group (5 participants)						A group (3 participants)						C group (8 participants)					
Section	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
Size of Chunk	0	4	**25**	6	**25**	2	8	0	**23**	11	**23**	1	0	0	26	4	**38**	1
Oracle Sections Matched	0	4	**12**	5	**12**	2	6	0	**10**	8	**10**	0	0	0	12	4	**13**	1
Oracle Sections Not Matched	14	10	**2**	9	**2**	12	8	14	**4**	6	**4**	14	14	14	2	10	**1**	13
False Sections in Chunk	0	0	**13**	1	**13**	0	2	0	**13**	3	**13**	1	0	0	14	0	**25**	0
Recall	0.00	0.29	**0.86**	0.36	**0.86**	0.14	0.43	0.00	**0.71**	0.57	**0.71**	0.00	0.00	0.00	0.86	0.29	**0.93**	0.07
Precision	0.00	1.00	**0.48**	0.83	**0.48**	1.00	0.75	0.00	**0.43**	0.73	**0.43**	0.00	0.00	0.00	0.46	1.00	**0.34**	1.00
Goodness Measure	/	Av	**Av**	Av	**Av**	P	Av	/	**P**	Av	**P**	VP	/	/	Av	Av	**Av**	P

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 7.3: Attributes of Chunks (WCT Task 2)

C Group's chunk found using Factor R3 (recall = 0.86, precision = 0.46, size = 26, goodness = 'Average') also has high recall (containing 12 of the 14 sections in the oracle set). It has fewer (14) false sections but it does not include the most important section which is Section "2.1.1 Modularity/Plugability". C Group's chunk found using Factor A|H|R3 (recall = 0.29, precision = 1.00, size = 4, goodness = 'Average') does not contain the complete information required especially for Part 2 of the task. Factor A|H|R3's chunk is the most basic chunk for C Group (instead of Factor AveR3F's chunk which is the smallest chunk but insufficient for the task as mentioned earlier).

For I Group, the goodness of chunks found using Factors H, R3, A|H|R3 and AveR3 are 'Average'. The chunk found using Factor AveR3F is 'Poor' and no chunk is found when Factor A is used. Chunks produced by Factors R3 and AveR3 are identical (recall = 0.86, precision = 0.48, size = 25, goodness = 'Average'). It contains 12 of the 14 sections in the oracle set. It misses the most important section ("2.1.1 Modularity/Plugability") and section on process view description. I Group's chunk found using Factors R3 and AveR3 contains Section "4.4.2 UC5 – Submit to Archive" which has the information on what to do to use a different digital archive system, which is for Part 1 of the task. It has quite sufficient information to access the change impact for Part 2 of the task. It includes Section "6.3 Logical Deployment" which is important for Part 2 of the task. Nevertheless it has 13 false sections, which are mainly related to common functionalities, detailed description of some use-cases and scheduling the distributed Harvesters. Sections related to details such as these made it into the chunk probably because the task was attempted from the perspective of a developer. In terms of composition and recall measure, Factor R3's or AveR3's chunk is better when compared to the other two 'Average' chunks (explained in next paragraph) and is therefore the best chunk for I Group.

I Group's chunks found using Factors H and A|H|R3 contain Section "4.4.2 UC5 – Submit to Archive" which has the information for Part 1 of the task. Factor H's chunk has 4 sections (recall = 0.29, precision = 1.00, size = 4, goodness = 'Average') from the oracle set and is the most basic chunk for I Group. It contains sections on textual description of main logical components, detailed description of 'Submit to Archive' use-case, descriptions of actors and use-cases. Factor H's chunk is similar to Factor A|H|R3's chunk (recall = 0.36, precision = 0.83, size = 6, goodness = 'Average'). The latter has one additional oracle set's section (on the main logical components diagram) and one false section (on store file server). However chunks found using Factors H and A|H|R3 do not contain sufficient information to assess the change impact, for Part 2 of the task.

It is important to note that with the absence of Section "2.1.1 Modularity/Plugability", all I Group's chunks miss some pieces of information needed for WCT Task 2, namely, changing the Spring configuration files.

For A Group, the chunks found using Factors A and A|H|R3 are 'Average', those found using Factor sR3 and AveR3 are 'Poor', chunk found using Factor AveR3F is 'Very Poor' and no chunk is found when Factor H is used. Chunks produced by Factors R3 and AveR3 are identical (recall = 0.71, precision = 0.43, size = 23, goodness = 'Poor'). Chunk found using Factor A (recall = 0.43, precision = 0.75, size = 8, goodness = 'Average') is the most basic chunk for A Group. It contains section on use-case diagram, description of actors and all the sections on deployment view. It matches 6 of the oracle set's sections and contains 2 false sections. Chunk found using Factor A|H|R3 (recall = 0.57, precision = 0.73, size = 11, goodness = 'Average') contains 2 additional oracle set's sections and 1 additional false section. Though chunks found using Factors A and A|H|R3 for A Group have the highest goodness measures, they do not contain either one of the two important sections for Part 1 of WCT Task 2. However, with the presence of Section "3.1 Actors" and those sections related to logical deployment in these chunks, those with knowledge on adapter pattern might be able to infer part of the answer for Part 1 of the task. Nevertheless, they would still have missed out the information on changing the Spring configuration file.

The identical *chunk (recall = 0.71, precision = 0.43, size = 23, goodness = 'Poor') found using Factors R3 and AveR3 is the best chunk for A Group.* It covers 10 of the 14 sections in the oracle set. It is the only A group's chunk that contains Section ("2.1.1 Modularity/Plugability") which is the most important section for WCT Task 2. It also has a good amount of information to assess the change impact. It includes Section "6.3 Logical Deployment" which is important for Part 2 of the task. It misses description of use-cases, packages of Java code, detailed description of 'Submit to Archive' use case and process view description. This A Group's chunk contains 13 false sections. They are mainly those related to key quality requirements, open source products used, operating system, database servers and, size and performance.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C Group is Factor AveR3, and for I and A Groups is either Factor R3 or Factor AveR3.

Commonly-missed Oracle Set's Sections and Common False Sections: Refer to Section "5.7.6 Determination of the Commonly-Missed Oracle Set's Sections and Common False Sections" of Chapter 5 on how common oracle set's sections missed by a group of participants, and common false sections for a group of participants are determined.

For each group of participants, we try to give possible reason for commonly-missed oracle set's sections and common false sections. If no reason is given, it means that we cannot think of any plausible reason other than a commonly-missed oracle set's section was generally 'not needed', and a commonly-included false section was generally 'needed', by the respective group of participants when they attempted the task.

Section "Process View Description" is not included by any chunk found for WCT Task 2. Inspection of raw data shows that it was not rated by any participant that attempted WCT Task 2. This section therefore did not qualify as commonly-missed section for any of the group of participants.

The oracle set's sections commonly missed by C Group are: Section "2.1.1 Modularity/Plugability" (missed by 5 chunk-identification factors); Sections "3. Use-Case View", "3.2 Use Cases", "4.2 Package and system decomposition", "5. Process View", and the four sections on logical deployment (missed by 4 factors).

The oracle set's sections commonly missed by I Group are: Section "2.1.1 Modularity/Plugability" (missed by all 6 factors); Sections "3. Use-Case View", "4.2 Package and system decomposition", "5. Process View", and the four sections on logical deployment (missed by 4 factors).

The oracle set's sections commonly missed by A Group are: Sections "2.1.1 Modularity/Plugability" and "5. Process View" (missed by 4 factors). Sections "3.2 Use Cases", "4.2 Package and system decomposition", "4.4.2 UC5 – Submit to Archive" are missed by all the factors. None of A Group's participants rated these sections and therefore they do not qualify as commonly-missed sections.

The reason that Section '2.1.1 Modularity/Plugability' (the most important section) is commonly missed by all the groups, could be its title does not suggest finding information related to digital archive system and subsequently the information needed for WCT Task 2. This could cause the section to be skipped by the participants on the onset of the process of finding the needed information. As for the other commonly-missed sections, we cannot think of any plausible reason other than they were generally 'not needed' by the respective group of participants, when they attempted the task.

There is no common false section for C and I Groups. The common false sections for A Group are: Sections "2.2 Architecturally Significant Open Source Products", "6.1 Operating Systems" and "6.2 Database Servers". The first common false section was probably needed by A Group to assess if the open source products used in WCT have any effect on changing the digital archive system. As for the other two common false sections, we cannot think of any plausible reason other than they were generally 'needed' by A Group when attempting WCT Task 2.

7.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for WCT Task 2 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. The larger chunks are usually supersets of the smaller chunks.

The most basic chunks for C, I and A Groups are found using Factor A|H|R3, Factor H and Factor A, respectively. Their compositions can be seen in Table 7.1. For WCT Task 2, the most basic chunk for a group is the chunk with the smallest number of section and the chunk contains at least information needed for Part 1 of the task. For I and A Groups, chunks found using Factor R3 and AveR3 are identical.

Part 1 of Table 7.4 summarises the main results of benchmarking of chunks. Recall that the number of participants in I and A Groups are 5 and 3, respectively. All the participants in each group provided some responses in terms of where the needed information (answers) can be found. Two academic participants did not highlight any information in the document. One academic participant did not rate sections visited in terms of their importance to the task.

In terms of how ‘good’ the chunks are for WCT Task 2, C Group has 3 chunks with goodness measures of ‘Average’ and 1 of ‘Poor’. No chunk is found for C Group when Factors A and H are used. I Group has 4 chunks with goodness measures of ‘Average’ and 1 of ‘Poor’. No chunk is found for I Group when Factor A is used. A Group has 2 chunks each with goodness measures of ‘Average’ and ‘Poor’, 1 of ‘Very Poor’, and no chunk is found when Factor H is used.

The best chunk for C Group is found using Factor AveR3 (with ‘Average’ goodness measure), and the best chunks for I Group (with ‘Average’ goodness measure) and A Group (with ‘Poor’ goodness measure) are discovered using Factor R3 or Factor AveR3. The factor that produces the best chunk for a group is the best chunk-identification factor for the group.

Part 2 of Table 7.4 summarises the information or sections needed by each group of participants. This is based on the groups’ chunks found using Factor A|H|R3. Refer to Section “5.8 Information Needed” in Chapter 5 for our reason. The goodness measures of chunks found using Factor A|H|R3 for all three groups are ‘Average’.

As shown in Table 7.4, both I and A Groups of participants needed the following information for WCT Task 2: description of actors, main logical components (textual description and diagram). I Group also needed description of use-cases, detailed description of ‘Submit to Archive’ use-case and store file server. A Group also needed open source products used in WCT, use-case diagram (Section “3. Use-Case View”), and deployment view (all sections).

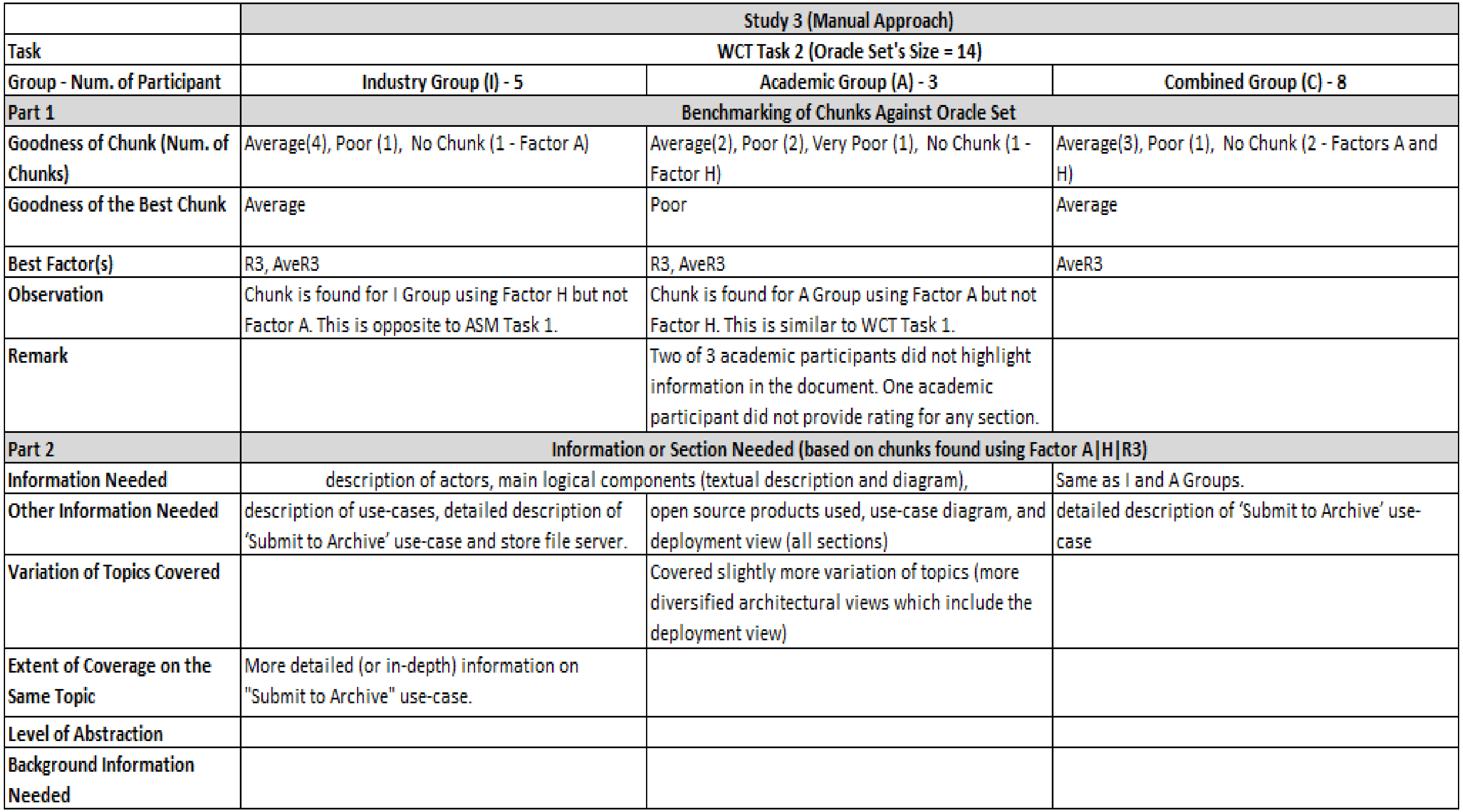
In summary, both industry practitioners and academics needed logical view (textual description of main logical components and the corresponding diagram) and use-case view (description of actors) while attempting WCT Task 2. In terms of use-case view, I Group also needed descriptions of use-cases, whereas A Group also needed the use-case diagram. A Group covered slightly more variation of topics (in particular more diversified architectural views which include the deployment view). I Group needed more detailed (or in-depth) information on ‘Submit to Archive’ use-case.

C Group’s chunk shows that when all the participants are combined, in addition to description of actors and main logical components (textual description and diagram), information on detailed description of ‘Submit to Archive’ use-case stands out as needed by the majority of the participants for WCT Task 2.

7.4 Summary

This chapter describes the chunking of architectural information for WCT Task 2. As with ASM Task 2 (i.e. a similar task performed on the second AD), the results show the discovery of usage-based chunks for the task of changing one part of a system and assessing the possible impact of change.

Chapter 13 contains the comparison of the results for WCT Task 2 and ASM Task 2, and the other tasks; the summary of the overall results for Studies 2 and 3; and threats to validity of findings. Chapter 10 presents the chunking of architectural information for ASM Task 2. The next chapter details the chunking of architectural information for WCT Task 3.



	Study 3 (Manual Approach)		
Task	WCT Task 2 (Oracle Set's Size = 14)		
Group - Num. of Participant	Industry Group (I) - 5	Academic Group (A) - 3	Combined Group (C) - 8
Part 1	Benchmarking of Chunks Against Oracle Set		
Goodness of Chunk (Num. of Chunks)	Average(4), Poor (1), No Chunk (1 - Factor A)	Average(2), Poor (2), Very Poor (1), No Chunk (1 - Factor H)	Average(3), Poor (1), No Chunk (2 - Factors A and H)
Goodness of the Best Chunk	Average	Poor	Average
Best Factor(s)	R3, AveR3	R3, AveR3	AveR3
Observation	Chunk is found for I Group using Factor H but not Factor A. This is opposite to ASM Task 1.	Chunk is found for A Group using Factor A but not Factor H. This is similar to WCT Task 1.	
Remark		Two of 3 academic participants did not highlight information in the document. One academic participant did not provide rating for any section.	
Part 2	Information or Section Needed (based on chunks found using Factor A|H|R3)		
Information Needed	description of actors, main logical components (textual description and diagram),		Same as I and A Groups.
Other Information Needed	description of use-cases, detailed description of 'Submit to Archive' use-case and store file server.	open source products used, use-case diagram, and deployment view (all sections)	detailed description of 'Submit to Archive' use-case
Variation of Topics Covered		Covered slightly more variation of topics (more diversified architectural views which include the deployment view)	
Extent of Coverage on the Same Topic	More detailed (or in-depth) information on "Submit to Archive" use-case.		
Level of Abstraction			
Background Information Needed			

Table 7.4: Summary of Results for WCT Task 2

---- Blank ----

8 Chunking of Architectural Information (WCT Task 3)

This chapter presents the chunking of architectural information for the third information-seeking task for the Web Curator Tool (WCT Task 3) software architecture document, for Study 2. The task was:

> "As a maintainer of Web Curator Tool, you would like to know how it was designed at the architectural level to achieve security."

The results presented in this chapter are based on the responses of 8 participants who completed WCT Task 3. Three were classified as industry participants and 5 were classified as academic participants. We recruited 4 industry practitioners and 4 academics for this task just as what we did for the tasks. Further inspection of the participants' data showed that one of the participants which was classified as industry participant during initial screening, actually has more years of SA teaching experience than industry experience. This participant was re-classified as academic participant.

This chapter adopts the same structure as Chapter 6. Chapter 5 provides the foundation to understand this chapter and should be read prior to reading this chapter.

8.1 Identification and Interpretation of Chunks

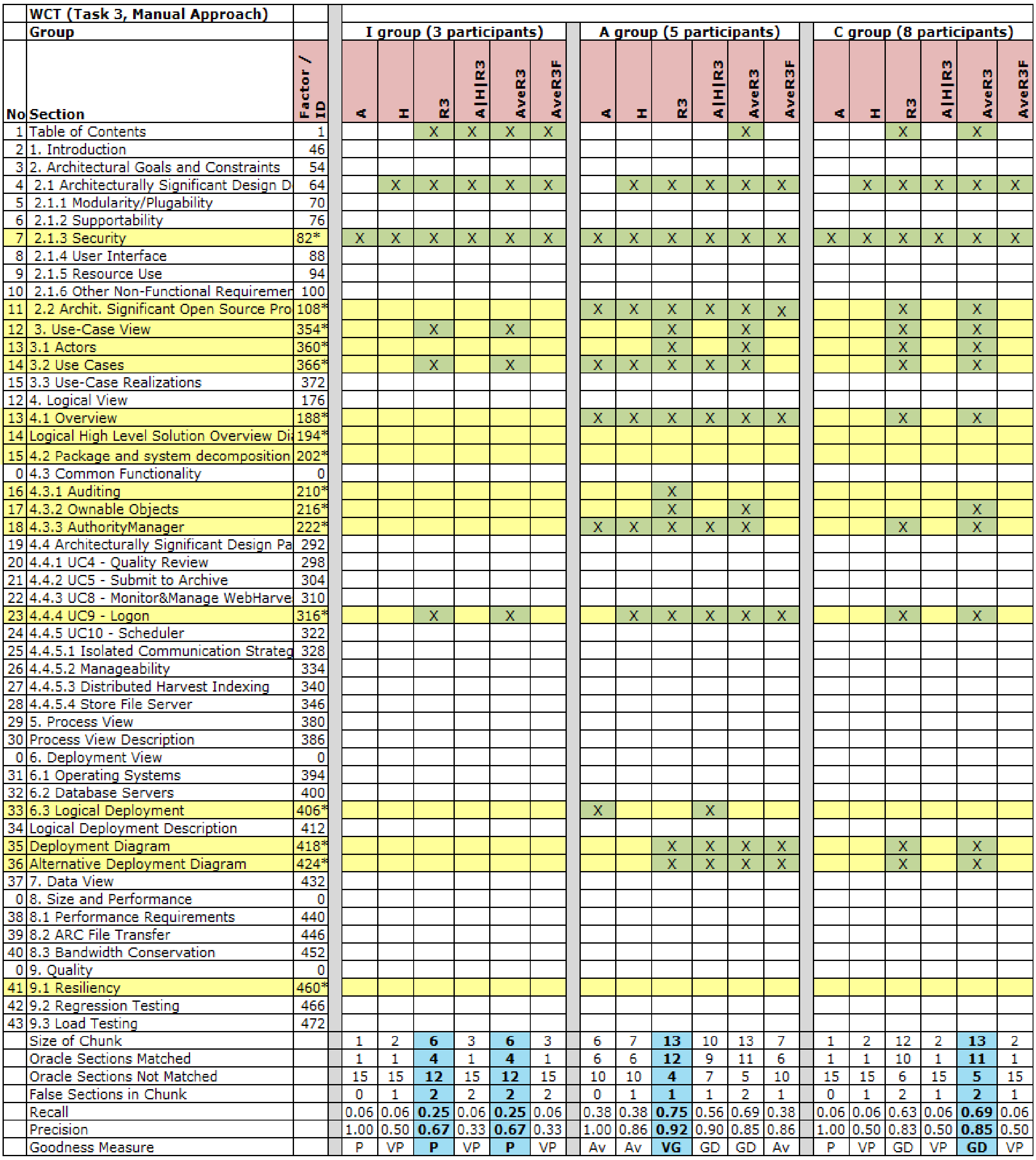
We used the 6 chunk-identification factors to identify the chunks for this task. This was done for the 3 groups of participants: the industry practitioner (I) group, the academic (A) group, and the combined (C) group of the previous two.

Table 8.1 shows the chunks discovered for WCT Task 3, for each group of participants. Refer to Section "6.1 Identification and Interpretation of Chunks" in Chapter 6 on how to interpret the chunks in the table. Recall that the size of a chunk is the number of the document's sections that the chunk comprises.

Factor A: A chunk comprising Section "2.1.3 Security" is discovered for C and I Groups. A Group's chunk comprises 6 sections: "2.1.3 Security", "2.2 Architecturally Significant Open Source Products", "3.2 Use Cases", "4.1 Overview", "4.3.3 Authority Manager" and "6.3 Logical Deployment".

Interpretation: Section "2.1.3 Security" contains information on the security framework used for WCT and some of its background information. C Group's chunk shows that this information was needed by the majority of the overall participants, for WCT Task 3. I Group's chunk shows that the same information was needed by the majority of the industry participants when they attempted the task.

The sections in A Group's chunk contain the following information: Section "2.1.3 Security" and "2.2 Architecturally Significant Open Source Products" contain complementary background information on the technologies used to achieve security for WCT. The use-case for logging into the system (3.2.9 UC9 Logon) is described in "3.2 Use Cases". It states the use of credential management system and enterprise directory services system for pass-through authentication. Section "4.1 Overview" contains the information about the use of interface to access the external authentication system, system accesses via web browser over HTTPS and the authorization of users' view of the system based on their privileges. Section “4.3.3 Authority Manager” describes the function of AuthorityManager which is to determine the actions that can be taken on Ownable objects based on the users’ privilege and privilege scope. It is related to the access control mechanisms used by WCT. The diagram on the logical mapping of components on the nodes for WCT (Section “6.3 Logical Deployment”) shows the ‘Authentication’ component interacting with the ‘Directory Server’ and ‘Database Server’ nodes.



	WCT (Task 3, Manual Approach)																				
	Group		I group (3 participants)						A group (5 participants)						C group (8 participants)					
No	Section	Factor / ID	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
1	Table of Contents	1			X	X	X	X					X				X		X	
2	1. Introduction	46																		
3	2. Architectural Goals and Constraints	54																		
4	2.1 Architecturally Significant Design D	64		X	X	X	X	X		X	X	X	X	X		X	X	X	X	X
5	2.1.1 Modularity/Plugability	70																		
6	2.1.2 Supportability	76																		
7	2.1.3 Security	82*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	2.1.4 User Interface	88																		
9	2.1.5 Resource Use	94																		
10	2.1.6 Other Non-Functional Requiremen	100																		
11	2.2 Archit. Significant Open Source Pro	108*							X	X	X	X	X	X			X		X	
12	3. Use-Case View	354*			X		X				X		X				X		X	
13	3.1 Actors	360*									X		X				X		X	
14	3.2 Use Cases	366*			X		X		X	X	X	X	X				X		X	
15	3.3 Use-Case Realizations	372																		
12	4. Logical View	176																		
13	4.1 Overview	188*							X	X	X	X	X	X			X		X	
14	Logical High Level Solution Overview Di	194*																		
15	4.2 Package and system decomposition	202*																		
0	4.3 Common Functionality	0																		
16	4.3.1 Auditing	210*									X									
17	4.3.2 Ownable Objects	216*									X		X						X	
18	4.3.3 AuthorityManager	222*							X	X	X	X	X				X		X	
19	4.4 Architecturally Significant Design Pa	292																		
20	4.4.1 UC4 - Quality Review	298																		
21	4.4.2 UC5 - Submit to Archive	304																		
22	4.4.3 UC8 - Monitor&Manage WebHarve	310																		
23	4.4.4 UC9 - Logon	316*			X		X			X	X	X	X	X			X		X	
24	4.4.5 UC10 - Scheduler	322																		
25	4.4.5.1 Isolated Communication Strateg	328																		
26	4.4.5.2 Manageability	334																		
27	4.4.5.3 Distributed Harvest Indexing	340																		
28	4.4.5.4 Store File Server	346																		
29	5. Process View	380																		
30	Process View Description	386																		
0	6. Deployment View	0																		
31	6.1 Operating Systems	394																		
32	6.2 Database Servers	400																		
33	6.3 Logical Deployment	406*							X			X								
34	Logical Deployment Description	412																		
35	Deployment Diagram	418*									X	X	X	X			X		X	
36	Alternative Deployment Diagram	424*									X	X	X	X			X		X	
37	7. Data View	432																		
0	8. Size and Performance	0																		
38	8.1 Performance Requirements	440																		
39	8.2 ARC File Transfer	446																		
40	8.3 Bandwidth Conservation	452																		
0	9. Quality	0																		
41	9.1 Resiliency	460*																		
42	9.2 Regression Testing	466																		
43	9.3 Load Testing	472																		
	Size of Chunk		1	2	**6**	3	**6**	3	6	7	**13**	10	13	7	1	2	12	2	**13**	2
	Oracle Sections Matched		1	1	**4**	1	**4**	1	6	6	**12**	9	11	6	1	1	10	1	**11**	1
	Oracle Sections Not Matched		15	15	**12**	15	**12**	15	10	10	**4**	7	5	10	15	15	6	15	**5**	15
	False Sections in Chunk		0	1	**2**	2	**2**	2	0	1	**1**	1	2	1	0	1	2	1	**2**	1
	Recall		0.06	0.06	**0.25**	0.06	**0.25**	0.06	0.38	0.38	**0.75**	0.56	0.69	0.38	0.06	0.06	0.63	0.06	**0.69**	0.06
	Precision		1.00	0.50	**0.67**	0.33	**0.67**	0.33	1.00	0.86	**0.92**	0.90	0.85	0.86	1.00	0.50	0.83	0.50	**0.85**	0.50
	Goodness Measure		P	VP	**P**	VP	**P**	VP	Av	Av	**VG**	GD	GD	Av	P	VP	GD	VP	**GD**	VP

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

Table 8.1: The Chunks for WCT Task 3 Identified Using Different Factors

In short, A Group's chunk found using Factor A shows that the majority of the academic participants needed the following information when they attempted WCT Task 3: background information on the security technologies, description of use-cases, textual description of main logical components, access control mechanism (AuthorityManager) and the mapping of the security component to the system's nodes and its interactions with other nodes.

Factor H: C and I Groups' chunks found using Factor H contain Sections "2.1 Architecturally Significant Design Decisions" and "2.1.3 Security". A Group's chunk found using Factor H is quite similar to the group's chunk found using Factor A. The former chunk excludes Section "6.3 Logical Deployment" but includes two additional sections namely Sections "2.1 Architecturally Significant Design Decisions" and "4.4.4 UC9 – Logon".

Interpretation: Inspection on the raw data shows that 1 of the 5 academic participants did not highlight information in the document when attempting WCT Task 3. This could have impacted the chunk found using Factor H for A Group.

C and I Groups' chunks found using Factor H show that the majority of the combined group and the majority of the industry participants needed information on 1) security as one of the categories of key quality requirements that affected the design of WCT (Sections "2.1 Architecturally Significant Design Decisions") and 2) security framework used for WCT and some of its background information (Section "2.1.3 Security"), when they attempted WCT Task 3.

Comparing to A Group's chunk found using Factor A, one of the additional section of A Group's chunk found using Factor H is Section "4.4.4 UC9 – Logon". A Group's chunk found using Factor A shows that the majority of the academic participants also needed the detailed description of the 'logon' use-case, for WCT Task 3. They also needed the section that introduces the categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decisions") of WCT.

Similar to ASM Task 2, chunks are found using Factors A and H for both I and A Groups. This is different from what is observed for WCT Task 2, where chunk is found for A Group using Factor A but not Factor H, and chunk is found for I Group using Factor H but not Factor A. The observation that academic participants were more keen to provide indications in terms of from which sections their answers were found (Factor A) than highlighting information (Factor H), and vice versa for the industry participants, does not apply for WCT Task 3 and ASM Task 2. In fact the overlapping of the sections in the chunks found using these two factors, reiterates the importance of those sections to the respective groups.

Factor R3: The chunks found using Factor R3 are relatively larger for all the groups (comprising 12, 6, 13 sections for C, I, A Groups respectively) when compared to the chunks found using the rest of the factors (with the exception of Factor AveR3) C Group's chunk contains Sections "Table of Contents", "2.1 Architecturally Significant Design Decisions", "2.1.3 Security", "2.2 Architecturally Significant Open Source Product", "3. Use-Case View", "3.1 Actors", "3.2 Use Cases", "4.1 Overview", "4.3.3 AuthorityManager", "4.4.4 UC9 – Logon", "Deployment Diagram" and "Alternative Deployment Diagram".

Chunk for I Group comprises Sections "Table of Contents", "2.1 Architecturally Significant Design Decisions", "2.1.3 Security", "3. Use-Case View", "3.2 Use Cases" and "4.4.4 UC9 – Logon". Chunk for A Group comprises Sections “2.1 Architecturally Significant Design Decisions”, “2.1.3 Security”, “2.2 Architecturally Significant Open Source Products”, “3. Use-Case View”, “3.1 Actors”, “3.2 Use Cases”, “4.1 Overview”, “4.3.1 Auditing”, “4.3.2 Ownable Objects”, “4.3.3 AuthorityManager”, “4.4.4 UC9 – Logon” and two sections on deployment diagram. This chunk is similar to C Group’s chunk with the exclusion of Section “Table of Contents” and the inclusion of 2 extra sections: Sections “4.3.1 Auditing” and “4.3.2 Ownable Objects”.

Interpretation: One of the 3 industry participants did not provide rating for any section, when attempting WCT Task 3. This could have affected I Group’s chunk found using Factor AveR3F but not the chunks found using Factors R3 and AveR3. This is because Factor AveR3F is based on the majority of a group, and Factors R3 and AveR3 are based on the majority of those who rated the respective sections.

C Group’s chunk found using Factor R3 shows that for WCT Task 3, sections that contain the following information were needed by the majority of the participants who rated the sections: organisation of the document (Section “Table of Contents”), key quality requirements that affected the design of WCT (where security is one of them), background information on the security technologies used, use-case diagram, description of actors, description of use-cases, textual description of main logical components (Section “4.1 Overview”), access control mechanism (AuthorityManager), detailed description of the ‘logon’ use-case and the deployment diagrams.

The overlapping sections in I and A Groups’ chunks show that the key quality requirements that affected the design of WCT (where security is one of them), background information on the security technologies used, use-case diagram, description of use-cases, and detailed description of the ‘logon’ use-case, were needed by the majority of the 2 groups who rated these sections.

In addition, for WCT Task 3, Table of Contents was needed by the majority of the industry participants who rated the respective section. On the other hand, the description of actors, textual description of main logical components, auditing, access control mechanisms (Section “4.3.2 Ownable Objects” and “4.3.3 AuthorityManager”), logical deployment diagram and deployment diagrams were also needed by the majority of the industry participants who rated the respective sections.

Factor A|H|R3: C Group’s chunk is identical to the group’s chunk found using Factor H. I Group’s chunk is almost identical to the group’s chunk found using Factor H with the addition of Section “Table of Contents”. A Group’s chunk resembles the group’s chunk found using Factor H, with 3 extra sections on logical deployment.

Interpretation: Since chunks found using Factor A|H|R3 for each group are either identical or resemble the respective groups’ chunks found using Factor H, the interpretations for Factor H’s chunks are applicable here. The additional observations here are: information related to logical deployment was also required by the majority of the academic participants, and Table of Contents was also required by the majority of the industry participants, for WCT Task 3.

Factor AveR3: Chunk found using Factor AveR3 for a group of participants is very similar to the group's chunk found using Factor R3. For I Group, chunks found using Factors AveR3 and R3 are identical. C Group's chunk found using Factor AveR3 is similar to the group's chunk found using Factor R3 with the addition of Section "4.3.2 Ownable Objects". A Group's chunk found using Factor AveR3 is similar to the group's chunk found using Factor R3 with the addition of Section "Table of Contents" and exclusion of Section "4.3.1 Auditing". The chunks found using Factor AveR3 for C and A Groups are identical.

Interpretation: The interpretations for C and A Groups' chunks are similar to the interpretation given for C Group's chunk found using Factor R3. An extra observation is that information on ownable objects (related to access control mechanisms for WCT) was also needed by some of the participants from the combined group and A Group who rated the section, when attempting WCT Task 3. The interpretation for I Group's chunk is the same as what is given for I Group's chunk found using Factor R3.

Factor AveR3F: The sizes of the chunks for I and A Groups reduce to half when compared to the respective chunks found using Factor AveR3. C Group's chunk found using Factor AveR3F is identical to the group's chunks found using Factors H and A|H|R3. I Group's chunk is identical to the group's chunk found using Factor A|H|R3. A Group's chunk found using Factor AveR3F is similar to the group's chunks found using Factors H and A|H|R3. It comprises Sections "2.1 Architecturally Significant Design Decisions", "2.1.3 Security", "2.2 Architecturally Significant Open Source Products", "4.1 Overview", "4.4.4 UC9 – Logon" and the 2 sections on deployment diagram.

Interpretation: Recall that for Factor AveR3F, chunk found using Factor AveR3 is filtered to arrive at new chunk containing only those sections rated by more than half of the participants in the group and with the average ratings of at least 3. The purpose is to remove those sections which have a small number of participants who rated them.

C Group's chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, information on security as one of the categories of key quality requirements and background information on the security technologies used, were needed for WCT Task 3. I Group's chunk found using Factor AveR3F shows that in addition to that, organisation of the document was also needed by I Group. A Group's chunk found using Factor AveR3F shows that information on security as one of the categories of key quality requirements, background information on the security technologies used, textual description of main logical components, detailed description on the 'logon' use-case and the deployment diagrams, were needed by A Group, for WCT Task 3.

8.2 Benchmarking Chunks against An Oracle Set

This section starts with a brief overview of the concept involved in WCT Task 3, which is security. This is followed by the details of the oracle set for WCT Task 3 and the results of benchmarking the discovered chunks against this oracle set.

8.2.1 What is Security?

WCT Task 3 is to find out from the perspective of a maintainer, how WCT is designed at the architectural level to achieve security. ”Security is a measure of the system's ability to resist unauthorized usage while still providing its services to legitimate users” (Bass et al., 2003). It can be characterized as a system providing non-repudiation (a transaction which has happened cannot be denied by any of the parties involved), confidentiality (protection of data or services from unauthorized access), integrity (delivery of data or services as intended), assurance (parties to a transaction are who they claim they are), availability (system available for legitimate use), and auditing (system activities tracking at sufficient levels for reconstruction) (Bass et al., 2003). The tactics (design decisions) to achieve security come in 3 categories (Bass et al., 2003): 1) Resisting Attack (authenticate users, authorize users, maintain data confidentiality, maintain integrity, limit exposure by limiting services on each host and limit access to known sources); 2) Detecting Attack (intrusion detection); 3) Recovering from Attack (restoration of state and identification of attacker by maintaining audit trails). “An audit trail is a copy of each transaction applied to the data in the system together with identifying information” (Bass et al., 2003).

The above serve as guidelines for us in the construction of the oracle set for WCT Task 3.

8.2.2 The Oracle Set for WCT Task 3

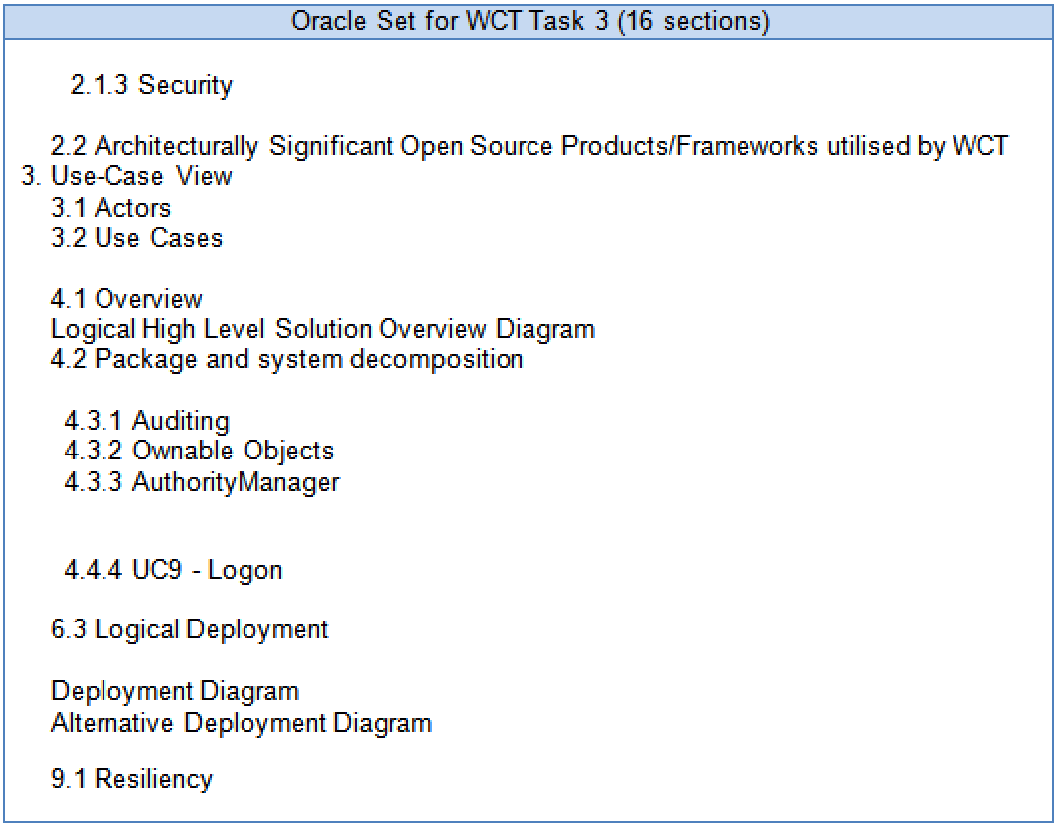
The oracle set for WCT Task 3 comprises 16 of the 47 sections of the WCT software architecture document (Table 8.2). These sections were agreed by both the researcher and a separate judge, as mandatory for WCT Task 3. In Table 8.1, these sections are denoted by yellow rows and their section IDs are marked with asterisks. This is to make it easier to compare the chunks found against the oracle set. The relevance of these sections to WCT Task 3 is explained next.

Section “2.1.3 Security” states the use of Acegi Security Framework for WCT and describes some background information of this framework. **Section “2.2 Architecturally Significant Open Source Products”** explains the open source products used in WCT. Of interest here is the Acegi Security System as a security mechanism. These 2 sections contain complementary background information on the technologies used to achieve security for WCT.

The remaining 14 sections contain specific information for WCT Task 3. Some of them contain redundant information but are included in the oracle set because they contain complementary information in one way or another.

The use-case diagram in **Section “3. Use Case View”** shows the ‘logon’ use-case and all the actors associated with it. **Section “3.1 Actors”** states the function of the 'Directory Services System' actor for authentication. It also describes role-based security access to restrict the functionalities available to different classes of ‘Nominator’ actor. The use-case for logging into the system (3.2.9 UC9 Logon) is described in **Section “3.2 Use Cases”**. It states that WCT will use a simple credential management system but will also support integration with an enterprise directory services system for pass-through authentication.

Section “4.1 Overview” contains the textual description of the main logical components of WCT. It contains the information about the use of interface to access the external authentication system. It also mentions that all the system accesses are via web browser over HTTPS (to protect logon credentials), and the authorization of users' view of the system based on their privileges. **Section "Logical High Level Solution Overview Diagram"** contains the high-level logical components diagram. This diagram shows authentication and authorisation as modules in one of the main logical components (Web Curator Tool Application Server) of the system. The diagram also shows the LDAP interaction of this component with the external directory authentication system and user accesses via web browser over HTTPS. The previous two sections formed the logical view of WCT.



Oracle Set for WCT Task 3 (16 sections)

2.1.3 Security

2.2 Architecturally Significant Open Source Products/Frameworks utilised by WCT
3. Use-Case View
3.1 Actors
3.2 Use Cases

4.1 Overview
Logical High Level Solution Overview Diagram
4.2 Package and system decomposition

4.3.1 Auditing
4.3.2 Ownable Objects
4.3.3 AuthorityManager

4.4.4 UC9 - Logon

6.3 Logical Deployment

Deployment Diagram
Alternative Deployment Diagram

9.1 Resiliency

Table 8.2: The Oracle Set for WCT Task 3

Section "4.2 Package and system decomposition" shows which Java package contains all the components required for the authentication of users in WCT. This information is important for the maintainer role assumed in WCT Task 3. **Section "4.3.1 Auditing"** describes how auditing (i.e. a security feature) is achieved in WCT.

Section "4.3.2 Ownable Objects" contains information about the concept of owners for objects, to control functional actions that can be taken on objects and screens based on the users' privilege and privilege scope. It also describes the two interfaces defined to support consistent implementation of ownable object in WCT. **Section "4.3.3 Authority Manager"** describes the function of AuthorityManager, which is to determine actions that can be taken on ownable objects. Both of these sections contain information about the access control mechanisms for WCT.

Section "4.4.4 UC9 – Logon" contains detailed and specific information about how authentication and authorisation are achieved in WCT. It explains the existence of the 'Authentication and Authorisation' module as a common component in the proposed architecture. This module relies on Acegi Security Framework which plugs into the Spring Application Framework to provide page-based, and if necessary, method-based authentication and authorisation. The section also contains information about the use of declarative security by Acegi Security Framework to define authorisation levels for functionality access. The access will be granted based on the users' privileges assigned through their roles. It also explains the hashing of passwords in database and session timeout to force logging out from WCT after a period of non-activity. This section contains much of the information needed for WCT Task 3. The information about access based on role is also described in Section "3.1 Actors". However Section "3.1 Actors" also contains description about the "Nominator" actor, the primary actor of WCT and how different classes of "Nominator" can be supported through role-based security access to restrict the functionalities available to them.

The diagram in **Section "6.3 Logical Deployment"** shows the logical mapping of components to the nodes for WCT. It shows the interaction of 'Authentication' component from the 'Web Curator Tool Server' node in the application layer with the 'Directory Server' node from external and 'Database Server' node in the data layer to support the 'logon' use-case. **Section "Deployment Diagram"** and **"Alternative Deployment Diagram"** shows the deployment of 'Authentication' component in WCT.

Section "9.1 Resiliency" contains resiliency requirements for the Harvester in WCT. It should be able to recover from exceptions, or stop independent components without halting the full system. This is related to the availability aspect of security.

We did not include Section "2.1.4 User Interface" which contains information about Spring Framework providing standard mechanism for implementing per page security in conjunction with Acegi Security Framework. The reason is this information can be found in Section "4.4.4 UC9 – Logon" which contains much more of the information needed for WCT Task 3.

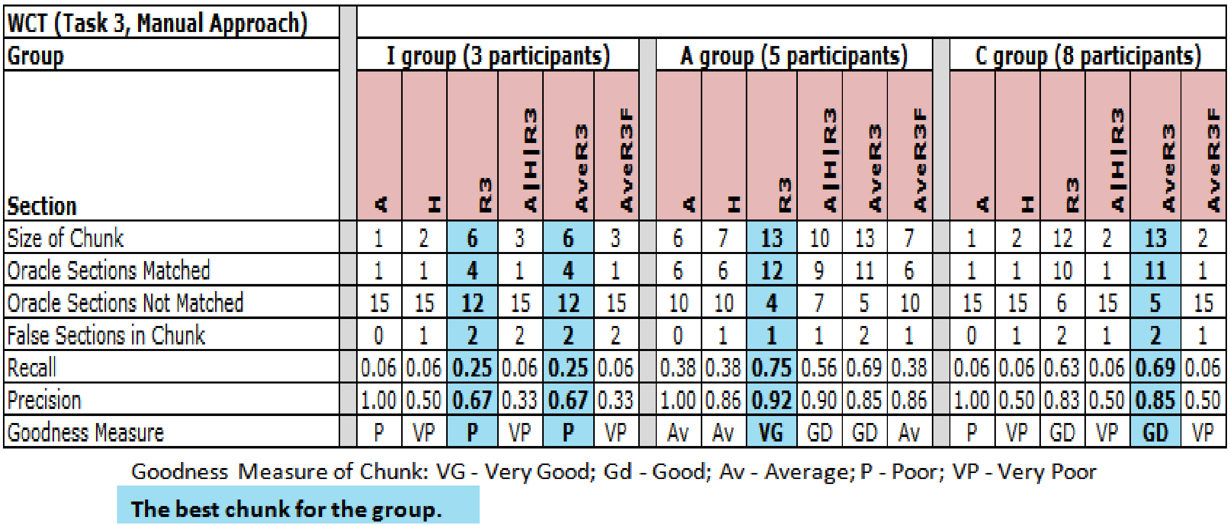
8.2.3 Benchmarking Results

Refer to Section "5.7 Benchmarking of Chunks against An Oracle Set" of Chapter 5 for all the concepts and terms needed to understand this section.

The attributes (size, numbers of oracle set's section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for WCT Task 3 are given in Table 8.3.

For C Group, the goodness of chunks found using Factors R3 and AveR3 are 'Good'. The chunk found using Factor A is 'Poor', the chunks found using Factors H, A|H|R3 and AveR3F are 'Very Poor'. We assessed the composition of the two 'Good' chunks to decide on the best chunk for C Group. The composition of Factor AveR3's chunk (recall = 0.69, precision = 0.85, size = 13, goodness = 'Good') and Factor R3's chunk (recall = 0.63, precision = 0.83, size = 12, goodness = 'Good') are almost identical. They contain the same 10 of the 16 sections in the oracle set. Factor AveR3's chunk comprises one additional section (Section "4.3.2 Ownable Objects") from the oracle set. Both chunks miss the following oracle set's sections: "Logical High Level Solution Overview Diagram", "4.2 Package and system decomposition", "4.3.1 Auditing", "6.3 Logical

Deployment” and “9.1 Resiliency”. Both chunks include two false sections: “Table of Contents” and “2.1 Architecturally Significant Design Decisions”. With almost the same composition, higher recall and precision, Factor AveR3’s chunk is the best chunk for C Group. Factor AveR3’s chunks for C and A Groups are identical.



WCT (Task 3, Manual Approach)																		
Group	I group (3 participants)						A group (5 participants)						C group (8 participants)					
Section	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
Size of Chunk	1	2	**6**	3	**6**	3	6	7	**13**	10	13	7	1	2	12	2	**13**	2
Oracle Sections Matched	1	1	**4**	1	**4**	1	6	6	**12**	9	11	6	1	1	10	1	**11**	1
Oracle Sections Not Matched	15	15	**12**	15	**12**	15	10	10	**4**	7	5	10	15	15	6	15	**5**	15
False Sections in Chunk	0	1	**2**	2	**2**	2	0	1	**1**	1	2	1	0	1	2	1	**2**	1
Recall	0.06	0.06	**0.25**	0.06	**0.25**	0.06	0.38	0.38	**0.75**	0.56	0.69	0.38	0.06	0.06	0.63	0.06	**0.69**	0.06
Precision	1.00	0.50	**0.67**	0.33	**0.67**	0.33	1.00	0.86	**0.92**	0.90	0.85	0.86	1.00	0.50	0.83	0.50	**0.85**	0.50
Goodness Measure	P	VP	**P**	VP	**P**	VP	Av	Av	**VG**	GD	GD	Av	P	VP	GD	VP	**GD**	VP

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 8.3: Attributes of Chunks (WCT Task 3)

The most basic chunk for C Group (recall = 0.06, precision = 1.00, size = 1, goodness = ‘Poor’) is found using Factor A. It comprises Section “2.1.3 Security”, which contains background information on the security technologies used in WCT. This chunk misses a significant number of sections from the oracle set and therefore is not really useful for WCT Task 3.

For I Group, the goodness of chunks found using Factors R3, AveR3 and A are ‘Poor’, whereas chunks found using Factors H, A|H|R3 and AveR3F are ‘Very Poor’. Recall that Factors R3’s and AveR3’s chunks are identical. We compare the composition of the identical chunk and Factor A’s chunk to decide on the best chunk for I Group. I Group’s chunk found using Factor R3 or Factor AveR3 (recall = 0.25, precision = 0.67, size = 6, goodness = ‘Poor’) *contains the most valuable section (“4.4.4 UC9 – Logon”) for WCT Task 3 but misses a considerable number (i.e. 12) of the sections in the oracle set and* has two false sections (Sections “Table of Contents” and “2.1 Architecturally Significant Design Decision”). Nevertheless, it is the best chunk for I Group. I Group’s chunk found using Factor A is the most basic chunk for the group. It is identical to the most basic chunk for C Group, which as mentioned in the previous paragraph is not really useful for WCT Task 3.

For A Group, the goodness of chunk found using Factor R3 is ‘Very Good’, those found using Factors AveR3 and A|H|R3 are ‘Good’, whereas chunks found using Factors A, H and AveR3F are ‘Average’. With the best level of good measure, Factor R3’s chunk (recall = 0.75, precision = 0.92, size = 13, goodness = ‘Very Good’) is the best chunk for A Group. Factor R3’s chunk matches 12 of the 16 sections in the oracle set. It misses Sections “Logical High Level Solution Overview Diagram”, “4.2 Package and system decomposition”, “6.3 Logical Deployment” and “9.1 Resiliency’. These are also missed by C and I Groups’ chunks found using

Factors AveR3 and R3. A Group’s chunk found using Factor R3 contains only 1 false section namely Section “2.1 Architecturally Significant Design Decisions”.

The most basic chunk for A Group is found using Factor A (recall = 0.38, precision = 1.00, size = 6, goodness = ‘Average’). Its precision is 1.00 but misses 10 of the oracle set’s sections, including Section “4.4.4 UC9 – Logon” (which contains much of the information needed for WCT Task 3 with some information such as passwords hashing not available in other sections).

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C Group is Factor AveR3, and for A Group is Factor R3. For I Group the best factor is Factor R3 or Factor AveR3.

Commonly-missed Oracle Set’s Sections and Common False Sections: Refer to Section “5.7.6 Determination of the Commonly-Missed Oracle Set’s Sections and Common False Sections“ of Chapter 5 on how common oracle set’s sections missed by a group of participants, and common false sections for a group of participants are determined.

For each group of participants, we try to give possible reason for commonly-missed oracle set’s sections and common false sections. If no reason is given, it means that we cannot think of any plausible reason other than a commonly-missed oracle set’s section was generally ‘not needed’, and a commonly-included false section was generally ‘needed’, by the respective group of participants when they attempted the task.

All the oracle set’s sections are commonly missed by C Group except for Section “2.1.3 Security”. They are missed by 4 factors. For I Group, oracle set’s sections commonly missed are: Sections “4.3.1 Auditing”, “4.3.2 Ownable Objects”, “4.3.3 AuthorityManager” (missed by all 6 factors); Sections “3. Use-Case View”, “3.2 Use Cases”, and “4.4.4 UC9 – Logon” (missed by 4 factors). The other nine oracle set’s sections were not rated by any participant in I Group and therefore do not qualify as commonly-missed sections. I Group’s chunks found using all the six chunk-identification factors contain Section “2.1.3 Security”.

For A Group, oracle set’s sections commonly missed are: Sections “Logical High Level Solution Overview Diagram”, “4.2 Package and system decomposition”, “9.1 Resiliency” (missed by all 6 factors); Sections “4.3.1 Auditing” (missed by 5 factors); Sections “3. Use-Case View”, “3.1 Actors”, “4.3.2 Ownable Objects”, and “6.3 Logical Deployment” (missed by 4 factors).

To summarise, none of the chunk-identification factors identifies Sections “Logical High Level Solution Overview Diagram”, “4.2 Package and system decomposition” and “9.1 Resiliency” as sections needed by any group of participants for WCT Task 3. In other words these sections do not appear in any chunk.

The common exclusion of Section “4.2 Package and system decomposition” is probably because the Java package for users authentication could be too low-level to be considered as architectural for some participants in the respective group. Section “9.1 Resiliency” describes resiliency requirements for the Harvester in WCT. These are related to the availability aspect of security but are on requirements and not the design .

As for the other commonly-missed sections, we cannot think of any plausible reason other than they were generally ‘not needed’ by the respective group of participants when attempting WCT Task 3.

Section “2.1 Architecturally Significant Design Decisions” appeared as common false section for all three groups of participants. It lists the categories of the key quality requirements that affect WCT design (where security is one of them). This section is needed by the respective group probably for getting an overall idea of the key quality requirements. I Group also includes Section “Table of Contents” as common false section, which was needed by the group probably for exploring the document.

8.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for WCT Task 3 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. The larger chunks are usually supersets of the smaller chunks.

The most basic chunks (comprising 1 to 5 sections) for C, I and A Groups are found using Factor A. The chunks found using Factors R3 and AveR3 for a group of participants are very similar if not identical (i.e. very similar for C and A Groups, and identical for I Group).

Part 1 of Table 8.4 summarises the main results of benchmarking of chunks. Recall that the number of participants in I and A Groups are 3 and 5, respectively. All the participants in each group provided some responses in terms of where the needed information (answers) can be found. One industry participant did not rate sections visited in terms of their importance to the assigned task. One academic participant did not highlight any information in the document.

In terms of how ‘good’ the chunks are for WCT Task 3, C Group has 2 chunks with goodness measures of ‘Good’, 1 of ‘Poor’ and 3 of ‘Very Poor’. I Group has 3 chunks with goodness measures of ‘Poor’ and 3 of ‘Very Poor’. A Group has 1 chunk with goodness measure of ‘Very Good’, 2 of ‘Good’ and 3 of ‘Average’.

The best chunk for C Group is found using Factor AveR3 (with goodness measure of ‘Good’), for A Group (with goodness measure of ‘Very Good’) is discovered using Factor R3, and for I Group (with goodness measure of ‘Poor’) is found using Factor R3 or Factor AveR3. The factor that produces the best chunk for a group is the best chunk-identification factor for the group.

Part 2 of Table 8.4 summarises the information or sections needed by each group of participants. This is based on the groups’ chunks found using Factor A|H|R3. Refer to Section “5.8 Information Needed” in Chapter 5 for our reason. The goodness measures for chunks found using Factor A|H|R3 for C and I Groups are ‘Very Poor’ and for A Group is ‘Good’.

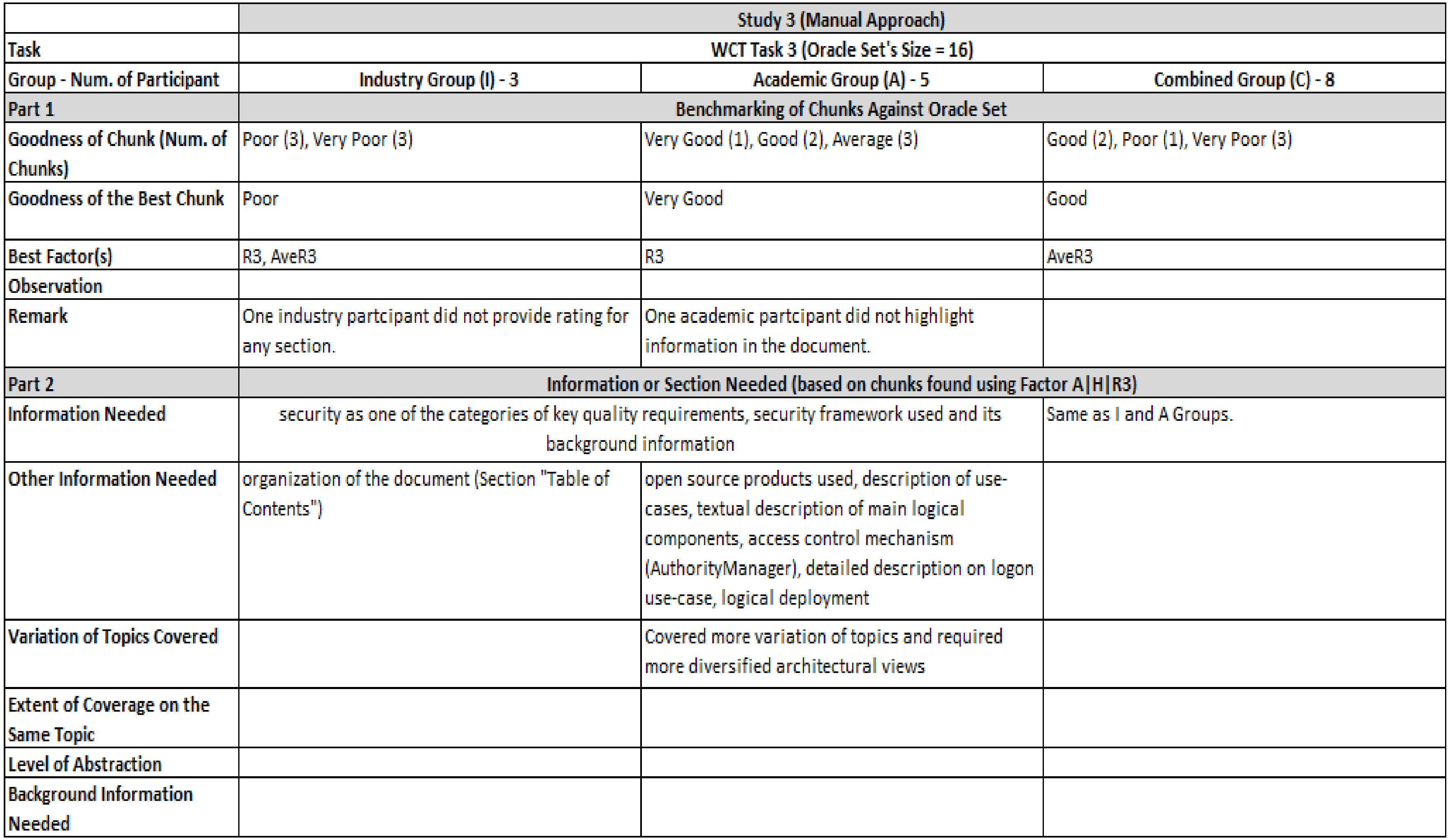
As shown in Table 8.4, both I, A and C Groups of participants needed the following information for WCT Task 3; security as one of the categories of key quality requirements, security framework used and its background information. I Group also needed organisation of the document (Section "Table of Contents"). A Group also needed open source products used, description of use-cases, textual description of main logical components, access control mechanism (AuthorityManager), detailed description on logon use-case and logical deployment. In summary, A Group covered more variation of topics (i.e. more diversified architectural views and more topics that are related to WCT Task 3).

A situation observed for WCT Task 3 is, to address security which is a cross-cutting concern that spans the entire system, both I and A Groups needed background information required by the task. In the case of WCT Task 3 this refers to the background information on the security framework used. For ASM Task 3 (which is also related to a cross-cutting concern of a system), I Group needed background information required by the task, in particular the background information on OSGi service model and Eclipse plug-in model.

8.4 **Summary**

This chapter describes the chunking of architectural information for WCT Task 3. As with ASM Task 3 (i.e. a similar task performed on the second AD), the results show the discovery of usage-based chunks for the task of how a system was designed at the architectural level to achieve a cross-cutting concern, namely security.

Chapter 13 contains the comparison of the results for WCT Task 3 and ASM Task 3, and the other tasks; the summary of the overall results for Studies 2 and 3; and threats to validity of findings. Chapter 11 presents the chunking of architectural information for ASM Task 3. The next chapter details the chunking of architectural information for ASM Task 1.



	Study 3 (Manual Approach)		
Task	WCT Task 3 (Oracle Set's Size = 16)		
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 5	Combined Group (C) - 8
Part 1	Benchmarking of Chunks Against Oracle Set		
Goodness of Chunk (Num. of Chunks)	Poor (3), Very Poor (3)	Very Good (1), Good (2), Average (3)	Good (2), Poor (1), Very Poor (3)
Goodness of the Best Chunk	Poor	Very Good	Good
Best Factor(s)	R3, AveR3	R3	AveR3
Observation			
Remark	One industry partcipant did not provide rating for any section.	One academic partcipant did not highlight information in the document.	
Part 2	Information or Section Needed (based on chunks found using Factor A|H|R3)		
Information Needed	security as one of the categories of key quality requirements, security framework used and its background information		Same as I and A Groups.
Other Information Needed	organization of the document (Section "Table of Contents")	open source products used, description of use-cases, textual description of main logical components, access control mechanism (AuthorityManager), detailed description on logon use-case, logical deployment	
Variation of Topics Covered		Covered more variation of topics and required more diversified architectural views	
Extent of Coverage on the Same Topic			
Level of Abstraction			
Background Information Needed			

Table 8.4: Summary of Results for WCT Task 3

---- Blank ----

9 Chunking of Architectural Information (ASM Task 1)

This chapter presents the chunking of architectural information for the first information-seeking task for the Aperi Storage Manager (ASM Task 1) software architecture document, for Study 2. The task was:

> “You are a software architect new to the Aperi Storage Manager project. You would like to know what the software architecture of the Aperi Storage Manager is.”

This task is similar to WCT Task 1.

We recruited 4 industry practitioners and 4 academics for this task just as we did for the other tasks. One of the industry participants had worked as a software engineer for 2 years but further inspection of the participant’s response indicated that he or she did not fulfil our selection criteria of at least 2 years industry experience related to SA. In addition, this participant did not attempt this task. Another industry participant has the same number of years of SA teaching and industry experiences. Further inspection of the participant’s data revealed that his or her background was more inclined towards the academic environment and therefore was classified as academic participant. Therefore, the results presented in this chapter are based on the responses of 7 participants who completed ASM Task 1. Two were classified as industry participants and 5 were classified as academic participants.

This chapter follows the same structure as Chapter 6. Chapter 5 provides the foundation to understand this chapter and should be read prior to reading this chapter.

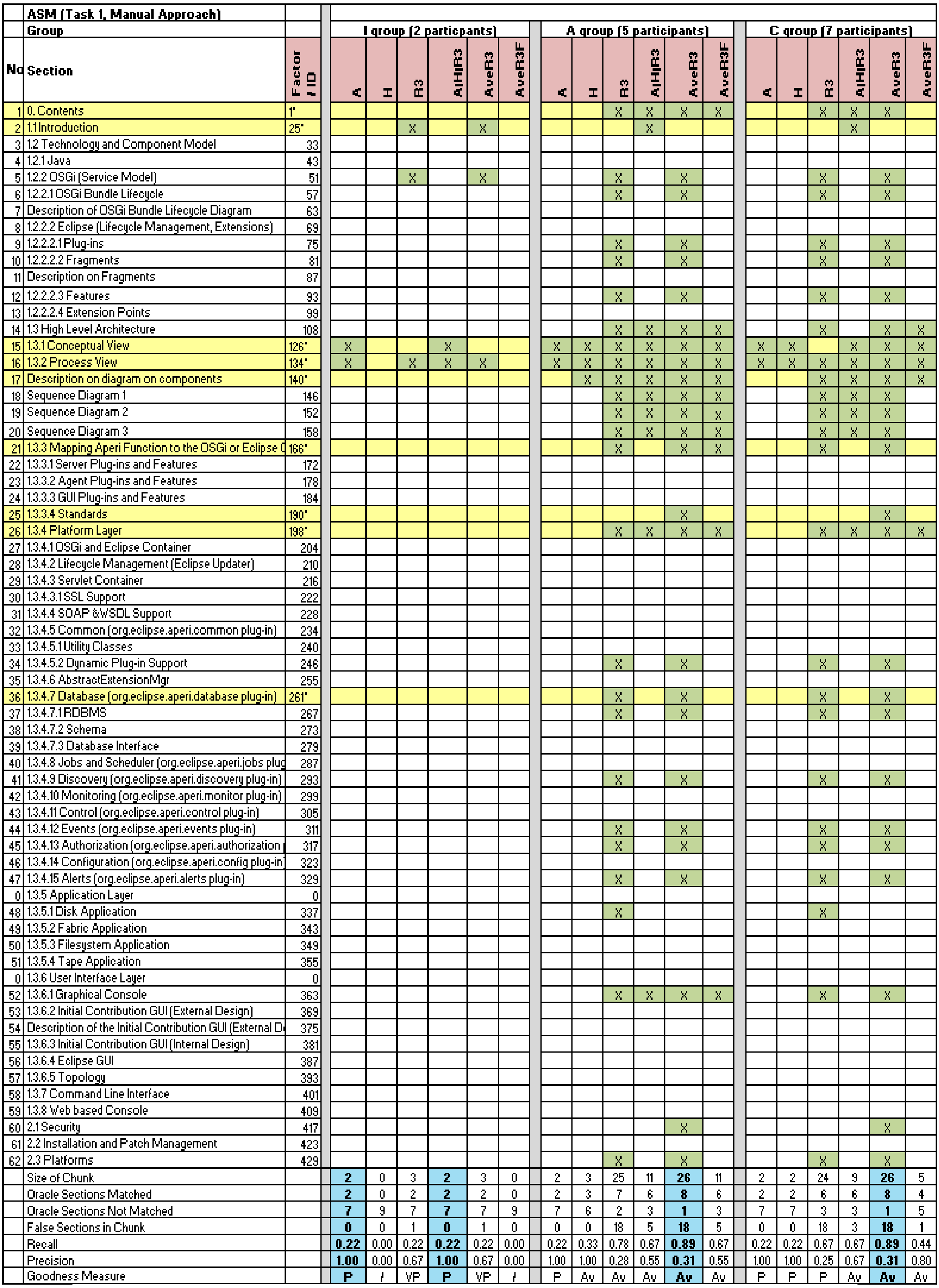
9.1 Identification and Interpretation of Chunks

We used the 6 chunk-identification to identify the chunks for this task. This was done for the 3 groups of participants: the industry practitioner (I) group, the academic (A) group and the combined (C) group of the previous two.

Table 9.1 shows the chunks discovered for ASM Task 1, for each group of participants. Refer to Section “6.1 Identification and Interpretation of Chunks” in Chapter 6 on how to interpret the chunks in the table. Recall that the size of a chunk is the number of the document’s sections that the chunk comprises.

Factor A: Chunks with two elements (comprising Section “1.3.1 Conceptual View” and “1.3.2 Process View”) are discovered for all the 3 groups of participants.

Interpretation: Section “1.3.1 Conceptual View” gives an overview of the conceptual view of the SA of ASM. It describes the two main conceptual layers of ASM. Section “1.3.2 Process View” gives an overview of the process view of the SA of ASM. It states the high-level processes and agents for deployment. It also contains a diagram that depicts the connections between the high-level processes of ASM. **Factor A’s chunks show that the majority of each group needed the overview of the conceptual view and the process view, for ASM Task 1.**



	ASM (Task 1, Manual Approach)																				
	Group		I group (2 particpants)						A group (5 participants)						C group (7 participants)					
No	Section	Factor / ID	A	H	R3	AIHR3	AveR3	AveR3F	A	H	R3	AIHR3	AveR3	AveR3F	A	H	R3	AIHR3	AveR3	AveR3F
1	0. Contents	1*									X	X	X	X			X	X	X	
2	1.1 Introduction	25*			X		X					X						X		
3	1.2 Technology and Component Model	33																		
4	1.2.1 Java	43																		
5	1.2.2 OSGi (Service Model)	51			X		X				X		X				X		X	
6	1.2.2.1 OSGi Bundle Lifecycle	57									X		X				X		X	
7	Description of OSGi Bundle Lifecycle Diagram	63																		
8	1.2.2.2 Eclipse (Lifecycle Management, Extensions)	69																		
9	1.2.2.2.1 Plug-ins	75									X		X				X		X	
10	1.2.2.2.2 Fragments	81									X		X				X		X	
11	Description on Fragments	87																		
12	1.2.2.2.3 Features	93									X		X				X		X	
13	1.2.2.2.4 Extension Points	99																		
14	1.3 High Level Architecture	108									X	X	X	X			X		X	X
15	1.3.1 Conceptual View	126*	X			X			X	X	X	X	X	X	X	X		X	X	X
16	1.3.2 Process View	134*	X		X	X	X		X	X	X	X	X	X	X	X	X	X	X	X
17	Description on diagram on components	140*								X	X	X	X	X			X	X	X	X
18	Sequence Diagram 1	146									X	X	X	X			X	X	X	
19	Sequence Diagram 2	152									X	X	X	X			X	X	X	
20	Sequence Diagram 3	158									X	X	X	X			X	X	X	
21	1.3.3 Mapping Aperi Function to the OSGi or Eclipse C	166*									X		X	X			X		X	
22	1.3.3.1 Server Plug-ins and Features	172																		
23	1.3.3.2 Agent Plug-ins and Features	178																		
24	1.3.3.3 GUI Plug-ins and Features	184																		
25	1.3.3.4 Standards	190*											X						X	
26	1.3.4 Platform Layer	198*									X	X	X	X			X	X	X	X
27	1.3.4.1 OSGi and Eclipse Container	204																		
28	1.3.4.2 Lifecycle Management (Eclipse Updater)	210																		
29	1.3.4.3 Servlet Container	216																		
30	1.3.4.3.1 SSL Support	222																		
31	1.3.4.4 SOAP &WSDL Support	228																		
32	1.3.4.5 Common (org.eclipse.aperi.common plug-in)	234																		
33	1.3.4.5.1 Utility Classes	240																		
34	1.3.4.5.2 Dynamic Plug-in Support	246									X		X				X		X	
35	1.3.4.6 AbstractExtensionMgr	255																		
36	1.3.4.7 Database (org.eclipse.aperi.database plug-in)	261*									X		X				X		X	
37	1.3.4.7.1 RDBMS	267									X		X				X		X	
38	1.3.4.7.2 Schema	273																		
39	1.3.4.7.3 Database Interface	279																		
40	1.3.4.8 Jobs and Scheduler (org.eclipse.aperi.jobs plug	287																		
41	1.3.4.9 Discovery (org.eclipse.aperi.discovery plug-in)	293									X		X				X		X	
42	1.3.4.10 Monitoring (org.eclipse.aperi.monitor plug-in)	299																		
43	1.3.4.11 Control (org.eclipse.aperi.control plug-in)	305																		
44	1.3.4.12 Events (org.eclipse.aperi.events plug-in)	311									X		X				X		X	
45	1.3.4.13 Authorization (org.eclipse.aperi.authorization	317									X		X				X		X	
46	1.3.4.14 Configuration (org.eclipse.aperi.config plug-in)	323																		
47	1.3.4.15 Alerts (org.eclipse.aperi.alerts plug-in)	329									X		X				X		X	
0	1.3.5 Application Layer	0																		
48	1.3.5.1 Disk Application	337									X						X			
49	1.3.5.2 Fabric Application	343																		
50	1.3.5.3 Filesystem Application	349																		
51	1.3.5.4 Tape Application	355																		
0	1.3.6 User Interface Layer	0																		
52	1.3.6.1 Graphical Console	363									X	X	X	X			X		X	
53	1.3.6.2 Initial Contribution GUI (External Design)	369																		
54	Description of the Initial Contribution GUI (External De	375																		
55	1.3.6.3 Initial Contribution GUI (Internal Design)	381																		
56	1.3.6.4 Eclipse GUI	387																		
57	1.3.6.5 Topology	393																		
58	1.3.7 Command Line Interface	401																		
59	1.3.8 Web based Console	409																		
60	2.1 Security	417											X						X	
61	2.2 Installation and Patch Management	423																		
62	2.3 Platforms	429									X		X				X		X	
	Size of Chunk		**2**	0	3	**2**	3	0	2	3	25	11	**26**	11	2	2	24	9	**26**	5
	Oracle Sections Matched		**2**	0	2	**2**	2	0	2	3	7	6	**8**	6	2	2	6	6	**8**	4
	Oracle Sections Not Matched		**7**	9	7	**7**	7	9	7	6	2	3	**1**	3	7	7	3	3	**1**	5
	False Sections in Chunk		**0**	0	1	**0**	1	0	0	0	18	5	**18**	5	0	0	18	3	**18**	1
	Recall		**0.22**	0.00	0.22	**0.22**	0.22	0.00	0.22	0.33	0.78	0.67	**0.89**	0.67	0.22	0.22	0.67	0.67	**0.89**	0.44
	Precision		**1.00**	0.00	0.67	**1.00**	0.67	0.00	1.00	1.00	0.28	0.55	**0.31**	0.55	1.00	1.00	0.25	0.67	**0.31**	0.80
	Goodness Measure		**P**	/	VP	**P**	VP	/	P	Av	Av	Av	**Av**	Av	P	P	P	Av	**Av**	Av

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

Table 9.1: The Chunks for ASM Task 1 Identified Using Different Factors

Factor H: C Group’s chunk found using Factor H is identical to the group’s chunk found using Factor A. No chunk is discovered for I Group. A Group’s chunk found using Factor H is similar to the group’s chunk found using Factor A, but with the addition of Section “Description on diagram on components”.

Interpretation: The fact that Factor H produces the same chunk as Factor A for C Group reinforces that the **majority of the group needed the overview of the conceptual view and the process view, for ASM Task 1**.

The inspection of the raw data reveals that the no chunk situation for I Group was due to 1 of the 2 industry participants did not highlight information in any section, when attempting ASM Task 1. Consequently, none of the section could meet the section inclusion criteria of Factor H when this factor is used to find the chunk for I Group. One of the 5 academic participants also did not highlight information in any section, when attempting ASM Task 1. This could have some impact on the chunk found using Factor H for A Group.

The additional section in A Group’s chunk provides the textual description of each of the high-level processes, the different agents and the relationships between the processes, as shown in the diagram in Section “1.3.2 Process View". **A Group’s chunk found using Factor H shows that the majority of the academic participants also needed the additional textual description of the process view, when they attempted ASM Task 1.**

Factor R3: C Group’s chunk found using Factor R3 contains 24 sections. These sections are related to Table of contents, brief overview of OSGi service model (section 1.2.2), diagram on the lifecycle of an OSGi bundle (Section “1.2.2.1 OSGi Bundle Lifecycle”), low-level detailed information on plug-ins (Section “1.2.2.2.1 Plug-ins”), brief description of plug-in fragment together with a diagram (Section “1.2.2.2.2 Fragments”), description of feature as a group of plug-ins that define a logical product feature (Section “1.2.2.2.3 Features”), Section “1.3 High Level Architecture” which contains only the titles of its sub-sections, overview of process view and the corresponding diagram (Section “1.3.2 Process View”), additional textual description of the process view (“Description on diagram on components”), the 3 sequence diagrams, general description of how the major components of Aperi are represented (which is as an OSGi bundle or a Eclipse plug-in or both) (Section “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model”), brief overview of platform layer (Section “1.3.4 Platform Layer”), dynamic plug-in support (section 1.3.4.5.2), database (section 1.3.4.7), the support of multiple Relational Database Management Systems (RDBMSs) (section 1.3.4.7.1), certain generic services which are discovery, events, authorization and alerts (section 1.3.4.9, 1.3.4.12, 1.3.4.13, 1.3.4.15), disk application (section 1.3.5.1), overview of Graphical User Interface (section 1.3.6.1), and initial development platforms (section 2.3).

I Group’s chunk comprises 3 sections: overview of ASM (Section “1.1 Introduction”), brief overview of OSGi service model (section 1.2.2), and overview of process view and the corresponding diagram (Section “1.3.2 Process View”). A Group’s chunk found using Factor R3 is nearly identical to C Group’s chunk found using the same factor. It contains an extra section which gives an overview of the conceptual view of the SA of ASM (Section “1.3.1 Conceptual View”).

Interpretation: C Group's chunk found using Factor R3 is large in terms of the number of sections that it contained. As with C Group's chunks found using some other factors, C Group's chunk found using Factor R3 contains Section "1.3 High Level Architecture", section on the overview of the process view and the corresponding diagram, section on additional textual description of the process view, and the sections on the 3 sequence diagrams. C Group's chunk found using Factor R3 contains a number of sections not presented in the group's chunks found using other factors (except for Factor AveR3). These sections are mainly related to OSGi service model, more detailed information related to Eclipse plug-in model (plug-ins, fragments and features), representation of the major components of Aperi (as OSGi bundle or Eclipse plug-in or both), dynamic plug-in support, database, the support of multiple Relational Database Management System (RDBMS), certain generic services, disk application, overview of Graphical User Interface (GUI), and initial development platform. **C Group's chunk found using Factor R3 shows that the majority of the group who rated the sections of the ASM document mentioned in this paragraph, needed them for ASM Task 1.**

The numbers of participants in I and A Groups differ significantly (2 industry participants versus 5 academic participants). One of the 2 industry participants, and 1 of the 5 academic participants did not provide ratings for any section of the ASM document, when attempting ASM Task 1. The 'not rating at all' by some participants could have affected the chunks found using Factor AveR3F but not the chunks found using Factors R3 and AveR3. This is because Factor AveR3F is based on the majority of a group, while Factors R3 and AveR3 are based on the majority of those who rated the respective sections.

I Group's chunk found using Factor R3 shows that **sections related to overview information (ASM, OSGi service model and process view) were needed by the majority of the industry participants who rated them, for ASM Task 1.**

Both of I and A Group's chunks found using Factor R3 contain the section on the overview of OSGi service model and the section on the overview of the process view. A Group's chunk also contains sections that have more detailed information on OSGi service model and Eclipse plug-in model, overview of the conceptual view, additional textual description of the process view, sequence diagrams, representation of major components of Aperi, brief overview of platform layer, dynamic plug-in support, database, RDBMS, certain generic services, disk application, overview of GUI, initial development platform and Section "1.3 High Level Architecture".

A Group's chunk found using Factor R3 shows that the majority of the academic participants who rated the respective sections, needed detailed (or in-depth) information on the following topics:

1) The background information (i.e. OSGi Service Model and Eclipse plug-in model) required for ASM Task 1.

2) The platform layer, in terms of the issues related to database and some generic services in the layer.

3) The process view, in terms of the additional textual description of the process view, and the sequence diagrams (which contain the lower-level information on the process view). In fact, the majority of the academic participants required comprehensive information on the process view. Verification with the raw data shows that the same 3 academic participants rated all the sections

related to the process view and gave these sections high ratings of either 4 or 5. These sections are on the overview of the process view, the additional textual description of the process view, and the sequence diagrams.

C Group's chunk is almost identical to A Group's chunk found using Factor R3. This means that the majority of the overall participants who rated the respective sections for ASM Task 1, needed the same information (except for the overview of the conceptual view) as the majority of the academic participants who rated the respective sections. This is not surprising because 5 out of the 7 participants in C Group are academic participants from A Group.

Factor A|H|R3: C Group's chunk comprises sections related to Table of contents, overview of ASM (Section "1.1 Introduction"), overview of conceptual view (Section "1.3.1 Conceptual View"), overview of process view and the corresponding diagram (Section "1.3.2 Process View"), additional textual description of the process view (Section "Description on diagram on components"), all the 3 sequence diagrams and brief overview of the platform layer (Section "1.3.4 Platform Layer"). I Group's chunk is identical to the group's chunk found using Factor A. The chunk for A Group is almost the same as C Group's with the additional sections of "1.3 High Level Architecture" and "1.3.6.1 Graphical Console".

Interpretation: Recall that this composite factor caters for the possibility that participants have different preferences in indicating the relevance of a section of the document to them, for the assigned task. They could have indicated that the section was where they found or searched for their answers, or they could have highlighted relevant information in the section or they could have rated the section with an importance rating of 3 and above, or they could have done a combination of these.

C Group's chunk found using Factor A|H|R3 shows that when all the different preferences in relevance indication were taken into consideration, **in addition to Table of Contents, the majority of the participants (C group) needed some overview information (related to ASM, conceptual view, process view, platform layer) to understand the SA of ASM. They also needed detailed information and comprehensive information on the process view by also requiring the additional textual description of the process view, and the sequence diagrams. The sequence diagrams contain lower-level information on the process view.**

The high similarity between the chunks for A and C Groups is because the majority (i.e. 5 out of 7) of the participants in C Group are from A Group. **A Group's chunk reveals that in addition to those information contained in C Group's chunk found using Factor A|H|R3, the majority of the academic participants also required overview of GUI and Section "1.3 High Level Architecture" (which contains only the titles of its sub-sections) for ASM Task 1.**

I Group's chunk found using Factor A|H|R3 shows that **the majority of the industry participants needed the overview of the conceptual view and the process view, just as what are revealed by the group's chunk found using Factor A.**

Factor AveR3: The chunk found using Factor AveR3 for a particular group of participants is very similar if not identical to the group’s chunk found using Factor R3. For C and A Groups, the chunks found using Factor AveR3 and Factor R3 are very similar. For I Group, they are identical. The chunks found using Factor AveR3 for A Group and C Group are identical.

Comparing with C Group’s chunk found using Factor R3, C Group’s chunk found using Factor AveR3 has the additional sections of “1.3.1 Conceptual View”, “1.3.3.4 Standards”, “2.1 Security” but excludes Section “1.3.5.1 Disk Application“. Comparing with A Group’s chunk found using Factor R3, A Group’s chunk found using Factor AveR3 also includes Sections “1.3.3.4 Standards” and “2.1 Security” but excludes Section “1.3.5.1 Disk Application“.

Interpretation: C Group’s chunk found using Factor AveR3 shows that in addition to those sections in the group’s chunk found using Factor R3, sections on the overview of the conceptual view and standards adopted were also needed by some of the participants who rated them. It is surprising that the section on security which does not contain any valuable information was needed by some of the participants who rated the section, when they attempted ASM Task 1. A Group’s chunk found using Factor AveR3 reveals that in addition to those sections in the group’s chunk found using Factor R3, sections on standards adopted and security, were needed by some academic participants who rated them, when they attempted ASM Task 1.

I Group’s chunk found using Factor AveR3 shows that sections related to overview information (ASM, OSGi service model and process view) were needed by the one industry participant who rated them, when he or she attempted ASM Task 1.

Factor AveR3F: C Group’s chunk found using Factor AveR3F comprises Sections “1.3 High Level Architecture”, “1.3.1 Conceptual View”, “1.3.2 Process View”, “Description on diagram on components” and “1.3.4 Platform Layer”. No chunk is discovered for I Group. The chunk for A Group found using Factor AveR3F is nearly identical to the group’s chunk found using Factor A|H|R3, with the addition of Section “1.3.3 Mapping Aperi function to the OSGi or Eclipse Component Model” and exclusion of Section “1.1 Introduction”.

Interpretation: This factor filters the chunks produced by Factor AveR3 to include only those sections rated by more than half of the participants in a group and with the average ratings of at least 3. The purpose is to remove those sections which have a small number of participants rating them.

C Group’s chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, overview information (on the conceptual view, process view and platform layer), additional textual description of the process view and Section “1.3 High Level Architecture”, were needed for ASM Task 1.

The no chunk situation for I Group is because 1 of the 2 industry participants did not provide rating for any section of the document, when attempting ASM Task 1. This made it impossible for any section of the ASM document to meet the section inclusion criteria of Factor AveR3F, when this factor is used to find chunk for I Group.

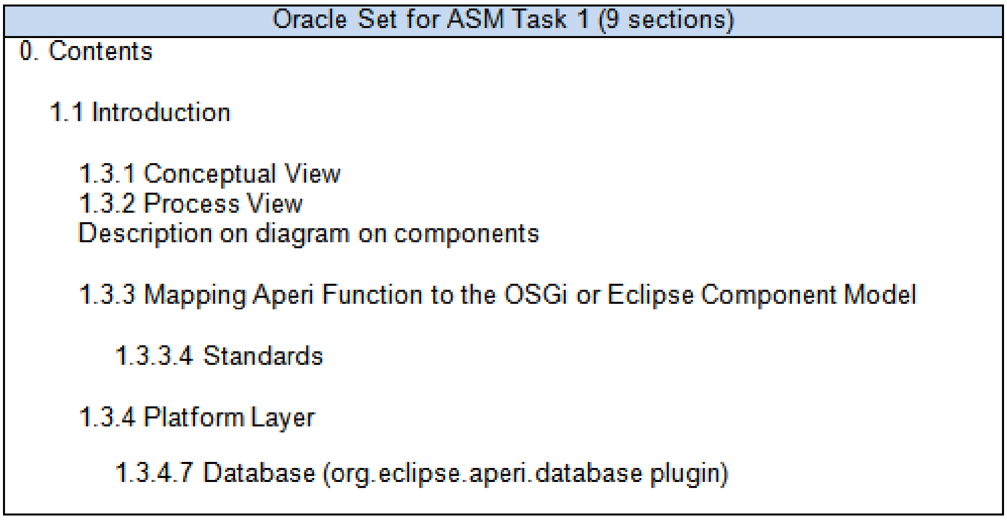
The resemblance of A Group's chunk found using Factor AveR3F to the group's chunk found using Factor A|H|R3 reinforces our point that the sections contained in Factor A|H|R3's chunk were needed by A Group for understanding the SA of ASM. In addition, section on representation of the major components of Aperi and the overview of ASM were also needed by A Group.

9.2 Benchmarking Chunks against An Oracle Set

This section describes the oracle set for ASM Task 1 and the results of benchmarking the discovered chunks against this oracle set. Refer to Section "6.2.1 What is Software Architecture?" in Chapter 6 on the notion of SA that we used as guidelines in our construction of the oracle set for ASM Task 1. We also included in the oracle set, sections that are helpful for understanding what the system is as that subsequently helps in understanding its SA. We also included section related to standards adopted that have some influences on the architectural design of ASM.

9.2.1 The Oracle Set for ASM Task 1

The oracle set for ASM Task 1 comprises the sections of the ASM document agreed by both the researcher and a separate judge, as mandatory to obtain an overall picture of the SA of ASM. It comprises 9 of the 62 sections of the ASM software architecture document, as shown in Table 9.2. In Table 9.1, these sections are denoted by yellow rows and their section IDs are marked with asterisks. This is to make it easier to compare the chunks found against the oracle set. The relevance of these sections to ASM Task 1 is explained next.



Oracle Set for ASM Task 1 (9 sections)
0. Contents
1.1 Introduction
1.3.1 Conceptual View 1.3.2 Process View Description on diagram on components
1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model
1.3.3.4 Standards
1.3.4 Platform Layer
1.3.4.7 Database (org.eclipse.aperi.database plugin)

Table 9.2: The Oracle Set for ASM Task 1

Section "0. Contents" gives a detailed organisation of the document. The table of content gives a general idea on the layered architecture of ASM: Platform layer, Application layer and User Interface (UI) layer. It is included in the oracle set because Section "1.3.1 Conceptual View" which gives an overview of the conceptual layers of ASM omits the UI layer and focuses only on the Platform and the Application layers.

Section "1.3.1 Conceptual View" gives an overview of the conceptual view. This section describes the two main conceptual layers of the system: the Platform layer which provides basic storage management functions and the basic server infrastructure; the Application layer which provides high-level functions (such as file system capacity reporting, fabric zone control). **Section "1.3.4 Platform Layer"** provides a brief description on the Platform layer, which complements the description of the same layer in Section "1.3.1 Conceptual View". These 2 sections of ASM software architecture document are important for understanding the module architectural structure of ASM and are included in the oracle set.

Section "1.3.2 Process View" gives an overview of the process view. This section states the high-level processes and agents for deployment. It also contains a diagram that depicts the connections between these processes. **Section "Description on diagram on components"** provides additional explanation on the elements shown in the diagram in Section "1.3.2 Process View". These 2 sections contain information related to the component-and-connector structure (or runtime architectural structure) of ASM. As a result, the 2 sections of the ASM software architecture document are included in the oracle set for ASM Task 1.

Section "1.3.4.7 Database" emphasises database as an essential component of the overall architecture, which is as a repository of information, and also essential data transfer mechanism between components.

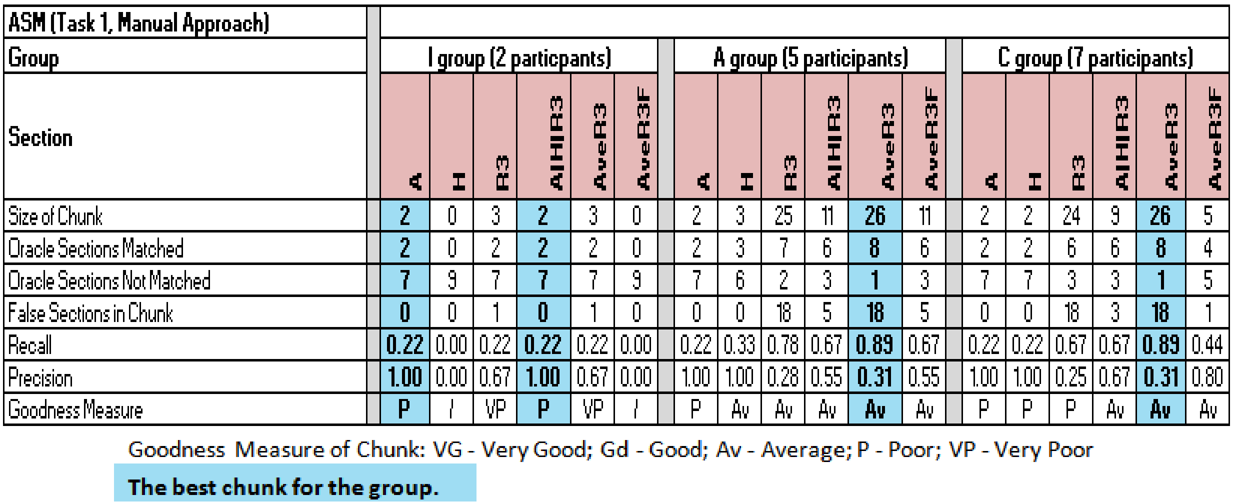
The 6 sections of the ASM software architecture document discussed prior to this contain specific information on the architectural design of ASM.

Section "1.1. Introduction" gives an overview of what Aperi Storage Manager is, a brief history on it, its goals, its leveraging the Eclipse Platform and the purpose of the document. The goals (extensible, componentized, commercial-friendly) and dependency (on Eclipse Platform) of ASM impacted its architectural design. Understanding this and what the system is, are helpful to understand the SA of ASM. **Section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model"** explicitly mentions that each major component of ASM is represented as an OSGi bundle or Eclipse plug-in or both. Understanding what are the building blocks used in ASM helps to understand its overall SA. **Section "1.3.3.4 Standards"** mentions that storage management standards are important for ASM. It states the standard used (SNIA SMI-S) and other standards (SNMP and GS3) that could be used. The use of standards has some influences on the architectural design of ASM. In particular, it makes ASM dependent on the SMI-S and SNMP agents (apart from the host agent) for deployment. This dependency of ASM is explained in Section "1.3.2 Process View". The 3 sections of the ASM software architecture document discussed in this paragraph contain supporting information that helps to understand ASM's software architecture, and therefore included in the oracle set.

9.2.2 Benchmarking Results

Refer to Section "5.7 Benchmarking of Chunks against An Oracle Set" of Chapter 5 for all the concepts and terms needed to understand this section. The attributes (size, numbers of oracle set's section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for ASM Task 1 are given in Table 9.3.

For C Group, the goodness of chunks found using Factors A|H|R3, AveR3 and AveR3F are ‘Average’, while the goodness of chunks found using the rest of the factors are ‘Poor’. We assessed the composition of the 3 ‘Average’ chunks to decide on the best chunk or the most favourable chunk for C Group.



ASM (Task 1, Manual Approach)																		
Group	I group (2 particpants)						A group (5 participants)						C group (7 participants)					
Section	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
Size of Chunk	**2**	0	3	**2**	3	0	2	3	25	11	**26**	11	2	2	24	9	**26**	5
Oracle Sections Matched	**2**	0	2	**2**	2	0	2	3	7	6	**8**	6	2	2	6	6	**8**	4
Oracle Sections Not Matched	**7**	9	7	**7**	7	9	7	6	2	3	**1**	3	7	7	3	3	**1**	5
False Sections in Chunk	**0**	0	1	**0**	1	0	0	0	18	5	**18**	5	0	0	18	3	**18**	1
Recall	**0.22**	0.00	0.22	**0.22**	0.22	0.00	0.22	0.33	0.78	0.67	**0.89**	0.67	0.22	0.22	0.67	0.67	**0.89**	0.44
Precision	**1.00**	0.00	0.67	**1.00**	0.67	0.00	1.00	1.00	0.28	0.55	**0.31**	0.55	1.00	1.00	0.25	0.67	**0.31**	0.80
Goodness Measure	**P**	/	VP	**P**	VP	/	P	Av	Av	Av	**Av**	Av	P	P	P	Av	**Av**	Av

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 9.3: Attributes of Chunks (ASM Task 1)

C Group’s chunk found using Factor AveR3 (recall = 0.89, precision = 0.31, size = 26, goodness = ‘Average’) contains 8 of the 9 sections in the oracle set but misses Section “1.1 Introduction”. It includes 18 false sections. These false sections are related to OSGi Service Model (including bundle lifecycle) and more detailed information related to Eclipse plug-in model (plug-ins, fragments, features), Section “1.3 High Level Architecture”, sequence diagrams, dynamic plug-in support, RDBMS, certain generic services (discovery, events, authorization and alerts), overview of GUI, security and initial development platform. C Group’s chunk found using Factor A|H|R3 (recall = 0.67, precision = 0.67, size = 9, goodness = ‘Average’) matches 6 of the 9 sections in the oracle set. It misses Sections “1.3.3 Mapping Aperi function to the OSGi or Eclipse Component Model”, “1.3.3.4 Standards” and “1.3.4.7 Database (org.eclipse.aperi.database plugin)”. It includes 3 false sections which are the sections on the 3 sequence diagrams. These sections are also in the Factor AveR3’s chunk found for C Group.

C Group’s chunk found using Factor AveR3F (recall = 0.44, precision = 0.80, size = 5, goodness = ‘Average’) matches 4 of the 9 sections in the oracle set. It misses 5 oracle set’s sections: “0. Contents”, “1.1 Introduction”, “1.3.3 Mapping Aperi function to the OSGi or Eclipse Component Model”, “1.3.3.4 Standards” and “1.3.4.7 Database (org.eclipse.aperi.database plugin)”. It includes 1 false section which is Section “1.3 High Level Architecture”.

Though having very high precision (0.80), Factor AveR3F’s chunk misses many more oracle set’s sections when compared to Factors AveR3’s and A|H|R3’s chunks found for C Group. Comparing the two latter chunks, Factor AveR3’s chunk has 6 times more false sections than Factor A|H|R3’s chunk but it matches 8 of the 9 sections in the oracle set. In addition, the only oracle set’s section Factor AveR3’s chunk excludes is Section “1.1 Introduction”. The exclusion of this section is relatively less critical (in terms of the information contained) compared to the collective effect of the exclusion of the 3 oracle set's sections from Factor A|H|R3's chunk. Consequently, we take Factor AveR3's chunk as the best chunk or the most favourable chunk for C Group.

The most basic chunk for C Group is the identical chunk found using Factors A and H (recall = 0.22, precision = 1.00, size = 2, goodness = 'Poor'). This chunk comprises 2 sections from the oracle set: section on the overview of conceptual view, and the overview of the process view together with the corresponding diagram. It misses a considerable number of sections from the oracle set and therefore is not useful for ASM Task 1.

For I Group, with goodness measures of 'Poor' the identical chunk found using Factors A and A|H|R3 (recall = 0.22, precision = 1.00, size = 2, goodness = 'Poor') is better than the identical 'Very Poor' chunk found using Factors R3 and AveR3. The former chunk is the best chunk, and also the most basic chunk for I Group. It is identical to the most basic chunk for C Group (found using Factors A and H), and the most basic chunk for A Group (found using Factor A).

For A Group, all the factors produce 'Average' chunks except for Factor A which produces 'Poor' chunk. Among the 'Average' chunks, Factor H's chunk (recall = 0.33, precision = 1.00, size = 3, goodness = 'Average') has the best precision (1.00) but the lowest recall, matching only 3 of the 9 sections in the oracle set. The other two 'Average' chunks found using Factor R3 (recall = 0.78, precision = 0.28, size = 25, goodness = 'Average') and Factor AveR3 (recall = 0.89, precision = 0.31, size = 26, goodness = 'Average') are very similar in terms of their compositions. Factor R3's chunk matches 7 of the 9 oracle set's section while Factor AveR3's chunk matches 8. Both chunks include 18 false sections. With higher recall and precision, Factor AveR3's chunk is better than Factor R3's.

The last two A Group's chunks with 'Average' goodness measures are produced by Factors A|H|R3 and AveR3F. These chunks have the same values for the 4 measures (recall = 0.67, precision = 0.55, size = 11, goodness = 'Average'). Their compositions are very similar. They differ only in one of the oracle set's sections that they miss. Both chunks match 6 of the 9 sections in the oracle set and contain 5 false sections. Both chunks include 5 of 6 oracle set's sections which contain specific information on the architectural design of ASM: Sections "0. Contents", "1.3.1 Conceptual View", "1.3.2 Process View", "Description on diagram on components" and "Platform Layer". Both miss Section "1.3.4.7 Database (org.eclipse.aperi.database plugin)" which also contains specific information on the architectural design of ASM. Both chunks also miss Section "1.3.3.4 Standards" which contains supporting information that helps in understanding ASM's software architecture. Factor A|H|R3's chunk also misses Section "1.3.3 Mapping Aperi function to the OSGi or Eclipse Component Model", whereas Factor AveR3F's chunk misses Section "1.1 Introduction". These two excluded sections contain supporting information that helps in understanding ASM's software architecture.

Factor A|H|R3's and Factor AveR3F's chunk for A Group contain the same 5 false sections: Section "1.3 High Level Architecture", the 3 sequence diagrams, and Section "1.3.6.1 Graphical Console".

Despite that Factor AveR3's chunk contains 3.6 times more false sections than Factor A|H|R3's and Factor AveR3F's chunk found for A Group, we chose chunk from Factor AveR3 as the best chunk for A Group. Our reasons are Factor AveR3's chunk contains more oracle set's sections and also it contains Section "1.3.4.7 Database" which contains specific information on the architectural design of ASM. The best chunk for A Group is identical to the best chunk for C Group, which are both found using Factor AveR3.

The most basic chunk for A Group is found using Factor A (recall = 0.22, precision = 1, size = 2, goodness = ‘Poor’). As mentioned earlier, it is identical to the most basic chunk for C Group (found using Factors A and H), and the most basic chunk for I Group (found using Factors A and A|H|R3).

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C and A Groups is Factor AveR3. For I Group the best factor is Factor A or Factor A|H|R3.

Commonly-missed Oracle Set’s Sections and Common False Sections: Refer to Section “5.7.6 Determination of the Commonly-Missed Oracle Set’s Sections and Common False Sections” of Chapter 5 on how common oracle set’s sections missed by a group of participants, and common false sections for a group of participants are determined.

For each group of participants, we try to give possible reason for commonly-missed oracle set’s sections and common false sections. If no reason is given, it means that we cannot think of any plausible reason other than a commonly-missed oracle set’s section was generally ‘not needed’, and a commonly-included false section was generally ‘needed’, by the respective group of participants when they attempted the task.

The oracle set’s sections commonly missed by C Group are: “1.1 Introduction” and “1.3.3.4 Standards” (missed by 5 of 6 chunk-identification factors); and “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” and “1.3.4.7 Database (org.eclipse.aperi.database plugin)” (missed by 4 factors).

Section “1.1 Introduction” gives an overview of ASM, its goals, its leveraging the Eclipse Platform and the purpose of the ASM software architecture document. It is commonly excluded probably because it does not contain specific information on the SA of ASM. Section “1.3.3.4 Standards” is excluded probably because most of the participants failed to make the connection that the use of standards influences the architectural design of ASM. In particular, it makes ASM dependent on the SMI-S and SNMP agents (apart from the host agent) for deployment. This dependency of ASM is explained in Section “1.3.2 Process View”. As for the exclusion of Sections “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” and “1.3.4.7 Database (org.eclipse.aperi.database plugin)”, we cannot think of any plausible reason other than they were generally ‘not needed’ by this group of participants, when they attempted the task.

There is no oracle set’s sections commonly missed by I Group. None of the chunk-identification factors includes the following oracle set’s sections in chunks found for I Group: “0. Contents”, “Description on diagram on components”, “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model”, “1.3.3.4 Standards”, “1.3.4 Platform Layer” and “1.3.4.7 Database (org.eclipse.aperi.database plugin)”. These sections were not rated by any of the participant of I Group and therefore do not qualify as commonly-missed sections.

There is no common false section for C and I Groups.

The oracle set’s sections commonly missed by A Group are: “1.1 Introduction” and “1.3.3.4 Standards” (missed by 5 factors); and “1.3.4.7 Database (org.eclipse.aperi.database plugin)” (missed by 4 factors). The probable reasons for the common exclusion of Sections “1.1 Introduction” and “1.3.3.4 Standards” are the same as what are given for C Group.

The common false sections for A Group are: Section “1.3 High Level Architecture” which contains the titles of its sub-sections, the 3 sections on the sequence diagram, and Section “1.3.6.1 Graphical Console”. These sections are included in chunks by 4 factors. Section “1.3 High Level Architecture” contains only the titles of its sub-sections. It indirectly gives a high-level overview of the organisation of the subsequent sections in the document. The 3 sequence diagrams show typical communication flow among various components, with regards to certain functionalities of ASM. These diagrams are related to the process view of the SA of ASM. Section “1.3.6.1 Graphical Console” gives an overview of possible graphical user interfaces. **This shows that A Group required the organisation of the document, the lower-level information on the process view in terms of sequence diagrams, and also the overview of GUI, for ASM Task 1.**

9.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for ASM Task 1 for each group of participants. Generally, two clusters of chunks are found for C and A Groups but not I Group. Cluster 1 consists of the smaller chunks found using Factors A, H, A|H|R3 and AveR3F. Cluster 2 consists of the larger chunks found using Factors R3 and AveR3.

The chunks found using Factors R3 and AveR3 for a group of participants, are very similar if not identical to each other (very similar for C and A Groups, and identical for I Group). All the chunks found for I Group are small, comprising 2 or 3 sections.

The most basic chunks (i.e. chunk with the smallest number of section) for the 3 groups are identical. They are produced by Factors A and H for C Group, Factors A and A|H|R3 for I Group, and Factor A for A Group. These two-element chunks contain sections on the overview of the conceptual view, and the overview of the process view.

Part 1 of Table 9.4 summarises the main results of benchmarking of chunks. It is important to bear in mind that the number of participants in I and A Groups differ significantly (2 industry participants versus 5 academic participants). One of the 2 industry participants did not provide rating for any section of ASM document when attempting the task and this affects I Group’s chunk found using Factor AveR3F. One of the industry participants did not highlight information in any section and this causes no chunk being found for I Group using Factor H. One of the 5 academic participants did not provide rating for any section of the document.

In terms of how ‘good’ the chunks are for ASM Task 1, C Group has 3 chunks with Average’ and 3 with ‘Poor’ goodness measures. I Group has 2 chunks with ‘Poor’ and 2 with ‘Very Poor’ goodness measures. No chunk is found for I Group using Factors H and AveR3F. A Group has 5 chunks with ‘Average’ and 1 with ‘Poor’ goodness measures.

Recall that for each group of participants, we assessed the composition of the chunks that are at the best level of goodness measure to decide the best chunk for a particular group. The best chunk for C and A Groups are found using Factor AveR3 (with goodness measure of ‘Average’). The best chunk for I Group is found using Factors A and A|H|R3 (with goodness measure of ‘Poor’). The factor that produces the best chunk for a group is the best chunk-identification factor for the group.

Part 2 of Table 9.4 summarises the information or sections needed by each group of participants. This is based on the groups' chunks found using Factor A|H|R3. Refer to Section "5.8 Information Needed" in Chapter 5 for our reason. C and A Groups' chunks found using Factor A|H|R3 are 'Average', and I Group's chunk is 'Poor'.

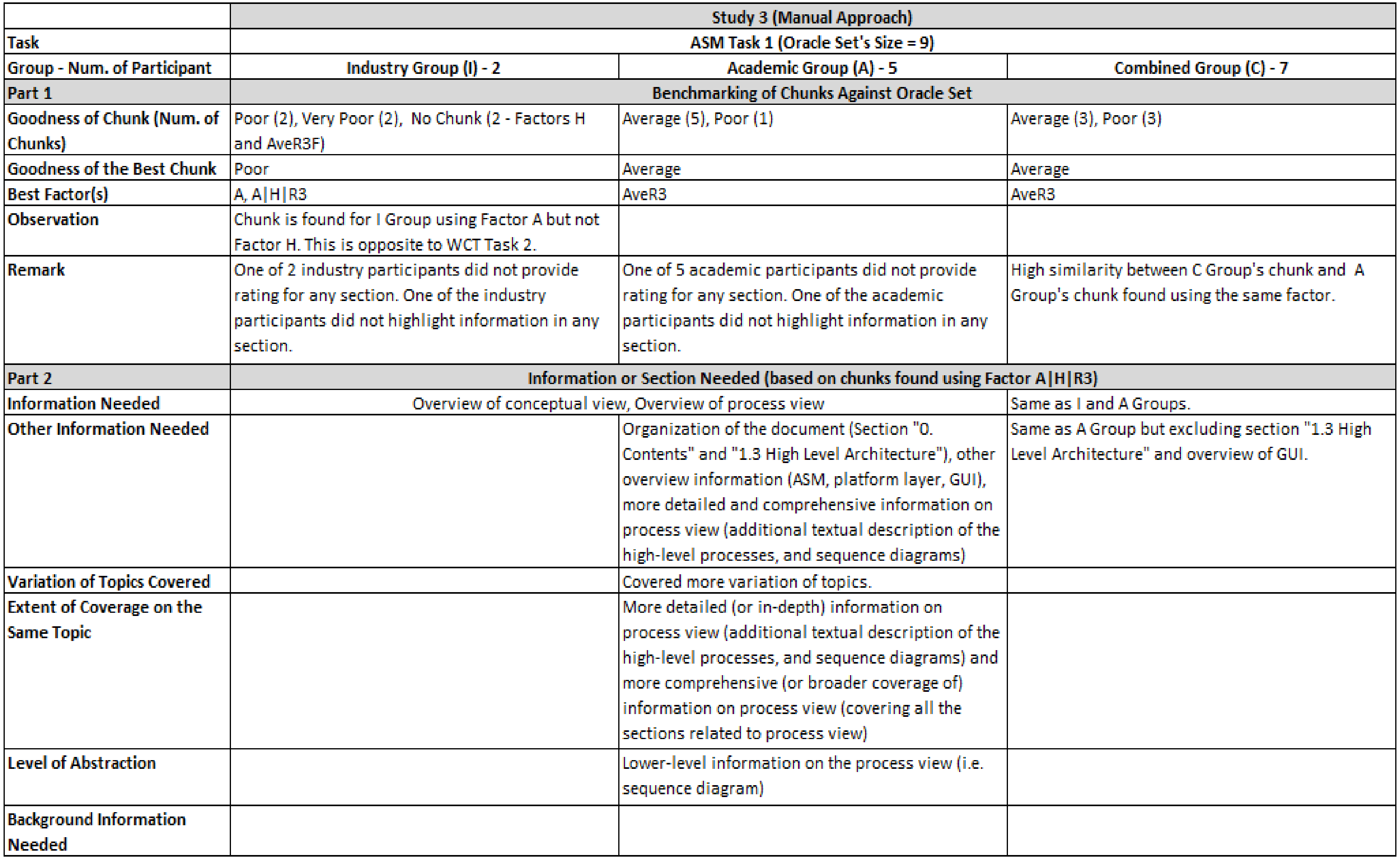
As shown in Table 9.4, both I and A Groups of participants needed the overview information on the conceptual view, and the process view (together with the corresponding diagram) for ASM Task 1. This is unsurprising for a task of getting an overview of the SA of a system. A Group covered more variation of topics. A Group also needed information on the organisation of the AD (Sections "0. Contents" and "1.3 High Level Architecture") and other overview information (ASM, platform layer and GUI). A Group also needed more detailed (or in-depth) information on process view (by requiring the additional textual description of the high-level processes, and sequence diagrams) and more comprehensive (or broader coverage of) information for process view (by requiring all the sections related to process view including the lower-level information in terms of the sequence diagrams). The comparison between I and A Groups for ASM Task 1 has to be taken with caution because of the imbalance number of participants in the groups (2 industry versus 5 academic participants).

When both I and A Groups are combined in C Group, information needed by C Group is the same as needed by A Group but excluding Section "1.3 High Level Architecture" and overview of GUI. This is unsurprising since 5 of the 7 participants in C Group are from A Group.

9.4 Summary

This chapter describes the chunking of architectural information for ASM Task 1. As with WCT Task 1 (i.e. a similar task performed on the first AD), the results show the discovery of usage-based chunks for the task of getting an overview of the SA of a system.

Chapter 13 contains the comparison of the results for ASM Task 1 and WCT Task 1, and the other tasks; the summary of the overall results for Studies 2 and 3; and threats to validity of findings. Chapter 6 presents the chunking of architectural information for WCT Task 1. The next chapter details the chunking of architectural information for ASM Task 2.



	Study 3 (Manual Approach)		
Task	ASM Task 1 (Oracle Set's Size = 9)		
Group - Num. of Participant	Industry Group (I) - 2	Academic Group (A) - 5	Combined Group (C) - 7
Part 1	Benchmarking of Chunks Against Oracle Set		
Goodness of Chunk (Num. of Chunks)	Poor (2), Very Poor (2), No Chunk (2 - Factors H and AveR3F)	Average (5), Poor (1)	Average (3), Poor (3)
Goodness of the Best Chunk	Poor	Average	Average
Best Factor(s)	A, A|H|R3	AveR3	AveR3
Observation	Chunk is found for I Group using Factor A but not Factor H. This is opposite to WCT Task 2.		
Remark	One of 2 industry participants did not provide rating for any section. One of the industry participants did not highlight information in any section.	One of 5 academic participants did not provide rating for any section. One of the academic participants did not highlight information in any section.	High similarity between C Group's chunk and A Group's chunk found using the same factor.
Part 2	Information or Section Needed (based on chunks found using Factor A|H|R3)		
Information Needed	Overview of conceptual view, Overview of process view		Same as I and A Groups.
Other Information Needed		Organization of the document (Section "0. Contents" and "1.3 High Level Architecture"), other overview information (ASM, platform layer, GUI), more detailed and comprehensive information on process view (additional textual description of the high-level processes, and sequence diagrams)	Same as A Group but excluding section "1.3 High Level Architecture" and overview of GUI.
Variation of Topics Covered		Covered more variation of topics.	
Extent of Coverage on the Same Topic		More detailed (or in-depth) information on process view (additional textual description of the high-level processes, and sequence diagrams) and more comprehensive (or broader coverage of) information on process view (covering all the sections related to process view)	
Level of Abstraction		Lower-level information on the process view (i.e. sequence diagram)	
Background Information Needed			

Table 9.4: Summary of Results for ASM Task 1

10 Chunking of Architectural Information (ASM Task 2)

This chapter presents the chunking of architectural information for the second information-seeking task for the Aperi Storage Manager (ASM Task 2) software architecture document, for Study 2. The task was:

> “As a developer you need to change the Aperi Storage Manager to dynamically unload a plug-in. You want to know what needs to be done and which parts of Aperi Storage Manager will be affected.”

This task is similar to WCT Task 2, in the sense that both tasks are about changing one part of a system and assessing the possible impact of such change.

The results presented in this chapter are based on the responses of 8 participants who completed ASM Task 2. Three were classified as industry participants and 5 were classified as academic participants. We recruited 4 industry practitioners and 4 academics for this task just as what we did for the other tasks. One industry practitioner have the same number of years of SA teaching and industry experiences but was more inclined towards the academic environment. This industry practitioner was therefore classified as academic participant.

This chapter adopts the same structure as Chapter 6. Chapter 5 provides the foundation to understand this chapter and should be read prior to reading this chapter.

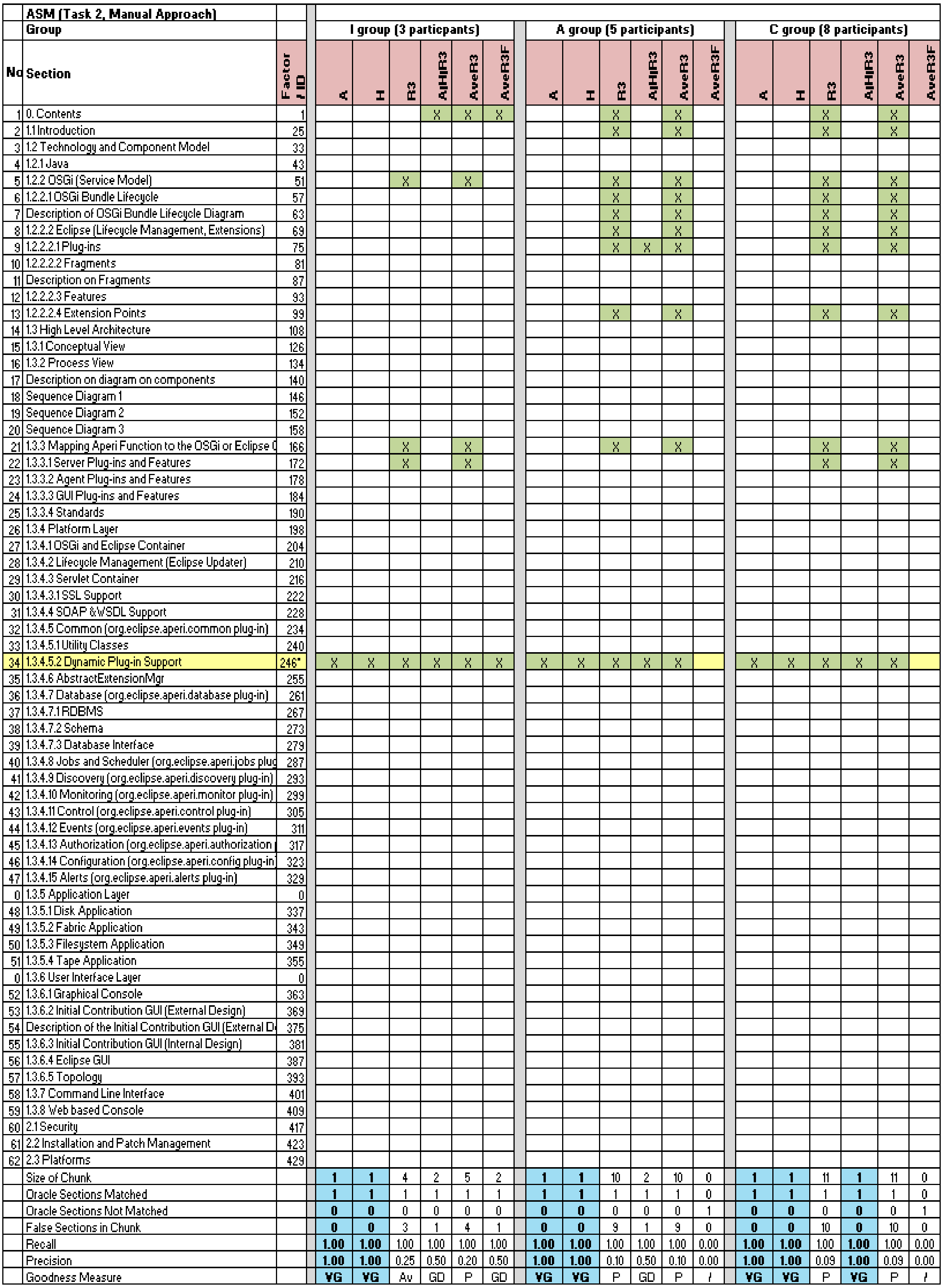
10.1 Identification and Interpretation of Chunks

We used the 6 chunk-identification factors to identify the chunks for this task. This was done for the 3 groups of participants: the industry practitioner (I) group, the academic (A) group and the combined (C) group of the previous two.

Table 10.1 shows the chunks discovered for ASM Task 2, for each group of participants. Refer to Section “6.1 Identification and Interpretation of Chunks” in Chapter 6 on how to interpret the chunks in the table. Recall that the size of a chunk is the number of the document’s sections that the chunk comprises.

Factor A and ***Factor H***: One-element chunks containing Section “1.3.4.5.2 Dynamic Plug-in Support” are found for all three groups of participants using Factors A and H.

Interpretation: Section “1.3.4.5.2 Dynamic Plug-in Support” explains the importance of supporting dynamic plug-ins, given that the nature of Aperi’s extensible architecture is based on plug-ins, and dynamic loading and unloading. It also explains the concept of dynamic plug-ins and the design paradigms ASM recommended to support them. It also gives the code needed to implement a dynamic plug-in. Factors A’s and H’s chunks show that the majority of each group needed this section when attempting ASM Task 2. Inspection of the raw data shows that 1 of the 3 industry participants did not highlight information in any section of the document, when attempting ASM Task 2. This could have affected I Group’s chunk found using Factor H.



	ASM (Task 2, Manual Approach)																			
	Group		I group (3 particpants)						A group (5 participants)						C group (8 participants)					
No	Section	Factor / ID	A	H	R3	AllR3	AveR3	AveR3F	A	H	R3	AllR3	AveR3	AveR3F	A	H	R3	AllR3	AveR3	AveR3F
1	0. Contents	1				X	X	X			X		X				X		X	
2	1.1 Introduction	25									X		X				X		X	
3	1.2 Technology and Component Model	33																		
4	1.2.1 Java	43																		
5	1.2.2 OSGi (Service Model)	51			X		X				X		X				X		X	
6	1.2.2.1 OSGi Bundle Lifecycle	57									X		X				X		X	
7	Description of OSGi Bundle Lifecycle Diagram	63									X		X				X		X	
8	1.2.2.2 Eclipse (Lifecycle Management, Extensions)	69									X		X				X		X	
9	1.2.2.2.1 Plug-ins	75									X	X	X				X		X	
10	1.2.2.2.2 Fragments	81																		
11	Description on Fragments	87																		
12	1.2.2.2.3 Features	93																		
13	1.2.2.2.4 Extension Points	99									X		X				X		X	
14	1.3 High Level Architecture	108																		
15	1.3.1 Conceptual View	126																		
16	1.3.2 Process View	134																		
17	Description on diagram on components	140																		
18	Sequence Diagram 1	146																		
19	Sequence Diagram 2	152																		
20	Sequence Diagram 3	158																		
21	1.3.3 Mapping Aperi Function to the OSGi or Eclipse (	166			X		X				X		X				X		X	
22	1.3.3.1 Server Plug-ins and Features	172			X		X										X		X	
23	1.3.3.2 Agent Plug-ins and Features	178																		
24	1.3.3.3 GUI Plug-ins and Features	184																		
25	1.3.3.4 Standards	190																		
26	1.3.4 Platform Layer	198																		
27	1.3.4.1 OSGi and Eclipse Container	204																		
28	1.3.4.2 Lifecycle Management (Eclipse Updater)	210																		
29	1.3.4.3 Servlet Container	216																		
30	1.3.4.3.1 SSL Support	222																		
31	1.3.4.4 SOAP &WSDL Support	228																		
32	1.3.4.5 Common (org.eclipse.aperi.common plug-in)	234																		
33	1.3.4.5.1 Utility Classes	240																		
34	1.3.4.5.2 Dynamic Plug-in Support	246*	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	
35	1.3.4.6 AbstractExtensionMgr	255																		
36	1.3.4.7 Database (org.eclipse.aperi.database plug-in)	261																		
37	1.3.4.7.1 RDBMS	267																		
38	1.3.4.7.2 Schema	273																		
39	1.3.4.7.3 Database Interface	279																		
40	1.3.4.8 Jobs and Scheduler (org.eclipse.aperi.jobs plug	287																		
41	1.3.4.9 Discovery (org.eclipse.aperi.discovery plug-in)	293																		
42	1.3.4.10 Monitoring (org.eclipse.aperi.monitor plug-in)	299																		
43	1.3.4.11 Control (org.eclipse.aperi.control plug-in)	305																		
44	1.3.4.12 Events (org.eclipse.aperi.events plug-in)	311																		
45	1.3.4.13 Authorization (org.eclipse.aperi.authorization (	317																		
46	1.3.4.14 Configuration (org.eclipse.aperi.config plug-in)	323																		
47	1.3.4.15 Alerts (org.eclipse.aperi.alerts plug-in)	329																		
0	1.3.5 Application Layer	0																		
48	1.3.5.1 Disk Application	337																		
49	1.3.5.2 Fabric Application	343																		
50	1.3.5.3 Filesystem Application	349																		
51	1.3.5.4 Tape Application	355																		
0	1.3.6 User Interface Layer	0																		
52	1.3.6.1 Graphical Console	363																		
53	1.3.6.2 Initial Contribution GUI (External Design)	369																		
54	Description of the Initial Contribution GUI (External D	375																		
55	1.3.6.3 Initial Contribution GUI (Internal Design)	381																		
56	1.3.6.4 Eclipse GUI	387																		
57	1.3.6.5 Topology	393																		
58	1.3.7 Command Line Interface	401																		
59	1.3.8 Web based Console	409																		
60	2.1 Security	417																		
61	2.2 Installation and Patch Management	423																		
62	2.3 Platforms	429																		
	Size of Chunk		1	1	4	2	5	2	1	1	10	2	10	0	1	1	11	1	11	0
	Oracle Sections Matched		1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0
	Oracle Sections Not Matched		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
	False Sections in Chunk		0	0	3	1	4	1	0	0	9	1	9	0	0	0	10	0	10	0
	Recall		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	0.00
	Precision		1.00	1.00	0.25	0.50	0.20	0.50	1.00	1.00	0.10	0.50	0.10	0.00	1.00	1.00	0.09	1.00	0.09	0.00
	Goodness Measure		VG	VG	Av	GD	P	GD	VG	VG	P	GD	P	/	VG	VG	P	VG	P	/

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

Table 10.1: The Chunks for ASM Task 2 Identified Using Different Factors

Factor R3: In comparison with chunks found using the rest of the factors (except for Factor AveR3), bigger chunks are found for all groups (11, 4, 10 for C, I, A Groups respectively) using Factor R3. In addition to Section "1.3.4.5.2 Dynamic Plug-in Support", C Group's chunk found using Factor R3 includes these sections: "0. Contents", "1.1 Introduction", "1.2.2 OSGi (Service Model)", "1.2.2.1 OSGi Bundle Lifecycle", "Description of OSGi Bundle Lifecycle Diagram", "1.2.2.2 Eclipse (Lifecycle Management, Extensions), "1.2.2.2.1 Plug-ins", "1.2.2.2.4 Extension Points", "1.3.3 Mapping Aperi function to the OSGi or Eclipse Component Model" and "1.3.3.1 Server Plug-ins and Features".

In addition to Section "1.3.4.5.2 Dynamic Plug-in Support", I Group's chunk found using Factor R3 contains Sections "1.2.2 OSGi (Service Model)", "1.3.3 Mapping Aperi function to the OSGi or Eclipse Component Model" and "1.3.3.1 Server Plug-ins and Features". A Group's chunk found using Factor R3 is very similar to C Group's chunk found using the same factor. The difference is A Group's chunk excludes Section "1.3.3.1 Server Plug-ins and Features".

Interpretation: It is important to note that not all the participants who completed ASM Task 2 provided ratings for sections of ASM SA document in terms of their importance to answering the task. One of the 3 industry participants, and 2 of the 5 academic participants did not provide ratings for any section of the document, when attempting ASM Task 2. The 'not rating at all' by some participants could have affected the chunks found using Factor AveR3F but not the chunks found using Factors R3 and AveR3. This is because Factor AveR3F is based on the majority of a group, while Factors R3 and AveR3 are based on the majority of those who rated the respective sections.

C Group's chunk found using Factor R3 shows that sections related to the following were needed by the majority of C Group who rated the respective sections: dynamic plug-in support, Table of Contents, overview information (on ASM, OSGi service model and Eclipse plug-in model), some detailed information on OSGi service model (provided by both the sections on the OSGi bundle lifecycle) and some detailed information on Eclipse plug-in model (i.e. plug-ins and extension points), representation of major components of Aperi (Section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model"), and Java packages related to server plug-ins and features. In short, C Group's chunk shows that some overview information (ASM and the underlying models used) and some detailed information (the underlying models used and Java packages) were needed by the majority of C Group who rated the respective sections.

The information on the underlying models provides some of the background information needed for ASM Task 2. C Group's chunk shows that both the overview of and some detailed information on this background information, were needed by the majority of C Group who rated the respective sections.

A Group's chunk found using Factor R3 is very similar to C Group's chunk found using the same factor. Consequently, sections (excluding Section "1.3.3.1 Server Plug-ins and Features") or information needed by the majority of C Group who rated the respective sections, were also needed by the majority of the academic participants who rated the sections. Probable reason is 5 of 8 participants in Group C are from Group A.I Group's chunk found using Factor R3 shows that information on dynamic plug-ins support, overview of OSGi service model, representation of major components of Aperi, Java packages related to server plug-ins and features, were needed by the majority of the industry participants who rated the respective sections. A Group's chunk is more than twice as large as I Group's chunk found using Factor R3. Most of the additional sections in Group A's chunk are related to detailed information of OSGi service model and Eclipse plug-in model.

Factor A|H|R3: C Group's chunk found using Factor A|H|R3 is the same one-element chunk found using Factors A and H for C Group. In addition to Section "1.3.4.5.2 Dynamic Plug-in Support", I Group's chunk found using Factor A|H|R3 contains Section "0. Contents" and A Group's chunk contains Section "1.2.2.2.1 Plug-ins".

Interpretation: C Group's chunk found using Factor A|H|R3 shows that **the majority of the participants needed information on dynamic plug-in support, when they attempted ASM Task 2. This is also revealed by C Group's chunks found using Factors A and H. I and A Groups' chunks found using Factor A|H|R3 show that when attempting ASM Task 2, in addition to information on the dynamic plug-in support, the majority of the industry participants also needed Table of Contents, and the majority of the academic participants also needed information on plug-ins which is not specifically on dynamic plug-ins.**

Factor AveR3: The chunk found using Factor AveR3 for a group of participant is very similar if not identical to the chunk found using Factor R3 for the same group. C Group's chunks found using Factors AveR3 and R3 are identical. Same thing happened for A Group. A Group's chunks found using Factors AveR3 and R3 are very similar to C Group's chunk found using Factor AveR3 (with the exclusion of Section "1.3.3.1 Server Plug-ins and Features"). I Group's chunk found using Factor AveR3 is almost identical to the group's chunk found using Factor R3 with the addition of Section "0. Contents".

Interpretation: C Group's chunk found using Factor AveR3 shows that sections related to dynamic plug-in support, Table of Contents, overview information (on ASM, OSGi service model and Eclipse plug-in model), some detailed information on OSGi service model (i.e. OSGi bundle lifecycle) and some detailed information on Eclipse plug-in model (i.e. plug-ins and extension points), representation of major components of Aperi (Section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model") and Java packages related to server plug-ins and features, were needed by some in C Group who rated them. The similarity between C and A Groups' chunks found using Factor AveR3, shows that all the sections described in this paragraph excluding Section "1.3.3.1 Server Plug-ins and Features" were also needed by some of the academic participants who rated the sections.

I Group's chunk found using Factor AveR3 shows that information on dynamic plug-ins support, overview of OSGi service model, representation of major components of Aperi, Java packages related to server plug-ins and features, and Table of Contents, were needed by some of the industry participants who rated the respective sections.

Factor AveR3F: No chunk is discovered for C and A Groups using Factor AveR3F. I Group's chunk found using Factor AveR3F is identical to the group's chunk found using Factor A|H|R3.

Interpretation: Recall that 2 of the 5 academic participants did not provide rating for any section of the document when attempting ASM Task 2. As a result, it is more difficult for A Group to meet the section inclusion criteria of Factor AveR3F (i.e. a section is rated by more than half of the participants in the group and its average rating is at least 3). This could be why no chunk is found for A Group. Recall also that 1 of the 3 industry participants did not provide rating for any section of the document. In total, 3 of the 8 participants who completed ASM Task 2 did not provide ratings for any section. This lack of rating activities among the participants could be the reason for no chunk being found using Factor AveR3F for C Group.

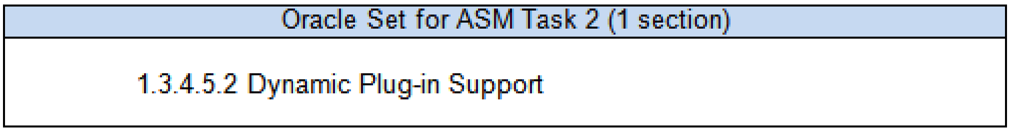
I Group's chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the industry participants, sections on Table of Contents and dynamic plug-in support were needed for ASM Task 2.

10.2 **Benchmarking Chunks Against An Oracle Set**

This section describes the oracle set for ASM Task 2 and the results of benchmarking the discovered chunks against this oracle set.

10.2.1 The Oracle Set for ASM Task 2

The oracle set for ASM Task 2 (Table 10.2) comprises 1 of the 62 sections of the ASM software architecture document. This section was agreed by both the researcher and a separate judge as mandatory for ASM Task 2. In Table 10.1, this section is denoted by a yellow row and its section ID is marked with an asterisk. This is to make it easier to compare the chunks found against the oracle set. The relevance of this section to ASM Task 2 is explained next.



Oracle Set for ASM Task 2 (1 section)
1.3.4.5.2 Dynamic Plug-in Support

Table 10.2: The Oracle Set for ASM Task 2

It is important to bear in mind that, similar to WCT Task 2, ASM Task 2 is a very specific task comprising two parts, both from the perspective of a developer: 1) what needs to be done to change ASM to dynamically unload a plug-in, and 2) identifying the parts of ASM that will be affected by the change. The difference is that WCT Task 2 states that the digital archive system is to be changed, whereas ASM Task 2 states that a plug-in is to be dynamically unloaded but does not state which plug-in. Without stating the specific plug-in to be dynamically unloaded, the requirement of ASM Task 2 is more inclined towards some general information related to all plug-ins.

Section "1.3.4.5.2 Dynamic Plug-in Support" is essential for both Parts 1 and 2 of ASM Task 2. It describes the importance of supporting dynamic plug-ins, given the nature of Aperi's extensible architecture which is based on plug-ins, and dynamic loading and unloading. It also explains the concept of dynamic plug-ins and the design paradigms recommended by Aperi to support them (i.e. use stateless object instead of instance variable, enforce lifecycle management whenever object caches are used, object caches should implement IDisposable interface and bundle activators should contain the provided logic). It also gives the code needed to implement dynamic plug-in.

Since dynamically unloading a plug-in is inherently supported by ASM through the recommended design paradigms and based on what is in the document, the only thing that we think could be affected when a plug-in is dynamically unloaded is other plug-ins that are dependent on its services. ASM Task 2 does not specify which plug-in to unload dynamically and therefore does not require information on any dependent plug-in that might be affected by dynamically unloading a plug-in.

10.2.2 Benchmarking Results

Refer to Section “5.7 Benchmarking of Chunks against An Oracle Set” of Chapter 5 for all the concepts and terms needed to understand this section.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for ASM Task 2 are given in Table 10.3.

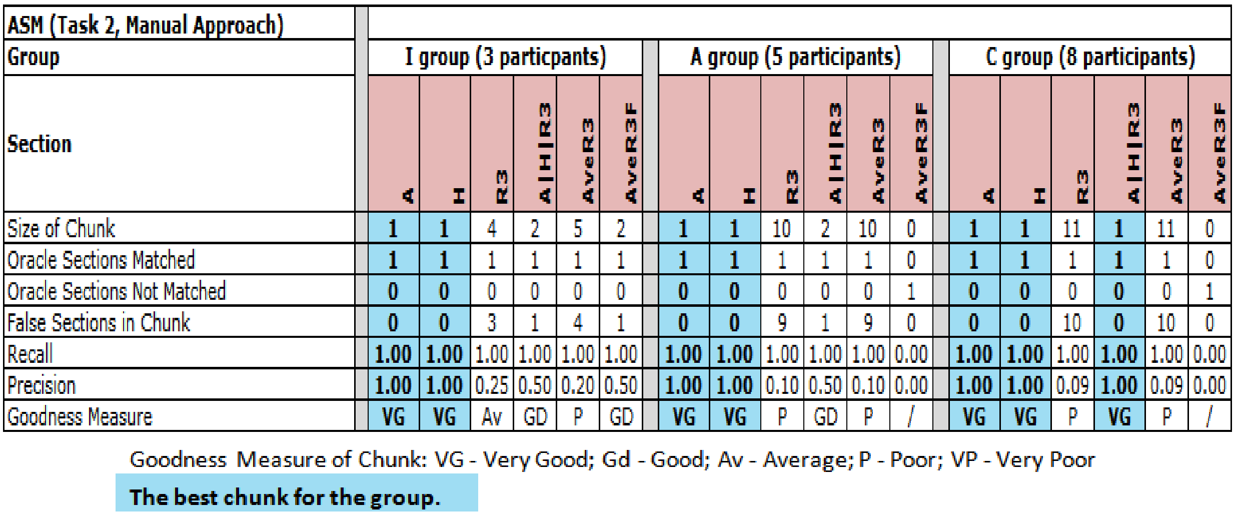
Two types of chunks are found for C Group:

1) The ‘Very Good’ chunk found using Factors A, H and A|H|R3 (recall = 1.00, precision = 1.00, size = 1, goodness = ‘Very Good’). With the maximum values of both recall and precision, this ‘Very Good’ chunk contains the oracle set’s section and only that section, and is the perfect chunk for ASM Task 2. Consequently, this chunk is the best chunk for C Group. This one-element chunk is also the most basic chunk for C Group.

2) The ‘Poor’ chunk found using Factors R3 and AveR3 (recall = 1.00; precision = 0.09, size = 11, goodness = ‘Poor’). This larger chunk has 100% recall but has 10 false sections, namely those sections related to Table of Contents, overview information (on ASM, OSGi service model and Eclipse plug-in model), some detailed information on OSGi service model (OSGi bundle lifecycle) and some detailed information on Eclipse plug-in model (plug-ins and extension points), representation of major components of Aperi, and Java packages related to server plug-ins and features.

No chunk is found for C Group using Factor AveR3F. The goodness measures for I Group’s chunks are: ‘Very Good’ for Factors A’s and H’s chunks, ‘Good’ for Factors A|H|R3’s and AveR3F’s chunks, ‘Average’ for Factor R3’s chunk and ‘Poor’ for Factor AveR3’s chunk. The two ‘Very Good’ chunks of I Group are identical, and the same as the perfect chunk found for C Group. The ‘Very Good’ chunk is therefore the best chunk and the most basic chunk for I Group.

The two ‘Good’ chunks of I Group (found using Factors A|H|R3 and AveR3F) are identical. It has 100% recall with Section “0.Contents” as false section (recall = 1.00; precision = 0.50, size = 2, goodness = ‘Good’). I Group’s chunks found using Factor R3 (recall = 1.00; precision = 0.25, size = 4, goodness = ‘Average’) and Factor AveR3 (recall = 1.00; precision = 0.20, size = 5, goodness = ‘Poor’) also have 100% recall but have low precision. Factor R3’s chunk contains 3 false sections on the overview of OSGi service model, representation of major components of Aperi, and Java packages related to server plug-ins and features. Factor AveR3’s chunk contains the same 3 false sections and also Section “0. Contents”.



ASM (Task 2, Manual Approach)																		
Group	I group (3 particpants)						A group (5 participants)						C group (8 participants)					
Section	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
Size of Chunk	**1**	**1**	4	2	5	2	**1**	**1**	10	2	10	0	**1**	**1**	11	**1**	11	0
Oracle Sections Matched	**1**	**1**	1	1	1	1	**1**	**1**	1	1	1	0	**1**	**1**	1	**1**	1	0
Oracle Sections Not Matched	**0**	**0**	0	0	0	0	**0**	**0**	0	0	0	1	**0**	**0**	0	**0**	0	1
False Sections in Chunk	**0**	**0**	3	1	4	1	**0**	**0**	9	1	9	0	**0**	**0**	10	**0**	10	0
Recall	**1.00**	**1.00**	1.00	1.00	1.00	1.00	**1.00**	**1.00**	1.00	1.00	1.00	0.00	**1.00**	**1.00**	1.00	**1.00**	1.00	0.00
Precision	**1.00**	**1.00**	0.25	0.50	0.20	0.50	**1.00**	**1.00**	0.10	0.50	0.10	0.00	**1.00**	**1.00**	0.09	**1.00**	0.09	0.00
Goodness Measure	**VG**	**VG**	Av	GD	P	GD	**VG**	**VG**	P	GD	P	/	**VG**	**VG**	P	**VG**	P	/

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 10.3: Attributes of Chunks (ASM Task 2)

The goodness measures for A Group’s chunks are: ‘Very Good’ for Factors A’s and H’s chunks, ‘Good’ for Factor A|H|R3’s chunk, and ‘Poor’ for Factors R3’s and AveR3’s chunks. No chunk is found for A Group using Factor AveR3F.

The two ‘Very Good’ chunks of A Group are identical, and the same as the perfect chunk found for C and I Groups. This ‘Very Good’ chunk is therefore the best chunk and the most basic chunk for A Group. Factor A|H|R3’s chunk for A Group (recall = 1.00, precision = 0.50, size = 2, goodness = ‘Good’) has 100% recall but 50% precision with Section “1.2.2.2.1 Plug-ins” as false section. The two ‘Poor’ chunk of A Group are identical (recall = 1.00; precision = 0.10, size = 10, goodness = ‘Poor’). It has 100% recall but low precision with 9 false sections. The ‘Poor’ chunk of A Group resembles the ‘Poor’ chunk of C Group, but with the exclusion of Section “1.3.3.1 Server Plug-ins and Features”.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for I and A Groups is Factor A or Factor H. For C Group the best factor is Factor A or H or A|H|R3.

Commonly-Missed Oracle Set’s Sections and Common False Sections: Refer to Section “5.7.6 Determination of the Commonly-Missed Oracle Set’s Sections and Common False Sections” of Chapter 5 on how common oracle set’s sections missed by a group of participants, and common false sections for a group of participants are determined.

For ASM Task 2, there is no oracle set’s section commonly missed by a group of participants, and there is no common false section found for any group of participants.

10.3 **Discussion of Chunk Discovery Results**

The results show the discovery of chunks for ASM Task 2 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. The larger chunks are usually supersets of the smaller chunks.

he most basic chunks (comprising 1 section) for the 3 groups are identical and produced by similar factors: Factors A and H for I and A Groups, Factors A, H and A|H|R3 for C Group. Chunks found using Factor A are identical to those found using Factor H. The chunks found using Factors R3 and AveR3 for a group of participants, are very similar if not identical to each other (i.e. very similar for I Group, and identical for A and C Groups).

C Group's chunk is very similar if not identical to A Group's chunk found using the same factor. This is probably because 5 of the 8 participants in C Group are academic participants from A Group.

The discovery of mostly small chunks is in line with the small oracle set for ASM Task 2. All the chunks include the one and only one oracle set's section ("1.3.4.5.2 Dynamic Plug-in Support"). Although the word 'dynamic' in its title could have contributed to the finding of Section "1.3.4.5.2 Dynamic Plug-in Support" by the participants, the fact that it appeared in all the chunks shows the emphasis of the participants on this section for ASM Task 2.

Part 1 of Table 10.4 summarises the main results of benchmarking of chunks. Recall that 1 of the 3 industry participants and 2 of the 5 academic participants did not provide ratings for any section of the document, and 1 of the 3 industry participants did not highlight information in any section of the document, when attempting the task.

C Group has 3 chunks with 'Very Good' and 2 chunks with 'Poor' goodness measures. I Group has 2 chunks with 'Very Good', 2 with 'Good', 1 with 'Average' and 1 with 'Poor' goodness measures. A Group has 2 chunks with 'Very Good', 1 with 'Good' and 2 chunks with 'Poor' goodness measures. No chunk is found for C and A Groups using Factor AveR3F.

For ASM Task 2, we did not assess the composition of the chunks that are at the best level of goodness measure in a group, to decide the best chunk for the group. This is because for each group, these chunks are identical to each other. The most basic chunk mentioned in the first paragraph of this section ("11.3 Discussion of Chunk Discovery Results") is also the best chunk for all the 3 groups of participants. With recall and precision of 1.00, the best chunk is the perfect chunk for ASM Task 2. **ASM Task 2 is the only task where a perfect chunk is found in Study 2 and it happened when the oracle set is very small.**

The factor that produces the best chunk for a group is the best chunk-identification factor for the group. Recall that the best (also the most basic and the perfect) chunk is found using Factors A and H for I and A Groups, and Factors A, H and A|H|R3 for C Group.

Factors that are based on those who rated only and not the majority of a group (i.e. Factors R3 and AveR3) have the tendency to produce large chunks. Benchmarking these chunks against a small oracle set as in the case of ASM Task 2 (with size of oracle set of 1) has the tendency to render many of the sections included in Factors R3's and AveR3's chunks as false sections. In other words, chunks

produced by Factor R3 and AveR3 fare much worse in terms of precision, when the respective oracle set is of small size.

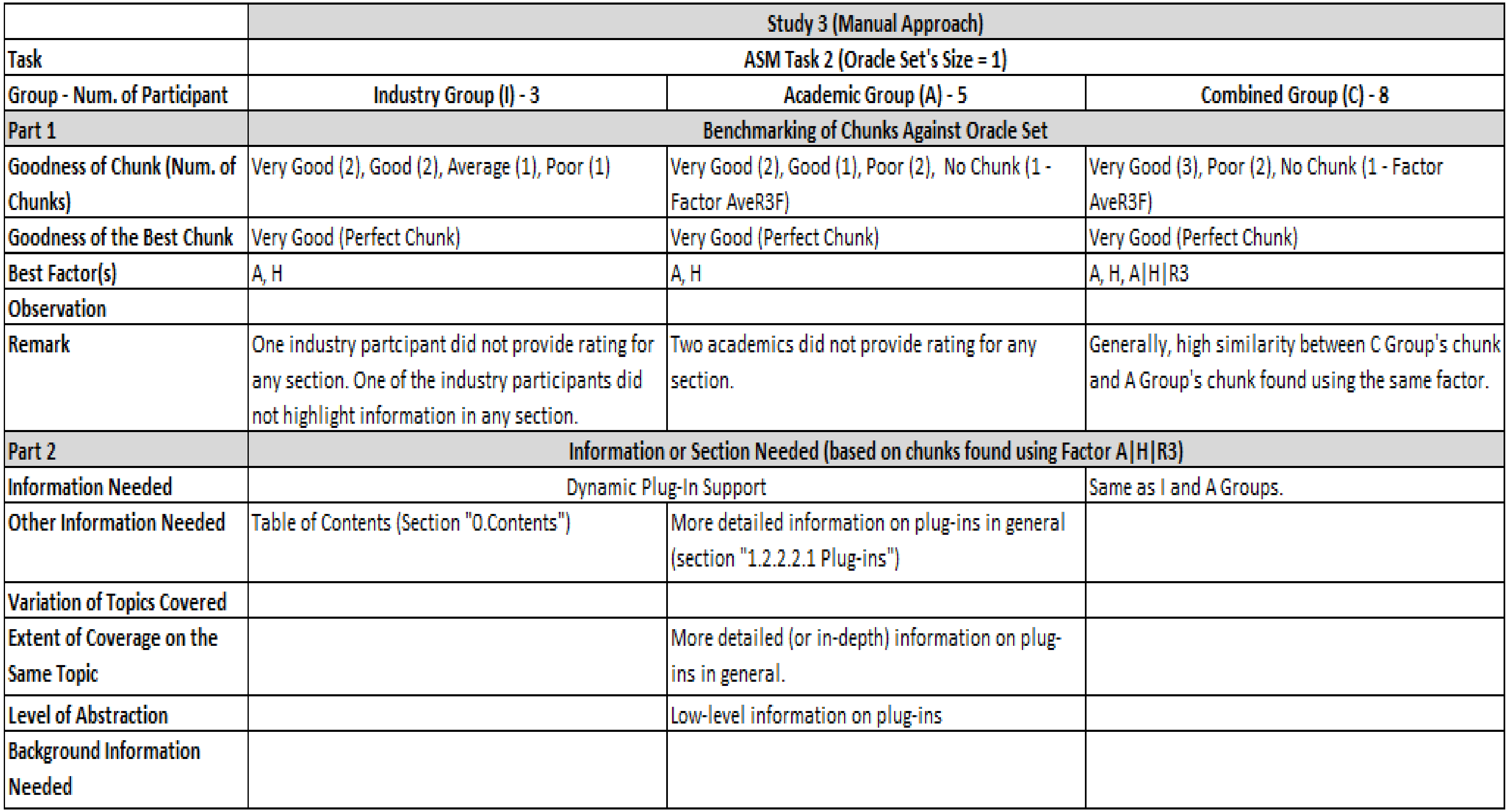
Part 2 of Table 10.4 summarises the information or sections needed by each group of participants. This is based on the groups' chunks found using Factor A|H|R3. Refer to Section "5.8 Information Needed" in Chapter 5 for our reason. C Group's chunk found using Factor A|H|R3 is 'Very Good', and I and A Groups' chunks are 'Good'.

Both I and A Groups needed the section on dynamic plug-in support which is critical for ASM Task 2. I Group also needed Table of Contents whereas A Group also needed more detailed information on plug-ins (Section "1.2.2.2.1 Plug-ins") which is not specifically on dynamic plug-ins. Section "1.2.2.2.1 Plug-ins" also includes low-level information on plug-ins. The comparison between I and A Groups involved in ASM Task 2 has to be treated cautiously because of the unbalanced number of participants in the groups (i.e. 3 industry versus 5 academic participants). When all the participants are combined in C Group, only section on dynamic plug-in support stands out as needed by the majority of the participants for ASM Task 2.

10.4 **Summary**

This chapter describes the chunking of architectural information for ASM Task 2. As with WCT Task 2 (i.e. a similar task performed on the first AD) the results show the discovery of usage-based chunks for the task of changing one part of a system and assessing the possible impact of change.

Chapter 13 contains the comparison of the results for ASM Task 2 and WCT Task 2, and the other tasks; the summary of the overall results for Studies 2 and 3; and threats to validity of findings. Chapter 7 presents the chunking of architectural information for WCT Task 2. The next chapter (Chapter 11) details the chunking of architectural information for ASM Task 3.



	Study 3 (Manual Approach)		
Task	ASM Task 2 (Oracle Set's Size = 1)		
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 5	Combined Group (C) - 8
Part 1	Benchmarking of Chunks Against Oracle Set		
Goodness of Chunk (Num. of Chunks)	Very Good (2), Good (2), Average (1), Poor (1)	Very Good (2), Good (1), Poor (2), No Chunk (1 - Factor AveR3F)	Very Good (3), Poor (2), No Chunk (1 - Factor AveR3F)
Goodness of the Best Chunk	Very Good (Perfect Chunk)	Very Good (Perfect Chunk)	Very Good (Perfect Chunk)
Best Factor(s)	A, H	A, H	A, H, A|H|R3
Observation			
Remark	One industry partcipant did not provide rating for any section. One of the industry participants did not highlight information in any section.	Two academics did not provide rating for any section.	Generally, high similarity between C Group's chunk and A Group's chunk found using the same factor.
Part 2	Information or Section Needed (based on chunks found using Factor A|H|R3)		
Information Needed	Dynamic Plug-In Support		Same as I and A Groups.
Other Information Needed	Table of Contents (Section "0.Contents")	More detailed information on plug-ins in general (section "1.2.2.2.1 Plug-ins")	
Variation of Topics Covered			
Extent of Coverage on the Same Topic		More detailed (or in-depth) information on plug-ins in general.	
Level of Abstraction		Low-level information on plug-ins	
Background Information Needed			

Table 10.4: Summary of Results for ASM Task 2

11 Chunking of Architectural Information (ASM Task 3)

This chapter presents the chunking of architectural information for the third information-seeking task for the Aperi Storage Manager (ASM Task 3) software architecture document, for Study 2. The task was:

> "As a maintainer of Aperi Storage Manager, you would like to know how it was designed at the architectural level to achieve modifiability."

This task is similar to WCT Task 3 in the sense that it is addressing a cross-cutting concern from the perspective of a maintainer. However, it focuses on modifiability instead of security issue. We were not able to specify the same issue for the two tasks due to the difference in the content of the two different ADs.

The results presented in this chapter are based on the responses of 7 participants who completed ASM Task 3. Three were classified as industry participants and 4 were classified as academic participants.

We recruited 4 industry practitioners and 4 academics for this task just as what we did for the other tasks. One of the industry participants had worked as a software engineer for 2 years but further inspection of the participant's response indicated that he or she did not fulfil our selection criteria of at least 2 years industry experience related to SA. Consequently, this participant's data was excluded from our analysis.

This chapter adopts the same structure as Chapter 6. Chapter 5 provides the foundation to understand this chapter and should be read prior to reading this chapter.

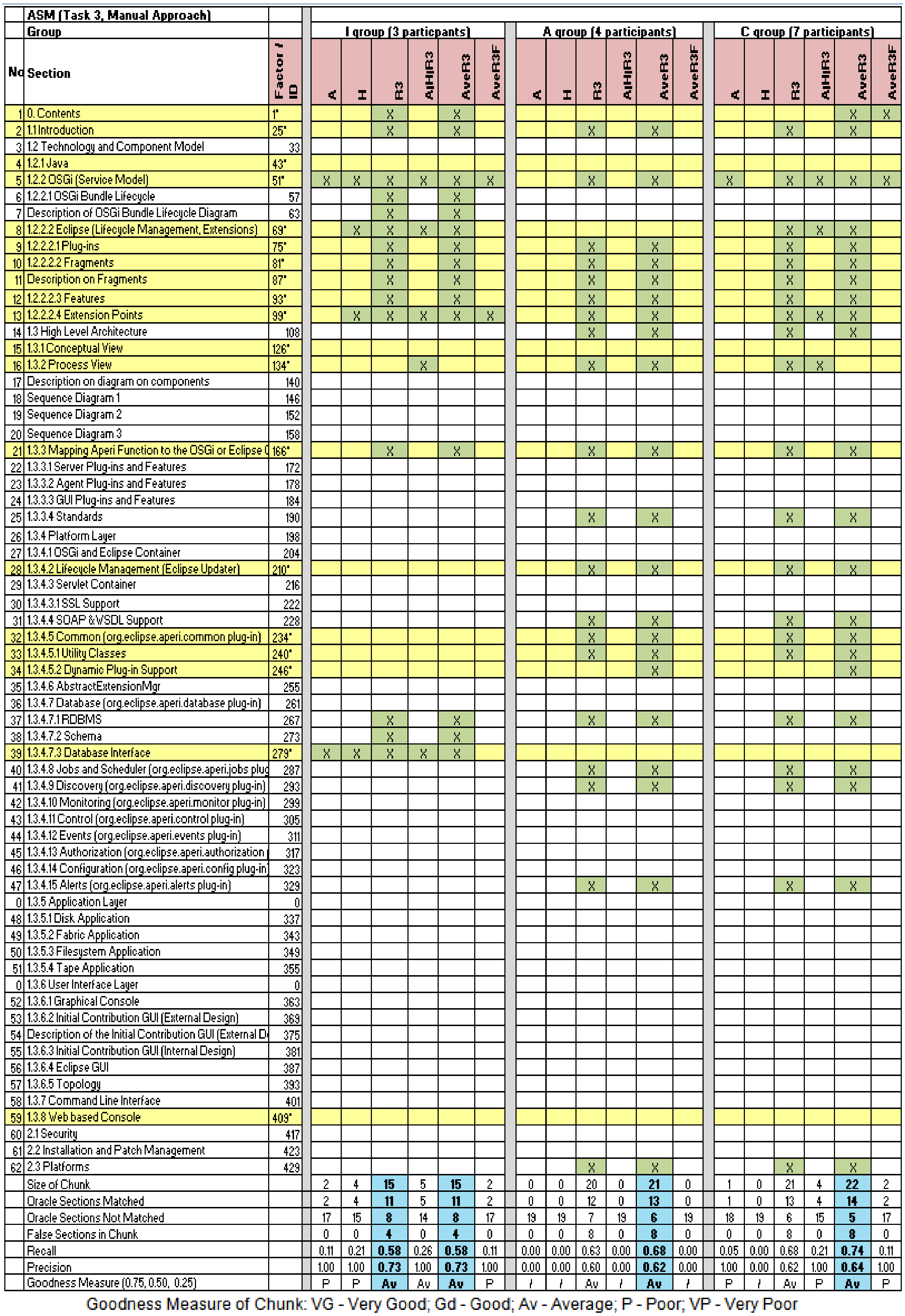
11.1 Identification and Interpretation of Chunks

We used the 6 chunk-identification factors to identify the chunks for this task. This was done for the 3 groups of participants: the industry practitioner (I) group, the academic (A) group and the combined (C) group of the previous two.

Table 11.1 shows the chunks discovered for ASM Task 3, for each group of participants. Refer to Section "6.1 Identification and Interpretation of Chunks" in Chapter 6 on how to interpret the chunks in the table. Recall that the size of a chunk is the number of the document's sections that the chunk comprises.

Factor A: C Group's chunk found using Factor A contains Section "1.2.2 OSGi (Service Model)" which gives a brief overview of OSGi Service model. I Group's chunk also contains this section and Section "1.3.4.7.3 Database Interface" (describing database interface component of the data and device servers). No chunk is found for A Group using Factor A.

Interpretation: C Group's chunk found using Factor A shows that the majority of the participants needed the overview of OSGi Service Model, when they attempted ASM Task 3. The majority of the industry participants also needed this in addition to the information related to database interface component.



	ASM (Task 3, Manual Approach)																			
	Group		I group (3 particpants)						A group (4 participants)						C group (7 participants)					
No	Section	Factor / ID	A	H	R3	AIHR3	AveR3	AveR3F	A	H	R3	AIHR3	AveR3	AveR3F	A	H	R3	AIHR3	AveR3	AveR3F
1	0. Contents	1*			X		X												X	X
2	1.1 Introduction	25*			X		X				X		X				X		X	
3	1.2 Technology and Component Model	33																		
4	1.2.1 Java	43*																		
5	1.2.2 OSGi (Service Model)	51*	X	X	X	X	X	X			X		X		X		X	X	X	X
6	1.2.2.1 OSGi Bundle Lifecycle	57			X		X													
7	Description of OSGi Bundle Lifecycle Diagram	63			X		X													
8	1.2.2.2 Eclipse (Lifecycle Management, Extensions)	69*		X	X	X	X										X	X	X	
9	1.2.2.2.1 Plug-ins	75*			X		X				X		X				X		X	
10	1.2.2.2.2 Fragments	81*			X		X				X		X				X		X	
11	Description on Fragments	87*			X		X				X		X				X		X	
12	1.2.2.2.3 Features	93*			X		X				X		X				X		X	
13	1.2.2.2.4 Extension Points	99*		X	X	X	X	X			X		X				X	X	X	
14	1.3 High Level Architecture	108									X		X				X		X	
15	1.3.1 Conceptual View	126*																		
16	1.3.2 Process View	134*				X					X		X				X	X		
17	Description on diagram on components	140																		
18	Sequence Diagram 1	146																		
19	Sequence Diagram 2	152																		
20	Sequence Diagram 3	158																		
21	1.3.3 Mapping Aperi Function to the OSGi or Eclipse C	166*			X		X				X		X				X		X	
22	1.3.3.1 Server Plug-ins and Features	172																		
23	1.3.3.2 Agent Plug-ins and Features	178																		
24	1.3.3.3 GUI Plug-ins and Features	184																		
25	1.3.3.4 Standards	190									X		X				X		X	
26	1.3.4 Platform Layer	198																		
27	1.3.4.1 OSGi and Eclipse Container	204																		
28	1.3.4.2 Lifecycle Management (Eclipse Updater)	210*									X		X				X		X	
29	1.3.4.3 Servlet Container	216																		
30	1.3.4.3.1 SSL Support	222																		
31	1.3.4.4 SOAP & WSDL Support	228									X		X				X		X	
32	1.3.4.5 Common (org.eclipse.aperi.common plug-in)	234*									X		X				X		X	
33	1.3.4.5.1 Utility Classes	240*									X		X				X		X	
34	1.3.4.5.2 Dynamic Plug-in Support	246*											X						X	
35	1.3.4.6 AbstractExtensionMgr	255																		
36	1.3.4.7 Database (org.eclipse.aperi.database plug-in)	261																		
37	1.3.4.7.1 RDBMS	267			X		X				X		X				X		X	
38	1.3.4.7.2 Schema	273			X		X													
39	1.3.4.7.3 Database Interface	279*	X	X	X	X	X													
40	1.3.4.8 Jobs and Scheduler (org.eclipse.aperi.jobs plug	287									X		X				X		X	
41	1.3.4.9 Discovery (org.eclipse.aperi.discovery plug-in)	293									X		X				X		X	
42	1.3.4.10 Monitoring (org.eclipse.aperi.monitor plug-in)	299																		
43	1.3.4.11 Control (org.eclipse.aperi.control plug-in)	305																		
44	1.3.4.12 Events (org.eclipse.aperi.events plug-in)	311																		
45	1.3.4.13 Authorization (org.eclipse.aperi.authorization	317																		
46	1.3.4.14 Configuration (org.eclipse.aperi.config plug-in)	323																		
47	1.3.4.15 Alerts (org.eclipse.aperi.alerts plug-in)	329									X		X				X		X	
0	1.3.5 Application Layer	0																		
48	1.3.5.1 Disk Application	337																		
49	1.3.5.2 Fabric Application	343																		
50	1.3.5.3 Filesystem Application	349																		
51	1.3.5.4 Tape Application	355																		
0	1.3.6 User Interface Layer	0																		
52	1.3.6.1 Graphical Console	363																		
53	1.3.6.2 Initial Contribution GUI (External Design)	369																		
54	Description of the Initial Contribution GUI (External De	375																		
55	1.3.6.3 Initial Contribution GUI (Internal Design)	381																		
56	1.3.6.4 Eclipse GUI	387																		
57	1.3.6.5 Topology	393																		
58	1.3.7 Command Line Interface	401																		
59	1.3.8 Web based Console	409*																		
60	2.1 Security	417																		
61	2.2 Installation and Patch Management	423																		
62	2.3 Platforms	429									X		X				X		X	
	Size of Chunk		2	4	**15**	5	**15**	2	0	0	20	0	**21**	0	1	0	21	4	**22**	2
	Oracle Sections Matched		2	4	**11**	5	**11**	2	0	0	12	0	**13**	0	1	0	13	4	**14**	2
	Oracle Sections Not Matched		17	15	**8**	14	**8**	17	19	19	7	19	**6**	19	18	19	6	15	**5**	17
	False Sections in Chunk		0	0	**4**	0	**4**	0	0	0	8	0	**8**	0	0	0	8	0	**8**	0
	Recall		0.11	0.21	**0.58**	0.26	**0.58**	0.11	0.00	0.00	0.63	0.00	**0.68**	0.00	0.05	0.00	0.68	0.21	**0.74**	0.11
	Precision		1.00	1.00	**0.73**	1.00	**0.73**	1.00	0.00	0.00	0.60	0.00	**0.62**	0.00	1.00	0.00	0.62	1.00	**0.64**	1.00
	Goodness Measure (0.75, 0.50, 0.25)		P	P	**Av**	Av	**Av**	P	/	/	Av	/	**Av**	/	P	/	Av	P	**Av**	P

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

Table 11.1: The Chunks for ASM Task 3 Identified Using Different Factors

Factor H: No chunk is discovered for C and A Groups using Factor H. I Group's chunk also includes Sections "1.2.2 OSGi (Service Model)" and "1.3.4.7.3 Database Interface" (as in I Group's chunk found using Factor A). In addition I Group's chunk contains Sections "1.2.2.2 Eclipse (Lifecycle Management, Extensions)" and "1.2.2.2.4 Extension Points".

Interpretation: Two of the 4 academic participants did not highlight any part of the document, when attempting ASM Task 3. This explains the no chunk for A Group. I Group's chunk found using Factor H shows that in addition to brief overview of OSGi Service Model and information on database interface component, the majority of the industry participants also needed overview of Eclipse plug-in model and information related to extension points (through which processing elements or elements can be added to the host plug-in), for ASM Task 3.

Factor R3: Factor R3 produces large chunks for all the groups (21, 15, 20 for C, I, A Groups respectively). C Group's chunk is made up of sections related to overview of ASM and the purpose of the document (Section "1.1 Introduction"), brief overview of OSGi Service Model, overview of Eclipse plug-in model and more detailed information related to it (plug-ins, fragments, features and extension points), overview of process view and the corresponding diagram (Section "1.3.2 Process View"), representation of major components of Aperi (section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model"), standards adopted, Eclipse updater for downloading updates (section 1.3.4.2), SOAP & WSDL support (section 1.3.4.4), common plug-in which is shared across implementation plug-ins (section 1.3.4.5), utility classes containing helper logic (section 1.3.4.5.1), Relational Database Management System (RDBMS) supported (section 1.3.4.7.1), certain generic services namely jobs and scheduler, discovery, alert (section 1.3.4.8, 1.3.4.9, 1.3.4.15), initial development platform (section 2.3) and Section "1.3 High Level Architecture" (containing the titles of its sub-sections).

I Group's chunk found using Factor R3 comprises sections related to the organisation of the document (Section "0. Contents"), overview of ASM and the purpose of the document (Section "1.1 Introduction"), brief overview of OSGi Service Model (section 1.2.2), lifecycle of OSGi bundle (both sub-sections of section 1.2.2), overview of Eclipse plug-in model and more detailed information related to it (plug-ins, fragments, features and extension points), representation of major components of Aperi (section 1.3.3), RDBMS supported (section 1.3.4.7.1), database schema (section 1.3.4.7.2) and database interface component.

A Group's chunk is very similar to C Group's but excludes Section "1.2.2.2 Eclipse (Lifecycle Management, Extensions) which is on the overview of Eclipse plug-in model.

Interpretation: C Group's chunk found using Factor R3 shows that for ASM Task 3 the sections needed by the majority of C Group who rated the respective sections, are related to overview information (on ASM, OSGi Service Model, Eclipse plug-in model, process view), detailed information related to Eclipse plug-in model, detailed information related to platform layer (i.e. some sub-sections under section 1.3.4), organisation of the document (Section "1.3 High Level Architecture"), representation of major components of Aperi, standards adopted and initial development platform. A Group's chunk found using Factor R3 shows that all the aforementioned sections were also needed by the majority of A Group who rated the respective sections except for the overview of Eclipse plug-in model.

The overlapping sections between I and A Groups' chunks found using Factor R3 show that the majority of those who rated the respective sections in both groups needed the following for ASM Task 3: overview information (on ASM and OSGi Service Model), detailed information on Eclipse plug-in model (section 1.2.2.2.1 to 1.2.2.2.4), representation of major components of Aperi and RDBMS. The obvious difference between the two chunks is A Group's chunk includes section on process view, standards adopted and many more sub-sections under Section "1.3.4 Platform Layer" which contain some detailed information related to platform layer. A Group's chunk found using Factor R3 shows that **the majority of the academic participants who rated the respective sections, also needed overview of process view, standards adopted and detailed information on platform layer when attempting ASM Task 3.**

Factor A|H|R3: C Group's chunk found using Factor A|H|R3 includes Section "1.2.2 OSGi (Service Model)", "1.2.2.2 Eclipse (Lifecycle Management, Extensions)", "1.2.2.2.4 Extension Points" and "1.3.2 Process View". I Group's chunk is very similar to C Group's chunk with the addition of Section "1.3.4.7.3 Database Interface". No chunk is found for A Group.

Interpretation: C Group's chunk found using Factor A|H|R3 shows that the majority of the participants needed overview information (of OSGi Service Model, Eclipse plug-in model, and process view) and information on extension points, for ASM Task 3. The majority of the industry participants also needed the aforementioned information and also information on database interface component.

Factor AveR3: Chunk found using Factor AveR3 for a group of participants is very similar if not identical to the group's chunk found using Factor R3. C Group's chunk found using Factor AveR3 is very similar to the group's chunk found using Factor R3 with the addition of Section "0. Contents" and exclusion of Section "1.3.2 Process View". I Group's chunk found using Factor AveR3 is identical to I Group's chunk found using Factor R3. A Group's chunk found using Factor AveR3 is very similar to A Group's chunk found using Factor R3 with the addition of Section "1.3.4.5.2 Dynamic Plug-in Support".

Interpretation: C Group's chunk found using Factor AveR3 reveals that in addition to those sections (except for section on process view) in the group's chunk found using Factor R3, sections on Table of Contents and dynamic plug-in support were also needed by some of the participants who rated them. I Group's chunk found using Factor AveR3 shows that all the sections in the group's chunk found using Factor R3 were needed by some of the industry participants who rated the sections. In fact, these sections were needed by the majority of the industry participants who rated the sections as shown by I Group's chunk found using Factor R3. A Group's chunk found using Factor AveR3 reveals that in addition to those sections in the group's chunk found using Factor R3, section on dynamic plug-in support was also needed by some of the academic participants who rated it.

Factor AveR3F: C Group's chunk found using Factor AveR3F comprises Sections "0. Contents" and "1.2.2 OSGi (Service Model)". I Group's chunk includes Sections "1.2.2 OSGi (Service Model)" and "1.2.2.2.4 Extension Points". No chunk is found for A Group.

Interpretation: This factor filters the chunks produced by Factor AveR3 to include only those sections rated by more than half of the participants in a group and with the average ratings of at least 3. The purpose is to remove those sections which have a small number of participants rating them.

C Group's chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, Table of Contents and brief overview of OSGi Service Model were needed for ASM Task 3. I Group's chunk found using Factor AveR3F shows that brief overview of OSGi Service Model, and information related to extension points, were needed by the majority of the industry participants, for the task.

11.2 **Benchmarking Chunks against An Oracle Set**

This section starts with a brief overview of the concept involved in ASM Task 3, which is modifiability. This is followed by the details of the oracle set for ASM Task 3 and the results of benchmarking the discovered chunks against this oracle set.

11.2.1 What is Modifiability?

ASM Task 3 is to find out from the perspective of a maintainer, how ASM was designed at the architectural level to achieve modifiability. "Modifiability is about the cost of change" (Bass et al., 2003). Two things need to be considered: 1) what can change (artefact), and 2) when is the change made and who makes the change (the environment) (Bass et al., 2003). A change (either in terms of add, delete, or modify) can happen to any aspect of a system: functions, platform, environment, exhibited qualities, capacity and user interface. It can be made to the implementation (source code change), during compile (using compile-time switches), during build (by choice of libraries), during configuration setup (such as parameter setting) or during execution (by parameter setting). It can be carried out by developer, end user, or system administrator. "Once a change has been specified, the new implementation must be designed, implemented, tested, and deployed" (Bass et al., 2003).

The tactics (design decisions) to achieve modifiability are available in 3 categories: localize changes, prevention of ripple effect and defer binding time (Bass et al., 2003). The first category comprises the following tactics: maintain semantic coherence, abstract common services, anticipate expected changes, generalize the module and limit possible options. The second category's tactics are hide information, maintain existing interfaces, restrict communication paths and use an intermediary. The last category aims to control deployment time and cost. It contains the following tactics: runtime registration (supports plug-and-play), configuration files, polymorphism, component replacement and adherence to defined protocols (runtime binding of independent processes).

Extensibility is a special form of modifiability (Agrawal, Imieliński, & Swami, 1993) (Agarwal, Aggarwal, & Prasad, 2000). It is the ease of extending the services' capabilities without affecting other parts of the system

(Agrawal et al., 1993). Portability is also a special form of modifiability (Agarwal et al., 2000). "Portability is captured as a platform modification" (Bass et al., 2003). It is the "the ability of the system to run under different computing systems: hardware, software or combination of the two" (Agarwal et al., 2000).

The above serve as guidelines for us in the construction of the oracle set for ASM Task 3. ASM adopts a plug-in architecture by leveraging other models. Consequently, we also included in the oracle set some sections related to these underlying models as they have impact on how ASM can be modified.

11.2.2 The Oracle Set for ASM Task 3

The oracle set for ASM Task 3 comprises 19 of the 62 sections of the ASM software architecture document (Table 11.2). These sections were agreed by both the researcher and a separate judge as mandatory for ASM Task 3. In Table 11.1, these sections are denoted by yellow rows and their section IDs are marked with asterisks. This is to make it easier to compare the chunks found against the oracle set. The relevance of these sections to ASM Task 3 is explained next.

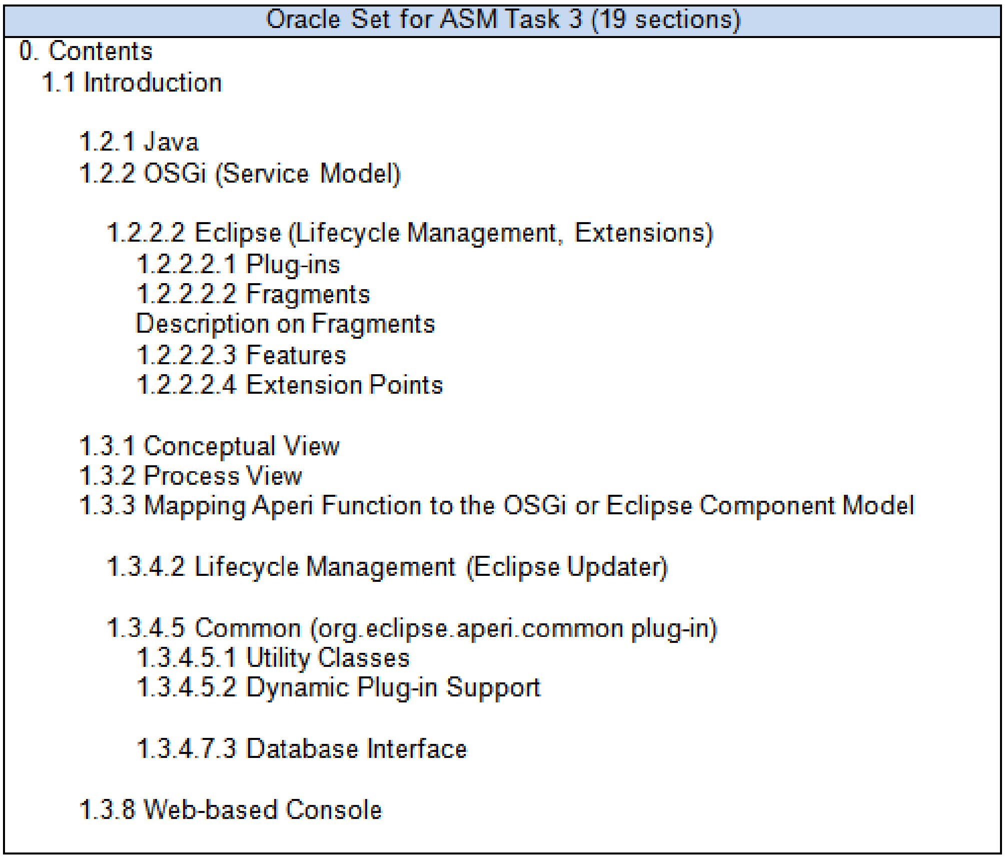
ASM is built upon Eclipse plug-in model. This model adopts deployment time registration of the components or plug-ins in this case. This is one of the defer binding time's tactics (i.e. runtime registration that supports plug-and-play operation) to achieve modifiability (Bass et al., 2003).

Section "0. Contents" gives a detailed organisation of the document. It gives a general idea on the layered architecture of ASM: Platform layer, Application layer and User Interface (UI) layer. Knowing this helps to understand the portability aspect of ASM since, a layered view informs about a system's portability (Bass et al., 2003). It is included in the oracle set because Section "1.3.1 Conceptual View" which explains about the conceptual layers of ASM omits the UI layer and focuses only on the Platform and the Application layers.

Section "1.1. Introduction" gives an overview of what Aperi Storage Manager is, a brief history on it, its goals, its leveraging the Eclipse Platform and the purpose of the document. The goals (extensible, componentized, commercial-friendly) and dependency (on Eclipse Platform) of ASM impacted its architectural design in supporting modifiability. Consequently, this section is included in the oracle set.

Section "1.2.1 Java" contains information on the development languages (Java, and C for native code) used and possible compatibility issues between Java Runtime Environments (JREs). The choice of Java as main development language reduces platform dependency (portability) issue of the system.

ASM leverages the Eclipse platform and Eclipse model uses OSGi component model as the basis for its plug-in architecture. Because of this we included the overview sections on Eclipse plug-in model, OSGi Service Model and plug-in in the oracle set, as these sections in a way reveal the scope of their uses in ASM. This is relevant even to people who have background knowledge on these aspects. In addition, the underlying models have some impact on how ASM was designed to achieve modifiability. These overview sections are: 1) **Section "1.2.2 OSGi (Service Model)"**. It provides a brief overview of the OSGi Service Model. It describes the function of the model in developing, deploying and managing services in a coordinated fashion and the notion of 'bundles' as entities for deployment. 2) **Section "1.2.2.2 Eclipse (Lifecycle Management, Extensions)"**. It provides general information on Eclipse platform for components integration and its use of OSGi components model for its plug-in architecture. It also explains that Eclipse plug-in model extends the base OSGi bundle interfaces with additional features such as ‘extension’ and ‘extension points’ (through which some processing elements or elements can be added to the host plug-in). This informs how ASM supports the extensibility (a special form of modifiability) of plug-ins, which are fundamental building blocks used in ASM. 3) **Section “1.2.2.2.1 Plug-ins”**. It explains the difference between plug-ins and bundles, the possibility of bundles leveraging plug-ins via extensions and extension points, and the possibility of a component to be defined as a plug-in, a bundle or both. It also contains some Java code on Bundle activator class.



Oracle Set for ASM Task 3 (19 sections)
0. Contents
1.1 Introduction
1.2.1 Java
1.2.2 OSGi (Service Model)
1.2.2.2 Eclipse (Lifecycle Management, Extensions)
1.2.2.2.1 Plug-ins
1.2.2.2.2 Fragments
Description on Fragments
1.2.2.2.3 Features
1.2.2.2.4 Extension Points
1.3.1 Conceptual View
1.3.2 Process View
1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model
1.3.4.2 Lifecycle Management (Eclipse Updater)
1.3.4.5 Common (org.eclipse.aperi.common plug-in)
1.3.4.5.1 Utility Classes
1.3.4.5.2 Dynamic Plug-in Support
1.3.4.7.3 Database Interface
1.3.8 Web-based Console

Table 11.2: The Oracle Set for ASM Task 3

Section “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” explicitly states that each major component of ASM is represented as an OSGi bundle or Eclipse plug-in or both. Both Section “1.2.2.2.1 Plug-ins” and “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” provide information on the building blocks used in ASM. Understanding the building blocks used helps to understand how ASM was designed to achieve modifiability.

The following 3 sections of ASM software architecture document are about how plug-ins could be extended (one form of modifiability): **Section “1.2.2.2.2 Fragments”** explains the use of fragment to provide additional functionality to an installed plug-in. **Section “Description on Fragments”** shows the ‘how’ of using fragments (including code snippet) to provide additional functionality to an installed plug-in. **Section “1.2.2.2.4 Extension Points”** provides more detailed explanation on how plug-in could be extended through ‘extension points’ with some example code.

Section “1.3.1 Conceptual View” explains the two conceptual layers of ASM. The Platform layer provides basic storage management functions and the basic server infrastructure. The Application layer provides high-level functions (such as file system capacity reporting and fabric zone control). A layered view informs about a system’s portability (Bass et al., 2003). Section “1.3.1 Conceptual View” contains the information about the layered architecture of ASM which is helpful to understand the portability aspect of ASM.

Section “1.3.2 Process View” gives an overview of the high-level processes of ASM. It also contains a diagram that depicts the connections between the processes. It shows the core components of ASM and which components could be substituted by alternatives (in particular, the GUI component). The latter gives some hints on the modifiable parts of ASM.

Section “1.3.4.5.2 Dynamic Plug-in Support” explains the importance of supporting dynamic plug-ins given the nature of ASM’s extensible architecture which is based on plug-ins, and dynamic loading and unloading. It also explains the concept of dynamic plug-ins and the design paradigms recommended by Aperi to support them (i.e. use stateless object instead of instance variable, enforce lifecycle management whenever object caches are used, object caches should implement IDisposable interface and bundle activators should contain the provided logic). This section is relevant to ASM Task 3 as it contains information on dynamic plug-in to support extensible (which is related to modifiability) architecture.

The following sections are related to deployment of possible changes made, which is one issue related to modifiability: 1) **Section “1.2.2.2.3 Features”** explains that feature is a group of plug-ins that defines a logical product feature and it is a logical entity for remote plug-in deployment and updates. 2) **Section “1.3.4.2 Lifecycle Management (Eclipse Updater)”** describes leveraging Eclipse Updater to simplify deployment and patch management.

The sections of ASM software architecture document mentioned in this paragraph are related to the tactic of abstracting common services to achieve modifiability. This tactic supports localization of modifications (i.e. modifications made only once to the common services instead of each module where the services are used) and prevention of ripple effects (i.e. modification to the modules using those services will not impact other users) (Bass et al., 2003). The first section is **“1.3.4.5 Common (org.eclipse.aperi.common plug-in)”**. It provides brief description on the org.eclipse.aperi.common plug-in which contains common code shared across implementation plug-ins. The second section is **“1.3.4.5.1 Utility Classes”** which provides brief description on utility classes (i.e. helper classes used by the rest of the platform).

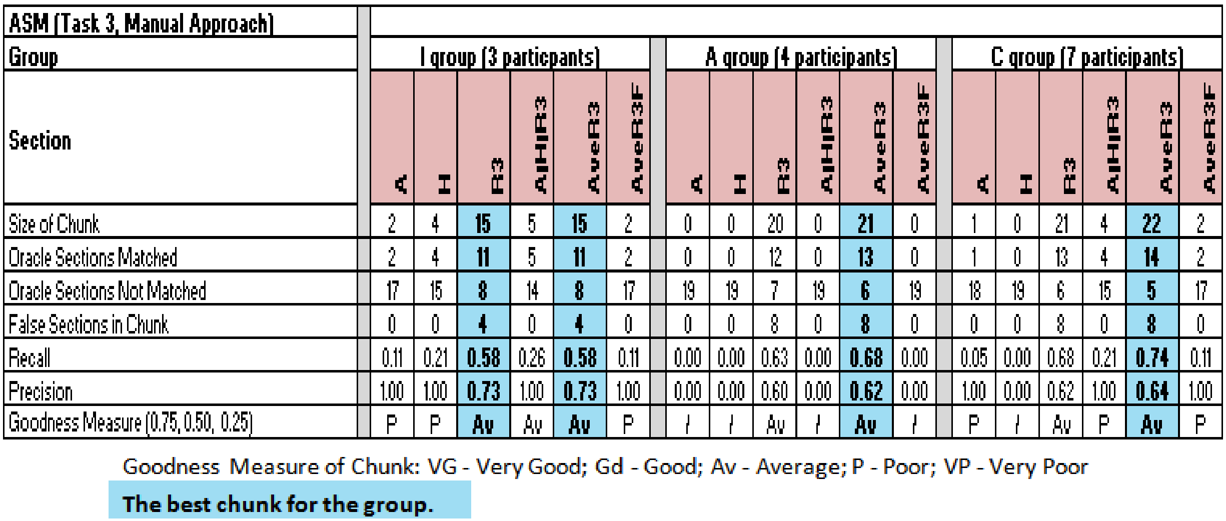
Section “1.3.4.7.3 Database Interface” explains the use of database interface component to minimize the coupling between the details of the table schema and other parts of the system. This is another aspect which supports modifiability and is relevant to ASM Task 3.

ASM also takes into consideration the ease of changing its user interface. **Section “1.3.8 Web based Console”** mentions that other layers of the architecture apart from the Graphical User Interface (GUI) layer should be usable by any user interface, for example the web-based GUI.

11.2.3 Benchmarking Results

Refer to Section “5.7 Benchmarking of Chunks against An Oracle Set” of Chapter 5 for all the concepts and terms needed to understand this section.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for ASM Task 3 are given in Table 11.3.



ASM (Task 3, Manual Approach)																		
Group	I group (3 particpants)						A group (4 participants)						C group (7 participants)					
Section	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F	A	H	R3	A|H|R3	AveR3	AveR3F
Size of Chunk	2	4	**15**	5	**15**	2	0	0	20	0	**21**	0	1	0	21	4	**22**	2
Oracle Sections Matched	2	4	**11**	5	**11**	2	0	0	12	0	**13**	0	1	0	13	4	**14**	2
Oracle Sections Not Matched	17	15	**8**	14	**8**	17	19	19	7	19	**6**	19	18	19	6	15	**5**	17
False Sections in Chunk	0	0	**4**	0	**4**	0	0	0	8	0	**8**	0	0	0	8	0	**8**	0
Recall	0.11	0.21	**0.58**	0.26	**0.58**	0.11	0.00	0.00	0.63	0.00	**0.68**	0.00	0.05	0.00	0.68	0.21	**0.74**	0.11
Precision	1.00	1.00	**0.73**	1.00	**0.73**	1.00	0.00	0.00	0.60	0.00	**0.62**	0.00	1.00	0.00	0.62	1.00	**0.64**	1.00
Goodness Measure (0.75, 0.50, 0.25)	P	P	**Av**	Av	**Av**	P	/	/	Av	/	**Av**	/	P	/	Av	P	**Av**	P

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 11.3: Attributes of Chunks (ASM Task 3)

For C Group, the goodness of chunks found using Factors R3 and AveR3 are ‘Average’, the chunks found using Factors A, A|H|R3 and AveR3F are ‘Poor’. No chunk is found using Factor H. We assessed the composition of the two ‘Average’ chunks to decide on the best chunk for C Group. The composition of Factor R3’s chunk (recall = 0.68, precision = 0.62, size = 21, goodness = ‘Average’) and Factor AveR3’s chunk (recall = 0.74, precision = 0.64, size = 22, goodness = ‘Average’) are very similar. Factor R3’s chunk matches 13 and Factor AveR3’s chunk matches 14, of the 19 oracle set’s sections. They contain the same 12 sections from the oracle set, with Factor R3’s chunk also including oracle set’s section “1.3.2 Process View” and Factor AveR3’s chunk also including oracle set’s sections “0. Contents” and “1.3.4.5.2 Dynamic Plug-in Support”. Both miss these 4 oracle set’s sections: “1.2.1 Java”, “1.3.1 Conceptual View”, “1.3.4.7.3 Database Interface”, “1.3.8 Web based Console”. Factor R3’s chunk also misses Section “0. Contents” and Section “1.3.4.5.2 Dynamic Plug-in Support”, whereas Factor AveR3’s chunk also misses Section “1.3.2 Process View”. Both chunks include 8 false sections: “1.3 High Level Architecture”, “1.3.3.4 Standards”, “1.3.4.4 SOAP & WSDL Support”, “1.3.4.7.1 RDBMS” and some sections on generic services (sections 1.3.4.8, 1.3.4.9, 1.3.4.15) and “2.3 Platforms”. With almost the same composition, higher recall and precision, Factor AveR3’s chunk is the best chunk for C Group.

The most basic chunk for C Group (recall = 0.05, precision = 1.00, size = 1, goodness = ‘Poor’) is found using Factor A. It comprises only Section “1.2.2 OSGi (Service Model)” that gives a brief overview of OSGi Service Model. This chunk misses a significant number of sections (i.e. 18) from the oracle set and therefore is not really useful for ASM Task 3.

For I Group, the goodness of chunks found using Factors R3, AveR3 and A|H|R3 are ‘Average’, and the goodness of chunks found using Factors A, H and AveR3F are ‘Poor’. Factors R3’s chunk and AveR3’s chunk are identical. We assessed the composition of this identical chunk and Factor A|H|R3’s chunk to decide on the best chunk for I Group.

Factor A|H|R3’s chunk (recall = 0.26, precision = 1.00, size = 5, goodness = ‘Average’) matches 5 of the oracle set’s sections (i.e. overview of OSGi Service Model, overview of Eclipse plug-in model, extension points, overview of process view, and database interface component) and has no false section. However, it misses 14 of the oracle set’s sections. Factor R3’s or Factor AveR3’s chunk (recall = 0.58, precision = 0.73, size = 15, goodness = ‘Average’) matches 11 of the sections in the oracle set. It misses 8 sections from the oracle set: “1.2.1 Java”, “1.3.8 Web based Console”, “1.3.1 Conceptual View”, “1.3.2 Process View”, “1.3.4.2 Lifecycle Management (Eclipse Updater)”, “1.3.4.5 Common (org.eclipse.aperi.common plug-in)”, “1.3.4.5.1 Utility Classes” and “1.3.4.5.2 Dynamic Plug-in Support”. It has 4 false sections:“1.2.2.1 OSGi Bundle Lifecycle”, “Description of OSGi Bundle Lifecycle Diagram”, “1.3.4.7.1 RDBMS” and “1.3.4.7.2 Schema”. With higher recall, Factor R3’s (or Factor AveR3’s) chunk is the best chunk for I Group.

For I Group, both Factors A and AveR3F produce chunks with the smallest number of sections (i.e. the most basic chunk). Both chunks (recall = 0.11, precision = 1.00, size = 2, goodness = ‘Poor’) contain Section “1.2.2 OSGi (Service Model)”. Factor A’s chunk also includes Section “1.3.4.7.3 Database Interface” whereas Factor AveR3F also includes Section “1.2.2.2.4 Extension Points”. They exclude 17 sections from the oracle set. These chunks are relevant but miss out too much information for ASM Task 3.

For A Group, the goodness of chunks found using Factors R3 and A|H|R3 are ‘Average’. No chunk is found using the other 4 factors. We assessed the composition of the two ‘Average’ chunks to decide on the best chunk for A Group. The composition of Factor R3’s chunk (recall = 0.63, precision = 0.60, size = 20, goodness = ‘Average’) and Factor AveR3’s chunk (recall = 0.68, precision = 0.62, size = 21, goodness = ‘Average’) are almost identical. They contain the same 12 sections from the oracle set, with Factor AveR3’s chunk including one additional oracle set’s section (“1.3.4.5.2 Dynamic Plug-in Support”). They have the same false sections. With almost the same composition, higher recall and precision, Factor AveR3’s chunk is the best chunk for A Group. The best chunk for A Group is very similar to the best chunk for C Group which is also found using Factor AveR3. The difference is the best chunk for A Group excludes oracle set’s sections “0. Contents” and “1.2.2.2 Eclipse (Lifecycle Management, Extensions)” but includes oracle set’s section “1.3.2 Process View”. They have the same false sections.

The most basic chunk for A Group is found using Factor R3.

The best chunk for C and A Groups are found using Factor AveR3 (with goodness measure of ‘Average’), and for I Group is found using Factor R3 or Factor AveR3 (with goodness measure of ‘Average’). The factor that produces the best chunk for a group is the best chunk-identification factor for the group.

Commonly-missed Oracle Set’s Sections and Common False Sections: Refer to Section “5.7.6 Determination of the Commonly-Missed Oracle Set’s Sections and Common False Sections” of Chapter 5 on how common oracle set's sections missed by a group of participants, and common false sections for a group of participants are determined.

For each group of participants, we try to give possible reason for commonly-missed oracle set's sections and common false sections. If no reason is given, it means that we cannot think of any plausible reason other than a commonly-missed oracle set's section was generally 'not needed', and a commonly-included false section was generally 'needed', by the respective group of participants when they attempted the task.

For C Group, most of the oracle set's sections were commonly missed except for Sections "1.2.2 OSGi (Service Model)", "1.2.2.2 Eclipse (Lifecycle Management, Extensions)" and "1.2.2.2.4 Extension Points". Sections "1.2.1 Java", "1.3.1 Conceptual View", "1.3.4.7.3 Database Interface", "1.3.8 Web based Console" were missed by all 6 chunk-identification factors. Section "1.3.4.5.2 Dynamic Plug-in Support" was missed by 5 factors and the other 11 commonly-missed oracle set's sections were missed by 4 factors.

For I Group, commonly-missed oracle set's sections are: "1.3.2 Process View" (missed by 5 factors); "0. Contents", "1.1 Introduction", "1.2.2.2.1 Plug-ins", "1.2.2.2.2 Fragments", "Description on Fragments", "1.2.2.2.3 Features" and "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model" (missed by 4 factors). Sections "1.2.1 Java", "1.3.1 Conceptual View", "1.3.4.2 Lifecycle Management (Eclipse Updater)", "1.3.4.5 Common (org.eclipse.aperi.common plug-in)", "1.3.4.5.1 Utility Classes", "1.3.4.5.2 Dynamic Plug-in Support" and "1.3.8 Web based Console" were missed by all the factors but they were not rated by any participant in I Group, and therefore do not qualify as commonly-missed sections for this group.

For A Group, chunks are discovered using Factors R3 and AveR3 only. Consequently, all the oracle set's sections were commonly missed except for Section "1.2.2.2 Eclipse (Lifecycle Management, Extensions)" which does not qualify as a commonly-missed section for this group since it was not rated by any participant in A Group. Sections "0. Contents", "1.2.1 Java", "1.3.1 Conceptual View", "1.3.4.7.3 Database Interface" and "1.3.8 Web based Console" were missed by all 6 chunk-identification factors. Section "1.3.4.5.2 Dynamic Plug-in Support" was missed by 5 factors and the other 12 commonly-missed oracle set's sections were missed by 4 factors.

For each group of participants, none of the chunk-identification factor includes Sections "1.2.1 Java", "1.3.8 Web based Console" and "1.3.1 Conceptual View" in chunk. The total exclusion of the first 2 sections coincides with the suggestion that these two parts (i.e. platform and user interface) of a system are quite distinct and subject to change, and changes to them are considered separately (Bass et al., 2003) from other genres of modifiability of a system. As for the conceptual view, it shows some aspects of the SA but does not contain specific information for ASM Task 3.

Section "1.3.2 Process View" also shows some aspects of the SA, and also the substitutable parts of ASM but is commonly excluded probably because it does not contain specific information for ASM Task 3. Section "0. Contents" gives a general idea on the layered architecture of ASM but is commonly excluded probably because this section does not contain specific information for ASM Task 3. As for the other commonly-missed sections, we cannot think of any plausible reason other than they were generally 'not needed' by the respective group of participants, when they attempted ASM Task 3.

There is no common false section for each group of participants.

11.3 **Discussion of Chunk Discovery Results**

The results show the discovery of chunks for ASM Task 3 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. The most basic chunks (i.e. chunk with the smallest number of section) for the 3 groups are produced by different factors: Factor A for C Group, Factors A and AveR3F for I Group, and Factor R3 for A Group. Their compositions can be seen in Table 11.1. For C and A Groups, chunks found using Factors R3 and AveR3 for the respective group are very similar to each other. For I Group, chunks found using these two factors are identical.

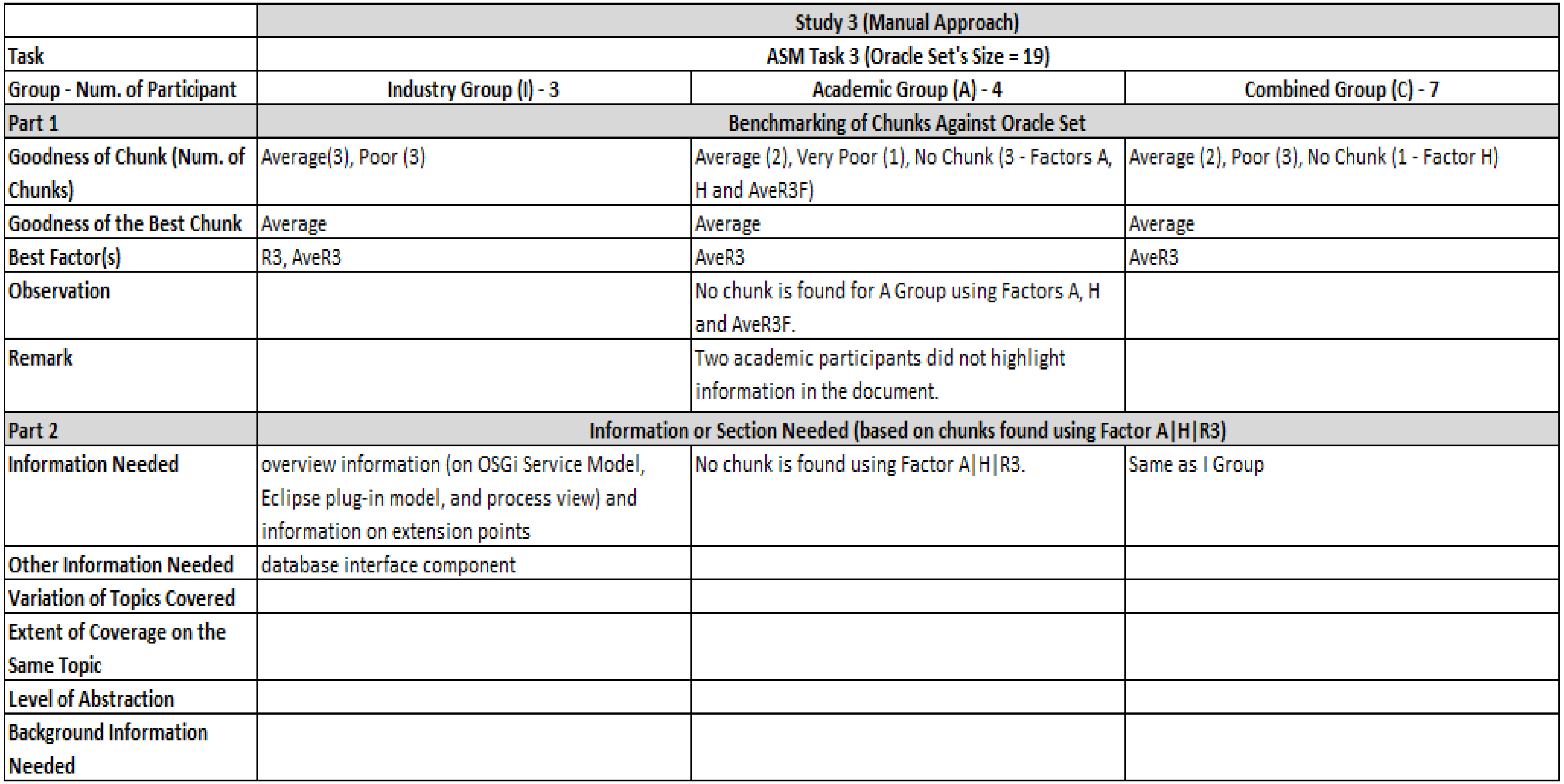
Part 1 of Table 11.4 summarises the main results of benchmarking of chunks. Recall that the number of participants in I and A Groups are 3 and 4, respectively. All the participants in each group provided some responses in terms of 1) where the needed information (answers) can be found and, 2) ratings of some sections visited in terms of their importance to the assigned task. Two of the academic participants did not highlight information in any section for this task and caused no chunk being found for A Group using Factor H.

In terms of how ‘good’ the chunks are for ASM Task 3, C Group has 2 chunks with ‘Average’ and 3 with ‘Poor’ goodness measures. No chunk is found for C Group using Factor H. I Group has 3 chunks with ‘Average’ and 3 with ‘Poor’ goodness measures. A Group has 2 chunks with ‘Average’ goodness measures. No chunk is found for A Group using the other 4 factors. The best chunk for C and A Groups are found using Factor AveR3 (with goodness measure of ‘Average’). The best chunk for I Group is found using Factor R3 or Factor AveR3 (with goodness measure of ‘Average’). The factor that produces the best chunk for a group is the best chunk-identification factor for the group.

The Part 2 of Table 11.4 summarises the information or sections needed by each group of participants. This is based on the groups’ chunks found using Factor A|H|R3. Refer to Section “5.8 Information Needed” in Chapter 5 for our reason. C group’s chunk found using Factor A|H|R3 is ‘Poor’, I Group’s chunk is ‘Average’. No chunk is found for A Group using Factor A|H|R3. As shown in Table 11.4, both C and I Groups of participants needed overview information (on OSGi Service Model, Eclipse plug-in model, and process view) and information on extension points, when attempting ASM Task 3. I Group also needed information on database interface component. We do not mention the information needed by A Group since no chunk is found for this group using Factor A|H|R3.

11.4 **Summary**

This chapter describes the chunking of architectural information for ASM Task 3. As with WCT Task 3 (i.e. a similar task performed on the second AD), the results show the discovery of usage-based chunks for the task of how a system was designed at the architectural level to achieve a cross-cutting concern, namely modifiability. Chapter 13 contains the comparison of the results for ASM Task 3 and WCT Task 3, and the other tasks; the summary of the overall results for Studies 2 and 3; and threats to validity of findings. Chapter 8 presents the chunking of architectural information for WCT Task 3. Chapter 8 presents the chunking of architectural information for WCT Task 3, which is also related to cross-cutting concern. The next chapter is on Study 3 (Online Approach).



	Study 3 (Manual Approach)		
Task	ASM Task 3 (Oracle Set's Size = 19)		
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 4	Combined Group (C) - 7
Part 1	Benchmarking of Chunks Against Oracle Set		
Goodness of Chunk (Num. of Chunks)	Average(3), Poor (3)	Average (2), Very Poor (1), No Chunk (3 - Factors A, H and AveR3F)	Average (2), Poor (3), No Chunk (1 - Factor H)
Goodness of the Best Chunk	Average	Average	Average
Best Factor(s)	R3, AveR3	AveR3	AveR3
Observation		No chunk is found for A Group using Factors A, H and AveR3F.	
Remark		Two academic participants did not highlight information in the document.	
Part 2	Information or Section Needed (based on chunks found using Factor A|H|R3)		
Information Needed	overview information (on OSGi Service Model, Eclipse plug-in model, and process view) and information on extension points	No chunk is found using Factor A|H|R3.	Same as I Group
Other Information Needed	database interface component		
Variation of Topics Covered			
Extent of Coverage on the Same Topic			
Level of Abstraction			
Background Information Needed			

Table 11.4: Summary of Results for ASM Task 3

---- Blank ----

12 Study 3 - Exploration of Software Architecture Documents (Online Approach Study)

This chapter describes Study 3, the third study we conducted to investigate the chunking of architectural information based on ADs' usage data. This chapter contains two main parts: 1) Study 3 itself and 2) chunking of architectural information for the six information-seeking tasks of Study 3.

Chapter 5 provides the foundation to understand this chapter and should be read prior to reading it. A discussion of threats to validity of findings can be found in Chapter 13, where the overall results for Studies 3 and 2 are summarised.

12.1 **Overview of Study 3**

We explain the differences between Studies 1, 2 and 3 in Section "5.1 Overview of Study 2" of Chapter 5. We recap here the main differences between these studies with more details on Study 3:

a) Recall that we simplified and improved the study design of the User Evaluation Study (Study 1) for Studies 2 and 3. The improvements made for Studies 2 and 3 were quite similar except for those parts that involved the use of KaitoroCap in Study 3.

   Study 3 (Online Approach Study) is the *online* version of Study 2 (Manual Approach Study). The reason is its study design is similar to Study 2 but the collection of data in Study 3 was on-line instead of off-line. In Study 3, participants explored the ADs in the form of wiki pages using KaitoroCap Version 2 (i.e. a version improved based on the findings from Study 1). Participants used KaitoroCap to capture their exploration paths through the documents, as well as all their other responses (such as answers, ratings, tags, comments, keywords, and so on).

   The pre-questionnaire of Study 3 is very similar to the pre-questionnaire of Study 1. The post-questionnaire of Study 3 is similar to the post-questionnaire of Study 1, except that the former has the additional questions on common navigation patterns. Both pre- and post- questionnaires of Study 3 were administered online, using SurveyMonkey (SurveyMonkey, 2012), embedded in KaitoroCap. There is only one questionnaire in Study 2 which is a combination of the pre- and post-questionnaires of Study 3, but without the questions related to educational background, wiki, tool similar to KaitoroCap, KaitoroCap's features, navigation paths and patterns.

b) The participants recruited for Study 1 (User Evaluation Study) were predominantly 'novices' in SA. They were only required to have some SA background, which could be having taken a course related to SA. The participants recruited for Study 2 (Manual Approach Study) must have at least two years of industry experience related to SA, or at least two years of teaching (or training) experience related to SA. Collectively participants in Study 2 were therefore more 'expert' in SA than those recruited in Study 1. The participants who took part in Study 3 were a mixture of 'novices' and 'experts' in SA.

c) As with Study 2, we employed quantitative analysis to identify chunks in Study 3. However, only factors that involved ratings only (i.e. Factors R3, AveR3, AveR3F) were used to identify chunks in Study 3. The reason was in Study 3 we did not solicit the data required by the other three chunk-identification factors (Factors A, H and A|H|R3) because of the use of KaitoroCap.

d) For Study 3, we also recruited students who have taken course or were doing research related to SA. Consequently during our analysis, in addition to the total combined group (C Group), we studied the three different sub-groups of participants: industry practitioner (I Group), academic (A Group), and student (S Group). There was no S Group in Study 2.

Study 3 was also conducted as a quasi-experiment (Easterbrook et al., 2008). The reason was the same as for Study 2: we did not have full control in assigning the study participants (i.e. subjects) randomly to the ADs and the information-seeking tasks (the treatments). The recruitment of highly specialized participants was difficult and we could not predict the actual number of participants that would participate. To ensure that we obtained the target number of participants we wanted for at least one of the documents we used, we allocated the participants we recruited to one document first. Subsequent participants that we recruited we then assigned to the second document.

12.2 **Study Design**

As mentioned earlier, the study design of Study 3 is similar to Study 2. Basically, participants with SA background were recruited to explore ADs to find the information needed for the given information-seeking tasks (or questions). The participants' usage data of the documents was collected for analysis to find chunks.

Specifically, Study 3 and Study 2 are similar in these aspects:

1. The use of the same ADs related to Web Curator Tool (WCT) and Aperi Storage Manager (ASM), and the same information-seeking tasks (or questions). Refer to Section "5.2.1 Choice of Documents" and Section "5.2.2 Determination of Information-seeking tasks and Roles" of Chapter 5 for the details on these aspects of the studies.
2. The same estimated time of participation for each participant (i.e. one hour and 15 minutes).
3. Two tasks were given in a different sequence for each alternate participant. This resulted in 6 sets of tasks (or 6 sets of questions) for each AD: Set A (Task 1, Task 2), Set B (Task 2, Task 1), Set C (Task 1, Task 3), Set D (Task 3, Task 1), Set E (Task 2, Task 3), Set F (Task 3, Task 2). The rationale was to balance-off the influence of the participants' familiarity with the document acquired during the first task, on the second task. We gave two information-seeking tasks in Studies 2 and 3, instead of three tasks as in the User Evaluation Study (Study 1) because we observed that with three tasks participants were less focused on each task.

The designs of the two studies differed in terms of the data collected, data collection instruments and the preparation of the documents. We explain these differences in the following sub-sections.

12.2.1 Data Collected

For each information-seeking task (or question) assigned, a participant was asked to explore the given AD to find the information needed or the answer for the task. In Study 3, the AD was given in the form of wiki pages in Atlassian Confluence Enterprise Wiki (Atlassian, 2013a). A participant was asked to capture his or her exploration path through the document during the exploration process by using KaitoroCap, which was installed as a plug-in in Atlassian Confluence (Atlassian, 2013a). To capture an exploration path, a participant has to initiate an exploration path capturing session for the information-seeking task and has to provide the metadata of the path. Five metadata of an exploration path were solicited: 1) what you try to look for, 2) keywords, 3) role, 4) reason of exploration, and 5) in relation to task. The values chosen for the metadata should be relevant to the information-seeking task as they served as contextual information for an exploration. The main purpose of the metadata was to allow the exploration paths to be searched by users using KaitoroCap.

A participant was also asked to provide the set and the number of the information-seeking task (or question). A task's set shows the prescribed sequence to attempt the two information-seeking tasks given to a participant. The prescribed sequence was not followed by some participants. A task's number shows an exploration path is for which information-seeking task.

In Study 3, a participant was asked to provide the answer to the information-seeking task as specific as he or she could, and not in bullet-point form as in Study 2. The answer was to be entered in KaitoroCap and could be entered anytime during the particular exploration session.

During an exploration session, a participant could navigate to any wiki page (containing one or more sections) of the document. For each section of the document visited for the task, a participant was asked to provide two ratings (in terms of its importance to the task and in terms of its importance to his or her overall understandability of the described SA), and also to provide tags and comments. These requirements were the same as in Study 2. The two ratings were compulsory but not the tags and comments. Recall that a tag is a user-defined keyword reflecting a section's content and a comment is a participant's more elaborated opinion on a section.

To save his or her exploration data of the current wiki page, a participant has to navigate away from the page to another page. The exploration data of a page consists of a participant's interactions with the page which include expanding ('Read More') and collapsing ('Collapse') sections, mouse entering and leaving sections, providing ratings and tags as well as comments, page loading and unloading, and clicking on hyperlinks to go to other pages.

The answer, ratings, tags and comments could be altered any time before the stop of an exploration path capturing session. The participant was instructed to stop an exploration path capturing session when he or she felt that he or she had found the answer. He or she then finalized the answer. The participant was also asked to provide ratings (5-point Likert-scale ranging from one the lowest to five the highest) on his or her own answer, satisfaction level with own exploration of the document to find his or her best answer to the task, and his or her expertise level in terms of the role undertaken when attempting the task. The participant was also asked to specify the amount of information (10-point Likert scale ranging from 10% to 100%) he or she thought, he or she found for the task.

The participant was asked to look at the tree view visualisation of the exploration path captured. As the owner of the path, it was compulsory for the participant to rate the path in terms of how good he or she thought the exploration path was. The participant was also asked to use KaitoroCap's search feature to find the exploration path he or she captured for the first information-seeking task.

Different from Study 2, the following data was not solicited from a participant in Study 3: 1) The start and stop time of attempting the information-seeking task. This information was captured automatically by KaitoroCap at the start of an exploration path capturing session and at the end of it. 2) The section(s) of the document where each bullet-point answer was found (since in Study 3 one overall answer was required), and whether it was found by looking at the section's title, reading the section or it was not obvious from the document but came from his or her past experience and knowledge. 3) Highlighting information relevant to the assigned task. 4) Suggested reading sequence (for those sections relevant to the assigned task) which in the participant's opinion would support a better understanding of the described SA. Instead in Study 3, KaitoroCap captured the actual exploration path of a participant through the document. In Study 3, a participant completed a pre-questionnaire (on background and experience) at the beginning of the participation session and a post-questionnaire after completing both information-seeking tasks. The main content of these questionnaires are explained in the next sub-section. The pre- and post-questionnaires were administered using SurveyMonkey (SurveyMonkey, 2012), but embedded in and accessible from KaitoroCap. In Study 2, a participant was given only one questionnaire, which was to be completed after finishing both information-seeking tasks. The questionnaire in Study 2 was a combination of the pre- and post-questionnaires of Study 3, but without the questions related to educational background, wiki, tool similar to KaitoroCap, KaitoroCap's features, navigation paths and patterns. No question on educational background was asked in Study 2 because what mattered was the SA-related experience of the industry and academic participants. For Study 3, we also targeted students as participants and therefore included questions on educational background.

12.2.2 Data Collection Instruments

The data collection instruments (Appendix 12.1) for Study 3 comprised of:

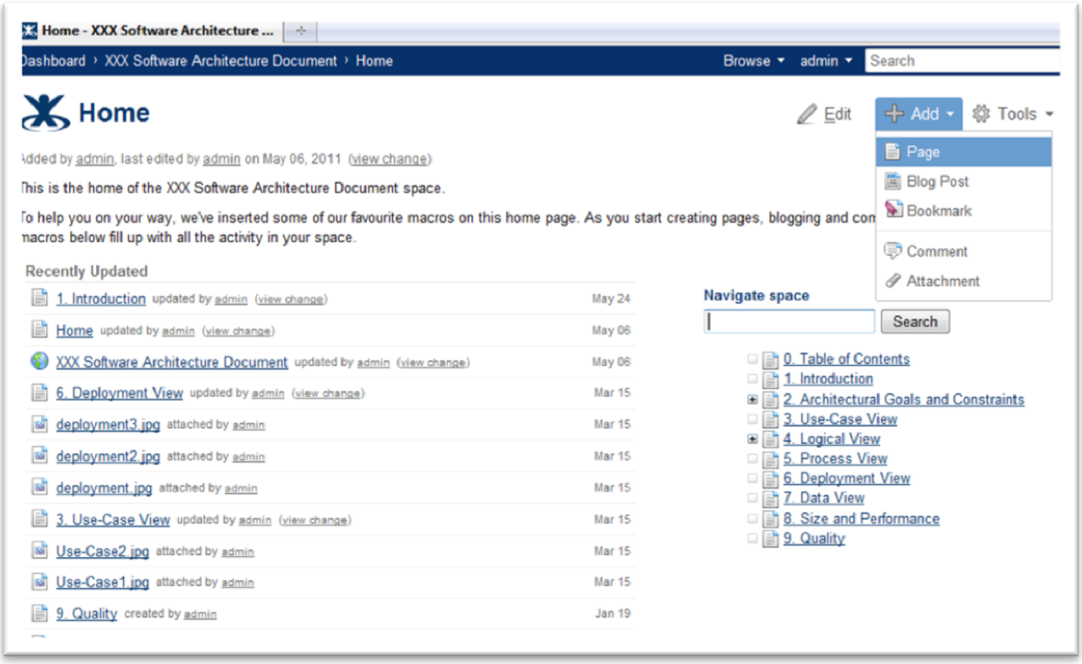
- Participant Information Sheets that detailed the terms and condition of participation in the study, and Consent Forms for participants to indicate consent to take part in the study.
- The instructions on how to access KaitoroCap Version 2, the tasks to be completed, and the information-seeking tasks to be attempted.
- Help file on how to use KaitoroCap Version 2. The help file was accessible from KaitoroCap.
- KaitoroCap Version 2. For Study 3, the ADs were created as wiki pages in Atlassian Confluence (Atlassian, 2013a). The participants used KaitoroCap which was installed as a plug-in in Atlassian Confluence (Atlassian, 2013a), to capture their exploration paths through the documents, as well as all their other responses (such as answers, keywords and so on) and their annotations (ratings, tags and comments) on the documents' sections. Refer to Section "4.5 KaitoroCap Version 2" of Chapter 4 for further details on the features of KaitoroCap Version 2.
- Pre-questionnaire and post-questionnaire administered using SurveyMonkey (SurveyMonkey, 2012), but embedded in and accessible from KaitoroCap Version 2:

1) The pre-questionnaire contained multiple-choice, structured Likert-scale and un-structured free-text questions. It included questions mainly on educational background, education and industry experience related to SA, experience with ADs, level of proficiency in English, experience with wiki environment, and exposure to the system described by the document.

2) The post-questionnaire contained structured Likert-scale and un-structured free-text questions. The questionnaire included questions mainly on experience using tool similar to KaitoroCap, perceptions of the main features of KaitoroCap, perceptions of the content of the given document, navigation characteristics, perception of the usage of textual descriptions and diagrams as well as their usefulness in supporting understanding, perception of the usefulness of navigation paths and common navigation patterns, ways the given document supported and hindered the understanding of the described SA, and suggestions for prototype improvement.

12.2.3 Preparation of Documents

As mentioned earlier, in Study 3 the two ADs were created as wiki pages in Atlassian Confluence (Atlassian, 2013a). Creation of a new wiki page was done by selecting the ‘Page’ menu item under the ‘Add’ menu provided by Atlassian Confluence (Figure 12.1). This produced a screen that prompted for the title and the content of the wiki page (Figure 12.2). When the ‘Save’ button was clicked, the wiki page was created in Atlassian Confluence.



Added by admin, last edited by admin on May 06, 2011 (view change)

This is the home of the XXX Software Architecture Document space.

To help you on your way, we've inserted some of our favourite macros on this home page. As you start creating pages, blogging and con
macros below fill up with all the activity in your space.

Recently Updated

1. Introduction updated by admin (view change)	May 24
Home updated by admin (view change)	May 06
XXX Software Architecture Document updated by admin (view change)	May 06
6. Deployment View updated by admin (view change)	Mar 15
deployment3.jpg attached by admin	Mar 15
deployment2.jpg attached by admin	Mar 15
deployment.jpg attached by admin	Mar 15
3. Use-Case View updated by admin (view change)	Mar 15
Use-Case2.jpg attached by admin	Mar 15
Use-Case1.jpg attached by admin	Mar 15
9. Quality created by admin	Jan 19

- 0. Table of Contents
- 1. Introduction
- 2. Architectural Goals and Constraints
- 3. Use-Case View
- 4. Logical View
- 5. Process View
- 6. Deployment View
- 7. Data View
- 8. Size and Performance
- 9. Quality

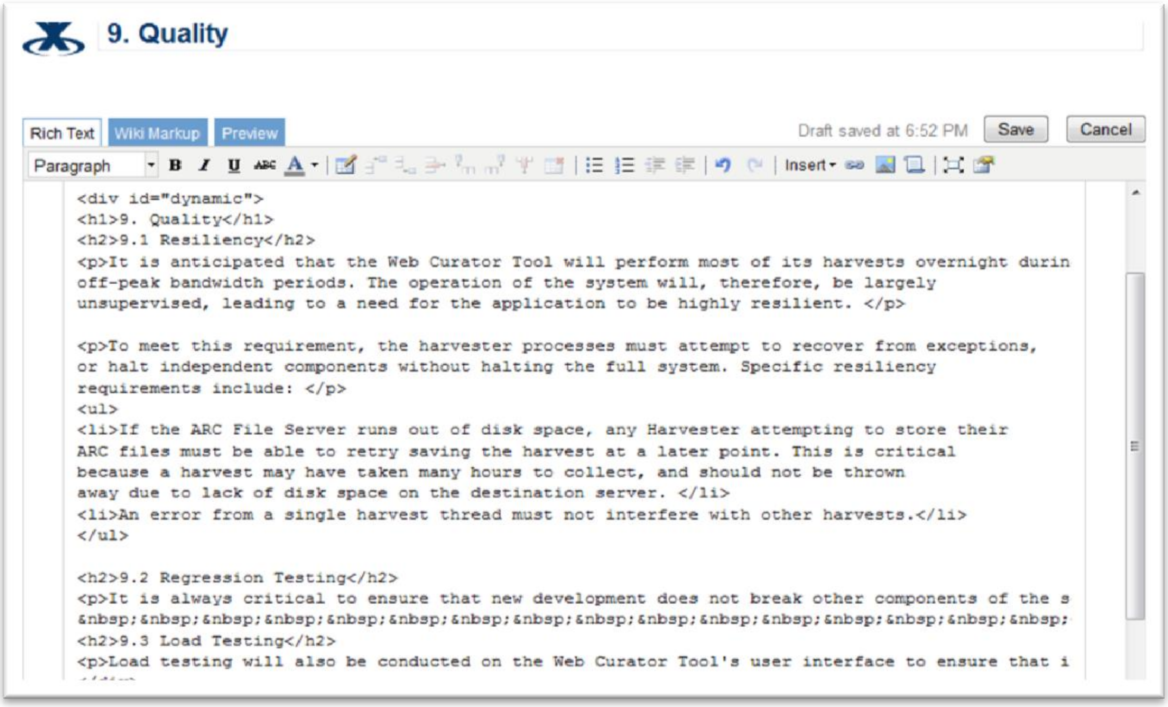
Figure 12.1: Creation of A New Page in Atlassian Confluence (Part 1)

KaitoroCap extended Atlassian Confluence’s functionality in creating and editing pages, to automatically create a page model for each page created or edited. The page model is not visible to the user. The page model improves performance of KaitoroCap by minimizing the amount of details saved during the exploration path capturing and for restructuring of the AD. Refer to Section “4.3 Usage Example” of Chapter 4 for further details.

As with Study 2, we wanted to gather the participants’ perceptions of the content of the sections of the ADs with regards to the assigned information-seeking task. To cater for that, KaitoroCap was built to dynamically insert annotation fields (to capture ratings, tags and comments) into each section of a page opened for viewing. KaitoroCap also inserts other features such as expanding and collapsing of section, and a surrounding border, to each section of a page opened for viewing. The on-demand insertion of all these features enables a clean separation of these features from the content of the pages.

In the creation of a page, a section is denoted by a ‘level 2 html heading (<H2>)’ (Figure 12.2). When the page is opened for viewing, KaitoroCap constructs the corresponding annotatable section (Figure 12.3) comprising this heading and all succeeding elements before the next ‘level 2 html heading (<H2>)’.

The number of annotatable sections for the two documents in Study 3 remained unchanged as in Study 2: WCT document contained 47 annotatable sections, with 6 containing diagrams (one section containing 2 closely-related diagrams), and ASM document contained 62 annotatable sections, with 7 sections containing diagrams.

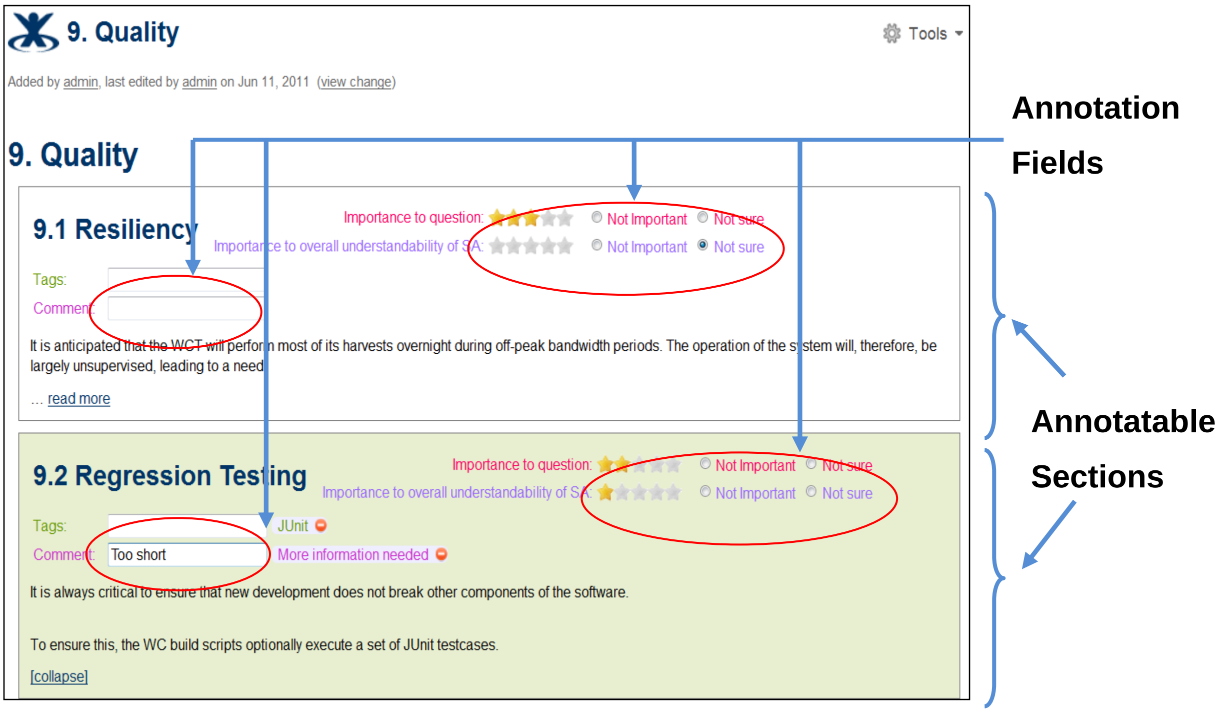


```
<div id="dynamic">
<h1>9. Quality</h1>
<h2>9.1 Resiliency</h2>
<p>It is anticipated that the Web Curator Tool will perform most of its harvests overnight durin
off-peak bandwidth periods. The operation of the system will, therefore, be largely
unsupervised, leading to a need for the application to be highly resilient. </p>

<p>To meet this requirement, the harvester processes must attempt to recover from exceptions,
or halt independent components without halting the full system. Specific resiliency
requirements include: </p>
<ul>
<li>If the ARC File Server runs out of disk space, any Harvester attempting to store their
ARC files must be able to retry saving the harvest at a later point. This is critical
because a harvest may have taken many hours to collect, and should not be thrown
away due to lack of disk space on the destination server. </li>
<li>An error from a single harvest thread must not interfere with other harvests.</li>
</ul>

<h2>9.2 Regression Testing</h2>
<p>It is always critical to ensure that new development does not break other components of the s
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<h2>9.3 Load Testing</h2>
<p>Load testing will also be conducted on the Web Curator Tool's user interface to ensure that i
```

Figure 12.2: Creation of A New Page in Atlassian Confluence (Part 2)



9. Quality

Added by admin, last edited by admin on Jun 11, 2011 (view change)

9. Quality

9.1 Resiliency

Importance to question: ★★★☆☆ ○ Not Important ○ Not sure

Importance to overall understandability of SA: ☆☆☆☆☆ ○ Not Important ● Not sure

Tags:

Comment:

It is anticipated that the WCT will perform most of its harvests overnight during off-peak bandwidth periods. The operation of the system will, therefore, be largely unsupervised, leading to a need

... read more

9.2 Regression Testing

Importance to question: ★★☆☆☆ ○ Not Important ○ Not sure

Importance to overall understandability of SA: ★☆☆☆☆ ○ Not Important ○ Not sure

Tags: JUnit

Comment: Too short More information needed

It is always critical to ensure that new development does not break other components of the software.

To ensure this, the WC build scripts optionally execute a set of JUnit testcases.

[collapse]

Figure 12.3: Example of Annotatable Sections of One of the Documents in KaitoroCap

12.3 **Participant Recruitment**

The focus of Study 3 was to collect more usage data (particularly exploration paths) by using KaitoroCap. The selection criteria for the recruitment of the participants for Study 3 were: (i) having some SA background and (ii) the willingness and ability to commit the required time (one hour and 15 minutes) and effort. The SA background could be either having taken or were taking course related to SA, having research experience in SA, having teaching (or training), or industry experience in SA.

The requirements of the specific background in SA and the considerable amount of time and effort to take part in the study discouraged the use of random sampling to recruit the participants for Study 3. As with Study 2, non-probabilistic sampling techniques, in particular convenience and snowball sampling were used to invite potential participants. Convenience sampling involves recruiting participants who meet the selection criteria and are available and willing to participate in the study (Barbara & Shari Lawrence, 2002). Snowball sampling refers to asking participants of the study to recommend other potential respondents (Barbara & Shari Lawrence, 2002).

Recall that we conducted Studies 2 and 3 at about the same time, and we recruited the participants for both studies simultaneously since we required participants of similar background. Refer to Section “5.3 Participant Recruitment” of Chapter 5 for further details of our strategies in the recruitment of participants for both studies, the total number of participants we invited for both studies, the number of those who responded and so on.

Recall (from Chapter 5) that we classified students who have industry experience in SA as industry participants, students with teaching experience in SA as academic participants. For those who have experiences in both, the length of SA experiences in industry and teaching decided their classifications. For Study 3 but not Study 2, we also targeted students who have taken or were taking course related to SA, and students having research experience related to SA. We classified this group of students as student participants.

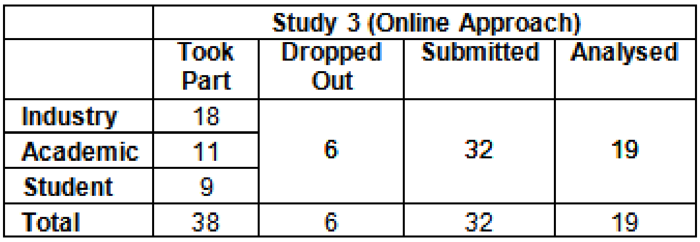
We explain here, our other means of recruiting students for Study 3 apart from personally emailing invitations to them following the introduction by our referrals. We put up recruitment advertisement at the Department of Computer Science and Department of Electrical and Computer Engineering (which offered the Software Engineering Programme) at University of Auckland, New Zealand. A twenty dollar voucher would be given to students who could be presented for observation to be made during the participation. Only one student was able to be physically presented.

We also engaged the help of our acquaintances at Auckland University of Technology (New Zealand), University of Malaya (Malaysia), University Putra Malaysia (Malaysia) and Swinburne University of Technology (Australia), to invite students with SA background from their universities.

Recall (from Chapter 5) that we personally invited 24 students (mostly PhD students) for Study 3. Eleven students responded out of which two were excluded as they did not have SA background, and nine took part in Study 3.

For Study 3 (Table 12.1), in total 38 participants (18 industry practitioners, 11 academics and 9 students) took part with 6 drop-outs, and 32 submitted responses out of which 19 were analysed. Those responses which were excluded from our analysis were either incomplete (not having any exploration path captured or not completing the pre-questionnaire or post-questionnaire) or showed mediocre attempt of the exploration tasks (such as not providing the question set and question number causing the problem in identifying the information-seeking task an exploration path was captured for).

The participants who took part in Study 3 were a mixture of ‘novices’ and ‘experts’ in SA as we could not recruit an adequate number of either group for this study. The novices included students who have taken or were taking course related to SA, and students having research experience in SA, or working professionals having less than two years of industry, teaching or training experience related to SA. The ‘experts’ were working professionals having at least two years of industry, teaching or training experience related to SA.



	Study 3 (Online Approach)			
	Took Part	Dropped Out	Submitted	Analysed
Industry	18	6	32	19
Academic	11			
Student	9			
Total	38	6	32	19

Table 12.1: Participants of Study 3 (Online Approach Study)

12.4 **Administration of the Study**

A set of the data collection instruments was given to those who agreed to take part in Study 3. Some of the data collection instruments (such as help file on using KaitoroCap, pre-questionnaire and post-questionnaire) were accessible through KaitoroCap.

A face-to-face meeting was conducted with physically accessible participants to explain the tasks to be completed and to clarify any question. For more distant participants, clarification was either done through email or online chat discussion.

12.5 **Grouping of Participants' Responses**

We segregated the responses of participants who have industry experience into an industry practitioner (I) group, and those who have academic teaching or training experience into an academic (A) group. The reason was the same as what was given for Study 2: there were considerable differences between industry and academics in their perception of SA and reusable assets (Bosch, 1999). For those who have experience from both, the length of experience decided to which group their responses were assigned. We also separated the responses of student participants who have neither of these experiences but have taken or were taking course or having research experience related to SA, into a student (S) group. We then analysed the responses of I, A and S Groups, in addition to the whole combined group (C Group).

12.6 **Chunking of Architectural Information**

This section presents the chunking of architectural information for the six information-seeking tasks in Study 3. For each of the information-seeking tasks, we present 1) the identification and interpretation of chunks, 2) benchmarking chunks against the respective oracle set, and 3) the discussion of chunk discovery results. To avoid repetition, we give the background of these three aspects at the beginning of this section.

1) Identification and Interpretation of Chunks: As mentioned earlier in this chapter, we used the 3 chunk-identification factors that involved ratings data only (i.e. Factors R3, AveR3 and AveR3F) to identify chunks for the information-seeking tasks in Study 3. Refer to Section "5.6.2 Factors Used to Identify Chunks" of Chapter 5 for further details on these factors. Recall that for Study 3, we identified chunks for the combined (C) group and the sub-groups of industry (I Group) and academic (A Group) participants. In addition, we identified chunks for the sub-group of student (S Group) participants where applicable.

The discovered chunks for each information-seeking task are shown in Table 12.2, 12.4, 12.6, 12.8, 12.10, and 12.12, respectively. Refer to Section "6.1 Identification and Interpretations of Chunks" in Chapter 6, on how to interpret the chunks shown in these tables. Recall that the size of a chunk is the number of the document's sections that the chunk comprises.

2) Benchmarking Chunks Against An Oracle Set: For each of the information-seeking tasks, the discovered chunks are benchmarked against the respective oracle set. The explanation on the oracle set for each task, can be found in the chapter written for each task in Study 2: Chapter 6, 7 and 8 for WCT Task 1, 2 and 3, and Chapter 9, 10 and 11 for ASM Task 1, 2 and 3. In Table 12.2, 12.4, 12.6, 12.8, 12.10 and 12.12, the sections that are in the respective oracle set are denoted by yellow rows and their section IDs are marked with asterisks. This is to make it easier to compare the chunks found against the oracle set.

Refer to Section "5.7 Benchmarking of Chunks against An Oracle Set" of Chapter 5 for all the concepts and terms needed to understand benchmarking of chunks against an oracle set.

The attributes (size, numbers of oracle set's section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for each task are given in Table 12.2, 12.4, 12.6, 12.8, 12.10 and 12.12, respectively.

We also identified 'the most basic chunk' for each group of participants. We do not elaborate the oracle set's sections missed by, and false sections included by each chunk as these sections can be seen in the respective table (Table 12.2, 12.4, 12.6, 12.8, 12.10 and 12.12).

Commonly-Missed Oracle Set's Sections and Common False Sections: For the information-seeking tasks in Study 3, in identifying the common oracle set's sections missed by a group of participants, we only consider oracle set's sections missed by all the three chunk-identification factors. For common false sections in chunks found for a group, we only consider those false sections included by all the three chunk-identification factors.

We cannot use the same criteria of 'missed by or included by, more than half of the chunk-identification factors' as in Study 2. This is because Study 2's criteria translates to 'at least two of the three chunk-identification factors' in Study 3. Due to the similarity between chunks found by Factors R3 and AveR3 for a group, many oracle set's sections excluded by Factor R3 and most likely Factor AveR3 (or vice versa) would be rendered commonly-missed oracle set's sections. Similarly, many false sections included by Factor R3 and most likely Factor AveR3 (or vice versa) would be rendered common false sections.

As with Study 2, oracle set's sections not rated at all by any participant in a group do not qualify as commonly-missed oracle set's sections for the group. The reason is we made no assumption about the value to assign to a section not rated by a participant and do not interpret such a section as 'not needed' by the participant for the task.

We also do not identify commonly-missed oracle set's sections and common false sections for a group that has only one participant. The reason is with one participant, the three factors would produce identical chunks and any oracle set's section excluded would automatically become commonly-missed oracle set's section. The same argument applies to false sections.

For each group of participants, we try to give possible reason for commonly-missed oracle set's sections and common false sections. If no reason is given, it means that we cannot think of any plausible reason other than a commonly-missed oracle set's section was generally 'not needed', and a commonly-included false section was generally 'needed', by the respective group of participants when they attempted the task.

3) Discussion of Chunk Discovery Results: We summarise the main results of benchmarking of chunks (best chunk, goodness of best chunk, and best chunk-identification factor for each group of participants). We also state any interesting observation made.

In addition, we summarise the information or sections needed by each group of participants. This is based on the groups' chunks found using Factor AveR3F. Factor AveR3F filters Factor AveR3's chunk to exclude those sections rated by a small number of participants in a group. Comparing Factor AveR3F and Factor R3, the former is based on average ratings contributed by the majority of a group, whereas the latter is based on the majority of those who rated only.

12.6.1 WCT Task 1

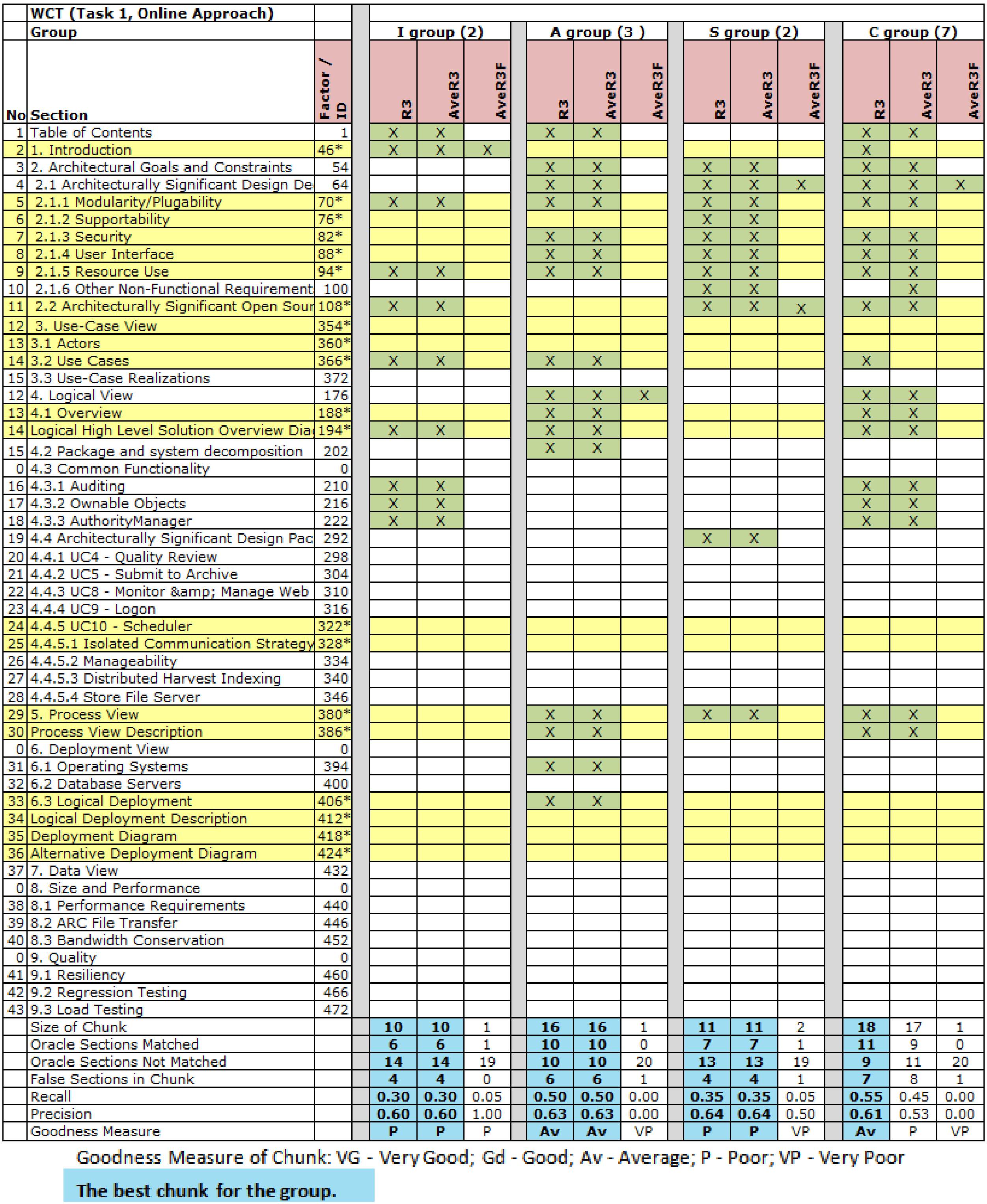
Table 12.2 shows the chunks discovered for WCT Task 1, for industry practitioner (I), academic (A), student (S), and combined (C) groups of participants. The number of participants in I, A, S and C Groups are 2, 3, 2 and 7, respectively.

12.6.1.1 Identification and Interpretation of Chunks

Factor R3: C Group's chunk found using Factor R3 contains 18 sections. These sections are related to the organisation of the document (Section "Table of Contents", and Section "2. Architectural Goals and Constraints" containing hyperlinks to its two sub-sections), overview of the document and WCT (Section "1. Introduction"), categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decisions"), 4 of the 5 key quality requirements (Sections "2.1.1 Modularity/Plugability", "2.1.3 Security", "2.1.4 User Interface", "2.1.5 Resource Use"), open source products used (Section "2.2 Architecturally Significant Open Source Products"), description of use-cases (Section "3.2 Use Cases"), brief overview of sub-sections related to logical view (Section "4. Logical View"), textual description of main logical components (Section "4.1 Overview"), high-level logical components diagram (Section "Logical High Level Solution Overview Diagram"), auditing (Section "4.3.1 Auditing"), access control mechanism (Sections "4.3.2 Ownable Objects" and "4.3.3 AuthorityManager"), process view diagram (Section "5. Process View"), and textual description of the main processes and threads of execution (Section "Process View Description").

The chunk found using Factor R3 for I Group excludes a number of the sections in C Group's chunk. I Group's chunk comprises 10 sections related to the organisation of the document (Section "Table of Contents"), overview of the document and WCT, 2 of the 5 key quality requirements (Sections "2.1.1 Modularity/Plugability" and "2.1.5 Resource Use"), open source products used, description of use-cases, high-level logical components diagram, auditing, and access control mechanism (Sections "4.3.2 Ownable Objects" and "4.3.3 AuthorityManager").

Chunk (with16 sections) found using Factor R3 for A Group includes many of the sections in C Group's chunk. It also comprises sections related to the organisation of the document (Sections "Table of Contents" and "2. Architectural Goals and Constraints"), categories of the key quality requirements, 4 of the 5 key quality requirements (Sections "2.1.1 Modularity/Plugability", "2.1.3 Security", "2.1.4 User Interface", "2.1.5 Resource Use"), description of use-cases, brief overview of sub-sections related to logical view (Section "4. Logical View"), textual description of main logical components, high-level logical components diagram, packages of Java code (Section "4.2 Package and system decomposition"), process view diagram, textual description of the main processes and threads of execution. In addition, Factor R3's chunk for A Group contains two sections related to operating systems (Section "6.1 Operating Systems") and logical deployment diagram which shows the logical mapping of components to nodes (Section "6.3 Logical Deployment").



	WCT (Task 1, Online Approach)													
	Group		I group (2)			A group (3)			S group (2)			C group (7)		
No	Section	Factor / ID	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F
1	Table of Contents	1	X	X		X	X					X	X	
2	1. Introduction	46*	X	X	X							X		
3	2. Architectural Goals and Constraints	54				X	X		X	X		X	X	
4	2.1 Architecturally Significant Design De	64				X	X		X	X	X	X	X	X
5	2.1.1 Modularity/Plugability	70*	X	X		X	X		X	X		X	X	
6	2.1.2 Supportability	76*							X	X				
7	2.1.3 Security	82*				X	X		X	X		X	X	
8	2.1.4 User Interface	88*				X	X		X	X		X	X	
9	2.1.5 Resource Use	94*	X	X		X	X		X	X		X	X	
10	2.1.6 Other Non-Functional Requirement	100							X	X			X	
11	2.2 Architecturally Significant Open Sour	108*	X	X					X	X	X	X	X	
12	3. Use-Case View	354*												
13	3.1 Actors	360*												
14	3.2 Use Cases	366*	X	X		X	X					X		
15	3.3 Use-Case Realizations	372												
12	4. Logical View	176				X	X	X				X	X	
13	4.1 Overview	188*				X	X					X	X	
14	Logical High Level Solution Overview Dia	194*	X	X		X	X					X	X	
15	4.2 Package and system decomposition	202				X	X							
0	4.3 Common Functionality	0												
16	4.3.1 Auditing	210	X	X								X	X	
17	4.3.2 Ownable Objects	216	X	X								X	X	
18	4.3.3 AuthorityManager	222	X	X								X	X	
19	4.4 Architecturally Significant Design Pac	292							X	X				
20	4.4.1 UC4 - Quality Review	298												
21	4.4.2 UC5 - Submit to Archive	304												
22	4.4.3 UC8 - Monitor &amp; Manage Web	310												
23	4.4.4 UC9 - Logon	316												
24	4.4.5 UC10 - Scheduler	322*												
25	4.4.5.1 Isolated Communication Strategy	328*												
26	4.4.5.2 Manageability	334												
27	4.4.5.3 Distributed Harvest Indexing	340												
28	4.4.5.4 Store File Server	346												
29	5. Process View	380*				X	X		X	X		X	X	
30	Process View Description	386*				X	X					X	X	
0	6. Deployment View	0												
31	6.1 Operating Systems	394				X	X							
32	6.2 Database Servers	400												
33	6.3 Logical Deployment	406*				X	X							
34	Logical Deployment Description	412*												
35	Deployment Diagram	418*												
36	Alternative Deployment Diagram	424*												
37	7. Data View	432												
0	8. Size and Performance	0												
38	8.1 Performance Requirements	440												
39	8.2 ARC File Transfer	446												
40	8.3 Bandwidth Conservation	452												
0	9. Quality	0												
41	9.1 Resiliency	460												
42	9.2 Regression Testing	466												
43	9.3 Load Testing	472												
	Size of Chunk		10	10	1	16	16	1	11	11	2	18	17	1
	Oracle Sections Matched		6	6	1	10	10	0	7	7	1	11	9	0
	Oracle Sections Not Matched		14	14	19	10	10	20	13	13	19	9	11	20
	False Sections in Chunk		4	4	0	6	6	1	4	4	1	7	8	1
	Recall		0.30	0.30	0.05	0.50	0.50	0.00	0.35	0.35	0.05	0.55	0.45	0.00
	Precision		0.60	0.60	1.00	0.63	0.63	0.00	0.64	0.64	0.50	0.61	0.53	0.00
	Goodness Measure		P	P	P	Av	Av	VP	P	P	VP	Av	P	VP

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 12.2: The Chunks for WCT Task 1 Identified Using Different Factors (Study 3)

S Group's chunk found using Factor R3 comprises 11 sections. These sections are related to the key quality requirements (including Section 2.1 and all its 6 sub-sections), open source products used, brief introduction to architecturally significant use-cases of WCT (Section "4.4 Architecturally Significant Design Packages"), process view diagram, and Section "2. Architectural Goals and Constraints".

Interpretation: All participants provided ratings for some sections of WCT document in terms of the importance of these sections to answering WCT Task 1.

C Group's chunk found using Factor R3 shows that for WCT Task 1 the sections needed by the majority of C Group who rated the respective sections, are mainly related to the organisation of the document, overview of the document and WCT, key quality requirements (almost all sections on this topic), open source products used, description of use-cases, logical view (all sections), auditing, access control mechanism, and process view (all sections).

I Group's chunk shows that for WCT Task 1 the sections needed by the majority of I Group who rated the respective sections, are related to the organisation of the document, overview of the document and WCT, certain key quality requirements (modularity/plugability and resource use), open source products used, description of use-cases, high-level logical components diagram, auditing, and access control mechanism.

A Group's chunk shows that for WCT Task 1 the sections needed by the majority of A Group who rated the respective sections, are related to the organisation of the document, most of the key quality requirements, description of use-cases, logical view (all sections on this topic), packages of Java code, process view (all sections), operating systems and logical deployment diagram.

S Group's chunk shows that for WCT Task 1 the sections needed by the majority of S Group who rated the respective sections, are related to the key quality requirements (all sections on this topic), open source products used, brief introduction to architecturally significant use-cases of WCT, process view diagram, and Section "2. Architectural Goals and Constraints" (containing hyperlinks to its two sub-sections).

Factor AveR3: C Group's chunk found using Factor AveR3 is very similar to the group's chunk found using Factor R3, with the exclusion of Sections "1. Introduction" and "3.2 Use Cases", and the inclusion of Section "2.1.6 Other Non-Functional Requirements". For I, A and S Groups, chunk found using Factor AveR3 is identical to chunk found using Factor R3 for the respective group.

Interpretation: Factor AveR3's chunk found for C Group shows that, in addition to those sections contained in Factor R3's chunk found for C Group, Section "2.1.6 Other Non-Functional Requirements" was needed by some of the participants who rated it. On the other hand, Sections "1. Introduction" and "3.2 Use Cases" were not needed for WCT Task 1, by some of the participants who rated them.

I Group's chunk found using Factor AveR3 shows that all sections in the group's chunk found using Factor R3 were needed by some of the industry participants who rated them (in fact needed by the majority of the industry participants who rated them as revealed by Factor R3's chunk). Similar interpretations are applicable to A and S Groups' chunks found using Factor AveR3 since each is identical to the respective group's chunk found using Factor R3.

Factor AveR3F: Chunks found for each group of participants using Factor AveR3F are very small: C Group's chunk comprises 1 section related to the categories of the key quality requirements; I Group's chunk comprises 1 section related to the overview of the document and WCT; A Group's chunk comprises 1 section which gives a brief overview of sub-sections related to logical view (Section “4. Logical View”); S Group’s chunk comprises 2 sections related to the categories of the key quality requirements, and open source products used.

Interpretation: C Group’ chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, categories of the key quality requirements was needed for WCT Task 1. I, A, S Groups’ chunks found using Factor AveR3F show that, the information on the overview of the document and WCT was needed by I Group, the brief overview of sub-sections related to logical view was needed by A Group, and the categories of the key quality requirements and open source products used were needed by S Group.

12.6.1.2 Benchmarking Chunks Against An Oracle Set

For the concept of ‘Software Architecture’ involved in WCT Task 1, refer to Section “6.2.1 What is Software Architecture?” of Chapter 6.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for WCT Task 1 are given at the bottom of Table 12.2. In Table 12.2, the sections that are part of the oracle set are denoted by yellow rows and their section IDs are marked with asterisks. This makes it easy to compare the chunks found against the oracle set.

Recall that chunks found using Factors R3 and AveR3 for a group of participants, are identical except for C Group where they are very similar.

For C Group, Factor R3’s chunk has goodness measure of ‘Average’ (recall = 0.55; precision = 0.61, size = 18, goodness = ‘Average’) which is better than Factor AveR3’s chunk with goodness measure of ‘Poor’ (recall = 0.45; precision = 0.53, size = 17, goodness = ‘Poor’) and AveR3F’s chunk with goodness measure of ‘Very Poor’ (recall = 0.00; precision = 0.00, size = 1, goodness = ‘Very Poor’). With the best goodness measure, Factor R3’s chunk is the best chunk for C Group. With zero recall and precision measures, Factor AveR3F’s chunk misses all the oracle set’s sections and contains only false section. It is therefore not useful for WCT Task 1.

The goodness measures of all the 3 chunks found for I Group are the same, which is ‘Poor’. Therefore, we assessed their composition to decide the best chunk for I Group. We chose Factor R3’s chunk (identical to Factor AveR3’s chunk) (recall = 0.30; precision = 0.60, size = 10, goodness = ‘Poor’) as the best chunk for I Group instead of Factor AveR3F’s chunk (recall = 0.05; precision = 1.00, size = 1, goodness = ‘Poor’). This is because Factor AveR3F’s chunk contains only 1 (Section “1. Introduction”) of the 20 oracle set’s sections.

For A Group, the identical chunk found using Factors R3 and AveR3 (recall = 0.50; precision = 0.63, size = 16, goodness = ‘Average’) has better goodness measure than the chunk found using Factor AveR3F (recall = 0.00; precision = 0.00, size = 1, goodness = ‘Very Poor’). The identical chunk is therefore the best chunk for A Group. With zero recall and precision measures, Factor AveR3F’s chunk found for A Group misses all the oracle set’s sections and contains only false section. It is not useful for WCT Task 1.

For S Group, the identical chunk found using Factors R3 and AveR3 (recall = 0.35; precision = 0.64, size = 11, goodness = ‘Poor’) is the best chunk for S Group because its goodness measure is better than Factor AveR3F’s chunk (recall = 0.05; precision = 0.50, size = 2, goodness = ‘Very Poor’).

The chunk found using Factor AveR3F for each of the 4 groups of participants is the smallest chunk for the group, and therefore is the most basic chunk for the respective group.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C Group is Factor R3. For each of I, A and S Groups the best factor is either Factor R3 or Factor AveR3.

Commonly-Missed Oracle Set’s Sections and Common False Sections: The oracle set’s sections commonly missed by C Group are Sections “2.1.2 Supportability”, “3. Use-Case View”, “3.1 Actors” and all the sections on logical deployment (Sections “6.3 Logical Deployment”, “Logical Deployment Description”, “Deployment Diagram”, “Alternative Deployment Diagram”). Sections “4.4.5 UC10 – Scheduler” and “4.4.5.1 Isolated Communication Strategy” are excluded by all factors because of not being rated by any participant in C Group. These sections do not qualify as commonly-missed oracle set’s sections for C Group.

The oracle set’s section commonly missed by I Group are mainly related to key quality requirements (Sections “2.1.2 Supportability”, “2.1.3 Security”, “2.1.4 User Interface”) and use-case view (Sections “3. Use-Case View” and “3.1 Actors”). The other 9 oracle set’s section are excluded by all factors because of not being rated by any participant in I Group. These sections do not qualify as commonly-missed oracle set’s sections for I Group.

The oracle set’s section commonly missed by A Group are: Sections “1. Introduction”, “2.1.2 Supportability”, “3. Use-Case View”, “3.1 Actors”, 3 of the sections on logical deployment (Sections “Logical Deployment Description”, “Deployment Diagram”, “Alternative Deployment Diagram”). Sections “2.2 Architecturally Significant Open Source Products”, “4.4.5 UC10 – Scheduler” and “4.4.5.1 Isolated Communication Strategy” are excluded by all factors because of not being rated by any participant in A Group. These sections do not qualify as commonly-missed oracle set’s sections for A Group.

The oracle set’s section commonly missed by S Group are: Section “1. Introduction”, 3 sections on use-case view (Sections “3. Use-Case View”, “3.1 Actors”, “3.2 Use Cases”) and all the sections on logical deployment (Sections “6.3 Logical Deployment”, “Logical Deployment Description”, “Deployment Diagram”, “Alternative Deployment Diagram”). The other 5 oracle set’s section are excluded by all factors because of not being rated by any participant in S group. These sections do not qualify as commonly-missed oracle set’s sections for S Group.

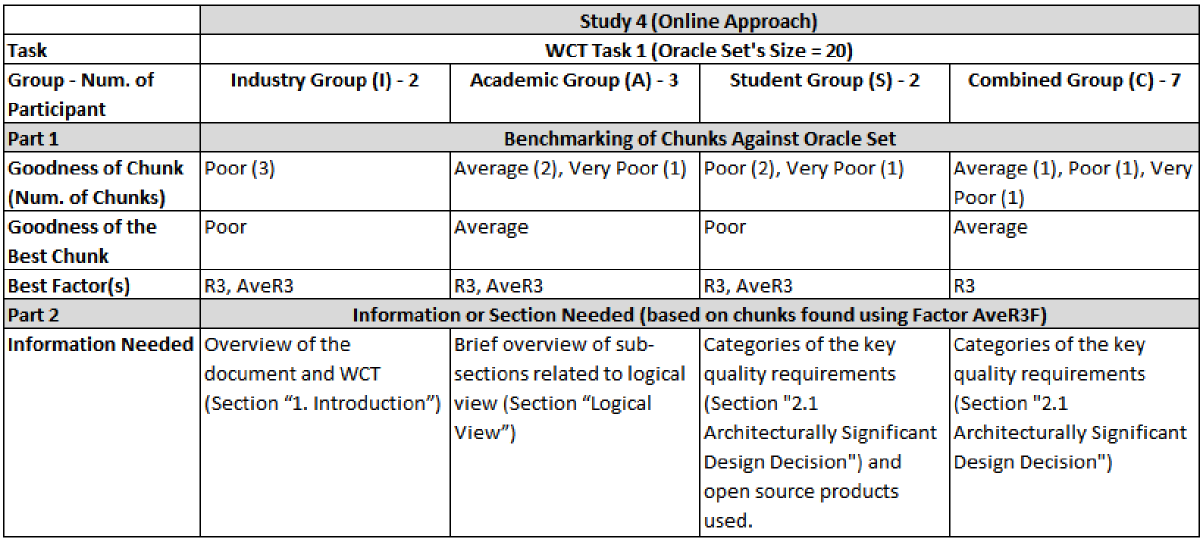
Section “1.1 Introduction” gives an overview of the document and WCT. The common exclusion of this section by A Group could be due to this section does not specifically contain information on the SA aspect of WCT. We could not think of any plausible reason for the common exclusion of other identified oracle set’s sections by the respective group of participants, other than they were generally ‘not needed’ by the group, when attempting WCT Task 1.

The common false section for each C and S Groups is Section “2.1 Architecturally Significant Design Decision”, and for A Group is Section “4. Logical View”. There is no common false section for I Group. Sections “2.1 Architecturally Significant Design Decisions” lists the categories of the key quality requirements which are described by its sub-sections. This section was probably useful for C and S Groups to obtain an overview of the subsequent materials. Section “4. Logical View” contains a brief description of its sub-sections and hyperlinks to them, which was probably useful for A Group to navigate to its sub-sections.

12.6.1.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for WCT Task 1 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. For WCT Task 1, chunks found using Factors R3 and AveR3 are identical for I, A and S Groups, and very similar for C Group.

Part 1 of Table 12.3 summarises the main results of benchmarking of chunks. The number of participants in I, A and S Groups are quite balance, which is 2, 3 and 2, respectively. In terms of how ‘good’ the chunks are for WCT Task 1, C Group has one each of ‘Average’, ‘Poor’ and ‘Very Poor’ chunk. I Group has 3 ‘Poor’ chunks. A Group has 2 ‘Average’ chunks and 1 ‘Very Poor’ chunk. S Group has 2 ‘Poor’ chunks and 1 ‘Very Poor’ chunk.



	Study 4 (Online Approach)			
Task	WCT Task 1 (Oracle Set's Size = 20)			
Group - Num. of Participant	Industry Group (I) - 2	Academic Group (A) - 3	Student Group (S) - 2	Combined Group (C) - 7
Part 1	Benchmarking of Chunks Against Oracle Set			
Goodness of Chunk (Num. of Chunks)	Poor (3)	Average (2), Very Poor (1)	Poor (2), Very Poor (1)	Average (1), Poor (1), Very Poor (1)
Goodness of the Best Chunk	Poor	Average	Poor	Average
Best Factor(s)	R3, AveR3	R3, AveR3	R3, AveR3	R3
Part 2	Information or Section Needed (based on chunks found using Factor AveR3F)			
Information Needed	Overview of the document and WCT (Section “1. Introduction”)	Brief overview of sub-sections related to logical view (Section “Logical View”)	Categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision") and open source products used.	Categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision")

Table 12.3: Summary of Results for WCT Task 1 (Study 3)

For WCT Task 1, the best chunk for C Group is found using Factor R3. The best chunks for I, A and S Groups are found using either Factor R3 or Factor AveR3. Following that, the best chunk-identification factor for C Group is Factor R3. For each of I, A and S Groups the best factor is either Factor R3 or Factor AveR3. These factors produce identical chunk for each of the 3 sub-groups.

Part 2 of Table 12.3 summarises the information or sections needed by each group of participants for WCT Task 1. This is based on the groups' chunks found using Factor AveR3F. We give our rationale for this at the beginning of Section "12.6 Chunking of Architectural Information" in this chapter.

For WCT Task 1, the three sub-groups of participants needed different information: I Group needed the overview of the document and WCT, A Group needed brief overview of sub-sections related to logical view, and S Group needed the categories of the key quality requirements and open source products used. This shows that the sub-groups needed different overview information for WCT Task 1, and this information might not be specifically on the architectural design of WCT. When all the participants are combined into C Group, the group needed information on the categories of the key quality requirements for WCT Task 1.

12.6.2 WCT Task 2

Table 12.4 shows the chunks discovered for WCT Task 2, for industry practitioner (I), academic (A), student (S), and combined (C) groups of participants. The number of participants in I, A, S and C Groups are 2, 1, 3 and 6, respectively.

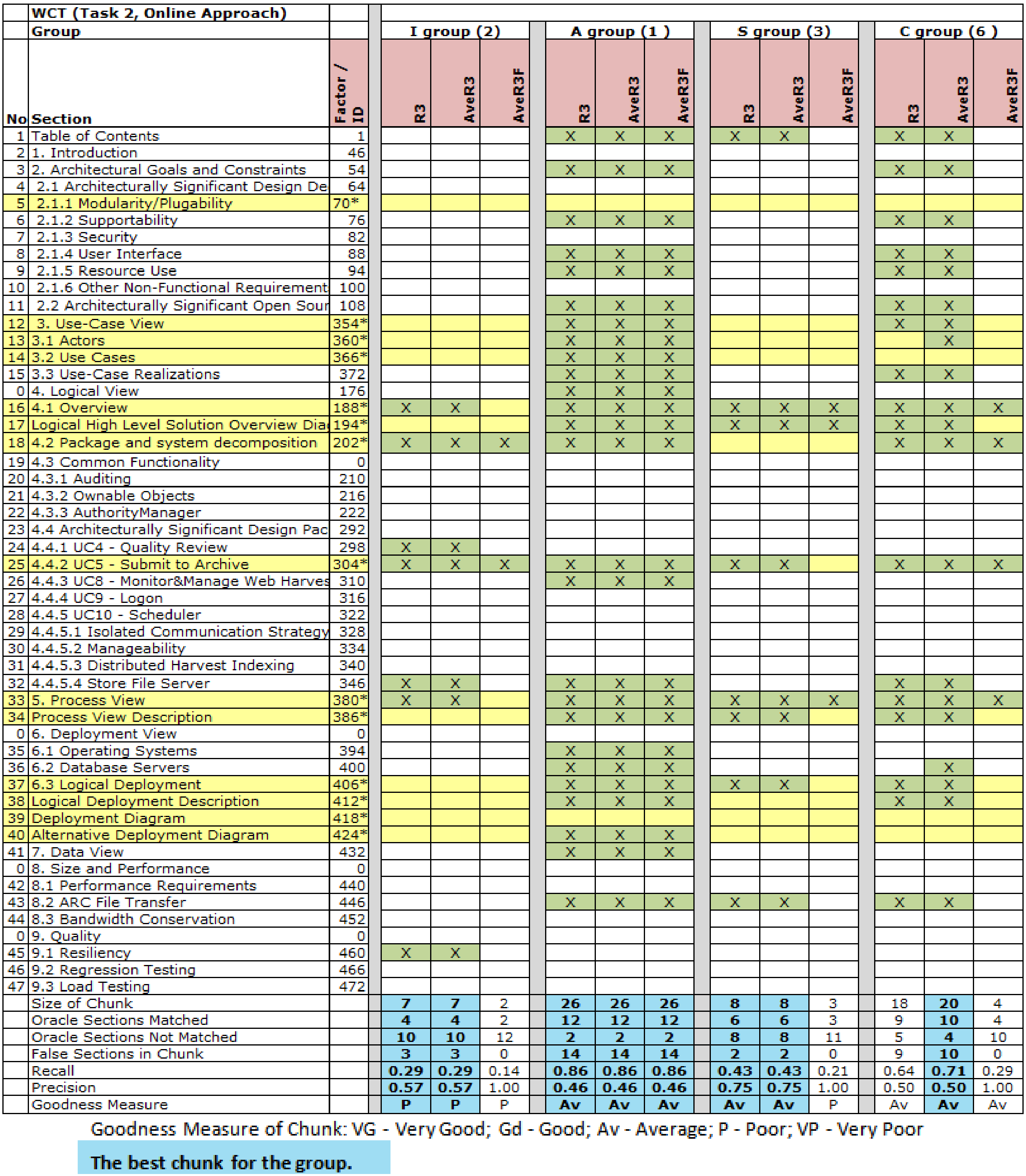
12.6.2.1 Identification and Interpretation of Chunks

Factor R3: C Group's chunk found using Factor R3 contains 18 sections. These sections are related to the organisation of the document (Section "Table of Contents", and Section "2. Architectural Goals and Constraints" containing hyperlinks to its two sub-sections), 3 of the 5 key quality requirements (Sections "2.1.2 Supportability", "2.1.4 User Interface" and "2.1.5 Resource Use"), open source products used (Section "2.2 Architecturally Significant Open Source Products"), use-case diagram (Section "3. Use-Case View"), Section "3.3 Use-Case Realizations", textual description of main logical components (Section "4.1 Overview"), high-level logical components diagram (Section "Logical High Level Solution Overview Diagram"), packages of Java code (Section "4.2 Package and system decomposition"), detailed description of 'Submit to Archive' use-case (Section "4.4.2 UC5 – Submit to Archive"), store file server (Section "4.4.5.4 Store File Server"), process view diagram (Section "5. Process View") and textual description of the main processes and threads of execution (Section "Process View Description"), logical mapping of components to nodes (diagram and its textual description) (Section "6.3 Logical Deployment" and Section "Logical Deployment Description") and support of large ARC file transfer (Section "8.2 ARC File Transfer").

I Group's chunk found using Factor R3 comprises only 7 sections: Sections "4.1 Overview", "4.2 Package and system decomposition", "4.4.1 UC4 – Quality Review" (detailed description of use-case related to quality review), "4.4.2 UC5 – Submit to Archive", "4.4.5.4 Store File Server", "5. Process View" and "9.1 Resiliency" (system resiliency requirement).

The chunk found using Factor R3 for A Group includes all the sections in C Group's chunk. In addition A Group's chunk includes 8 sections related to description of actors (Section "3.1 Actors"), description of use-cases (Section "3.2 Use Cases"), brief overview of sub-sections related to logical view (Section "4. Logical View"), detailed description of use-case UC8 on monitoring and managing Web Harvester System (Section "4.4.3 UC8 – Monitor & Manage Web Harvester"), operating systems (Section "6.1 Operating Systems"), database servers (Section “6.2 Database Servers”), alternative deployment diagram (Section “Alternative Deployment Diagram”) and data view (Section “7. Data View”).

S Group’s chunk found using Factor R3 comprises 8 sections: Sections “Table of Contents”, “4.1 Overview”, “Logical High Level Solution Overview Diagram”, “4.4.2 UC5 – Submit to Archive”, “5. Process View”, “Process View Description”, “6.3 Logical Deployment” and “8.2 ARC File Transfer”.



	WCT (Task 2, Online Approach)													
	Group		I group (2)			A group (1)			S group (3)			C group (6)		
No	Section	Factor / ID	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F
1	Table of Contents	1				X	X	X	X	X		X	X	
2	1. Introduction	46												
3	2. Architectural Goals and Constraints	54				X	X	X				X	X	
4	2.1 Architecturally Significant Design De	64												
5	2.1.1 Modularity/Plugability	70*												
6	2.1.2 Supportability	76				X	X	X				X	X	
7	2.1.3 Security	82												
8	2.1.4 User Interface	88				X	X	X				X	X	
9	2.1.5 Resource Use	94				X	X	X				X	X	
10	2.1.6 Other Non-Functional Requirement	100												
11	2.2 Architecturally Significant Open Sour	108				X	X	X				X	X	
12	3. Use-Case View	354*				X	X	X				X	X	
13	3.1 Actors	360*				X	X	X					X	
14	3.2 Use Cases	366*				X	X	X						
15	3.3 Use-Case Realizations	372				X	X	X				X	X	
0	4. Logical View	176				X	X	X						
16	4.1 Overview	188*	X	X		X	X	X	X	X	X	X	X	X
17	Logical High Level Solution Overview Dia	194*				X	X	X	X	X	X	X	X	
18	4.2 Package and system decomposition	202*	X	X	X	X	X	X				X	X	X
19	4.3 Common Functionality	0												
20	4.3.1 Auditing	210												
21	4.3.2 Ownable Objects	216												
22	4.3.3 AuthorityManager	222												
23	4.4 Architecturally Significant Design Pac	292												
24	4.4.1 UC4 - Quality Review	298	X	X										
25	4.4.2 UC5 - Submit to Archive	304*	X	X	X	X	X	X	X	X		X	X	X
26	4.4.3 UC8 - Monitor&Manage Web Harves	310				X	X	X						
27	4.4.4 UC9 - Logon	316												
28	4.4.5 UC10 - Scheduler	322												
29	4.4.5.1 Isolated Communication Strategy	328												
30	4.4.5.2 Manageability	334												
31	4.4.5.3 Distributed Harvest Indexing	340												
32	4.4.5.4 Store File Server	346	X	X		X	X	X				X	X	
33	5. Process View	380*	X	X		X	X	X	X	X	X	X	X	X
34	Process View Description	386*				X	X	X	X	X		X	X	
0	6. Deployment View	0												
35	6.1 Operating Systems	394				X	X	X						
36	6.2 Database Servers	400				X	X	X					X	
37	6.3 Logical Deployment	406*				X	X	X	X	X		X	X	
38	Logical Deployment Description	412*				X	X	X				X	X	
39	Deployment Diagram	418*												
40	Alternative Deployment Diagram	424*				X	X	X						
41	7. Data View	432				X	X	X						
0	8. Size and Performance	0												
42	8.1 Performance Requirements	440												
43	8.2 ARC File Transfer	446				X	X	X	X	X		X	X	
44	8.3 Bandwidth Conservation	452												
0	9. Quality	0												
45	9.1 Resiliency	460	X	X										
46	9.2 Regression Testing	466												
47	9.3 Load Testing	472												
	Size of Chunk		**7**	**7**	2	**26**	**26**	**26**	**8**	**8**	3	18	**20**	4
	Oracle Sections Matched		**4**	**4**	2	**12**	**12**	**12**	**6**	**6**	3	9	**10**	4
	Oracle Sections Not Matched		**10**	**10**	12	**2**	**2**	**2**	**8**	**8**	11	5	**4**	10
	False Sections in Chunk		**3**	**3**	0	**14**	**14**	**14**	**2**	**2**	0	9	**10**	0
	Recall		**0.29**	**0.29**	0.14	**0.86**	**0.86**	**0.86**	**0.43**	**0.43**	0.21	0.64	**0.71**	0.29
	Precision		**0.57**	**0.57**	1.00	**0.46**	**0.46**	**0.46**	**0.75**	**0.75**	1.00	0.50	**0.50**	1.00
	Goodness Measure		**P**	**P**	P	**Av**	**Av**	**Av**	**Av**	**Av**	P	Av	**Av**	Av

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 12.4: The Chunks for WCT Task 2 Identified Using Different Factors (Study 3)

Interpretation: All participants provided ratings for some sections of WCT document in terms of the importance of these sections to answering WCT Task 2.

C Group's chunk found using Factor R3 shows that for WCT Task 2, the sections needed by the majority of C Group who rated the respective sections are related to organisation of the document, some key quality requirements (supportability, user interface and resource use), open source products used, use-case diagram, use-case realizations, logical view, packages of Java code, detailed description of 'Submit to Archive' use-case, store file server, process view, logical mapping of components to nodes (diagram and its textual description) and support of large ARC file transfer. It is surprising that Section "3.3 Use-Case Realizations" that contains little information was needed for ASM Task 2.I Group's chunk shows that for WCT Task 2, the sections needed by the majority of I Group who rated the respective sections are related to textual description of main logical components, packages of Java code, detailed descriptions of 'Quality Review' and 'Submit to Archive' use-cases, store file server, process view diagram and system resiliency requirements.

A Group's chunk shows that in addition to those sections contained in C Group's chunk, the majority of A Group (which consists of only one participant) who rated the respective sections, needed description of actors and use-cases, brief overview of sub-sections related to logical view, detailed description of use-case UC8 on monitoring and managing Web Harvester System, operating systems, database servers, alternative deployment diagram and data view, for WCT Task 2.

S Group's chunk shows that for WCT Task 2, the sections needed by the majority of S Group who rated the respective sections are related to main logical components (both textual description and the diagram), detailed description of 'Submit to Archive' use-case, process view (both textual description and the diagram), logical mapping of components to nodes (diagram), and support of large ARC file transfer.

Factor AveR3: C Group's chunk found using AveR3 is very similar to the group's chunk found using Factor R3, but with the addition of Section "3.1 Actors" and "6.2 Database Servers". For I, A and S Groups, chunk found using Factor AveR3 is identical to chunk found using Factor R3 for the respective group. A Group's chunks found using all the 3 factors are identical because there was only one participant in A Group.

Interpretation: Factor AveR3's chunk found for C Group shows that, in addition to the sections contained in Factor R3's chunk found for C Group, sections on description of actors and databases were needed by some of the participants who rated them.

I Group's chunk found using Factor AveR3 shows that all the sections in the group's chunk found using Factor R3 were needed by some of the industry participants who rated them (in fact needed by the majority of the industry participants who rated them as revealed by Factor R3's chunk). Similar interpretations are applicable to A and S Groups' chunks found using Factor AveR3 since each is identical to the respective group's chunk found using Factor R3.

Factor AveR3F: C Group's chunk found using Factor AveR3F comprises Sections "4.1 Overview", "4.2 Package and system decomposition", "4.4.2 UC5 - Submit to Archive" and "5. Process View". I Group's chunk comprises Sections “4.2 Package and system decomposition” and “4.4.2 UC5 - Submit to Archive”. S Group’s chunk comprises Sections “4.1 Overview”, “Logical High Level Solution Overview Diagram” and “5. Process View”. A Group’s chunk found using Factor AveR3F is identical to the group’s chunks found using Factors R3 and AveR3.

Interpretation: C Group’s chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, textual description of main logical components, packages of Java code, detailed description of ‘Submit to Archive’ use-case and process view diagram, were needed for WCT Task 2. I and S Groups’ chunks found using Factor AveR3F show that packages of Java code and detailed description of ‘Submit to Archive’ use-case were needed by I Group, and main logical components (both textual description and diagram) and process view diagram were needed by S Group, when attempting WCT Task 2. The interpretation for A Group’s chunk is the same as the interpretation given for the group’s chunk found using Factor R3.

12.6.2.2 Benchmarking Chunks Against An Oracle Set

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for WCT Task 2 are given at the bottom of Table 12.4. In Table 12.4, the sections that are part of the oracle set are denoted by yellow rows and their section IDs are marked with asterisks. This makes it easy to compare the chunks found against the oracle set.

Recall that chunks found using Factors R3 and AveR3 for a group of participants, are identical except for C Group where they are very similar.

For C Group, all the 3 chunks have ‘Average’ goodness measures. With equal precision measure but higher recall measure, Factor AveR3’s chunk (recall = 0.71; precision = 0.50, size = 20, goodness = ‘Average’) is better than Factor R3’s chunk (recall = 0.64; precision = 0.50, size = 18, goodness = ‘Average’). Factor AveR3F’s chunk has precision measure of 1.00 but low recall measure of 0.29. We assessed the compositions of Factor AveR3’s and Factor AveR3F’s chunks to decide the best chunk for C Group. Factor AveR3’s chunk is the best chunk for C Group. Our main reason is Factor AveR3’s chunk contains Section “6.3 Logical Deployment” but Factor AveR3F’s chunk does not. Section “6.3 Logical Deployment” is important for Part 2 of WCT Task 2. Both chunks excludes the most important Section “2.1.1 Modularity/Plugability” but includes the second important section which is Section “4.4.2 UC5 - Submit to Archive” for Part 1 of WCT Task 2.

For I Group, we compared the composition of the identical chunk found using Factors R3 and AveR3 (recall = 0.29; precision = 0.57, size = 7, goodness = ‘Poor’), with the composition of the chunk found using Factor AveR3F (recall = 0.14; precision = 1.00, size = 2, goodness = ‘Poor’) to decide the best chunk for I Group. Both the identical chunk and Factor AveR3F’s chunk miss Section “2.1.1 Modularity/Plugability” but includes Section “4.4.2 UC5 – Submit to Archive”. These two sections of the document are very important sections for WCT Task 2, especially for Part 1 of the task. Refer to Section “7.2.1 The Oracle Set for WCT Task 2” of Chapter 7 for further details. Both chunks miss the important section for Part 2 of WCT Task 2 (i.e. Section “6.3 Logical Deployment”). Other than that, the identical chunk has lower precision measure but includes more oracle set’s sections that contain information on how digital archive system fits into WCT, compared to Factor

AveR3F's chunk. In order to change the digital archive system and to assess the possible change impact as required by WCT Task 2, it is important to understand how digital archive system fits into WCT. Consequently, we chose Factor R3's (or AveR3's) chunk as the best chunk for I Group.

For S Group, there is only one chunk with the best level of goodness measure: the identical chunk found using Factors R3 and AveR3 with goodness measure of 'Average' (recall = 0.43; precision = 0.75, size = 8, goodness = 'Average'). It is therefore the best chunk for S Group. The chunk found using Factor AveR3F has goodness measure of 'Poor' (recall = 0.21; precision = 1.00, size = 3, goodness = 'Poor').

The chunk found using Factor AveR3F for each of the C, I and S Groups is the smallest chunk for the particular group, and therefore the most basic chunk for the respective group. The only chunk (recall = 0.86; precision = 0.46, size = 26, goodness = 'Average') found for A Group is the best chunk and the most basic chunk for A Group.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C Group is Factor AveR3, for I and S Groups is Factor R3 or Factor AveR3, for A Group is either 1 of the 3 factors (since they produce the same chunk).

Commonly-Missed Oracle Set's Sections and Common False Sections: We did not identify commonly-missed oracle set's sections and common false sections for A Group which has only one participant.

The oracle set's sections commonly missed by C Group are Sections "2.1.1 Modularity/Plugability", "3.2 Use Cases", "Deployment Diagram" and "Alternative Deployment Diagram".

The oracle set's sections commonly missed by I Group are Sections "2.1.1 Modularity/Plugability", "3.1 Actors", "3.2 Use Cases" and 3 of the 4 sections related to deployment (Sections "6.3 Logical Deployment", "Deployment Diagram" and "Alternative Deployment Diagram"). Sections "3. Use-Case View", "Logical High Level Solution Overview Diagram", "Process View Description" and "Logical Deployment Description" are excluded by all factors because of not being rated by any participant in I Group. Therefore, these sections do not qualify as commonly-missed oracle set's sections for I Group.

The oracle set's sections commonly missed by S Group are Sections "4.2 Package and system decomposition", "Deployment Diagram" and "Alternative Deployment Diagram". Sections "2.1.1 Modularity/Plugability", "3. Use-Case View", "3.1 Actors", "3.2 Use Cases", "Logical Deployment Description" are excluded by all factors because of not being rated by any participant in S group, and therefore do not qualify as commonly-missed oracle set's sections for S Group.

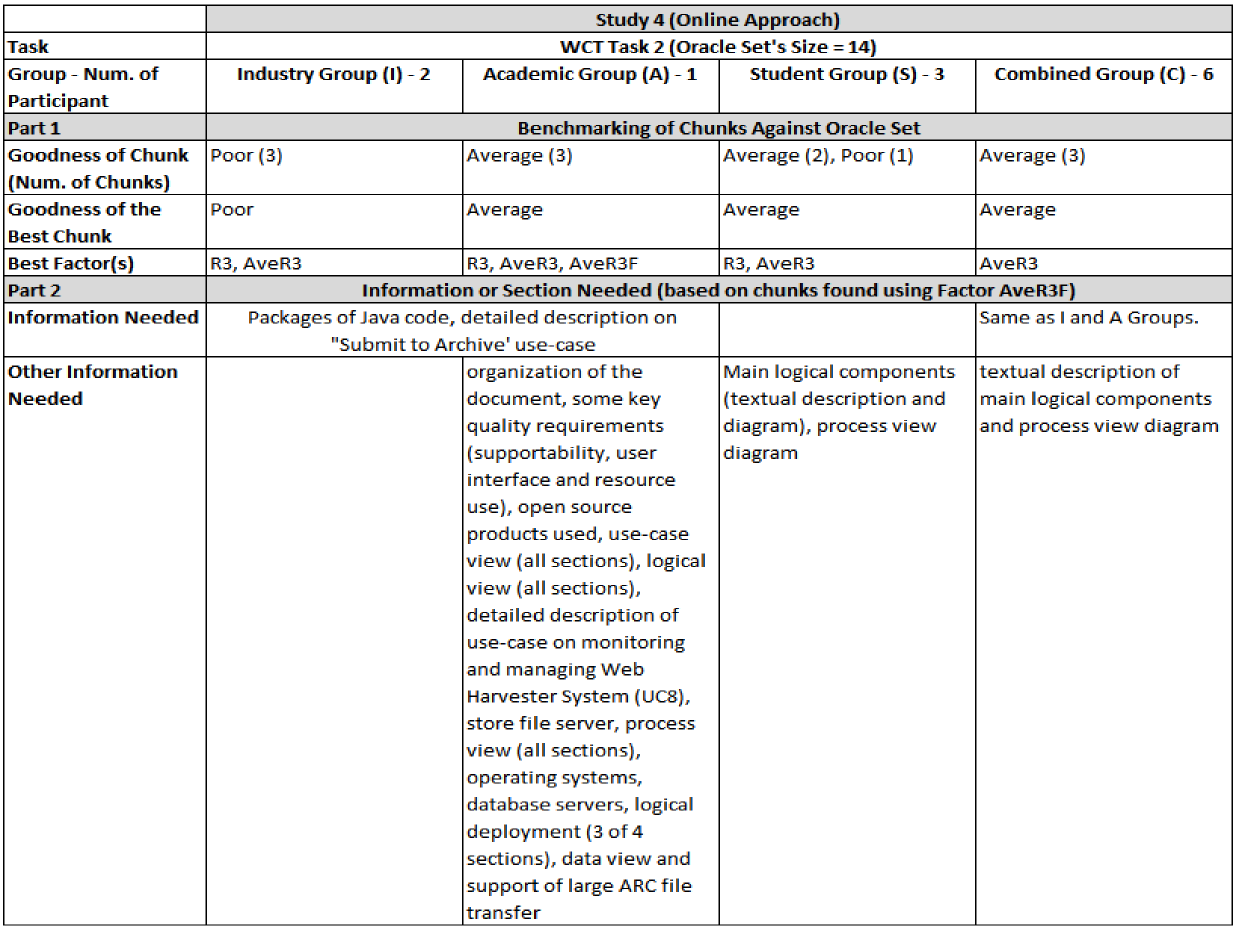
The reason for the common exclusion of Section "2.1.1 Modularity/Plugability" (the most important section) could be its title did not suggest to the participants that information related to digital archive system and subsequently the information required for WCT Task 2 could be found in the section. Consequently, the section might be skipped on the onset of the process of finding the needed information by the participants. We cannot think of any plausible reason for the common exclusion of other identified oracle set's sections by the respective group of participants, other than these sections were generally 'not needed' by the group when they attempted WCT Task 2.

There is no common false section for C, I and S Groups.

12.6.2.3 *Discussion of Chunk Discovery Results*

The results show the discovery of chunks for WCT Task 2 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. For WCT Task 2, chunks found using Factors R3 and AveR3 are identical for I, A and S Groups, and very similar for C Group.

Part 1 of Table 12.5 summarises the main results of benchmarking of chunks. There is only one participant in A Group. The number of participants in I and S Groups are quite balance (i.e. 2 and 3, respectively). In terms of how ‘good’ the chunks are for WCT Task 2, C Group has 3 ‘Average’ chunks. I Group has 3 ‘Poor’ chunks. A Group has 3 ‘Average’ identical chunks. S Group has 2 ‘Average’ chunks and 1 ‘Poor’ chunk.



	Study 4 (Online Approach)			
Task	**WCT Task 2 (Oracle Set's Size = 14)**			
Group - Num. of Participant	**Industry Group (I) - 2**	**Academic Group (A) - 1**	**Student Group (S) - 3**	**Combined Group (C) - 6**
Part 1	**Benchmarking of Chunks Against Oracle Set**			
Goodness of Chunk (Num. of Chunks)	Poor (3)	Average (3)	Average (2), Poor (1)	Average (3)
Goodness of the Best Chunk	Poor	Average	Average	Average
Best Factor(s)	R3, AveR3	R3, AveR3, AveR3F	R3, AveR3	AveR3
Part 2	**Information or Section Needed (based on chunks found using Factor AveR3F)**			
Information Needed	Packages of Java code, detailed description on "Submit to Archive' use-case			Same as I and A Groups.
Other Information Needed		organization of the document, some key quality requirements (supportability, user interface and resource use), open source products used, use-case view (all sections), logical view (all sections), detailed description of use-case on monitoring and managing Web Harvester System (UC8), store file server, process view (all sections), operating systems, database servers, logical deployment (3 of 4 sections), data view and support of large ARC file transfer	Main logical components (textual description and diagram), process view diagram	textual description of main logical components and process view diagram

Table 12.5: Summary of Results for WCT Task 2 (Study 3)

For WCT Task 2, the best chunk for C Group is found using Factor AveR3. Best chunks for I and S Groups are found using either Factor R3 or Factor AveR3. Best chunk for A Group is found using either 1 of the 3 factors (since they produce the same chunk). Following that, the best chunk-identification factor for C Group is

Factor AveR3, for each of I and S Groups is Factor R3 or Factor AveR3, for A Group is either 1 of the 3 factors.

Part 2 of Table 12.5 summarises the information or sections needed by each group of participants for WCT Task 2. This is based on the groups' chunks found using Factor AveR3F. We give our rationale for this at the beginning of Section "12.6 Chunking of Architectural Information" in this chapter.

For WCT Task 2, I and S Groups needed different information. I Group needed packages of Java code and detailed description of "Submit to Archive' use-case, and S Group needed main logical components (textual description and diagram) and process view diagram. The detailed description of "Submit to Archive' use-case (i.e. Section "4.4.2 UC5 – Submit to Archive" of WCT document) is one of the two very important sections for WCT Task 2, especially for Part 1 of the task. It contains specific information on what to do to use a different digital archive system. The packages of Java code contains code that needs to be considered to be changed to support different digital archive systems. This shows that I Group needed specific information related to Part 1 of WCT Task 2. On the other hand, S Group needed overview information (related to main logical components and process view), for WCT Task 2.

For WCT Task 2, we do not compare A Group with I and S Groups because the chunk found for A Group is based on only one academic participant. When all the participants are combined in C Group, the group needed textual description of main logical components, packages of Java code, detailed description of 'Submit to Archive' use-case and process view diagram for WCT Task 2.

12.6.3 WCT Task 3

Table 12.6 shows the chunks discovered for WCT Task 3, for industry practitioner (I), academic (A), student (S), and combined (C) groups of participants. The number of participants in I, A, S and C Groups are 3, 2, 3 and 8, respectively.

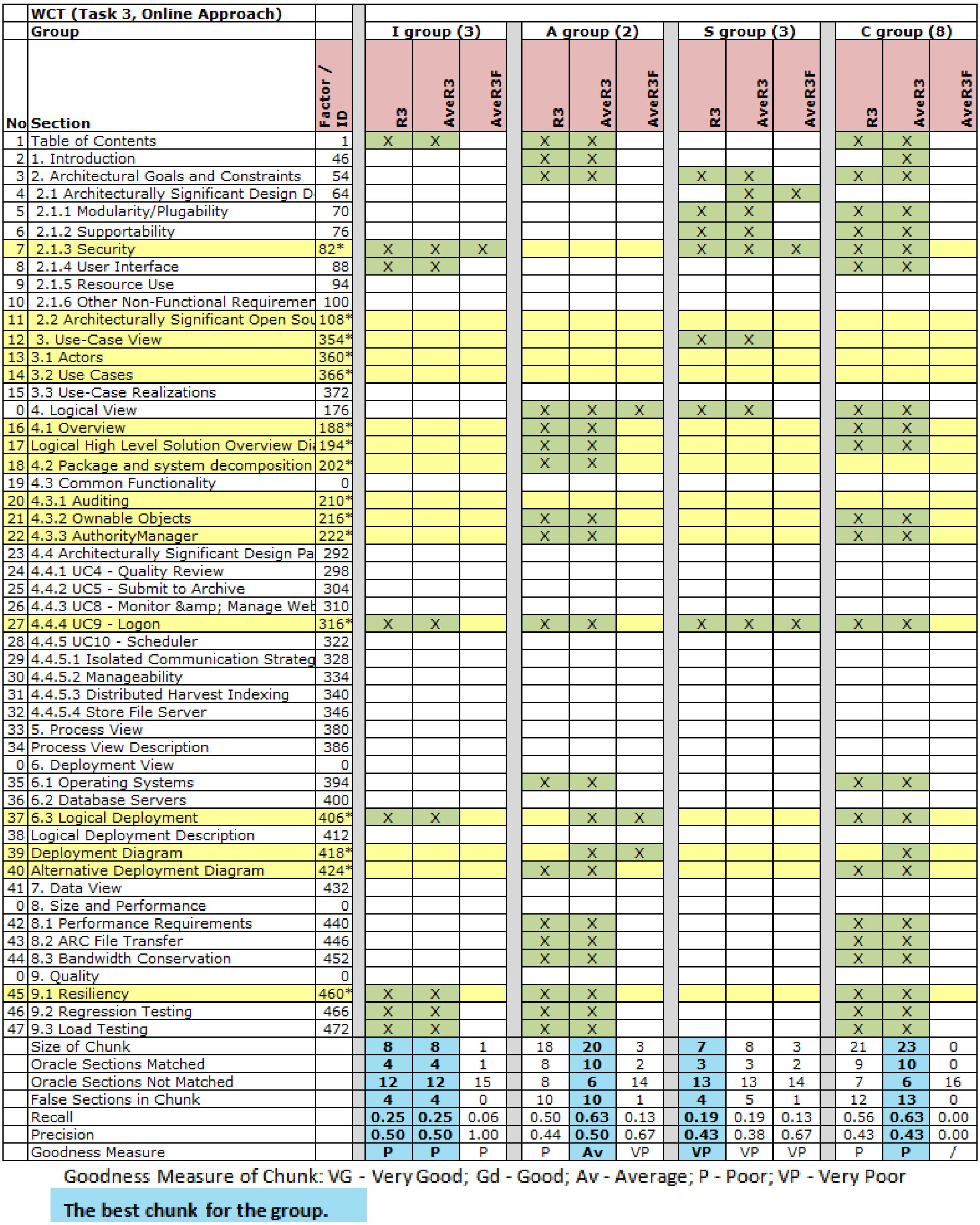
12.6.3.1 Identification and Interpretation of Chunks

Factor R3: C Group's chunk found using Factor R3 contains 21 sections. These sections are related to the organisation of the document (Section "Table of Contents", and Section "2. Architectural Goals and Constraints" containing hyperlinks to its two sub-sections), 4 of the 5 key quality requirements (Sections "2.1.1 Modularity/Plugability", "2.1.2 Supportability", "2.1.3 Security" and "2.1.4 User Interface"), brief overview of sub-sections related to logical view (Section "4. Logical View"), textual description of main logical components (Section "4.1 Overview"), high-level logical components diagram (Section "Logical High Level Solution Overview Diagram"), access control mechanism (Sections "4.3.2 Ownable Objects" and "4.3.3 AuthorityManager"), detailed description of 'logon' use-case (Section "4.4.4 UC9 - Logon"), operating systems (Section "6.1 Operating Systems"), logical deployment diagram (Section "6.3 Logical Deployment") and alternative deployment diagram (Section "Alternative deployment diagram").

C Group's chunk also includes all the 3 sections related to size and performance: performance requirements for login, report and page request (Section "8.1 Performance Requirements"), support of large ARC file transfer (Section "8.2 ARC File Transfer"), bandwidth conservation (Section "8.3 Bandwidth Conservation"). C

Group's chunk also includes all the 3 sections related to quality: system resiliency requirements (Section "9.1 Resiliency"), and testing (Sections "9.2 Regression Testing" and "9.3 Load Testing").

The chunk found using Factor R3 for I Group comprises 8 sections. These sections are related to Table of Contents, two key quality requirements which are security (Section "2.1.3 Security") and user interface (Section "2.1.4 User Interface"), detailed description of 'logon' use-case, logical deployment diagram, resiliency, and testing.



	WCT (Task 3, Online Approach)													
	Group		I group (3)			A group (2)			S group (3)			C group (8)		
No	Section	Factor / ID	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F
1	Table of Contents	1	X	X		X	X					X	X	
2	1. Introduction	46				X	X						X	
3	2. Architectural Goals and Constraints	54				X	X		X	X		X	X	
4	2.1 Architecturally Significant Design D	64								X	X			
5	2.1.1 Modularity/Plugability	70							X	X		X	X	
6	2.1.2 Supportability	76							X	X		X	X	
7	2.1.3 Security	82*	X	X	X				X	X	X	X	X	
8	2.1.4 User Interface	88	X	X								X	X	
9	2.1.5 Resource Use	94												
10	2.1.6 Other Non-Functional Requiremen	100												
11	2.2 Architecturally Significant Open Sou	108*												
12	3. Use-Case View	354*							X	X				
13	3.1 Actors	360*												
14	3.2 Use Cases	366*												
15	3.3 Use-Case Realizations	372												
0	4. Logical View	176				X	X	X	X	X		X	X	
16	4.1 Overview	188*				X	X					X	X	
17	Logical High Level Solution Overview Di	194*				X	X					X	X	
18	4.2 Package and system decomposition	202*				X	X							
19	4.3 Common Functionality	0												
20	4.3.1 Auditing	210*												
21	4.3.2 Ownable Objects	216*				X	X					X	X	
22	4.3.3 AuthorityManager	222*				X	X					X	X	
23	4.4 Architecturally Significant Design Pa	292												
24	4.4.1 UC4 - Quality Review	298												
25	4.4.2 UC5 - Submit to Archive	304												
26	4.4.3 UC8 - Monitor &amp; Manage Wel	310												
27	4.4.4 UC9 - Logon	316*	X	X		X	X		X	X	X	X	X	
28	4.4.5 UC10 - Scheduler	322												
29	4.4.5.1 Isolated Communication Strateg	328												
30	4.4.5.2 Manageability	334												
31	4.4.5.3 Distributed Harvest Indexing	340												
32	4.4.5.4 Store File Server	346												
33	5. Process View	380												
34	Process View Description	386												
0	6. Deployment View	0												
35	6.1 Operating Systems	394				X	X					X	X	
36	6.2 Database Servers	400												
37	6.3 Logical Deployment	406*	X	X			X	X				X	X	
38	Logical Deployment Description	412												
39	Deployment Diagram	418*					X	X					X	
40	Alternative Deployment Diagram	424*				X	X					X	X	
41	7. Data View	432												
0	8. Size and Performance	0												
42	8.1 Performance Requirements	440				X	X					X	X	
43	8.2 ARC File Transfer	446				X	X					X	X	
44	8.3 Bandwidth Conservation	452				X	X					X	X	
0	9. Quality	0												
45	9.1 Resiliency	460*	X	X		X	X					X	X	
46	9.2 Regression Testing	466	X	X		X	X					X	X	
47	9.3 Load Testing	472	X	X		X	X					X	X	
	Size of Chunk		**8**	**8**	1	18	**20**	3	**7**	8	3	21	**23**	0
	Oracle Sections Matched		**4**	**4**	1	8	**10**	2	**3**	3	2	9	**10**	0
	Oracle Sections Not Matched		**12**	**12**	15	8	**6**	14	**13**	13	14	7	**6**	16
	False Sections in Chunk		**4**	**4**	0	10	**10**	1	**4**	5	1	12	**13**	0
	Recall		**0.25**	**0.25**	0.06	0.50	**0.63**	0.13	**0.19**	0.19	0.13	0.56	**0.63**	0.00
	Precision		**0.50**	**0.50**	1.00	0.44	**0.50**	0.67	**0.43**	0.38	0.67	0.43	**0.43**	0.00
	Goodness Measure		**P**	**P**	P	P	**Av**	VP	**VP**	VP	VP	P	**P**	/

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 12.6: The Chunks for WCT Task 3 Identified Using Different Factors (Study 3)

The chunk found using Factor R3 for A Group includes many of the sections in C Group’s chunk. It comprises 18 sections. These sections are related to the organisation of the document (Section “Table of Contents”, and Section “2. Architectural Goals and Constraints” that contains hyperlinks to its two sub-sections), overview of the document and WCT (Section “1. Introduction”), all the 3 sections on logical view (Sections “4. Logical View”, “4.1 Overview”, “Logical High Level Solution Overview Diagram”), packages of Java code (Section “4.2 Package and system decomposition”), access control mechanism, detailed description of ‘logon’ use-case, operating systems, alternative deployment diagram, all the 3 sections related to size and performance, and all the 3 sections related to quality.

S Group’s chunk found using Factor R3 comprises 7 sections. These sections are related to the organisation of the document (Section “2. Architectural Goals and Constraints), key quality requirements (Sections “2.1.1 Modularity/Plugability”, “2.1.2 Supportability” and “2.1.3 Security”), use-case diagram (Section “3. Use-Case View”), brief overview of sub-sections related to logical view and detailed description of ‘logon’ use-case.

Interpretation: All participants provided ratings for some sections of WCT document in terms of the importance of these sections to answering WCT Task 3.

C Group’s chunk found using Factor R3 shows that for WCT Task 3, the sections needed by the majority of C Group who rated the respective sections are related to the organisation of the document, key quality requirements (modularity/plugability, supportability, security and user interface), logical view (all the sections on this topic), access control mechanism, detailed description of ‘logon’ use-case, operating systems, logical deployment diagram, alternative deployment diagram, size and performance, system resiliency requirements, and testing.

I Group’s chunk found using Factor R3 shows that for WCT Task 3, the sections needed by the majority of I Group who rated the respective sections are related to Table of Contents, key quality requirements (security and user interface), detailed description of ‘logon’ use-case, logical deployment diagram, resiliency, and testing.

A Group’s chunk shows that for WCT Task 3, the sections needed by the majority of A Group who rated the respective sections are related to the organisation of the document, overview of the document and WCT, logical view (all the sections on this topic), packages of Java code, access control mechanism, detailed description of ‘logon’ use-case, operating systems, alternative deployment diagram, size and performance, system resiliency requirements, and testing.

S Group’s chunk shows that for WCT Task 3, the sections needed by the majority of S Group who rated the respective sections are related to the organisation of the document, key quality requirements (modularity/plugability”, supportability and security), use-case diagram, brief overview of sub-sections related to logical view and detailed description of ‘logon’ use-case.

Factor AveR3: Chunks found using Factors AveR3 and R3 for a particular group of participants are very similar, except for I Group where they are identical. Chunks found using Factor AveR3 for C, A and S Groups have the following additional sections: Sections “1. Introduction” and “Deployment Diagram” for C Group’s chunk, Sections “6.3 Logical Deployment” and “Deployment Diagram” for A Group’s chunk, Section “2.1 Architecturally Significant Design Decision” for S Group’s chunk.

Interpretation: In addition to the sections in C Group’s chunk found using Factor R3, sections on overview of document and WCT, and deployment diagram were needed by some of the participants who rated the respective sections. In addition to the sections in A Group’s chunk found using Factor R3, sections on logical deployment diagram, and deployment diagram were needed by some of the academic participants who rated the respective sections. In addition to the sections in S Group’s chunk found using Factor R3, section on categories of key quality requirements was needed by some of the student participants who rated the respective section.

Factor AveR3F: Chunks found for each group of participants using Factor AveR3F are very small, with no chunk found for C Group. I Group’s chunk comprises only Section “2.1.3 Security”. A Group’s chunk comprises Sections “4. Logical View”, “6.3 Logical Deployment” and “Deployment Diagram”. S Group’s chunk comprises Sections “Architecturally Significant Design Decisions”, “2.1.3 Security” and “4.4.4 UC9 – Logon”.

Interpretation: C Group’ chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, Section “2.1.3 Security” (security as a key quality requirement together with the overview of Acegi Security Framework) was needed for WCT Task 3. A and S Groups’ chunks found using Factor AveR3F show that, brief overview of sub-sections related to logical view, logical deployment diagram and deployment diagram were needed by A Group, and the categories of the key quality requirements, Section “2.1.3 Security” and detailed description of ‘logon’ use-case were needed by S Group.

12.6.3.2 Benchmarking Chunks Against An Oracle Set

For the concept of ‘Security’ involved in WCT Task 3, refer to Section “8.2.1 What is Security?” of Chapter 8.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for WCT Task 3 are given at the bottom of Table 12.6. In Table 12.6, the sections that are part of the oracle set are denoted by yellow rows and their section IDs are marked with asterisks. This makes it easy to compare the chunks found against the oracle set.

Recall that chunks found using Factors AveR3 and R3 for a particular group of participants, are very similar except for I Group where they are identical.

For C Group, no chunk is found using Factor AveR3F. With equal precision but higher recall measures, Factor AveR3’s chunk (recall = 0.63; precision = 0.43, size = 23, goodness = ‘Poor’) is better than Factor R3’s chunk (recall = 0.56; precision = 0.43, size = 21, goodness = ‘Poor’) and therefore the best chunk for C Group. Factor R3’s chunk is the most basic chunk for C Group since it is the smallest chunk.

All the 3 chunks found for I Group have ‘Poor’ goodness measures, and Factor R3’s and Factor AveR3’s chunks are identical. Factor AveR3F’s chunk has precision measure of 1.00 (recall = 0.06; precision = 1.00, size = 1, goodness = ‘Poor’). However, it includes only 1 oracle set’s section, which describes the background information on the technologies used to achieve security for WCT. It misses all the 13 oracle set’s sections that contain specific information for WCT Task 3. Refer to Section “8.2.2 The Oracle Set for WCT Task 3” of Chapter 8 for further details on the oracle set for WCT Task 3. The identical chunk found using Factors R3 and AveR3 (recall = 0.25; precision = 0.50, size = 8, goodness = ‘Poor’) contains the section included by Factor AveR3F’s chunk. The identical chunk also includes 3 oracle set’s sections that contain specific information for WCT Task 3. Consequently, Factor R3’s or Factor AveR3’s chunk is the best chunk for I Group.

For A Group, Factor AveR3’s chunk (recall = 0.63; precision = 0.50, size = 20, goodness = ‘Average’) has better level of goodness measure compared to Factor R3’s chunk (recall = 0.50; precision = 0.44, size = 18, goodness = ‘Poor’) and Factor AveR3F’s chunk (recall = 0.13; precision = 0.67, size = 3, goodness = ‘Very Poor’). Factor AveR3’s chunk is therefore the best chunk for A Group.

All the 3 chunks found for S Group have ‘Very Poor’ goodness measures. With equal recall but higher precision, Factor R3’s chunk (recall = 0.19; precision = 0.43, size = 7, goodness = ‘Very Poor’) is better than Factor AveR3’s chunk (recall = 0.19; precision = 0.38, size = 8, goodness = ‘Very Poor’). Factor AveR3F’s chunk (recall = 0.13; precision = 0.67, size = 3, goodness = ‘Very Poor’) has higher precision measure but slightly lower recall measure than Factor R3’s chunk. The compositions of Factor R3’s and Factor AveR3F’s chunks show that both chunks include Sections “2.1.3 Security” and “4.4.4 UC9 – Logon”. Factor R3’s chunk includes an extra oracle set’s section (Section “3. Use-Case View’) on use-case diagram but includes 4 false sections, whereas Factor AveR3F’s chunk includes only 1 false section. We choose Factor R3’s chunk as the best chunk for S Group because the use-case diagram contains information about all the actors associated with ‘logon’ use-case, which is absented from Factor AveR3F’s chunk.

The chunk found using Factor AveR3F for each of I, A and S Groups is the smallest chunk for the particular group, and therefore the most basic chunk for the respective group.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C and A Groups is Factor AveR3, and for S Group is Factor R3. For I Group, the best factor is either Factor R3 or Factor AveR3.

Commonly-Missed Oracle Set’s Sections and Common False Sections: The oracle set’s sections commonly missed by C Group are Section “2.2 Architecturally Significant Open Source Products”, 3 sections on use-case view (Sections “3. Use-Case View”, “3.1 Actors”, “3.2 Use Cases”) and Section “4.2 Package and system decomposition”. The oracle set’s sections commonly missed by I Group is Section “2.2 Architecturally Significant Open Source Products”. The oracle set’s sections commonly missed by A Group are the 3 sections on use-case view (Sections “3. Use-Case View”, “3.1 Actors”, “3.2 Use Cases”). The oracle set’s sections commonly missed by S Group are Sections “2.2 Architecturally Significant Open Source Products” and “4.2 Package and system decomposition”. Other oracle set’s sections that are excluded by all chunk-identification factors for a particular group of participants do not qualify as commonly-missed oracle set’s sections for the group, because of not being rated by any of the participants in the group.

Section “2.2 Architecturally Significant Open Source Products” explains the open source products used in WCT. It contains information about Acegi Security Framework that complements the information in Section “2.1.3 Security”. Section “2.2 Architecturally Significant Open Source Products” is missed by all the chunks found for WCT Task 3 (including ‘no chunk’ cases) probably because most of the information about Acegi Security Framework can be found in Section “2.1.3 Security” or the participants have background knowledge on this framework. Section “4.2 Package and system decomposition” is commonly missed probably because Java package for authentication of users, could be too low level to be considered architectural for some participants. We cannot think of any plausible reason for the common exclusion of other oracle set’s sections by a particular group of participants, other than these sections were generally ‘not needed’ by the group when attempting WCT Task 3.

There is no common false section for C, I and S Groups. The common false section for A Group is Section “4. Logical View” (which contains brief description of its sub-sections and hyperlinks to them), which was probably useful for A Group to navigate to its sub-sections.

12.6.3.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for WCT Task 3 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. For WCT Task 3, chunks found using Factors R3 and AveR3 are identical for I Group and very similar for A, S and C Groups.

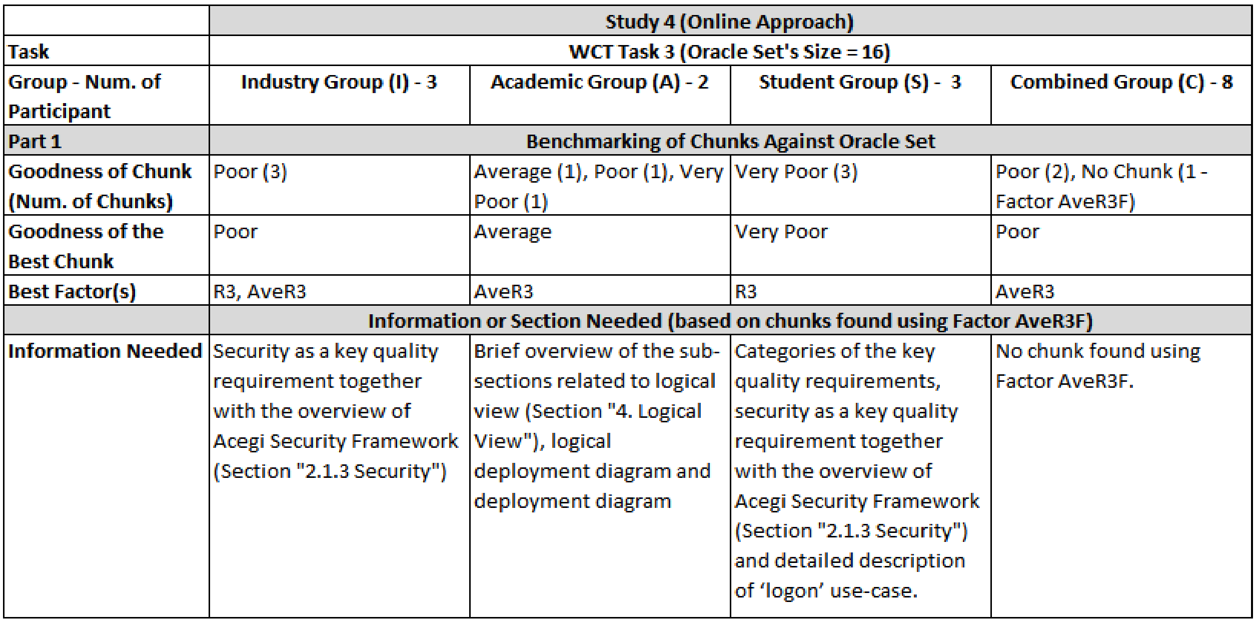
Part 1 of Table 12.7 summarises the main results of benchmarking of chunks. The number of participants in I, A and S Groups are quite balance, which is 3, 2 and 3, respectively. In terms of how ‘good’ the chunks are for WCT Task 3, C Group has 2 ‘Poor’ chunks. No chunk is found for C Group using Factor AveR3F. I Group has 3 ‘Poor’ chunks. A Group has one each of ‘Average’, ‘Poor’ and ‘Very Poor’ chunk. S Group has 3 ‘Very Poor’ chunks.

For WCT Task 3, the best chunk for C and A Groups are found using Factor AveR3. The best chunk for S Group is found using Factor R3. For I Group, the best chunk is found using Factor R3 or Factor AveR3. Following that, the best chunk-identification factor for C and A Groups is Factor AveR3, and for S Group is Factor R3. For I Group, the best factor is either Factor R3 or Factor AveR3.

Part 2 of Table 12.7 summarises the information or sections needed by each group of participants for WCT Task 3. This is based on the groups’ chunks found using Factor AveR3F. We give our rationale for this at the beginning of Section “12.6 Chunking of Architectural Information” in this chapter.

For WCT Task 3, the three sub-groups of participants needed different information: I Group needed the section that describes security as a key quality requirement together with the overview of Acegi Security Framework (Section “2.1.3 Security”); S Group needed categories of the key quality requirements, security as a key quality requirement together with the overview of Acegi Security Framework (Section “2.1.3 Security”) and detailed description of ‘logon’ use-case (Section “4.4.4 UC9 – Logon”); A Group needed brief overview of sub-sections related to logical view, logical deployment diagram and deployment diagram.

The detailed description of ‘logon’ use-case contains detailed and specific information about how authentication and authorisation are achieved in WCT, passwords hashing in database and session timeout. It contains much of the information needed for WCT Task 3. The logical deployment diagram shows the logical mapping of components to the nodes for WCT. It shows the interaction of ‘Authentication’ component with other nodes in the system to support the ‘logon’ use-case. The deployment diagram shows the deployment of ‘Authentication’ component in WCT.



	Study 4 (Online Approach)			
Task	**WCT Task 3 (Oracle Set's Size = 16)**			
Group - Num. of Participant	**Industry Group (I) - 3**	**Academic Group (A) - 2**	**Student Group (S) - 3**	**Combined Group (C) - 8**
Part 1	**Benchmarking of Chunks Against Oracle Set**			
Goodness of Chunk (Num. of Chunks)	Poor (3)	Average (1), Poor (1), Very Poor (1)	Very Poor (3)	Poor (2), No Chunk (1 - Factor AveR3F)
Goodness of the Best Chunk	Poor	Average	Very Poor	Poor
Best Factor(s)	R3, AveR3	AveR3	R3	AveR3
	Information or Section Needed (based on chunks found using Factor AveR3F)			
Information Needed	Security as a key quality requirement together with the overview of Acegi Security Framework (Section "2.1.3 Security")	Brief overview of the sub-sections related to logical view (Section "4. Logical View"), logical deployment diagram and deployment diagram	Categories of the key quality requirements, security as a key quality requirement together with the overview of Acegi Security Framework (Section "2.1.3 Security") and detailed description of ‘logon’ use-case.	No chunk found using Factor AveR3F.

Table 12.7: Summary of Results for WCT Task 3 (Study 3)

The comparison between the sub-groups for WCT Task 3 shows that both I and S Groups needed background information of Acegi Security Framework. In addition, S Group needed categories of the key quality requirements, and detailed and specific information about how security is achieved in WCT. On the other hand, A Group needed deployment aspect of the system, when attempting WCT Task 3. When all the participants are combined in C Group, no chunk is found using Factor AveR3F. This shows that none of the sections (or information) of WCT software architecture document stands out as needed by the group as a whole, when the group attempted WCT Task 3.

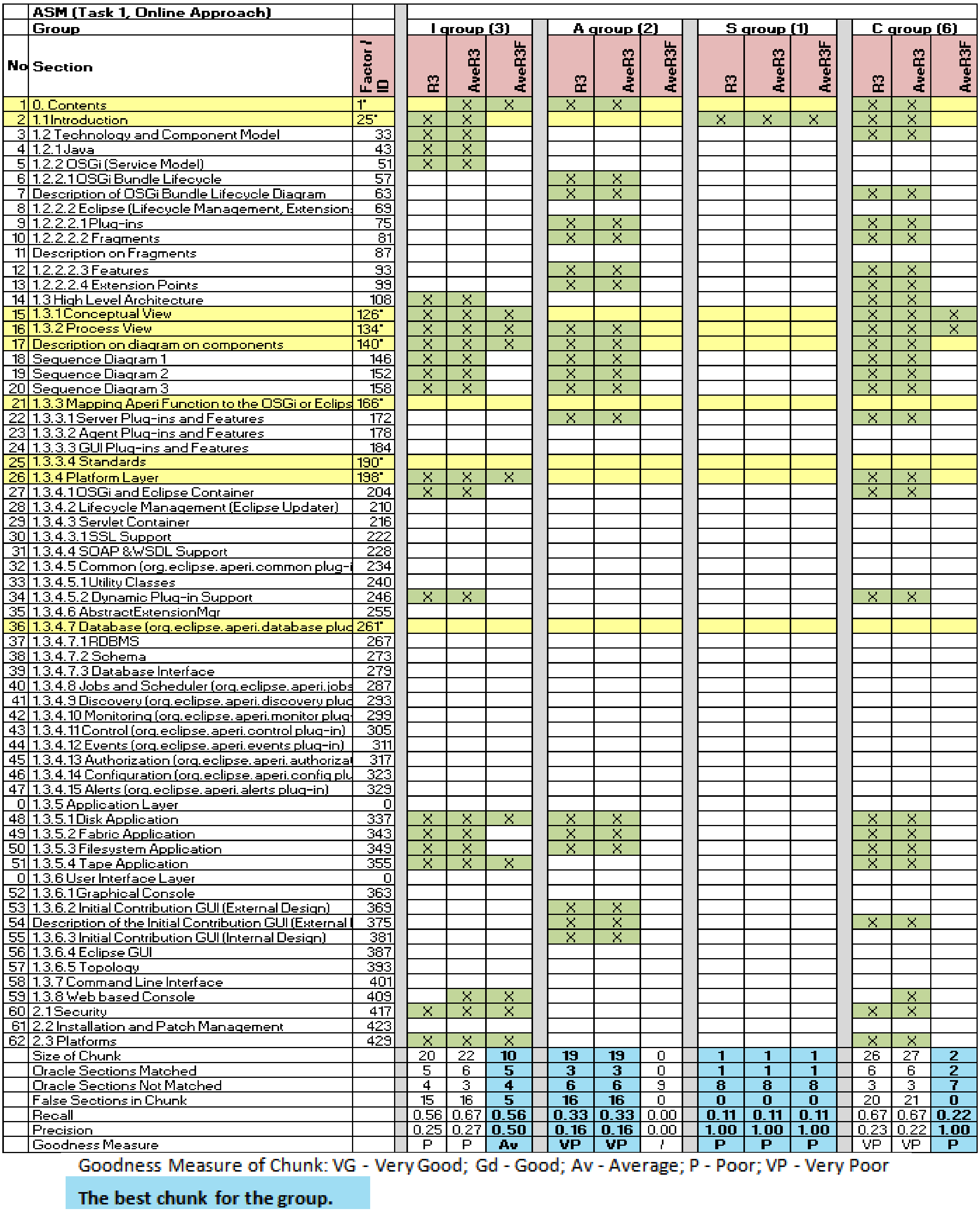
12.6.4 ASM Task 1

Table 12.8 shows the chunks discovered for ASM Task 1, for industry practitioner (I), academic (A), student (S), and combined (C) groups of participants. The number of participants in I, A, S and C Groups are 3, 2, 1 and 6, respectively.

12.6.4.1 Identification and Interpretation of Chunks

Factor R3: C Group’s chunk found using Factor R3 contains 26 sections. These sections are related to Table of Contents (Section “0. Contents”), overview of ASM (Section “1.1 Introduction”), Section “1.2 Technology and Component Model” (which contains hyperlinks to its two sub-sections), textual description of OSGi Bundle Lifecycle (Section “Description of OSGi Bundle Lifecycle Diagram”), low-level detailed information on plug-ins (Section “1.2.2.2.1 Plug-ins”), brief description of plug-in fragment together with a diagram (Section “1.2.2.2.2 Fragments”), description of feature as a group of plug-ins that define a logical product feature (Section

"1.2.2.2.3 Features"), detailed explanation on how plug-in can be extended through 'extension points' with some example code (Section "1.2.2.2.4 Extension Points"), Section "1.3 High Level Architecture" which contains only hyperlinks to its sub-sections.



	ASM (Task 1, Online Approach)													
	Group		I group (3)			A group (2)			S group (1)			C group (6)		
No	Section	Factor / ID	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F
1	0. Contents	1*		X	X	X	X					X	X	
2	1.1 Introduction	25*	X	X					X	X	X	X	X	
3	1.2 Technology and Component Model	33	X	X								X	X	
4	1.2.1 Java	43	X	X										
5	1.2.2 OSGi (Service Model)	51	X	X										
6	1.2.2.1 OSGi Bundle Lifecycle	57				X	X							
7	Description of OSGi Bundle Lifecycle Diagram	63				X	X					X	X	
8	1.2.2.2 Eclipse (Lifecycle Management, Extension:	69												
9	1.2.2.2.1 Plug-ins	75				X	X					X	X	
10	1.2.2.2.2 Fragments	81				X	X					X	X	
11	Description on Fragments	87												
12	1.2.2.2.3 Features	93				X	X					X	X	
13	1.2.2.2.4 Extension Points	99				X	X					X	X	
14	1.3 High Level Architecture	108	X	X								X	X	
15	1.3.1 Conceptual View	126*	X	X	X							X	X	X
16	1.3.2 Process View	134*	X	X	X	X	X					X	X	X
17	Description on diagram on components	140*	X	X	X	X	X					X	X	
18	Sequence Diagram 1	146	X	X		X	X					X	X	
19	Sequence Diagram 2	152	X	X		X	X					X	X	
20	Sequence Diagram 3	158	X	X		X	X					X	X	
21	1.3.3 Mapping Aperi Function to the OSGi or Eclips	166*												
22	1.3.3.1 Server Plug-ins and Features	172				X	X					X	X	
23	1.3.3.2 Agent Plug-ins and Features	178												
24	1.3.3.3 GUI Plug-ins and Features	184												
25	1.3.3.4 Standards	190*												
26	1.3.4 Platform Layer	198*	X	X	X							X	X	
27	1.3.4.1 OSGi and Eclipse Container	204	X	X								X	X	
28	1.3.4.2 Lifecycle Management (Eclipse Updater)	210												
29	1.3.4.3 Servlet Container	216												
30	1.3.4.3.1 SSL Support	222												
31	1.3.4.4 SOAP &WSDL Support	228												
32	1.3.4.5 Common (org.eclipse.aperi.common plug-i	234												
33	1.3.4.5.1 Utility Classes	240												
34	1.3.4.5.2 Dynamic Plug-in Support	246	X	X								X	X	
35	1.3.4.6 AbstractExtensionMgr	255												
36	1.3.4.7 Database (org.eclipse.aperi.database pluc	261*												
37	1.3.4.7.1 RDBMS	267												
38	1.3.4.7.2 Schema	273												
39	1.3.4.7.3 Database Interface	279												
40	1.3.4.8 Jobs and Scheduler (org.eclipse.aperi.jobs	287												
41	1.3.4.9 Discovery (org.eclipse.aperi.discovery pluc	293												
42	1.3.4.10 Monitoring (org.eclipse.aperi.monitor plug-	299												
43	1.3.4.11 Control (org.eclipse.aperi.control plug-in)	305												
44	1.3.4.12 Events (org.eclipse.aperi.events plug-in)	311												
45	1.3.4.13 Authorization (org.eclipse.aperi.authorizat	317												
46	1.3.4.14 Configuration (org.eclipse.aperi.config plu	323												
47	1.3.4.15 Alerts (org.eclipse.aperi.alerts plug-in)	329												
0	1.3.5 Application Layer	0												
48	1.3.5.1 Disk Application	337	X	X	X	X	X					X	X	
49	1.3.5.2 Fabric Application	343	X	X		X	X					X	X	
50	1.3.5.3 Filesystem Application	349	X	X		X	X					X	X	
51	1.3.5.4 Tape Application	355	X	X	X							X	X	
0	1.3.6 User Interface Layer	0												
52	1.3.6.1 Graphical Console	363												
53	1.3.6.2 Initial Contribution GUI (External Design)	369				X	X							
54	Description of the Initial Contribution GUI (External I	375				X	X					X	X	
55	1.3.6.3 Initial Contribution GUI (Internal Design)	381				X	X							
56	1.3.6.4 Eclipse GUI	387												
57	1.3.6.5 Topology	393												
58	1.3.7 Command Line Interface	401												
59	1.3.8 Web based Console	409		X	X								X	
60	2.1 Security	417	X	X	X							X	X	
61	2.2 Installation and Patch Management	423												
62	2.3 Platforms	429	X	X	X							X	X	
	Size of Chunk		20	22	**10**	**19**	**19**	0	**1**	**1**	**1**	26	27	**2**
	Oracle Sections Matched		5	6	**5**	**3**	**3**	0	**1**	**1**	**1**	6	6	**2**
	Oracle Sections Not Matched		4	3	**4**	**6**	**6**	9	**8**	**8**	**8**	3	3	**7**
	False Sections in Chunk		15	16	**5**	**16**	**16**	0	**0**	**0**	**0**	20	21	**0**
	Recall		0.56	0.67	**0.56**	**0.33**	**0.33**	0.00	**0.11**	**0.11**	**0.11**	0.67	0.67	**0.22**
	Precision		0.25	0.27	**0.50**	**0.16**	**0.16**	0.00	**1.00**	**1.00**	**1.00**	0.23	0.22	**1.00**
	Goodness Measure		P	P	**Av**	**VP**	**VP**	/	**P**	**P**	**P**	VP	VP	**P**

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 12.8: The Chunks for ASM Task 1 Identified Using Different Factors (Study 3)

C Group’s chunk also contains overview of conceptual view (Section 1.3.1 Conceptual View”), sections related to process view (Sections “1.3.2 Process View”, “Description on diagram on components”), the 3 sequence diagrams, Java packages related to server plug-ins and features (Section “1.3.3.1 Server Plug-in and Features”), brief overview of platform layer (Section “1.3.4 Platform Layer”), current base runtime container used (Section “1.3.4.1 OSGi and Eclipse Container”), dynamic plug-in support (Section 1.3.4.5.2), the 4 exemplary applications in application layer of ASM (Section 1.3.5.1 to Section 1.3.5.4), Section “Description of the Initial Contribution GUI (External Design) diagram” (explanation on the tree control nodes in the initial Graphical User Interface), security (Section 2.1), and initial development platforms (Section 2.3).

I Group’s chunk (comprising 20 sections) found using Factor R3 contains many of the sections in C Group’s chunk. However, I Group’s chunk excludes Sections “0. Contents”, “Description of OSGi Bundle Lifecycle Diagram”, “1.2.2.2.1 Plug-ins”, “1.2.2.2.2 Fragments”, “1.2.2.2.3 Features”, “1.2.2.2.4 Extension Points”, “1.3.3.1 Server Plug-in and Features” and “Description of the Initial Contribution GUI (External Design) diagram”. I Group’s chunk also includes section on the development languages (Section “1.2.1 Java”) and overview of OSGi service model (Section “1.2.2”).

A Group’s chunk found using Factor R3 contains 19 sections: “0. Contents”, both sections on OSGi Bundle Lifecycle (Section 1.2.2.1 and Section “Description of OSGi Bundle Lifecycle Diagram”), sections that contain more detailed information on Eclipse plug-in model which are excluded by I Group’s chunk (i.e. Sections “1.2.2.2.1 Plug-ins”, “1.2.2.2.2 Fragments”, “1.2.2.2.3 Features” and “1.2.2.2.4 Extension Points”), sections related to process view (Sections “1.3.2 Process View”, “Description on diagram on components”), and the 3 sequence diagrams, Section “1.3.3.1 Server Plug-in and Features”, 3 of the exemplary applications in application layer of ASM (Section 1.3.5.1 to Section 1.3.5.3), Section “1.3.6.2 Initial Contribution GUI (External Design)” that contains brief description and diagram on the initial Graphical User Interface (GUI), Section “Description of the Initial Contribution GUI (External Design) diagram”, Section “1.3.6.3 Initial Contribution GUI (Internal Design)” that explains the decisions related to internal design of GUI.

S Group’s chunk contains only section related to the overview of ASM (Section “1.1 Introduction”).

Interpretation: All participants provided ratings for some sections of ASM document in terms of the importance of these sections to answering ASM Task 1.

C Group’s chunk found using Factor R3 shows that sections needed by the majority of C Group who rated the respective sections, are mainly related to the organisation of the document (Sections “0. Contents”, “1.2 Technology and Component Model”, “1.3 High Level Architecture”), overview information (on ASM, conceptual view, process view, platform layer), some detailed information related to certain topics (OSGi service model, Eclipse plug-in model, process view), and exemplary applications in the Application Layer of ASM.

I and A Group’s chunks found using Factor R3 show that the majority of both groups who rated the respective sections needed comprehensive information on the process view (by requiring all the 5 sections on the process view which include the overview information and lower-level information in terms of sequence diagrams). They also needed 3 of the 4 exemplary applications. Other sections needed by the majority of I Group who rated the respective sections are mainly related to the organisation of the document (Sections “1.2 Technology and Component Model” and “1.3 High Level Architecture”) and overview information (on ASM, OSGi service model, conceptual view, platform layer). On the other hand, other sections needed by the majority of A Group who rated the respective sections are mainly related to the more detailed information on the underlying models used (OSGi Bundle Lifecycle and the Eclipse plug-in model) and the initial GUI.

S Group's chunk shows that the one student in this group only needed the overview of ASM (Section "1.1 Introduction") for ASM Task 1.

Factor AveR3: C Group's chunk found using Factor AveR3 is very similar to the group's chunk found using Factor R3 with the addition of Section "1.3.8 Web based Console" (possible development of alternative user interface). I Group's chunk found using Factor AveR3 is very similar to the group's chunk found using Factor R3 with the addition of Sections "0. Contents" and "1.3.8 Web based Console".

S Group's chunk found using Factor AveR3 is identical to the group's chunks found using Factors R3 and AveR3F. This is non-surprising since there is only one participant in S Group.

Interpretation: In addition to those sections presented in the respective group's chunk found using Factor R3, some of C Group's and I Group's participants also needed information about the possible development of alternative user interface, for ASM Task 1. Some of the industry participants also needed Table of Contents for ASM Task 1.

The interpretation for S Group's chunk is the same as the interpretation given for the group's chunk found using Factor R3.

Factor AveR3F: A small chunk comprising Sections "1.3.1 Conceptual View" and "1.3.2 Process View" is found for C Group using Factor AveR3F. I Group's chunk found using Factor AveR3F comprises Sections "0. Contents", "1.3.1 Conceptual View", "1.3.2 Process View", "Description on diagram on components", "1.3.4 Platform layer", "1.3.5.1 Disk Application", "1.3.5.4 Tape Application", "1.3.8 Web based Console", "2.1 Security" and "2.3 Platforms". No chunk is found for A Group. S Group's chunk is identical to the group's chunk found using Factors R3 and AveR3F.

Interpretation: C Group's chunk found using Factor AveR3F shows that based on the average ratings contributed by the majority of the participants, the overview of the conceptual view and overview of the process view were needed for ASM Task 1. I Group's chunk found using Factor AveR3F shows that, information related to Table of Contents, some overview information (conceptual view, process view and platform layer), additional textual description of the process view, some exemplary applications in the application layer of ASM, possible development of alternative user interface, security and initial development platforms were needed by I Group when attempting ASM Task 1. It is surprising that the section on security which does not contain any valuable information was needed by I Group.

The interpretation for S Group's chunk is the same as the interpretation given for the group's chunk found using Factor R3.

12.6.4.2 Benchmarking Chunks Against An Oracle Set

For the concept of ‘Software Architecture’ involved in ASM Task 1, refer to Section “6.2.1 What is Software Architecture?” of Chapter 6.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for ASM Task 1 are given at the bottom of Table 12.8. In Table 12.8, the sections that are part of the oracle set are denoted by yellow rows and their section IDs are marked with asterisks. This makes it easy to compare the chunks found against the oracle set.

Recall that chunks found using Factors R3 and AveR3 for a group of participants, are very similar except for A Group where they are identical.

For C Group, Factor AveR3F’s chunk with goodness measure of ‘Poor’ (recall = 0.22; precision = 1.00, size = 2, goodness = ‘Poor’) is better than Factor R3’s chunk (recall = 0.67; precision = 0.23, size = 26, goodness = ‘Very Poor’) and AveR3F’s chunk (recall = 0.67; precision = 0.22, size = 27, goodness = ‘Very Poor’) that have goodness measures of ‘Very Poor’. Factor R3’s and Factor AveR3F’s chunks have large number of false sections (20 and 21, respectively).

For I Group, Factor AveR3F’s chunk with goodness measure of ‘Average’ (recall = 0.56; precision = 0.50, size = 10, goodness = ‘Average’) is better than Factor R3’s chunk (recall = 0.56; precision = 0.25, size = 20, goodness = ‘Poor’) and AveR3’s chunk (recall = 0.67; precision = 0.27, size = 22, goodness = ‘Poor’) that have goodness measures of ‘Poor’.

Consequently, the best chunks for C and I Groups are both from Factor AveR3F. These best chunks are also the most basic chunks for the respective groups because they are the smallest chunks. For A Group, the best chunk (which is also the most basic chunk) is found using Factor R3 or Factor AveR3. These factors produced identical chunk (recall = 0.33; precision = 0.16, size = 19, goodness = ‘Very Poor’). No chunk is found for A Group using Factor AveR3F. For S Group, the 3 factors produce the same chunk (recall = 0.11; precision = 1.00, size = 1, goodness = ‘Poor’). This chunk is the best and also the most basic chunk for S Group.

For ASM Task 1, there is only one chunk with the best level of goodness measure for each group of participants. Therefore, it was unnecessary to assess the composition of chunk to decide the best chunk for a group.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C and I Groups is Factor AveR3F, for A Group is either Factor R3 or Factor AveR3, for S Group is either 1 of the 3 factors (since they produced the same chunk due only one participant in S Group).

Commonly-Missed Oracle Set’s Sections and Common False Sections: For the reason given at the beginning of Section “13.6 Chunking of Architectural Information”, we did not identify commonly-missed oracle set’s sections and common false sections for S Group that has only one participant.

The oracle set’s sections commonly missed by C Group are Sections “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” and “1.3.3.4 Standards”. These are also among the commonly-missed oracle set’s sections for C Group, for ASM Task 1 in Study 2. Oracle set’s Section “1.3.4.7 Database (org.eclipse.aperi.database plug-in)” is excluded by all factors but does not qualify as a commonly-missed section for C Group because of not being rated by any participant in C Group.

Sections “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model”, “1.3.3.4 Standards” and “1.3.4.7 Database (org.eclipse.aperi.database plug-in)” are excluded by all factors but do not qualify as a commonly-missed section for I Group because of not being rated by any participant in I Group.

For A Group, Sections “Introduction”, “1.3.1 Conceptual View”, “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model”, “1.3.3.4 Standards”, “1.3.4 Platform Layer”, and “1.3.4.7 Database (org.eclipse.aperi.database plug-in)” are excluded by all factors. However, only Sections “1.3.1 Conceptual View”, “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” and “1.3.3.4 Standards” qualified as commonly-missed sections for A Group because the rest of the sections were not rated by any participant in A Group. Section “1.3.3.4 Standards” is also among the commonly-missed oracle set’s sections for A Group, for ASM Task 1 in Study 2.

Section “1.3.3.4 Standards” is commonly missed probably because most of the participants failed to make the connection that the use of standards has some influences on the architectural design of ASM. In particular, it makes ASM dependent on SMI-S and SNMP agents (apart from the host agent) for deployment. This dependency of ASM is explained in Section “1.3.2 Process View”. As for the common exclusion of Section “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model” and Section “1.3.1 Conceptual View” by the respective group of participants, we cannot think of any plausible reason other than they were generally ‘not needed’ by the group when attempting ASM Task 1.

There is no common false section for C and A Groups. The common false sections for I Group are: Sections “1.3.5.1 Disk Application”, “1.3.5.4 Tape Application”, “2.1 Security” and “2.3 Platforms”. Sections “1.3.5.1 Disk Application” and “1.3.5.4 Tape Application” describe two of the exemplary applications in the Application Layer of ASM, and probably needed by the industry participants to get some idea on possible applications that can be developed using ASM. This might have helped in understanding ASM and subsequently its SA. Section “2.3 Platforms” describes the initial development platforms for ASM. We cannot think of any plausible reason other than this information was generally ‘needed’ by this group of industry participants, when trying to get an overview of the SA of ASM. It is surprising that Section “2.1 Security” which does not contain any valuable information was needed by I Group who attempted ASM Task 1.

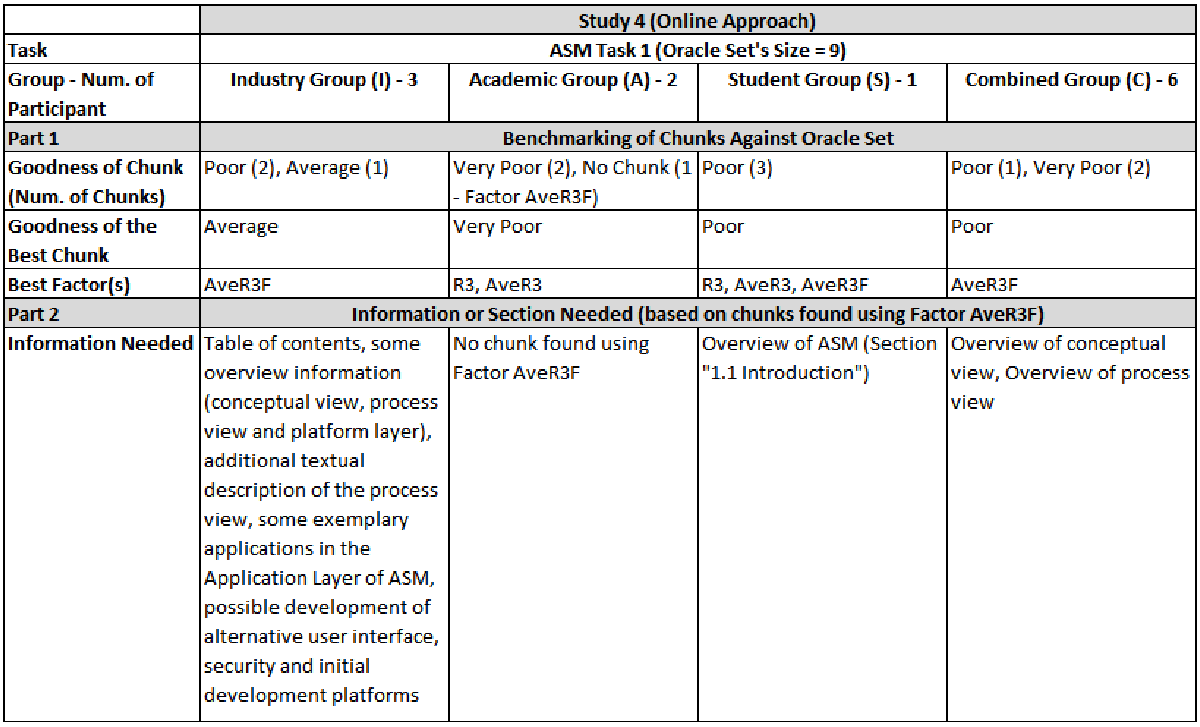
12.6.4.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for ASM Task 1 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. For ASM Task 1, chunks found using Factors R3 and AveR3 are identical for A and S Groups, and very similar for C and I Groups.

Part 1 of Table 12.9 summarises the main results of benchmarking of chunks. There is only one participant in S Group. The number of participants in I and A Groups are quite balance (i.e. 3 and 2, respectively). In terms of how ‘good’ the chunks are for ASM Task 1, C Group has 1 ‘Poor’ and 2 ‘Very Poor’ chunks. I Group has 1

‘Average’ and 2 ‘Poor’ chunks. A Group has 2 ‘Very Poor’ chunks. No chunk is found for A Group using Factor AveR3F. S Group has 3 ‘Poor’ identical chunks.

For ASM Task 1, the best chunk is found using Factor AveR3F for C and I Groups, Factor R3 or Factor AveR3 for A Group, and either 1 of the 3 factors (since they produce the same chunk) for S Group. Following that, the best chunk-identification factor for C and I Groups is Factor AveR3F. The best factor for A Group is either Factor R3 or Factor AveR3, and for S Group is either 1 of the 3 factors.



	Study 4 (Online Approach)			
Task	ASM Task 1 (Oracle Set's Size = 9)			
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 2	Student Group (S) - 1	Combined Group (C) - 6
Part 1	Benchmarking of Chunks Against Oracle Set			
Goodness of Chunk (Num. of Chunks)	Poor (2), Average (1)	Very Poor (2), No Chunk (1 - Factor AveR3F)	Poor (3)	Poor (1), Very Poor (2)
Goodness of the Best Chunk	Average	Very Poor	Poor	Poor
Best Factor(s)	AveR3F	R3, AveR3	R3, AveR3, AveR3F	AveR3F
Part 2	Information or Section Needed (based on chunks found using Factor AveR3F)			
Information Needed	Table of contents, some overview information (conceptual view, process view and platform layer), additional textual description of the process view, some exemplary applications in the Application Layer of ASM, possible development of alternative user interface, security and initial development platforms	No chunk found using Factor AveR3F	Overview of ASM (Section "1.1 Introduction")	Overview of conceptual view, Overview of process view

Table 12.9: Summary of Results for ASM Task 1 (Study 3)

Part 2 of Table 12.9 summarises the information or sections needed by each group of participants for ASM Task 1. This is based on the groups’ chunks found using Factor AveR3F. We give our rationale for this at the beginning of Section “12.6 Chunking of Architectural Information” in this chapter.

For ASM Task 1, we do not compare the three sub-groups of participants in terms of the information needed. The reason is no chunk is found using Factor AveR3F for A Group and the chunk found for S Group is based on only one participant. However, we present the information needed by groups which have more than one participant.

For ASM Task 1, I Group needed information related to Table of Contents, some overview information (conceptual view, process view and platform layer), additional textual description of the process view, some exemplary applications in the application layer of ASM, possible development of alternative user interface, security and initial development platforms. I Group covered a variation of topics, and needed detailed (or in-depth) information on process view (by requiring the additional textual description of the high-level processes).

When all the participants are combined in C Group, the group needed the overview of the conceptual view and overview of the process view for ASM Task 1.

12.6.5 ASM Task 2

Table 12.10 shows the chunks discovered for ASM Task 2, for industry practitioner (I), academic (A), student (S), and combined (C) groups of participants. The number of participants in I, A and C Groups are 3, 1 and 4, respectively. The background of the one participant in A Group is more towards research instead of academic.

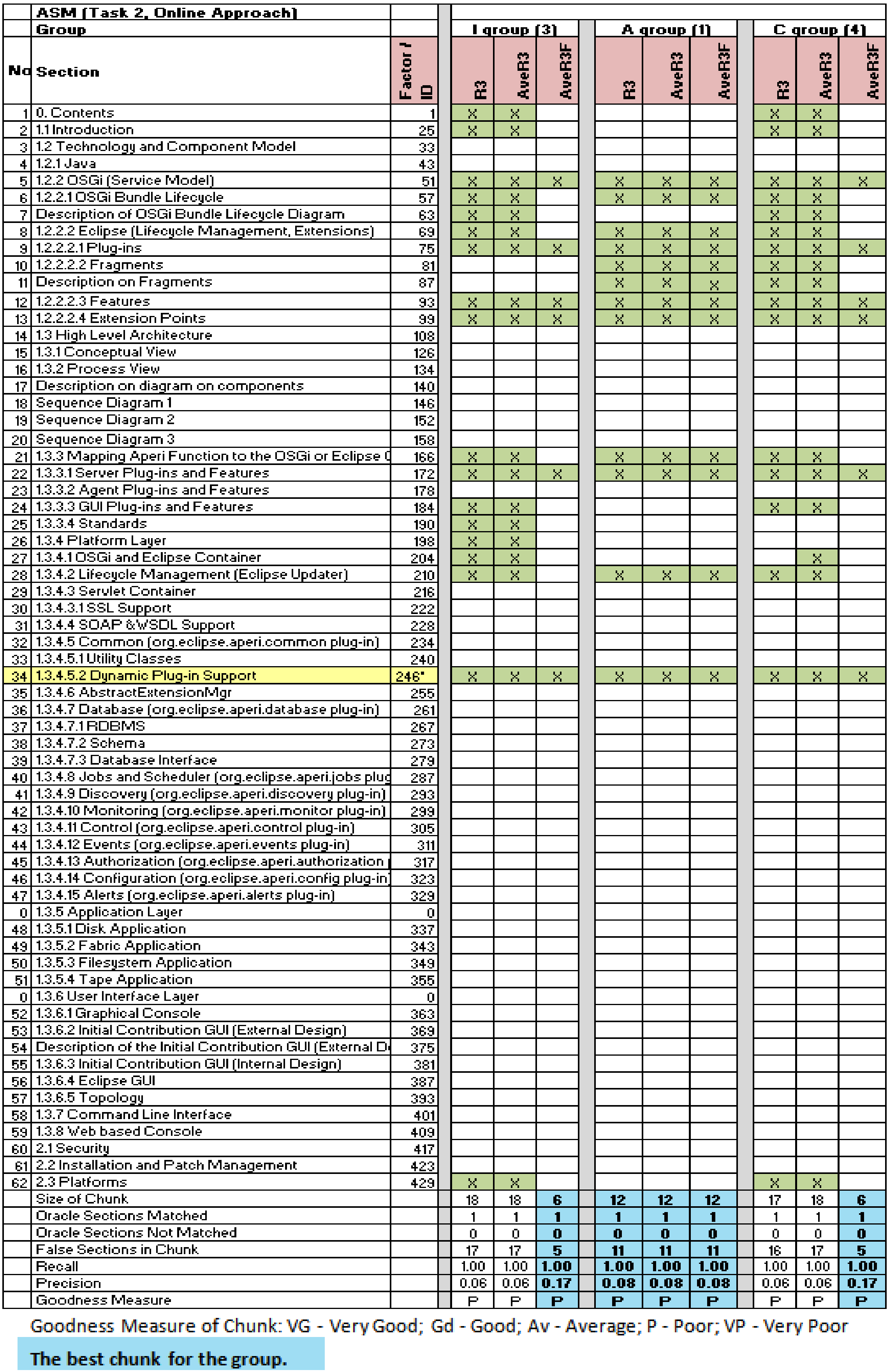
12.6.5.1 Identification and Interpretation of Chunks

Factor R3: C Group's chunk (17 sections) found using Factor R3 contains sections related to Table of Contents (Section "0. Contents"), overview of ASM (Section "1.1 Introduction"), all the 3 sections on OSGi service model (Sections "1.2.2 OSGi (Service Model)", "1.2.2.1 OSGi Bundle Lifecycle", and "Description of OSGi Bundle Lifecycle Diagram"), all the 6 sections on Eclipse plug-in model (Sections "1.2.2.2 Eclipse (Lifecycle Management, Extensions)", "1.2.2.2.1 Plug-ins", "1.2.2.2.2 Fragments", "Description on Fragments", "1.2.2.2.3 Features" and "1.2.2.2.4 Extension Points"), representation of major components of Aperi (Section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model"), Java packages related to Server and GUI plug-ins and features (Sections "1.3.3.1 Server Plug-in and Features" and "1.3.3.3 GUI Plug-in and Features"), Eclipse Updater for downloading updates (Section "1.3.4.2 Lifecycle Management (Eclipse Updater)"), dynamic plug-in support (Section "1.3.4.5.2 Dynamic Plug-in Support") and initial development platforms (Section "2.3 Platforms").

I Group's chunk (18 sections) found using Factor R3 is very similar to C Group's chunk found using the same factor, with the exclusion of Sections "1.2.2.2.2 Fragments" and "Description on Fragments", and the addition of Sections "1.3.3.4 Standards" (standards adopted), "1.3.4 Platform Layer" (overview of platform layer) and "1.3.4.1 OSGi and Eclipse Container" (base runtime container used).

Comparing A Group's chunk (12 sections) with I Group's chunk found using Factor R3 shows that both chunks include most of the sections related to OSGi service model and Eclipse plug-in model, Sections "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model", "1.3.3.1 Server Plug-in and Features", "1.3.4.2 Lifecycle Management (Eclipse Updater)" and "1.3.4.5.2 Dynamic Plug-in Support". In addition, the chunk for I Group contains Sections "0. Contents", "1.1 Introduction", "Description of OSGi Bundle Lifecycle Diagram", "1.3.3.3 GUI Plug-in and Features", "1.3.3.4 Standards", "1.3.4 Platform Layer", "1.3.4.1 OSGi and Eclipse Container", and "2.3 Platforms". The chunk for A Group also contains Sections "1.2.2.2.2 Fragments" and "Description on Fragments". A Group's chunk found using Factor R3 is identical to the group's chunks found using Factors AveR3 and AveR3F. This is non-surprising since there is only one participant in A Group.

Interpretation: All participants provided ratings for some sections of ASM document in terms of the importance of these sections to answering ASM Task 2. C Group's chunk found using Factor R3 shows that when attempting ASM Task 2, sections needed by the majority of C Group who rated the respective sections, are mainly related to the underlying models used by ASM (i.e. OSGi service model and Eclipse plug-in model). In addition, they needed Table of Contents, overview information on ASM, how major components of Aperi are represented, certain Java packages, Eclipse Updater, dynamic plug-in support and initial development platforms.



	ASM (Task 2, Online Approach)										
	Group		I group (3)			A group (1)			C group (4)		
No	Section	Factor / ID	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F
1	0. Contents	1	X	X					X	X	
2	1.1 Introduction	25	X	X					X	X	
3	1.2 Technology and Component Model	33									
4	1.2.1 Java	43									
5	1.2.2 OSGi (Service Model)	51	X	X	X	X	X	X	X	X	X
6	1.2.2.1 OSGi Bundle Lifecycle	57	X	X		X	X	X	X	X	
7	Description of OSGi Bundle Lifecycle Diagram	63	X	X					X	X	
8	1.2.2.2 Eclipse (Lifecycle Management, Extensions)	69	X	X		X	X	X	X	X	
9	1.2.2.2.1 Plug-ins	75	X	X	X	X	X	X	X	X	X
10	1.2.2.2.2 Fragments	81				X	X	X	X	X	
11	Description on Fragments	87				X	X	X	X	X	
12	1.2.2.2.3 Features	93	X	X	X	X	X	X	X	X	X
13	1.2.2.2.4 Extension Points	99	X	X	X	X	X	X	X	X	X
14	1.3 High Level Architecture	108									
15	1.3.1 Conceptual View	126									
16	1.3.2 Process View	134									
17	Description on diagram on components	140									
18	Sequence Diagram 1	146									
19	Sequence Diagram 2	152									
20	Sequence Diagram 3	158									
21	1.3.3 Mapping Aperi Function to the OSGi or Eclipse C	166	X	X		X	X	X	X	X	
22	1.3.3.1 Server Plug-ins and Features	172	X	X	X	X	X	X	X	X	X
23	1.3.3.2 Agent Plug-ins and Features	178									
24	1.3.3.3 GUI Plug-ins and Features	184	X	X					X	X	
25	1.3.3.4 Standards	190	X	X							
26	1.3.4 Platform Layer	198	X	X							
27	1.3.4.1 OSGi and Eclipse Container	204	X	X						X	
28	1.3.4.2 Lifecycle Management (Eclipse Updater)	210	X	X		X	X	X	X	X	
29	1.3.4.3 Servlet Container	216									
30	1.3.4.3.1 SSL Support	222									
31	1.3.4.4 SOAP &WSDL Support	228									
32	1.3.4.5 Common (org.eclipse.aperi.common plug-in)	234									
33	1.3.4.5.1 Utility Classes	240									
34	1.3.4.5.2 Dynamic Plug-in Support	246*	X	X	X	X	X	X	X	X	X
35	1.3.4.6 AbstractExtensionMgr	255									
36	1.3.4.7 Database (org.eclipse.aperi.database plug-in)	261									
37	1.3.4.7.1 RDBMS	267									
38	1.3.4.7.2 Schema	273									
39	1.3.4.7.3 Database Interface	279									
40	1.3.4.8 Jobs and Scheduler (org.eclipse.aperi.jobs plug	287									
41	1.3.4.9 Discovery (org.eclipse.aperi.discovery plug-in)	293									
42	1.3.4.10 Monitoring (org.eclipse.aperi.monitor plug-in)	299									
43	1.3.4.11 Control (org.eclipse.aperi.control plug-in)	305									
44	1.3.4.12 Events (org.eclipse.aperi.events plug-in)	311									
45	1.3.4.13 Authorization (org.eclipse.aperi.authorization p	317									
46	1.3.4.14 Configuration (org.eclipse.aperi.config plug-in)	323									
47	1.3.4.15 Alerts (org.eclipse.aperi.alerts plug-in)	329									
0	1.3.5 Application Layer	0									
48	1.3.5.1 Disk Application	337									
49	1.3.5.2 Fabric Application	343									
50	1.3.5.3 Filesystem Application	349									
51	1.3.5.4 Tape Application	355									
0	1.3.6 User Interface Layer	0									
52	1.3.6.1 Graphical Console	363									
53	1.3.6.2 Initial Contribution GUI (External Design)	369									
54	Description of the Initial Contribution GUI (External De	375									
55	1.3.6.3 Initial Contribution GUI (Internal Design)	381									
56	1.3.6.4 Eclipse GUI	387									
57	1.3.6.5 Topology	393									
58	1.3.7 Command Line Interface	401									
59	1.3.8 Web based Console	409									
60	2.1 Security	417									
61	2.2 Installation and Patch Management	423									
62	2.3 Platforms	429	X	X					X	X	
	Size of Chunk		18	18	**6**	**12**	**12**	**12**	17	18	**6**
	Oracle Sections Matched		1	1	**1**	**1**	**1**	**1**	1	1	**1**
	Oracle Sections Not Matched		0	0	**0**	**0**	**0**	**0**	0	0	**0**
	False Sections in Chunk		17	17	**5**	**11**	**11**	**11**	16	17	**5**
	Recall		1.00	1.00	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	**1.00**
	Precision		0.06	0.06	**0.17**	**0.08**	**0.08**	**0.08**	0.06	0.06	**0.17**
	Goodness Measure		P	P	**P**	**P**	**P**	**P**	P	P	**P**

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 12.10: The Chunks for ASM Task 2 Identified Using Different Factors (Study 3)

The chunk found using Factor R3 for I Group shows that the majority of industry participants who rated the respective sections, needed information on the underlying models used by ASM (i.e. OSGi service model and Eclipse plug-in model), representation of major components of Aperi, certain Java packages, Eclipse Updater and dynamic plug-in support. The aforementioned information was also needed by the one participant in A Group as shown by A Group's chunk found using Factor R3. In addition, the majority of I Group who rated the respective sections, needed information related to Table of Contents, overview of ASM and platform layer, standards adopted, current base runtime container used and initial development platforms, for ASM Task 2.

Factor AveR3: C Group's chunk found using Factor AveR3 is very similar to the group's chunk found using Factor R3 with the addition of Section ""1.3.4.1 OSGi and Eclipse Container". For I Group, the chunks found using Factors AveR3 and R3 are identical. A Group's chunk found using Factor AveR3 is identical to the group's chunks found using the other 2 factors.

Interpretation: In addition to information presented in C Group's chunk found using Factor R3, some of C Group's participants also needed information related to current base runtime container used, for ASM Task 2. The interpretation for I Group's chunk is the same as the interpretation given for the group's chunk found using Factor R3. The same chunk found for A Group shows that the one participant in A Group needed information on the underlying models used by ASM (i.e. OSGi service model and Eclipse plug-in model), representation of major components of Aperi, certain Java packages, Eclipse Updater and dynamic plug-in support, for ASM Task 2.

Factor AveR3F: The chunks found for C and I Groups using Factor AveR3F are identical, comprising Sections "1.2.2 OSGi (Service Model)", "1.2.2.2.1 Plug-ins", "1.2.2.2.3 Features", "1.2.2.2.4 Extension Points", "1.3.3.1 Server Plug-in and Features" and "1.3.4.5.2 Dynamic Plug-in Support". A Group's chunk found using all the 3 factors are the same.

Interpretation: Three of the 4 participants in C Group are industry practitioners from I Group. Therefore, it is non-surprising that same chunk is found for C and I Groups. C and I Groups' chunks found using Factor AveR3F show that based on the average ratings of the majority of the participants (i.e. the industry practitioners in the case of ASM Task 2), the overview of OSGi service model, some detailed information on Eclipse plug-in model (plug-ins but not specifically on dynamic plug-in, features, extension points), certain Java packages, and dynamic plug-in support, were needed for the task. The interpretation for A Group's chunk is the same as with the other 2 factors.

12.6.5.2 Benchmarking Chunks Against An Oracle Set

The attributes (size, numbers of oracle set's section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for ASM Task 2 are given at the bottom of Table 12.10. In Table 12.10, the section that is part of the oracle set is denoted by yellow row and its section ID is marked with asterisks. This makes it easy to compare the chunks found against the oracle set.

All the chunks found for ASM Task 2 have the same goodness measures of 'Poor' regardless of the group of participants and the chunk-identification factors used. All these 'Poor' chunks have recall measures of 1.00.

Nevertheless, the smaller chunks (recall = 1.00; precision = 0.17, size = 6, goodness = 'Poor') found using Factor AveR3F for C and I Groups, have slightly higher precision measures compared to the larger chunks found using Factors R3 and AveR3 for C and I Groups (recall = 1.00; precision = 0.06, size = 17, goodness = 'Poor'). Note that C Group's chunk found using Factor AveR3 has size 18 instead of 17, and I Group's chunks found using Factors R3 and AveR3 are identical.

Consequently, the best chunks (which are also the most basic chunks because they are the smallest chunks) for C and I Groups are both found using Factor AveR3F. For A Group, the 3 chunk-identification factors produce identical chunk (recall = 1.00; precision = 0.08, size = 12, goodness = 'Poor') because there is only one participant in this group. This chunk is the best and also the most basic chunk for A Group.

None of the chunks misses the one and only one oracle set's section.

The factor that produces the best chunk (i.e. the best chunk-identification factor) for C and I Groups is Factor AveR3F. The best factor for A Group is either 1 of the 3 factors since they produce identical chunks.

Commonly-Missed Oracle Set's Sections and Common False Sections: For the reason given at the beginning of Section "13.6 Chunking of Architectural Information", we did not identify commonly-missed oracle set's sections and common false sections for A Group, that has only one participant.

There is no oracle set's section commonly missed by C and I Groups. The common false sections for C and I Groups are the same. They are Sections "1.2.2 OSGi (Service Model)", "1.2.2.2.1 Plug-ins", "1.2.2.2.3 Features", "1.2.2.2.4 Extension Points", and "1.3.3.1 Server Plug-in and Features".

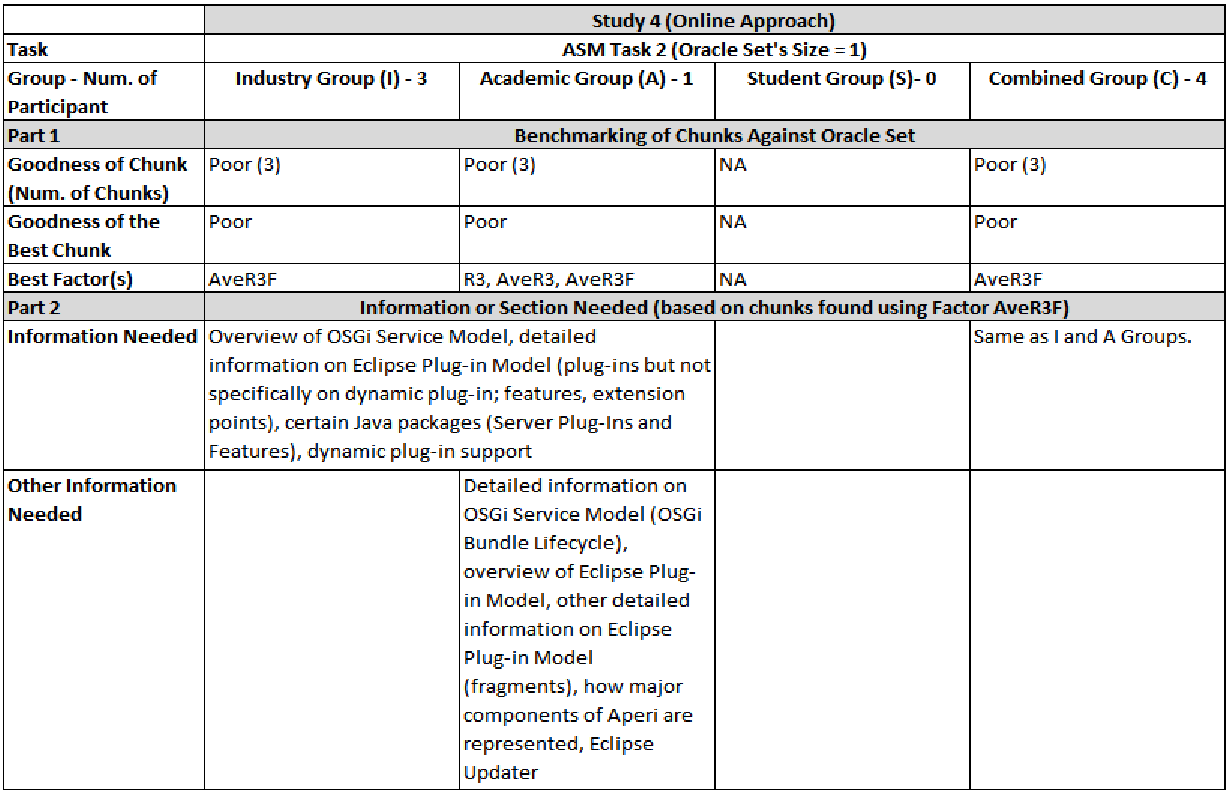
Section "1.2.2 OSGi (Service Model)" contains the overview of OSGi Service Model, what are bundles and deployment via bundles. Section "1.2.2.2.1 Plug-ins" contains low-level detailed information on plug-ins but not particularly on dynamic plug-in. The information in these two sections helps to understand the basic nature of the components in ASM. This could be important for some participants to find out how to dynamically unload a plug-in. Section "1.3.3.1 Server Plug-in and Features" contains Java packages related to server plug-ins and features, and probably needed by some participants to assess possible impact of dynamically unloading a plug-in.

Section "1.2.2.2.3 Features" contains description of feature as a group of plug-ins that define a logical product feature. Section "1.2.2.2.4 Extension Points" contains detailed explanation on how plug-in could be extended through 'extension points' with some example code. These two sections contain detailed information related to Eclipse plug-in model. We cannot think of any plausible reason for the common inclusion of these sections other than this information was generally 'needed', by C and I Groups of participants.

12.6.5.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for ASM Task 2 for each group of participants. Generally, there are two clusters of chunks found for each group of participants: the smaller chunks found using Factors A, H, A|H|R3 and AveR3F; and the larger chunks found using Factors R3 and AveR3. For ASM Task 2, chunks found using Factors R3 and AveR3 are identical for I and A Groups, and very similar for C Group. For A Group, the three factors produce identical chunks since there is only one participant in A Group.

Part 1 of Table 12.11 summarises the main results of benchmarking of chunks. The number of participants in I, and A Groups are imbalance, which is 3 and 1, respectively. There is no S Group for ASM Task 2. All the chunks found for ASM Task 2 have the same goodness measures of 'Poor' regardless of the group of participants and the chunk-identification factors used. They all have recall measure of 1.00.



	Study 4 (Online Approach)			
Task	**ASM Task 2 (Oracle Set's Size = 1)**			
Group - Num. of Participant	**Industry Group (I) - 3**	**Academic Group (A) - 1**	**Student Group (S)- 0**	**Combined Group (C) - 4**
Part 1	**Benchmarking of Chunks Against Oracle Set**			
Goodness of Chunk (Num. of Chunks)	Poor (3)	Poor (3)	NA	Poor (3)
Goodness of the Best Chunk	Poor	Poor	NA	Poor
Best Factor(s)	AveR3F	R3, AveR3, AveR3F	NA	AveR3F
Part 2	**Information or Section Needed (based on chunks found using Factor AveR3F)**			
Information Needed	Overview of OSGi Service Model, detailed information on Eclipse Plug-in Model (plug-ins but not specifically on dynamic plug-in; features, extension points), certain Java packages (Server Plug-Ins and Features), dynamic plug-in support			Same as I and A Groups.
Other Information Needed		Detailed information on OSGi Service Model (OSGi Bundle Lifecycle), overview of Eclipse Plug-in Model, other detailed information on Eclipse Plug-in Model (fragments), how major components of Aperi are represented, Eclipse Updater		

Table 12.11: Summary of Results for ASM Task 2 (Study 3)

In Study 3, the sizes of the chunks found for ASM Task 2 do not commensurate with the small size of the oracle set for the task. This is opposite to what is found for ASM Task 2 in Study 2.

For ASM Task 2, the best chunks for C and I Groups are found using Factor AveR3F, and for A Group is found using either 1 of the 3 factors (since they produce the same chunk). Following that, the best chunk-identification factor for C and I Groups is Factor AveR3F, and for A Group is either 1 of the 3 factors.

Part 2 of Table 12.11 summarises the information or sections needed by each group of participants for ASM Task 2. This is based on the groups' chunks found using Factor AveR3F. We give our rationale for this at the beginning of Section "12.6 Chunking of Architectural Information" in this chapter.

For ASM Task 2, we do not compare the two sub-groups of participants in terms of the information needed. The reason is the chunk found for A Group is based on only one participant. However, we present the information needed by groups which have more than one participant. There is no S Group for ASM Task 2.

For ASM Task 2, I Group needed information related to the overview of OSGi Service Model, detailed information on Eclipse plug-in model (plug-ins; features, extension points), certain Java packages (Server Plug-Ins and Features) and dynamic plug-in support. I Group covered a variation of topics (including the two underlying models used), needed detailed information on Eclipse plug-in model (including low-level information on plug-ins). When all the participants are combined in C Group, the group needed the same information as I Group. This is non-surprising since 3 of the 4 participants in C Group are from I Group.

12.6.6 ASM Task 3

Table 12.12 shows the chunks discovered for ASM Task 3, for industry practitioner (I), academic (A), student (S), and combined (C) groups of participants. The number of participants in I, A, S and C Groups are 2, 1, 1 and 4, respectively. The background of the one participant in A Group is more towards research instead of academic.

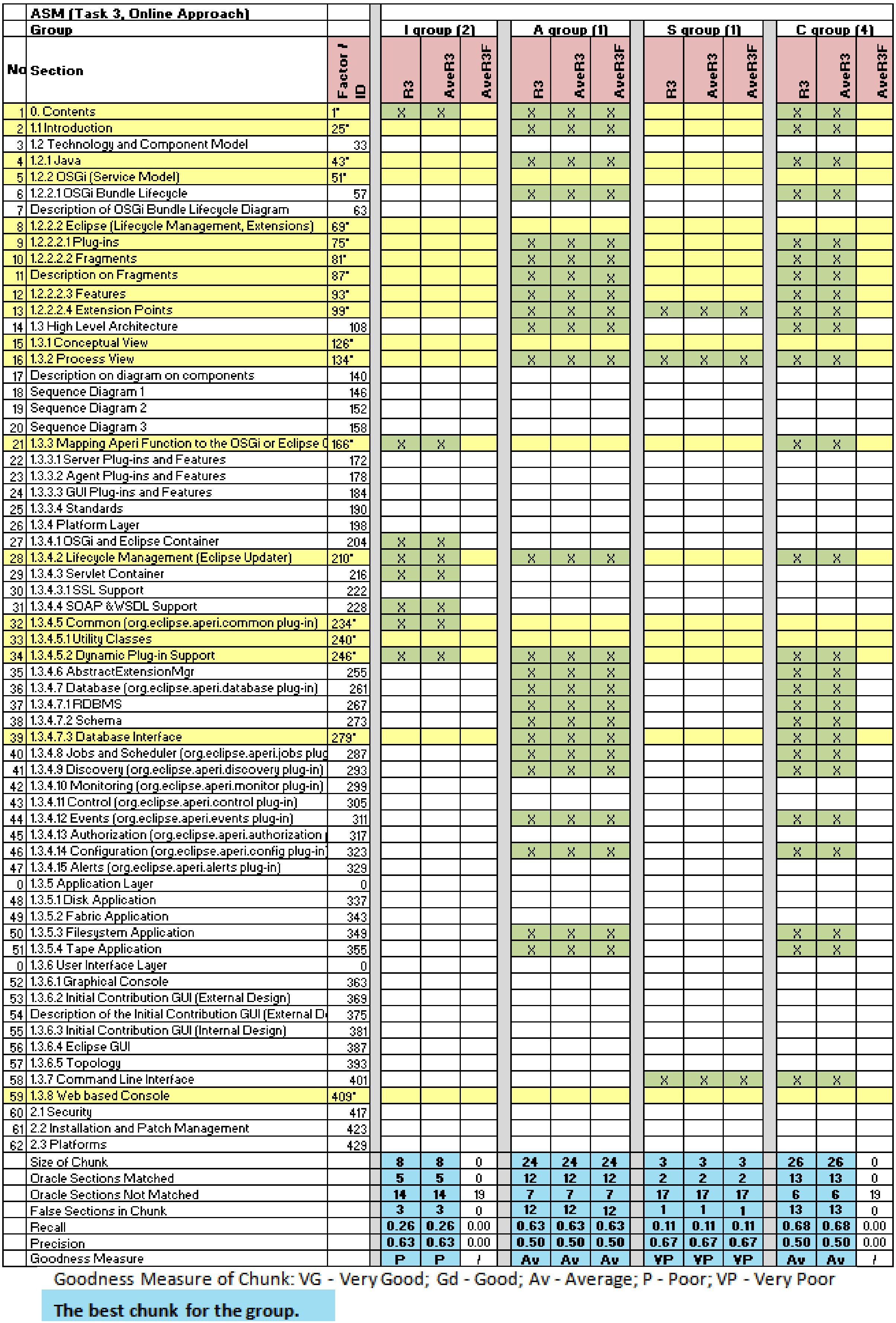
12.6.6.1 Identification and Interpretation of Chunks

Factor R3: C Group's chunk found using Factor R3 contains 26 sections. These sections are related to Table of Contents (Section "0. Contents"), overview of ASM (Section "1.1 Introduction"), development languages used (Section "1.2.1 Java"), diagram on the lifecycle of an OSGi bundle (Section "1.2.2.1 OSGi Bundle Lifecycle"), more detailed information on Eclipse plug-in model (in terms of plug-ins, fragment, features and extension points), Section "1.3 High Level Architecture" which contains only hyperlinks to its sub-sections, overview of process view (Section "1.3.2 Process View"), representation of major components of Aperi (Section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model"), Eclipse Updater for downloading updates (Section "1.3.4.2 Lifecycle Management (Eclipse Updater)"), dynamic plug-in support (Section "1.3.4.5.2 Dynamic Plug-in Support"), Java code to manage extensions (Section "1.3.4.6 AbstractExtensionMgr"), all sections on database (Sections "1.3.4.7 Database (org.eclipse.aperi.database plug-in)", "1.3.4.7.1 RDBMS", "1.3.4.7.2 Schema", and "1.3.4.7.3 Database Interface"), certain generic services (Sections 1.3.4.8, 1.3.4.9, 1.3.4.12", 1.3.4.14), two exemplary applications (Sections 1.3.5.3 and 1.3.5.4), and command line interface (CLI) (Section "1.3.7 Command Line Interface").

The chunk found using Factor R3 for I Group comprises sections related to Table of Contents, representation of major components of Aperi, base runtime container (Section "1.3.4.1 OSGi and Eclipse Container"), Eclipse Updater, servlet container used (Section "1.3.4.3 Servlet Container"), support of SOAP & WSDL (Section "1.3.4.4 SOAP & WSDL Support"), 'common' plug-in containing common code (Section "1.3.4.5 Common (org.eclipse.aperi.common plug-in)"), and dynamic plug-in support.

The chunks found using Factor R3 for A and C Groups are very similar. The difference is A Group's chunk excludes Section "1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model" and "1.3.7 Command Line Interface". S Group's chunk comprises sections related to extension points, overview of process view and command line interface.

Interpretation: All participants provided ratings for some sections of ASM document in terms of the importance of these sections to answering ASM Task 3.



	ASM (Task 3, Online Approach)														
	Group		I group (2)			A group (1)			S group (1)			C group (4)			
No	Section	Factor / ID	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F	R3	AveR3	AveR3F
1	0. Contents	1*	X	X		X	X	X				X	X	
2	1.1 Introduction	25*				X	X	X				X	X	
3	1.2 Technology and Component Model	33												
4	1.2.1 Java	43*				X	X	X				X	X	
5	1.2.2 OSGi (Service Model)	51*												
6	1.2.2.1 OSGi Bundle Lifecycle	57				X	X	X				X	X	
7	Description of OSGi Bundle Lifecycle Diagram	63												
8	1.2.2.2 Eclipse (Lifecycle Management, Extensions)	69*												
9	1.2.2.2.1 Plug-ins	75*				X	X	X				X	X	
10	1.2.2.2.2 Fragments	81*				X	X	X				X	X	
11	Description on Fragments	87*				X	X	X				X	X	
12	1.2.2.2.3 Features	93*				X	X	X				X	X	
13	1.2.2.2.4 Extension Points	99*				X	X	X	X	X	X	X	X	
14	1.3 High Level Architecture	108				X	X	X				X	X	
15	1.3.1 Conceptual View	126*												
16	1.3.2 Process View	134*				X	X	X	X	X	X	X	X	
17	Description on diagram on components	140												
18	Sequence Diagram 1	146												
19	Sequence Diagram 2	152												
20	Sequence Diagram 3	158												
21	1.3.3 Mapping Aperi Function to the OSGi or Eclipse C	166*	X	X								X	X	
22	1.3.3.1 Server Plug-ins and Features	172												
23	1.3.3.2 Agent Plug-ins and Features	178												
24	1.3.3.3 GUI Plug-ins and Features	184												
25	1.3.3.4 Standards	190												
26	1.3.4 Platform Layer	198												
27	1.3.4.1 OSGi and Eclipse Container	204	X	X										
28	1.3.4.2 Lifecycle Management (Eclipse Updater)	210*	X	X		X	X	X				X	X	
29	1.3.4.3 Servlet Container	216	X	X										
30	1.3.4.3.1 SSL Support	222												
31	1.3.4.4 SOAP &WSDL Support	228	X	X										
32	1.3.4.5 Common (org.eclipse.aperi.common plug-in)	234*	X	X										
33	1.3.4.5.1 Utility Classes	240*												
34	1.3.4.5.2 Dynamic Plug-in Support	246*	X	X		X	X	X				X	X	
35	1.3.4.6 AbstractExtensionMgr	255				X	X	X				X	X	
36	1.3.4.7 Database (org.eclipse.aperi.database plug-in)	261				X	X	X				X	X	
37	1.3.4.7.1 RDBMS	267				X	X	X				X	X	
38	1.3.4.7.2 Schema	273				X	X	X				X	X	
39	1.3.4.7.3 Database Interface	279*				X	X	X				X	X	
40	1.3.4.8 Jobs and Scheduler (org.eclipse.aperi.jobs plug	287				X	X	X				X	X	
41	1.3.4.9 Discovery (org.eclipse.aperi.discovery plug-in)	293				X	X	X				X	X	
42	1.3.4.10 Monitoring (org.eclipse.aperi.monitor plug-in)	299												
43	1.3.4.11 Control (org.eclipse.aperi.control plug-in)	305												
44	1.3.4.12 Events (org.eclipse.aperi.events plug-in)	311				X	X	X				X	X	
45	1.3.4.13 Authorization (org.eclipse.aperi.authorization p	317												
46	1.3.4.14 Configuration (org.eclipse.aperi.config plug-in)	323				X	X	X				X	X	
47	1.3.4.15 Alerts (org.eclipse.aperi.alerts plug-in)	329												
0	1.3.5 Application Layer	0												
48	1.3.5.1 Disk Application	337												
49	1.3.5.2 Fabric Application	343												
50	1.3.5.3 Filesystem Application	349				X	X	X				X	X	
51	1.3.5.4 Tape Application	355				X	X	X				X	X	
0	1.3.6 User Interface Layer	0												
52	1.3.6.1 Graphical Console	363												
53	1.3.6.2 Initial Contribution GUI (External Design)	369												
54	Description of the Initial Contribution GUI (External De	375												
55	1.3.6.3 Initial Contribution GUI (Internal Design)	381												
56	1.3.6.4 Eclipse GUI	387												
57	1.3.6.5 Topology	393												
58	1.3.7 Command Line Interface	401							X	X	X	X	X	
59	1.3.8 Web based Console	409*												
60	2.1 Security	417												
61	2.2 Installation and Patch Management	423												
62	2.3 Platforms	429												
	Size of Chunk		8	8	0	24	24	24	3	3	3	26	26	0
	Oracle Sections Matched		5	5	0	12	12	12	2	2	2	13	13	0
	Oracle Sections Not Matched		14	14	19	7	7	7	17	17	17	6	6	19
	False Sections in Chunk		3	3	0	12	12	12	1	1	1	13	13	0
	Recall		0.26	0.26	0.00	0.63	0.63	0.63	0.11	0.11	0.11	0.68	0.68	0.00
	Precision		0.63	0.63	0.00	0.50	0.50	0.50	0.67	0.67	0.67	0.50	0.50	0.00
	Goodness Measure		P	P	/	Av	Av	Av	VP	VP	VP	Av	Av	/

Goodness Measure of Chunk: VG - Very Good; Gd - Good; Av - Average; P - Poor; VP - Very Poor

The best chunk for the group.

Table 12.12: Chunks for ASM Task 3 Identified Using Different Factors (Study 3)

C Group’s chunk found using Factor R3 shows that for ASM Task 3 the sections needed by the majority of C Group who rated the respective sections, are mainly related to organisation of the document (Section “0. Contents” and Section “1.3 High Level Architecture”), overview information (on ASM and process view), development languages used, more detailed information on OSGi service model and Eclipse plug-in model, representation of major components of Aperi, Eclipse Updater, dynamic plug-in support, Java code to manage extensions, database, certain generic services, exemplary applications, and command line interface. These sections (except the sections on representation of major components of Aperi, and command line interface) were also needed by the one participant who rated the respective sections, as shown by A Group’s chunk found using Factor R3.

The chunk found using Factor R3 for I Group shows that the majority of I Group who rated the respective sections, needed Table of Contents, representation of major components of Aperi, base runtime container, Eclipse Updater, servlet container used, support of SOAP & WSDL, common plug, and dynamic plug-in support, when attempting ASM Task 3.

S Group’s chunk found using Factor R3 shows that the one student participant needed information on extension points, overview of process view, and command line interface, for ASM Task 3.

Factor AveR3: The chunk found using Factor AveR3 for a particular group of participants, is identical to the group’s chunk found using Factor R3. This happens for all the 4 groups of participants.

Interpretation: C Group’s chunk found using Factor AveR3 shows that all the sections in the group’s chunk found using Factor R3 were needed by some of the participants who rated them (in fact needed by the majority of the participants who rated them as revealed by Factor R3’s chunk). Similar interpretations are applicable to A, I and S Groups’ chunks found using Factor AveR3 since each of the chunks is identical to the respective group’s chunk found using Factor R3.

Factor AveR3F: No chunk is found for C and I Groups when Factor AveR3F is used. A Group’s chunk found using Factor AveR3F is identical to the group’s chunks found using Factors R3 and AveR3. Same thing happens for S Group. This is because there is only one participant, in A and S Groups.

Interpretation: The interpretations for A and S Groups’ chunks found using Factor R3 are applicable here.

12.6.6.2 Benchmarking Chunks Against An Oracle Set

For the concept of ‘Modifiability’ involved in ASM Task 3, refer to Section “11.2.1 What is Modifiability?” of Chapter 11.

The attributes (size, numbers of oracle set’s section matched and not matched, the number of false sections, recall, precision and goodness measures) of each chunk for ASM Task 3 are given at the bottom of Table 12.12. In Table 12.12, the sections that are part of the oracle set are denoted by yellow rows and their section IDs are marked with asterisks. This makes it easy to compare the chunks found against the oracle set.

For ASM Task 3, only one type of chunk is found for C and I Groups: the chunks found using Factors R3 and AveR3 (which are identical). No chunk is found for C and I Groups using Factor AveR3F. The only chunk found for C Group (using Factors R3 and AveR3) has goodness measure of 'Average' (recall = 0.68; precision = 0.50, size = 26, goodness = 'Average'). The only chunk found for I Group has goodness measure of 'Poor' (recall = 0.26; precision = 0.63, size = 8, goodness = 'Poor'). For both A and S Groups, the chunks found using the 3 factors are identical, since there is only one participant in each of these groups. Chunk for A Group has goodness measure of 'Average' (recall = 0.63; precision = 0.50, size = 24, goodness = 'Average'), and S Group's chunk is 'Very Poor' (recall = 0.11; precision = 0.67, size = 3, goodness = 'Very Poor').

The only chunk found for a particular group is the best and also the most basic chunk for the group. The factor that produces the best chunk (i.e. the best chunk-identification factor) for C and I Groups is either Factor R3 or Factor AveR3. The best factor for A and S Groups is either 1 of the 3 factors.

Commonly-Missed Oracle Set's Sections and Common False Sections: For the reason given at the beginning of Section "13.6 Chunking of Architectural Information", we did not identify commonly-missed oracle set's sections and common false sections for A and S Groups which have only one participant.

The oracle set's sections commonly missed by C Group are Sections "1.2.2 OSGi (Service Model)", "1.2.2.2 Eclipse (Lifecycle Management, Extensions)", "1.3.1 Conceptual View", "1.3.4.5 Common (org.eclipse.aperi.common plug-in)", "1.3.4.5.1 Utility Classes". The oracle set's section commonly missed by I Group are Sections "1.3.1 Conceptual View" and "1.3.4.5.1 Utility Classes". Other oracle set's sections that are excluded by all chunk-identification factors do not qualify as commonly-missed oracle set's sections for I Group, because of not being rated by any of the participants in I Group.

Section "1.3.1 Conceptual View" shows some aspects of the SA but do not contain specific information for ASM Task 3. Sections "1.2.2 OSGi (Service Model)" and "1.2.2.2 Eclipse (Lifecycle Management, Extensions)" contain the overview of OSGi service model and Eclipse plug-in model. They were commonly missed by C Group probably because the participants in C Group generally have background knowledge on these models.

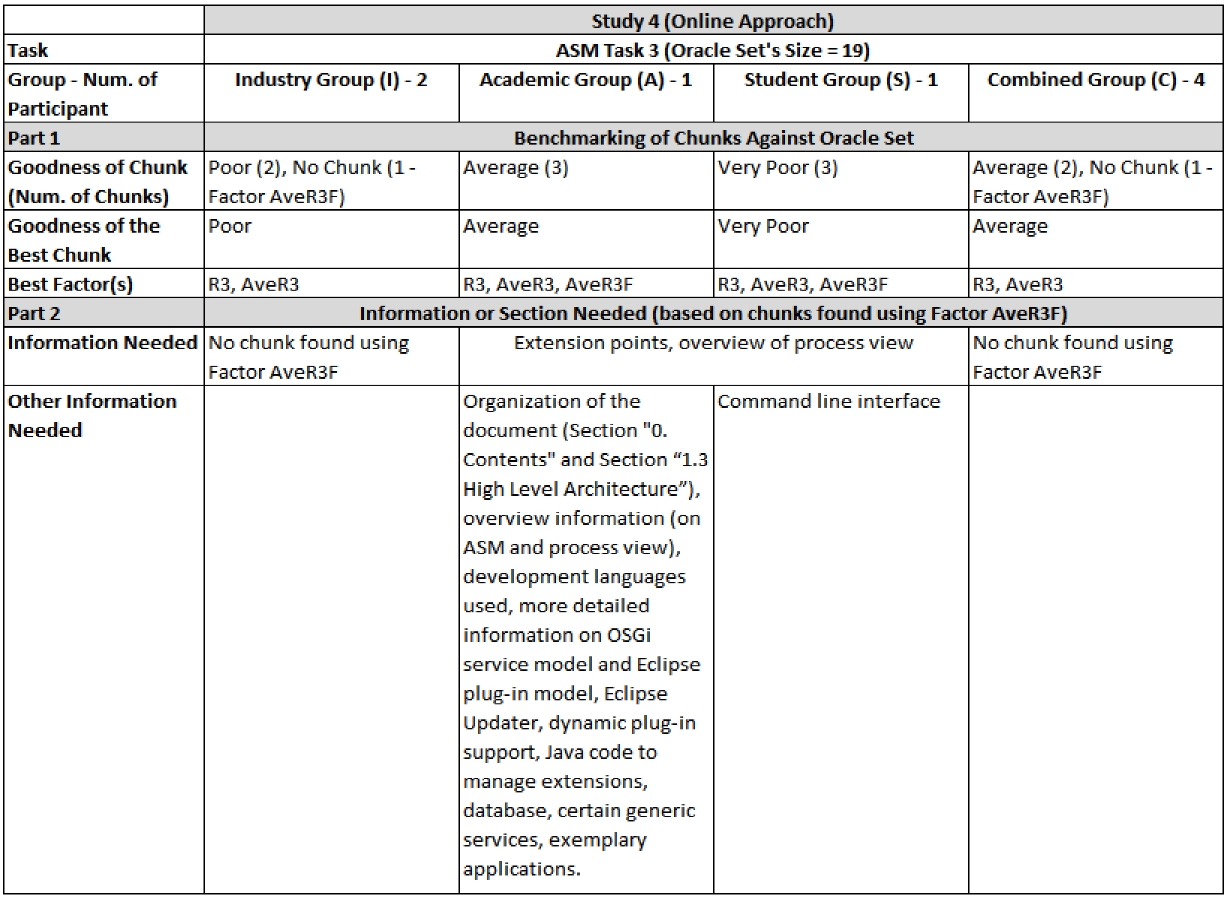
Section "1.3.4.5 Common (org.eclipse.aperi.common plug-in)" describes the specific plug-in which contains common code shared across implementation plug-ins. Section "1.3.4.5.1 Utility Classes" describes the utility classes which served as helper classes used by the rest of the platform. Both the sections mentioned in this paragraph relate to the tactic of abstracting common services to achieve modifiability. We cannot think of any plausible reason for their common exclusion by the respective group of participants, other than they were generally 'not needed' by the respective group.

There is no common false section for C and I Groups.

12.6.6.3 Discussion of Chunk Discovery Results

The results show the discovery of chunks for ASM Task 3 for each group of participants. For ASM Task 3, only one type of chunk is found for each group of participants. This is different from the other 5 information-seeking tasks where chunks found can be grouped into two clusters (i.e. the larger chunks found using Factors R3 and AveR3, and the smaller chunk found using Factor AveR3F). For ASM Task 3, chunks found using Factors R3 and AveR3 for each of the 4 groups, are identical.

Part 1 of Table 12.13 summarises the main results of benchmarking of chunks. There is only one participant in A and S Groups. There are 2 participants in I Group. In terms of how 'good' the chunks are for ASM Task 1, C Group has 2 'Average' chunks. I Group has 2 'Poor' chunks. A Group has 3 'Average' identical chunks. S Group has 3 'Very Poor' identical chunks. For ASM Task 3, the best chunks for C and I Groups are found using Factor R3 or Factor AveR3. Best chunks for A and S Groups are found using either 1 of the 3 factors. Following that, the best chunk-identification factor for C and I Groups is either Factor R3 or Factor AveR3. The best factor for A and S Groups is either 1 of the 3 factors.



	Study 4 (Online Approach)			
Task	**ASM Task 3 (Oracle Set's Size = 19)**			
Group - Num. of Participant	**Industry Group (I) - 2**	**Academic Group (A) - 1**	**Student Group (S) - 1**	**Combined Group (C) - 4**
Part 1	**Benchmarking of Chunks Against Oracle Set**			
Goodness of Chunk (Num. of Chunks)	Poor (2), No Chunk (1 - Factor AveR3F)	Average (3)	Very Poor (3)	Average (2), No Chunk (1 - Factor AveR3F)
Goodness of the Best Chunk	Poor	Average	Very Poor	Average
Best Factor(s)	R3, AveR3	R3, AveR3, AveR3F	R3, AveR3, AveR3F	R3, AveR3
Part 2	**Information or Section Needed (based on chunks found using Factor AveR3F)**			
Information Needed	No chunk found using Factor AveR3F	Extension points, overview of process view		No chunk found using Factor AveR3F
Other Information Needed		Organization of the document (Section "0. Contents" and Section "1.3 High Level Architecture"), overview information (on ASM and process view), development languages used, more detailed information on OSGi service model and Eclipse plug-in model, Eclipse Updater, dynamic plug-in support, Java code to manage extensions, database, certain generic services, exemplary applications.	Command line interface	

Table 12.13: Summary of Results for ASM Task 3 (Study 3)

Part 2 of Table 12.13 summarises the information or sections needed by each group of participants for ASM Task 3. This is based on the groups' chunks found using Factor AveR3F. We give our rationale for this at the beginning of Section "12.6 Chunking of Architectural Information" in this chapter.

For ASM Task 3, we do not compare the three sub-groups of participants in terms of the information needed. The reason is no chunk is found for I Group using Factor AveR3F, and the chunks found for A and S Groups are based on only one participant. When all the participants are combined in C Group, no chunk is found using Factor AveR3F. This shows that none of the sections (or information) of ASM software architecture document stands out as needed by the group as a whole, when the group attempted ASM Task 3.

12.7 **Summary**

This chapter describes the chunking of architectural information for all 6 information-seeking tasks of Study 3. The results show the discovery of chunks based on consumers' usage of a document, for each of the tasks. The next chapter contains the comparison of the results for all the 6 information-seeking tasks, the summary of the overall results for Studies 2 and 3, and threats to validity of findings.

13 Summary and Discussion of Chunking Results

This chapter presents the overall results of our investigation into the chunking of architectural information in ADs, based on consumers' usages of the architectural information in the documents when engaged with certain information-seeking tasks. This chapter starts with the answering of our research questions by drawing upon the findings from both Studies 2 and 3. Next the chapter presents the summaries of how 'good' the discovered chunks are for the respective task, chunk-identification factors that produced better chunks, best chunks and factors that produced the best chunks, goodness of best chunks, and information-seeking tasks having better benchmarking results of chunks. This chapter also presents our own assessment of our approach (boundary values used in the assessment of the goodness of chunks; and correlation between size of chunk, recall and precision measures). This is followed by the comparison between the different groups of participants in terms of chunks found and information needed; the comparison of the six oracle sets; totally-excluded oracle set's sections and totally-included false sections; the threats that might have affected the validity of our findings in all three studies. The chapter then compares our studies on chunking with the related work in the literature, and concludes with the discussion of the advantages and limitations of our approach.

13.1 Answering of Research Questions

Recall that a chunk is defined in this research as below:

A chunk for an information-seeking task performed on a software architecture document is a collection of related pieces of architectural information needed for the task by the majority of a group of users. Specifically it comprises section(s) of a software architecture document needed for the task by the majority of a group of users.

Our research questions are:

RQ1: Do chunks exist?

If yes,

RQ2: Do chunks contain information (or sections of document) compulsory for information-seeking tasks?

RQ3: Do chunks contain information (or sections of document) not needed for information-seeking tasks?

To find the chunk, we used chunk-identification factors that made use of participants' usage data of the ADs we collected when they attempted certain information-seeking tasks. In Study 2 (Manual Approach Study), we used six factors to find chunks for total combined group (C) and the two sub-groups of industry practitioner (I) and academic (A). For Study 3 (Online Approach Study), we used three factors to find chunks for C Group, I and A sub-groups and the sub-group of students (S). In both studies, two ADs were involved, with three information-seeking tasks specified for each document.

RQ1

Most of the factors produced chunks, except for those cases denoted by '/' in Table 13.1 and Table 13.2. I, A, S and C refer to industry, academic, student sub-groups and combined group, respectively. S sub-group only existed in Study 3 and there is no S sub-group for ASM Task 2. Taking WCT Task 1 from Study 2 (Table 13.1) as an example, no chunk is found for A and C Groups when Factor H is used.

For each task, the number of cases where no chunk is found is low. For Study 2, the most is 5 of 18 cases (i.e. 6 factors for 3 groups of participants) or 27.78% for ASM Task 3. For Study 3, the most is 2 of 12 cases (3 factors for 4 groups) or 16.67% for ASM Task 3. Regardless of that, **chunks are found for each group of participants involved in the three information-seeking tasks specified for the two ADs, in both studies. So the answer to RQ1 is 'yes'.**

RQ2

We benchmarked the chunks found for an information-seeking task against an oracle set and calculated their recall and precision measures. The oracle set contains all the sections of the AD that are compulsory for the task. It was constructed using a vigorous relevance judgments process as explained in Chapter 5.

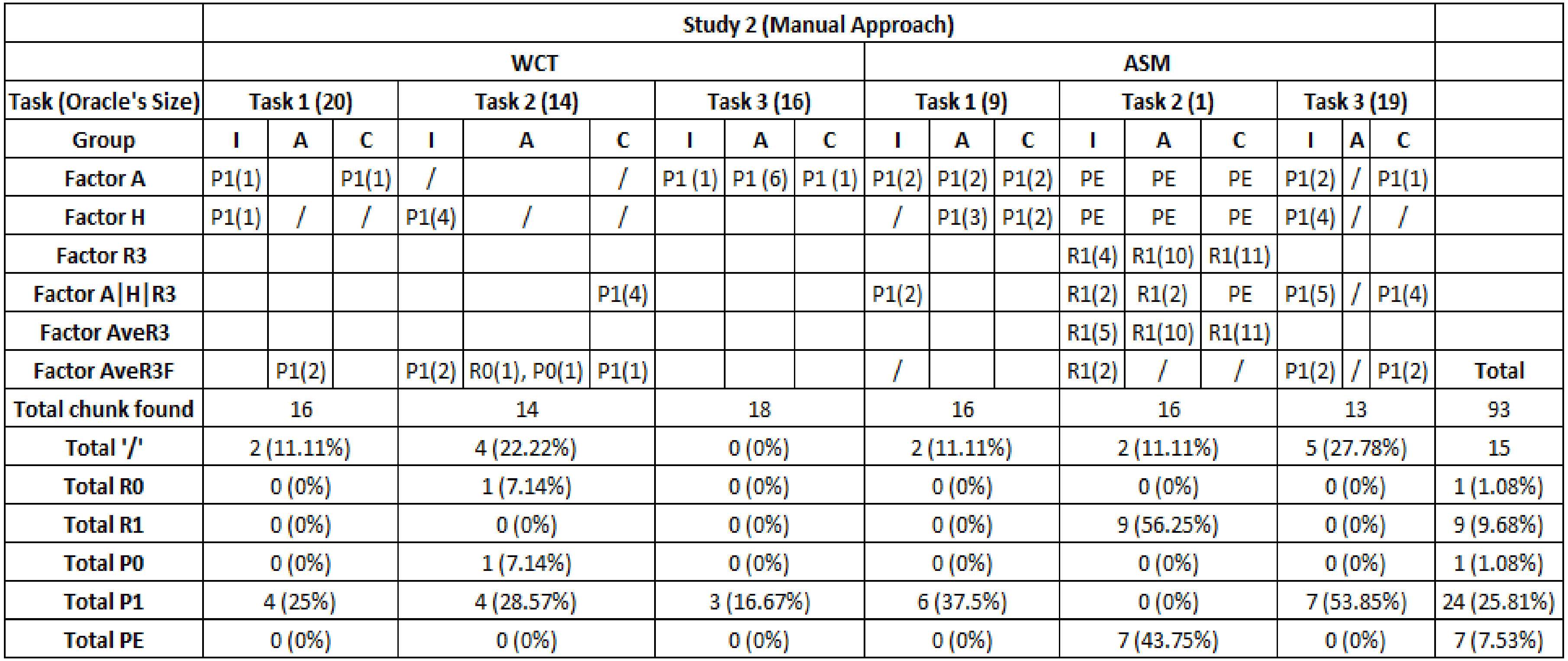
Recall measures of chunks are used to answer RQ2. Recall measure reveals the percentage of the number of sections in the oracle set that a chunk comprises.

Chunks with recall 0.0: The recall measures of all the chunks found regardless of factor used, task, AD, and study (Study 2 or 3) are more than 0.0 with the exception of the following three chunks (denoted by 'R0' in Table 13.1 and 13.2) which have recall measures of 0.0 (and also precision measures of 0.0):

a) Chunk for A Group found using Factor AveR3F for WCT Task 2 in Study 2 (size 1, recall and precision measures 0.0). It contains a section not relevant to the task.

b) A and C Groups' chunks found using Factor AveR3F for WCT Task 1 in Study 3 (size 1, recall and precision measures 0.0). They contain introductory sections not compulsory for the task.

Thus, all the discovered chunks except 1.08% (1) of the 93 chunks found for Study 2 and 3.08% (2) of the 65 chunks found for Study 3, contain sections compulsory for the respective task. So, w**ith reference to our oracle sets, our answer to RQ2 is 'yes'. In other words, most of the chunks found contain information or sections of document compulsory for the respective task.**



	Study 2 (Manual Approach)																		
	WCT									ASM									
Task (Oracle's Size)	Task 1 (20)			Task 2 (14)			Task 3 (16)			Task 1 (9)			Task 2 (1)			Task 3 (19)			
Group	I	A	C	I	A	C	I	A	C	I	A	C	I	A	C	I	A	C	
Factor A	P1(1)		P1(1)	/		/	P1 (1)	P1 (6)	P1 (1)	P1(2)	P1(2)	P1(2)	PE	PE	PE	P1(2)	/	P1(1)	
Factor H	P1(1)	/	/	P1(4)	/	/				/	P1(3)	P1(2)	PE	PE	PE	P1(4)	/	/	
Factor R3													R1(4)	R1(10)	R1(11)				
Factor A|H|R3						P1(4)				P1(2)			R1(2)	R1(2)	PE	P1(5)	/	P1(4)	
Factor AveR3													R1(5)	R1(10)	R1(11)				
Factor AveR3F		P1(2)		P1(2)	R0(1), P0(1)	P1(1)				/			R1(2)	/	/	P1(2)	/	P1(2)	Total
Total chunk found	16			14			18			16			16			13			93
Total '/'	2 (11.11%)			4 (22.22%)			0 (0%)			2 (11.11%)			2 (11.11%)			5 (27.78%)			15
Total R0	0 (0%)			1 (7.14%)			0 (0%)			0 (0%)			0 (0%)			0 (0%)			1 (1.08%)
Total R1	0 (0%)			0 (0%)			0 (0%)			0 (0%)			9 (56.25%)			0 (0%)			9 (9.68%)
Total P0	0 (0%)			1 (7.14%)			0 (0%)			0 (0%)			0 (0%)			0 (0%)			1 (1.08%)
Total P1	4 (25%)			4 (28.57%)			3 (16.67%)			6 (37.5%)			0 (0%)			7 (53.85%)			24 (25.81%)
Total PE	0 (0%)			0 (0%)			0 (0%)			0 (0%)			7 (43.75%)			0 (0%)			7 (7.53%)

'/' - No chunk was found

PE- The perfect chunk with recall and precision measures 1.0

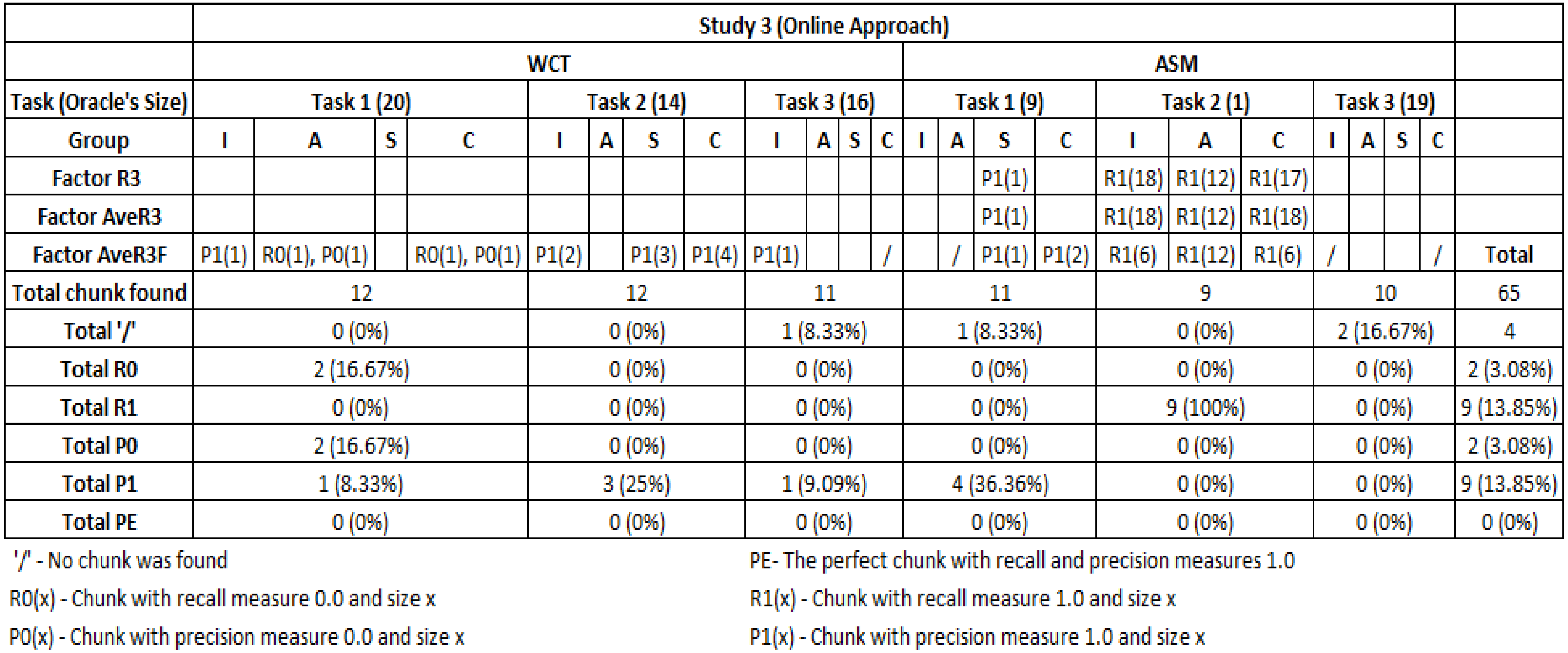
R0(x) - Chunk with recall measure 0.0 and size x

R1(x) - Chunk with recall measure 1.0 and size x

P0(x) - Chunk with precision measure 0.0 and size x

P1(x) - Chunk with precision measure 1.0 and size x

Table 13.1: No Chunk Cases and Special Chunks (Study 2)



	Study 3 (Online Approach)																							
	WCT												ASM											
Task (Oracle's Size)	Task 1 (20)				Task 2 (14)				Task 3 (16)				Task 1 (9)				Task 2 (1)			Task 3 (19)				
Group	I	A	S	C	I	A	S	C	I	A	S	C	I	A	S	C	I	A	C	I	A	S	C	
Factor R3															P1(1)		R1(18)	R1(12)	R1(17)					
Factor AveR3															P1(1)		R1(18)	R1(12)	R1(18)					
Factor AveR3F	P1(1)	R0(1), P0(1)		R0(1), P0(1)	P1(2)		P1(3)	P1(4)	P1(1)			/		/	P1(1)	P1(2)	R1(6)	R1(12)	R1(6)	/			/	Total
Total chunk found	12				12				11				11				9			10				65
Total '/'	0 (0%)				0 (0%)				1 (8.33%)				1 (8.33%)				0 (0%)			2 (16.67%)				4
Total R0	2 (16.67%)				0 (0%)				0 (0%)				0 (0%)				0 (0%)			0 (0%)				2 (3.08%)
Total R1	0 (0%)				0 (0%)				0 (0%)				0 (0%)				9 (100%)			0 (0%)				9 (13.85%)
Total P0	2 (16.67%)				0 (0%)				0 (0%)				0 (0%)				0 (0%)			0 (0%)				2 (3.08%)
Total P1	1 (8.33%)				3 (25%)				1 (9.09%)				4 (36.36%)				0 (0%)			0 (0%)				9 (13.85%)
Total PE	0 (0%)				0 (0%)				0 (0%)				0 (0%)				0 (0%)			0 (0%)				0 (0%)

'/' - No chunk was found

R0(x) - Chunk with recall measure 0.0 and size x

P0(x) - Chunk with precision measure 0.0 and size x

PE- The perfect chunk with recall and precision measures 1.0

R1(x) - Chunk with recall measure 1.0 and size x

P1(x) - Chunk with precision measure 1.0 and size x

Table 13.2: No Chunk Cases and Special Chunks (Study 3)

Chunks with recall 1.0: Chunks that contain all the sections from the respective oracle set for a task (i.e. those with recall measures 1.0) are found for ASM Task 2 only. These chunks are denoted by 'R1' in Table 13.1 and 13.2. In fact all the chunks found for ASM Task 2 in Studies 2 and 3, have recall measures of 1.0.

Perfect Chunks: Chunks found using Factors A and H for I, A and C Groups, and also Factor A|H|R3 for C Group who attempted ASM Task 2 in Study 2, have recall and precision of 1.0. These chunks are identical. This chunk is the perfect chunk (denoted by "PE' in Table 13.1) for ASM Task 2. With 100% recall and precision, it contains all the sections from the oracle set, and contains only these sections. In our studies, a perfect chunk is found only for ASM Task 2 which has a very small oracle set consisting of one section. It is found for ASM Task 2 in Study 2 but not Study 3. **Probable reason is to arrive at perfect chunk (despite that it comprises only one section), usage data of participants with stronger SA background (such as in Study 2) is required. Inspection of participants' background shows that participants involved in ASM Task 2 in Study 2 indeed have much longer number of years of experience in SA than those involved with the same task in Study 3.**

In summary, of the 93 chunks found for Study 2, only 9.68% (9) of them have recall measures of 1.0 but generally with low precision measures between 0.09 to 0.25 (though three of them have precision measures 0.50), and another 7.53% (7) of them have recall measures of 1.0 and precision measures of 1.0 (i.e. the perfect chunk). Of the 65 chunks found for Study 3, only 13.85% (9) of them have recall measures of 1.0 but with low precision measures in between 0.06 to 0.17.

RQ3

Precision measures of chunks are used to answer RQ3. Precision measure reveals the percentage of the total number of sections in a chunk that match with sections in the oracle set. These sections are compulsory for the task and therefore needed for the task. The rest of the sections in the chunk are not needed for the task. The higher the precision measure of a chunk, the fewer the sections not needed for the task, are in the chunk.

Chunks with precision 1.0: We inspected chunks with precision measures 1.0 as they contain only sections from the respective oracle set and do not contain section not needed for the task. These are denoted by 'P1' in Table 13.1 and Table 13.2. A number of them are found in Study 2: 4 (25% of the chunks discovered for the task), 4 (28.57%), 3 (16.67%), 6 (37.5%) and 7 (53.85%) for WCT Task 1, Task 2, Task 3, ASM Task 1 and Task 3, respectively. As for ASM Task 2 in Study 2, there are 7 (43.75%) chunks with precision measures of 1.0. These chunks also have recall measures of 1.0 and they are the perfect chunks (denoted by 'PE' in Table 13.1). For Study 3, the numbers of chunks with precision measures of 1.0 are: 1 (8.33% of the chunks discovered for the task), 3 (25%), 1 (9.09%) and 4 (36.36%) for WCT Task 1, Task 2, Task 3 and ASM Task 1, respectively.

In summary, of the 93 chunks found for Study 2, only 25.81% (24) of them have precision measures of 1.0 (but generally with low recall measures from 0.05 to 0.38), and another 7.53% (7) of them have precision and also recall measures of 1.0 (i.e. the perfect chunk). Of the 65 chunks found for Study 3, only 13.85% (9) of them have precision measures of 1.0 (but generally with low recall measures from 0.05 to 0.29).

The findings show that a significant number of chunks found (i.e. 66.66% in Study 2 and 86.15% in Study 3) do not have 100% precision. **Therefore, with reference to our oracle sets, our answer to RQ3 is 'yes'. In other words, a significant number of chunks found contain section(s) not needed for the respective task.**

Chunks with precision 0.0: A chunk with precision measure 0.0 contains no section from the respective oracle set and all its section are not needed for the task. Such chunks are denoted by 'P0' in Table 13.1 and 14.2. There are only three chunks with precision 0.0 and their recall measures are also 0.0. Refer to '**Chunks with recall 0.0'** explained earlier.

Summary: Almost all the chunks (i.e. 98.92% and 96.92% of chunks found in Studies 2 and 3, respectively) contain some compulsory sections, namely sections from the respective oracle set. These chunks have recall measures greater than 0.0. The number of times of finding chunk that contains all the sections from the respective oracle set (i.e. those with recall 1.0) is small (i.e. 17.21% and 13.85% of chunks found in Studies 2 and 3, respectively). They are found for ASM Task 2 only. The finding of the perfect chunk (i.e. those with recall and precision of 1.0) that matches the respective oracle set exactly is even less frequent and happens only for ASM Task 2 in Study 2 (i.e. only 7.53% of chunks found in Study 2).

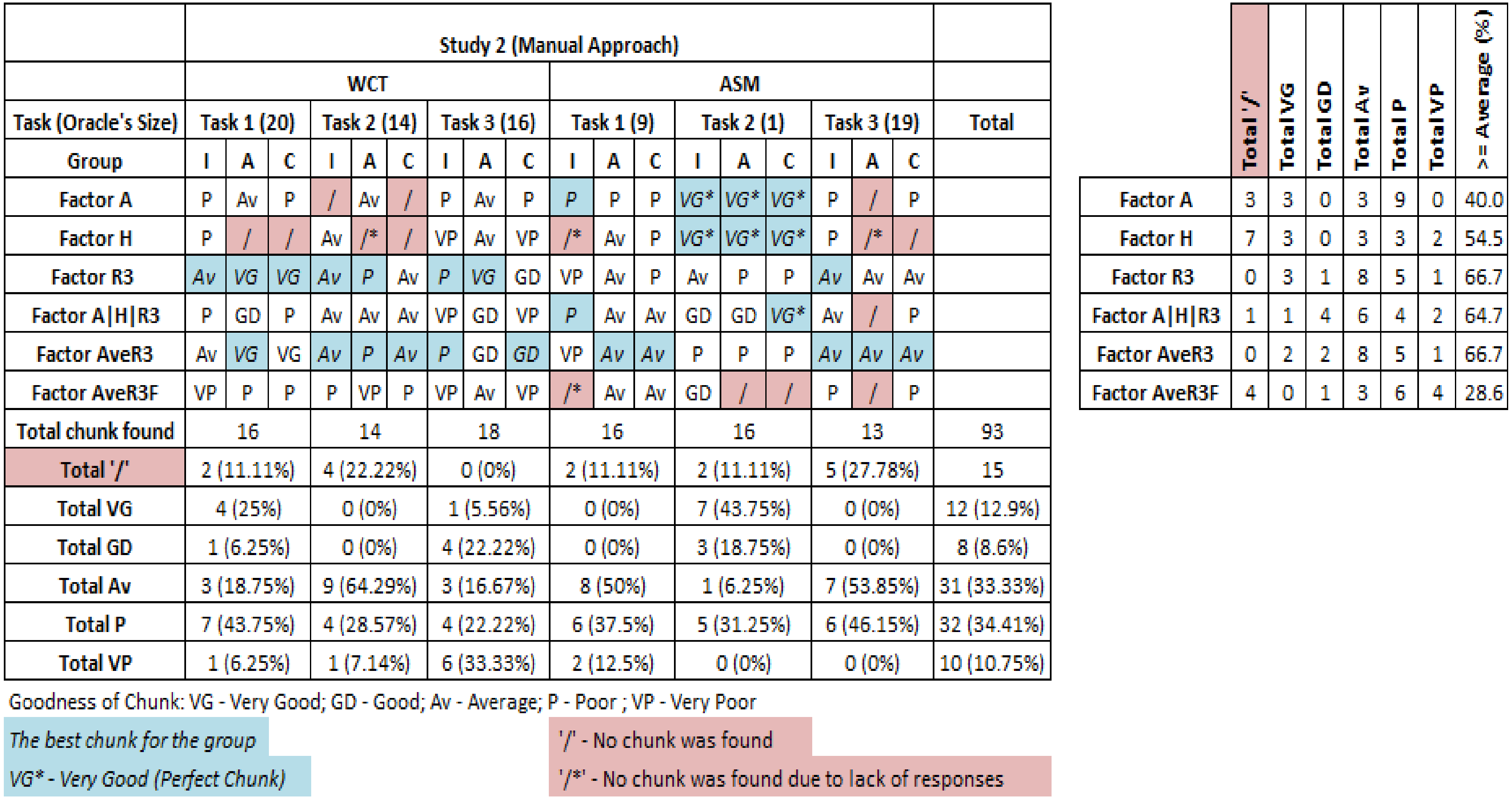
A significant number of chunks found (i.e. 66.66% in Study 2 and 86.15% in Study 3) do not have 100% precision. In other words, they contain section(s) not needed for the respective task. However, we did find chunks that have precision 1.0 that do not contain section not needed for the task. At the same time, chunks which are totally different from their respective oracle set (i.e. containing only sections not needed for a task and having recall and precision of 0.0) are very rare (i.e. 1.08% and 3.08% of chunks found in Studies 2 and 3, respectively).

13.2 **Goodness of Chunks**

Our studies show that chunks based on consumers' usage data of ADs exist. Almost all the discovered chunks contain some compulsory sections for the respective task (i.e. recall measures more than 0.0). However, at the same time a significant number of the discovered chunks contain some sections not needed for the respective task (i.e. precision measures less than 1.0). This means that amidst the sections compulsory for the respective task, sections not needed for the task are also presented in chunks and this happened for a significant number of the discovered chunks.

As mentioned in Section "5.7.3 Criteria to Assess the Goodness of a Chunk" in Chapter 5, the information-seeking tasks in our studies require chunks with high recall measures but acceptable chunks for each task should at the same time do not have too low precision measures (i.e. contain too many 'not needed' sections). Consequently, we used criteria which take into consideration the trade-off between recall and precision measures to determine how 'good' the discovered chunks are, for the respective task. These criteria are explained in Section "5.7.3 Criteria to Assess the Goodness of a Chunk" in Chapter 5.

Table 13.3 and 13.4 show the goodness of the chunks found in Study 2 and 3, respectively.



	Study 2 (Manual Approach)																		
	WCT									ASM									
Task (Oracle's Size)	Task 1 (20)			Task 2 (14)			Task 3 (16)			Task 1 (9)			Task 2 (1)			Task 3 (19)			Total
Group	I	A	C	I	A	C	I	A	C	I	A	C	I	A	C	I	A	C	
Factor A	P	Av	P	/	Av	/	P	Av	P	*P*	P	P	*VG**	*VG**	*VG**	P	/	P	
Factor H	P	/	/	Av	/*	/	VP	Av	VP	/*	Av	P	*VG**	*VG**	*VG**	P	/*	/	
Factor R3	*Av*	*VG*	*VG*	*Av*	*P*	Av	*P*	*VG*	GD	VP	Av	P	Av	P	P	*Av*	Av	Av	
Factor A|H|R3	P	GD	P	Av	Av	Av	VP	GD	VP	*P*	Av	Av	GD	GD	*VG**	Av	/	P	
Factor AveR3	Av	*VG*	VG	*Av*	*P*	*Av*	*P*	GD	*GD*	VP	*Av*	*Av*	P	P	P	*Av*	*Av*	*Av*	
Factor AveR3F	VP	P	P	P	VP	P	VP	Av	VP	/*	Av	Av	GD	/	/	P	/	P	
Total chunk found	16			14			18			16			16			13			93
Total '/'	2 (11.11%)			4 (22.22%)			0 (0%)			2 (11.11%)			2 (11.11%)			5 (27.78%)			15
Total VG	4 (25%)			0 (0%)			1 (5.56%)			0 (0%)			7 (43.75%)			0 (0%)			12 (12.9%)
Total GD	1 (6.25%)			0 (0%)			4 (22.22%)			0 (0%)			3 (18.75%)			0 (0%)			8 (8.6%)
Total Av	3 (18.75%)			9 (64.29%)			3 (16.67%)			8 (50%)			1 (6.25%)			7 (53.85%)			31 (33.33%)
Total P	7 (43.75%)			4 (28.57%)			4 (22.22%)			6 (37.5%)			5 (31.25%)			6 (46.15%)			32 (34.41%)
Total VP	1 (6.25%)			1 (7.14%)			6 (33.33%)			2 (12.5%)			0 (0%)			0 (0%)			10 (10.75%)

Goodness of Chunk: VG - Very Good; GD - Good; Av - Average; P - Poor ; VP - Very Poor

The best chunk for the group

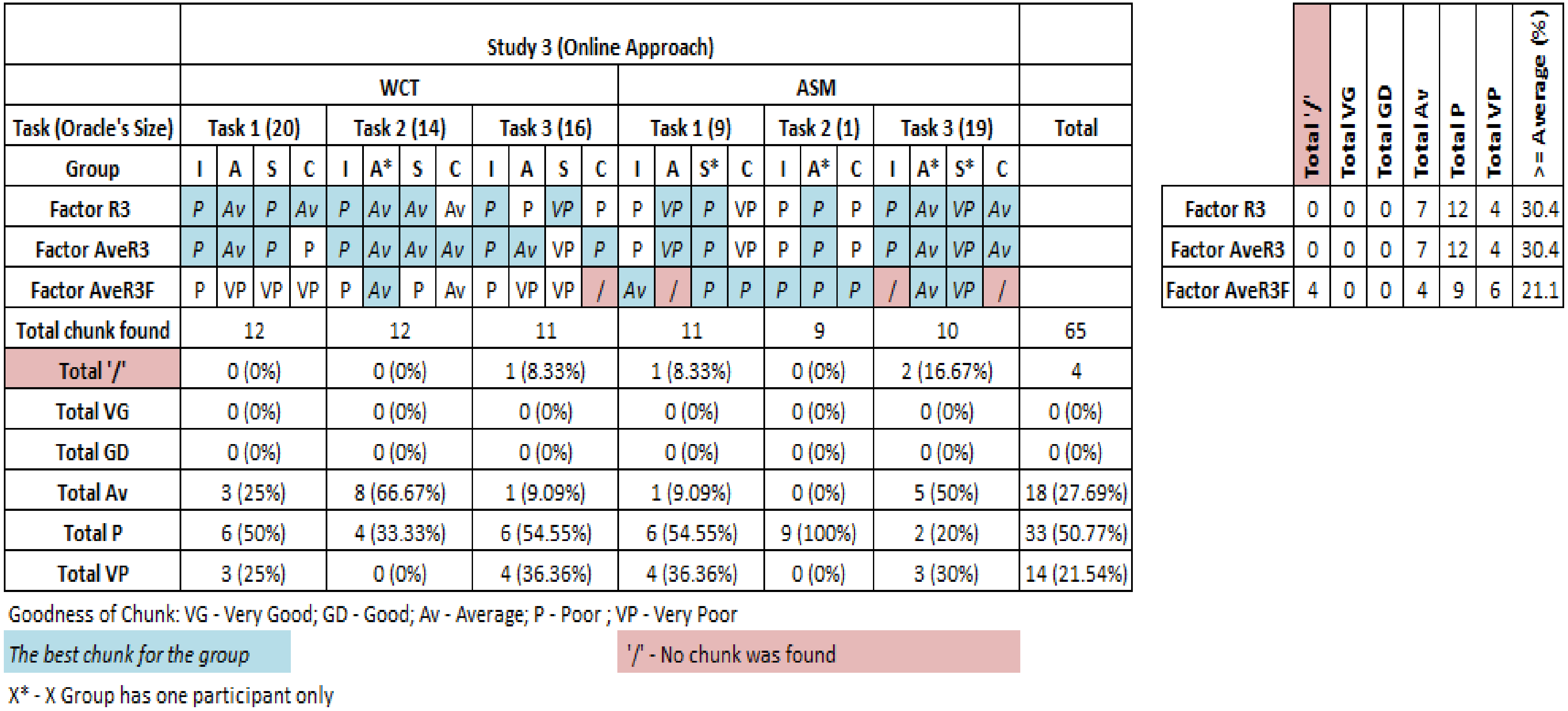
VG - Very Good (Perfect Chunk)*

'/' - No chunk was found

'/*' - No chunk was found due to lack of responses

	Total '/'	Total VG	Total GD	Total Av	Total P	Total VP	>= Average (%)
Factor A	3	3	0	3	9	0	40.0
Factor H	7	3	0	3	3	2	54.5
Factor R3	0	3	1	8	5	1	66.7
Factor A|H|R3	1	1	4	6	4	2	64.7
Factor AveR3	0	2	2	8	5	1	66.7
Factor AveR3F	4	0	1	3	6	4	28.6

Table 13.3: Goodness of Chunks and Best Chunks (Study 2)



	Study 3 (Online Approach)																							
	WCT												ASM											
Task (Oracle's Size)	Task 1 (20)				Task 2 (14)				Task 3 (16)				Task 1 (9)				Task 2 (1)			Task 3 (19)				Total
Group	I	A	S	C	I	A*	S	C	I	A	S	C	I	A	S*	C	I	A*	C	I	A*	S*	C	
Factor R3	*P*	*Av*	*P*	*Av*	*P*	*Av*	*Av*	Av	*P*	P	*VP*	P	P	*VP*	*P*	VP	P	*P*	P	*P*	*Av*	*VP*	*Av*	
Factor AveR3	*P*	*Av*	*P*	P	*P*	*Av*	*Av*	*Av*	*P*	*Av*	VP	*P*	P	*VP*	*P*	VP	P	*P*	P	*P*	*Av*	*VP*	*Av*	
Factor AveR3F	P	VP	VP	VP	P	*Av*	P	Av	P	VP	VP	*/*	*Av*	*/*	*P*	*P*	*P*	*P*	*P*	*/*	*Av*	*VP*	*/*	
Total chunk found	12				12				11				11				9			10				65
Total '/'	0 (0%)				0 (0%)				1 (8.33%)				1 (8.33%)				0 (0%)			2 (16.67%)				4
Total VG	0 (0%)				0 (0%)				0 (0%)				0 (0%)				0 (0%)			0 (0%)				0 (0%)
Total GD	0 (0%)				0 (0%)				0 (0%)				0 (0%)				0 (0%)			0 (0%)				0 (0%)
Total Av	3 (25%)				8 (66.67%)				1 (9.09%)				1 (9.09%)				0 (0%)			5 (50%)				18 (27.69%)
Total P	6 (50%)				4 (33.33%)				6 (54.55%)				6 (54.55%)				9 (100%)			2 (20%)				33 (50.77%)
Total VP	3 (25%)				0 (0%)				4 (36.36%)				4 (36.36%)				0 (0%)			3 (30%)				14 (21.54%)

	Total '/'	Total VG	Total GD	Total Av	Total P	Total VP	>= Average (%)
Factor R3	0	0	0	7	12	4	30.4
Factor AveR3	0	0	0	7	12	4	30.4
Factor AveR3F	4	0	0	4	9	6	21.1

Goodness of Chunk: VG - Very Good; GD - Good; Av - Average; P - Poor ; VP - Very Poor

The best chunk for the group

'/' - No chunk was found

X* - X Group has one participant only

Table 13.4: Goodness of Chunks and Best Chunks (Study 3)

In Study 2 (Table 13.3), twelve (12.9%) of 93 chunks found have ‘Very Good’ goodness measures: WCT Task 1 (4 chunks), WCT Task 3 (1 chunk) and ASM Task 2 (7 chunks). Eight (8.6%) of the chunks found have ‘Good’ goodness measures: WCT Task 1 (1 chunk), WCT Task 3 (4 chunks) and ASM Task 2 (3 chunks). The results show that the ‘Very Good’ and Good’ chunks are found for the same tasks namely, WCT Task 1, WCT Task 3 and ASM Task 2.

In Study 2, one third of the chunks found are of ‘Average’ goodness measures: WCT Task 1 (3 chunks), WCT Task 2 (9 chunks), WCT Task 3 (3 chunks), ASM Task 1 (8 chunks), ASM Task 2 (1 chunk) and ASM Task 3 (7 chunks). Another about one third of the chunks are of ‘Poor’ goodness measures: WCT Task 1 (7 chunks), WCT Task 2 (4 chunks), WCT Task 3 (4 chunks), ASM Task 1 (6 chunks), ASM Task 2 (5 chunks) and ASM Task 3 (6 chunks). Ten (10.75%) of the chunks found are of ‘Very Poor’ goodness measures: WCT Task 1 (1 chunk), WCT Task 2 (1 chunk), WCT Task 3 (6 chunks) and ASM Task 1 (2 chunks).

In Study 3 (Table 13.4), there is no ‘Very Good’ or ‘Good’ chunk. Eighteen (27.69%) of 65 chunks found in Study 3 are ‘Average’: WCT Task 1 (3 chunks), WCT Task 2 (8 chunks), WCT Task 3 (1 chunk), ASM Task 1 (1 chunk) and ASM Task 3 (5 chunks). About half of the chunks found are of ‘Poor’ goodness measures: WCT Task 1 (6 chunks), WCT Task 2 (4 chunks), WCT Task 3 (6 chunks), ASM Task 1 (6 chunks), ASM Task 2 (9 chunks) and ASM Task 3 (2 chunks). Fourteen (21.54%) of the chunks found are ‘Very Poor’: WCT Task 1 (3 chunks), WCT Task 3 (4 chunk), ASM Task 1 (4 chunks) and ASM Task 3 (3 chunks).

Recall from our answering of RQ1 that for each task, the number of cases where no chunk is found is low. The ‘no chunk’ cases are denoted by ‘/’ or ‘/*’ in Table 13.3, and ‘/’ in Table 13.4. In Study 2, there are 15 of 108 cases where no chunk is found: WCT Task 1 (2 cases of 18 cases), WCT Task 2 (4 cases), ASM Task 1 (2 cases), ASM Task 2 (2 cases) and ASM Task 3 (5 cases). Four of these cases (denoted by ‘/*’ in Table 13.3) are due to a lack of the sought responses from the participants in the respective group (i.e. half or more of the participants did not provide the sought responses at all). In Study 3, there are four of 69 cases where no chunk is found: WCT Task 3 (1 of 12 cases), ASM Task 1 (1 case) and ASM Task 3 (2 cases).

In summary, 54.83% of the chunks found in Study 2 have goodness measures of average and above, whereas only 27.69% of the chunks found in Study 3 have goodness measures of average (which is the best level of goodness measure for chunks in Study 3). The rest of the chunks in both studies have below average goodness measures. In other words, **in terms of how ‘good’ the discovered chunks are for the respective task, only slightly more than half of the chunks found in Study 2 are of ‘average’ or above goodness, and only slightly more than a quarter of the chunks found in Study 3 are of ‘average’ goodness, for their respective tasks. Could the aforementioned situation be due to some of the chunk-identification factors are not suitable for finding chunks, in other words they are not able to find chunks with at least ‘average’ goodness measures? We address this issue in the next section of this chapter.**

As for the significantly fewer chunks with goodness measures of average or above in Study 3, it could be due to participants in Study 3 are made up of both ‘novices’ and ‘experts’ in SA, whereas participants in Study 2 are all ‘experts’ in SA. Novices were those with less than two years and experts were those with at least two years of experience in SA. Another reason could be reading behaviour is different on line (Study 3) compared to with local electronic or printed environment (Study 2), with on line reading promoting a more superficial approach to understanding, and therefore reducing the goodness of chunks found in Study 3.

13.3 **Factors that Produced Better Chunks**

How many chunks with goodness measures of average and above are produced by each factor? The answers in decreasing percentages for Study 2 (Table 13.3) are: Factors R3 and AveR3 (66.7% or 12 of 18 chunks), Factor A|H|R3 (64.7% or 11 of 17 chunks), Factor H (54.5% or 6 of 11 chunks), Factor A (40% or 6 of 15 chunks), and Factor AveR3F (28.6% or 4 of 14 chunks). The answers in decreasing percentages for Study 3 (Table 13.4) are: Factors R3 and AveR3 (30.4% or 7 of 23 chunks) and Factor AveR3F (21.1% or 4 of 19 chunks).

Which factors are more inclined to produce no chunk? The answers for Study 2 are Factor H (7 'no chunk' cases) followed by Factor AveR3F (4 cases), Factor A (3 cases) and Factor A|H|R3 (1 case). Three 'no chunk' cases produced by Factor H and one 'no chunk' case produced by Factor AveR3F, are due to a lack of the sought responses from the participants in the respective group. As for Study 3, the four 'no chunk' cases are all produced by Factor AveR3F.

In summary, two-third of the chunks produced by Factors R3 and AveR3 in Study 2 have goodness measures of average and above. Though almost two-third (64.7%) of chunks produced by Factor A|H|R3 also have goodness measures of average and above, this factor produced one 'no chunk' case. The other three factors (i.e. Factors A, H and R3) have more tendencies of producing below average chunks or no chunk. As for Study 3, one-third of the chunks produced by Factors R3 and AveR3 have goodness measures of average. About one-fifth of the chunks produced by Factor AveR3F have goodness measures of average and all the four 'no chunk' cases are produced by this factor. In Study 3, all the three factors (i.e. Factors R3 and AveR3 and AveR3F) have more tendencies of producing below average chunks.

The lower tendencies of Factors R3 and AveR3 in producing chunks with goodness measures of average and above in Study 3 comparing to Study 2, could be due to both 'novices' and 'experts' in SA participated in Study 3, but only 'experts' in SA participated in Study 2. The other possible reason is reading behaviour is different on line compared to with local electronic or printed environment, with on line reading promoting a more superficial approach to understanding. Nevertheless, the results show that across different tasks and different participants, the numbers of better chunks (i.e. those with goodness measures of average and above) produced by Factors R3 and AveR3 (and also Factor A|H|R3 in Study 2) are relatively more than other factors. At the same time, none of the 'no chunk' cases are produced by Factors R3 and AveR3. Consequently, Factors R3 and AveR3 are generally more suitable than other factors for finding better chunks. Although Factor A|H|R3 in Study 2 also shows the potential of finding better chunks, it makes use of two other forms of participants' responses (i.e. specifying the sections where the answer was found or search from, and highlighting relevant information in the section) which required more effort from the participants.

13.4 **Factors that Produced Best Chunks**

Recall that for each group of participants, we compared the composition of the chunks that are at the best level of goodness measure to decide the best (or the most favourable) chunk for a particular group. We did so by assessing the criticality of the inclusion and the omission of the different sections in the chunks, to the task. This is explained in Section “5.7.4 The Best Chunk and The Best Chunk-Identification Factor” in Chapter 5.

The goodness measures of the best chunks in Study 2 and 3 are presented in ‘italic’ format and shaded with ‘light blue’ background in Table 13.3 and Table 13.4, respectively. Recall that factor that produces the best chunk for a group is the best chunk-identification factor for the group, for the respective task.

For WCT Task 1, WCT Task 2 and WCT Task 3, and ASM Task 3 in Study 2, the best chunk for each group of participants is found using either Factor R3 or AveR3, or both. For ASM Task 1, the best chunk for I Group is discovered using Factors A and A|H|R3 (which produce identical chunk), and the best chunks for A and C Groups are found using Factor AveR3. For ASM Task 2, the best chunks (which are in fact the ‘perfect chunk’) are discovered using Factors A and H for all three groups of participants and also Factor A|H|R3 for C Group. In Study 2, the frequencies of each factor producing best chunk are 11, 8, 4, 3, 2 for Factors AveR3, R3, A, H and A|H|R3, respectively.

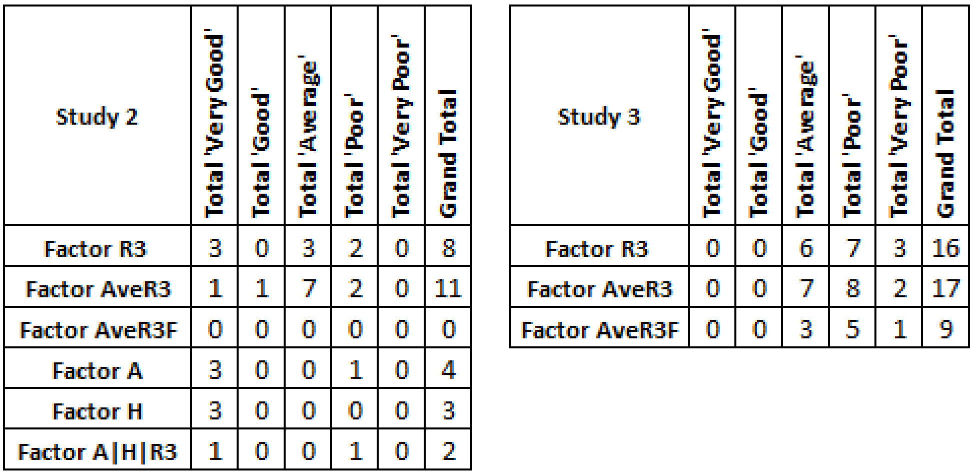
Recall that in Study 3, chunks produced by all three factors for a group of participants are identical when there is only one participant in the group. The identical chunk automatically becomes the best chunk and the three factors appear concurrently as the best chunk-identification factors. In Table 13.4, a group with only one participant is denoted by ‘X*’ where X is the abbreviation of the group. In our discussion in the next two paragraphs, we exclude one-participant groups because the chunks produced from these groups are based on only one participant’s responses.

For WCT Task 1, 2 and 3, and ASM Task 3 in Study 3, the best chunk for each group of participants is found using either Factor R3 or AveR3, or both. This is similar to Study 2. For ASM Task 1 and Task 2, the best chunks for I and C Groups are discovered using Factor AveR3F. This is different from Study 2 where none of the best chunks for I and C Groups attempting ASM Task 1 and Task 2 are found using Factor AveR3F. In Study 3, the frequencies of each factor producing best chunk are 17, 16 and 8 for Factors AveR3, R3 and AveR3F, respectively.

In summary, Factor AveR3 has the highest frequency of producing best chunks in both Studies 2 and 3. This is followed by Factor R3. Recall that in both Studies 2 and 3, Factors R3’s and AveR3’s chunks for a group of participants are very similar if not identical to each other (except for C Group involved in WCT Task 2 in Study 2). Consequently, we are not distinguishing between Factor AveR3 and Factor R3 as to which of them is the most suitable factor for finding chunks but posit that factors based on the preference of the majority of those who rated (be it Factor AveR3 or Factor R3) are suitable for finding chunks. Nevertheless, when the oracle set for an information-seeking task is very small (such as ASM Task 2), factors having more stringent section inclusion criteria based on the preference of the majority of the participants in a group (such as Factor A, Factor H or Factor A|H|R3 in Study 2, or Factor AveR3F in Study 3), are more suitable for finding chunks.

13.5 **Goodness Measures of Best Chunks**

We found that in both studies, the numbers of better chunks (i.e. those with goodness measures of average and above) produced by Factors R3 and AveR3 are relatively more than other factors. In addition in both studies, Factors R3 and AveR3 are at the top of the list in terms of the frequency of each factor in producing best chunks. Here, we narrow down to look at how 'good' the best chunks are for the respective tasks, in particular those produced by Factors R3 and AveR3. Table 13.5 shows the number of best chunks produced by each factor and the distribution of their goodness measures across the five levels, for Studies 2 and 3. For example, 3, 3, 2 of the best chunks produced by Factor R3 are 'Very Good', 'Average' and 'Poor', respectively.



Study 2	Total 'Very Good'	Total 'Good'	Total 'Average'	Total 'Poor'	Total 'Very Poor'	Grand Total
Factor R3	3	0	3	2	0	8
Factor AveR3	1	1	7	2	0	11
Factor AveR3F	0	0	0	0	0	0
Factor A	3	0	0	1	0	4
Factor H	3	0	0	0	0	3
Factor A|H|R3	1	0	0	1	0	2

Study 3	Total 'Very Good'	Total 'Good'	Total 'Average'	Total 'Poor'	Total 'Very Poor'	Grand Total
Factor R3	0	0	6	7	3	16
Factor AveR3	0	0	7	8	2	17
Factor AveR3F	0	0	3	5	1	9

Table 13.5: Number of Best Chunks Produced by Each Factor and the Distribution of Their Goodness Measures (Studies 2 and 3)

As mentioned earlier, in both studies Factors R3 and AveR3 are at the top of the list in terms of the frequencies of them in producing best chunks. In Study 2, 75% (6) of 8 best chunks produced by Factor R3 and 81.8% (9) of 11 best chunks produced by Factor AveR3 have goodness measures of average and above. In Study 3, only 37.5% (6) of 16 best chunks produced by Factor R3 and 41.2% (7) of 17 best chunks produced by Factor AveR3 have goodness measures of average and above. **The findings here further support the suitability of Factors R3 and AveR3 in finding chunks, with the performance of these factors being affected by the participants' background in SA and the different reading behaviours in on line, versus local electronic or printed environments.**

Take note that the 'Very Good' chunks produced by Factors A, H and A|H|R3 in Study 2 are all found for ASM Task 2, which has a very small oracle set. **This re-iterates our earlier point that when the oracle set is very small, factors with more stringent section inclusion criteria based on the preference of the majority of the participants in a group are more suitable for finding chunks.**

13.6 **Tasks Having Better Benchmarking Results of Chunks**

The bottom rows of Table 13.3 and Table 13.4 show the numbers (and percentages) of chunks that fell under each category of goodness measures for each information-seeking task, for Studies 2 and 3, respectively.

For Study 2, ASM Task 2 has the best benchmarking results of chunks, followed by WCT Task 1 and WCT Task 3. ASM Task 2 has the highest percentages of chunks with ‘Very Good’ and ‘Good’ goodness measures, and the lowest percentage of chunks with below average goodness measures. For Study 3, WCT Task 2 has the best benchmarking results of chunks, followed by ASM Task 3 and WCT Task 1. WCT Task 2 has the highest percentage of chunks with ‘Average’ goodness measures (i.e. the best level of goodness measure for chunks in Study 3), and the lowest percentage of chunks with below average goodness measures.

The oracle set of the task with the best benchmarking results has the best matching by chunks. In other words, ASM Task 2’s and WCT Task 2’s oracle set is the best-matched oracle set for Study 2, and Study 3, respectively.

The inconsistent results between the two studies could be due to SA novices also participated in Study 3 while only SA experts participated in Study 2. Another possible reason is reading behaviour is different on line (Study 3) compared to with local electronic or printed environment (Study 2), with on line reading promoting a more superficial approach to understanding.

13.7 **Assessment of Our Approach**

In the following, we discuss our own assessment of our approach.

13.7.1 Boundary Values Used in Assessment of Goodness of Chunks

Recall that the boundary values of the 4 levels of recall and precision measures used in the categorization of the goodness measures of chunks were arbitrarily chosen (refer to Section “5.7.3 Criteria to Assess the Goodness of a Chunk” in Chapter 5). We tried to choose boundary values that are of minimal sensitivity to the categorization of the ‘goodness’ of the chunks in our studies. In other words, if we were to make some small adjustments to these boundary values, the number of the chunks that fell under each category of ‘goodness’ would not be much affected.

We used a set of boundary values with equal ranges (i.e. 0.25) between each successive level of recall and precision: (0, 0.25, 0.50, 0.75, 1.00). Despite that our main purpose of defining the notion of ‘goodness’ of chunks is to help us in making consistent assessment of the chunks found across different information-seeking tasks, we show here that changing the boundary values to a different set (0, 0.30, 0.50, 0.70, 1.00) does not have much effect on our findings.

Figure 13.1 shows the goodness of all the chunks found in Study 2, when the boundary values chosen are (0, 0.25, 0.50, 0.75, 1.00). The numbers of chunk under each goodness category are: ‘Very Good’ (12), ‘Good’ (8), ‘Average’ (31), ‘Poor’ (32) and ‘Very Poor’ (10). There are 15 instances when no chunk is found in Study 2. Figure 13.2 shows the goodness of all the chunks found in Study 2, when the boundary values are changed to (0, 0.30, 0.50, 0.70, 1.00). The number of chunks which are ‘Very Good’ still remains at 12, the number of ‘Good’ chunks increases by 5 to 13 but the number of ‘Average’ chunks decreases by 7 to 24. The number of ‘Poor’ chunks decreases by one to 31, whereas the number of ‘Very Poor’ chunks increases by 3 to 13.

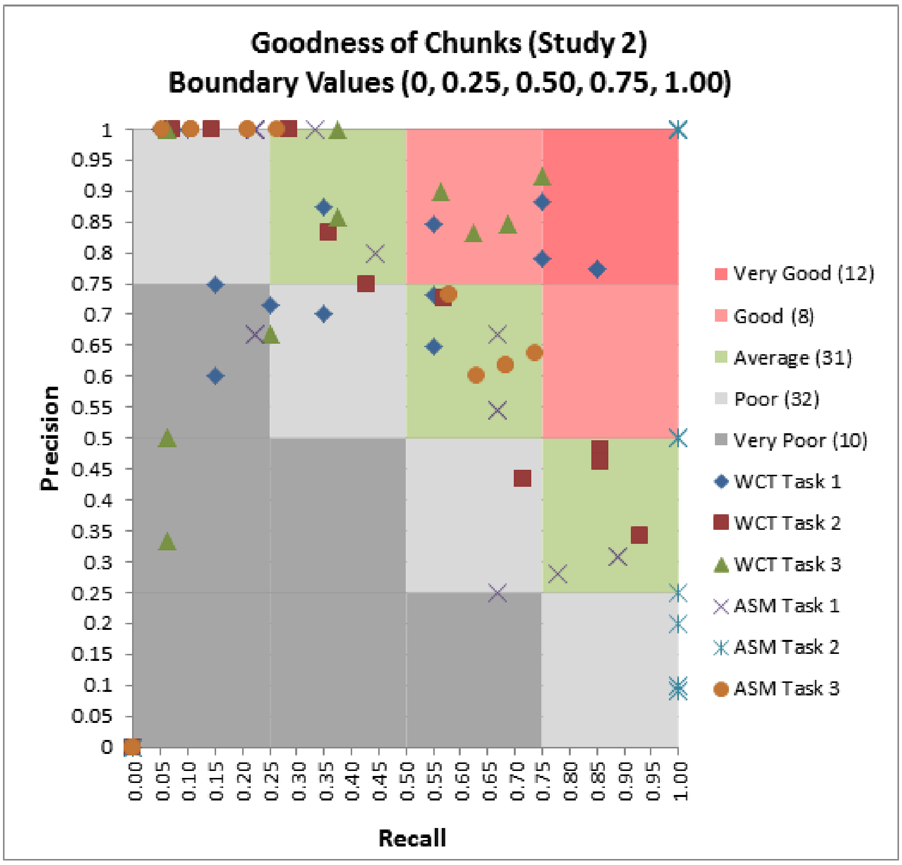


Figure 13.1: Goodness of Chunks (Study 2) – Boundary Values (0, 0.25, 0.50, 0.75, 1.00)

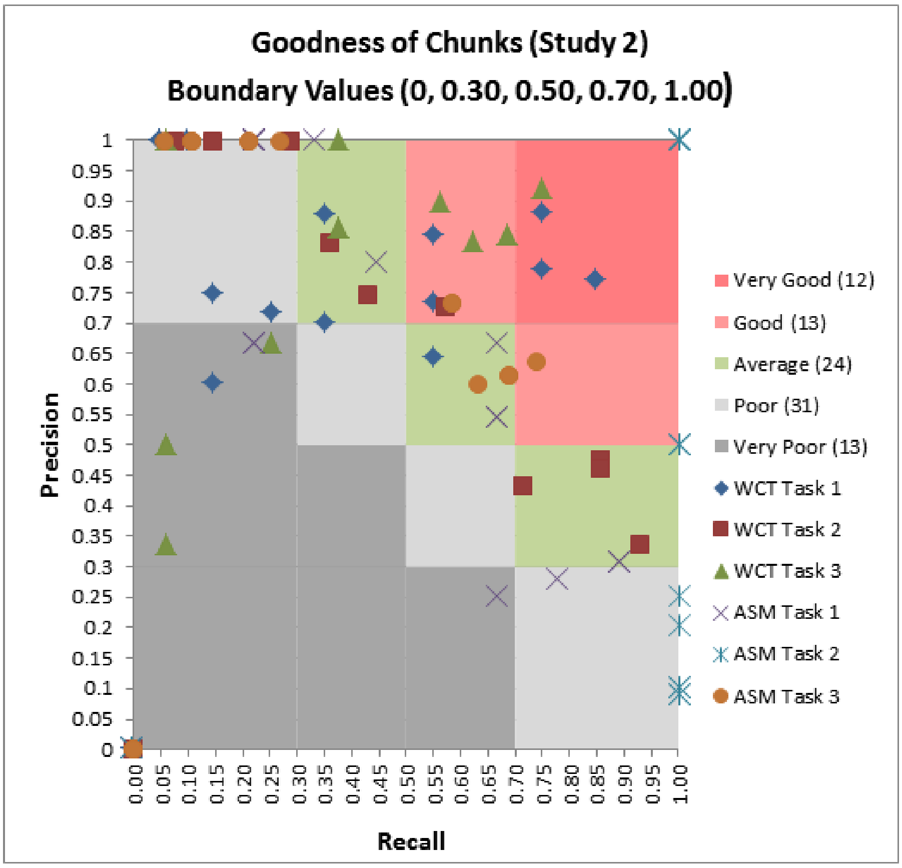


Figure 13.2: Goodness of Chunks (Study 2) – Boundary Values (0, 0.30, 0.50, 0.70, 1.00)

Figure 13.3 shows the goodness of all the chunks found in Study 3, when the boundary values chosen are (0, 0.25, 0.50, 0.75, 1.00). The numbers of chunk under each goodness category are: 'Very Good' (0), 'Good' (0), 'Average' (18), 'Poor' (33) and 'Very Poor' (14). There are 4 instances when no chunk is found in Study 3. Figure 13.4 shows the goodness of all the chunks found in Study 3, when the boundary values are changed to (0, 0.30, 0.50, 0.70, 1.00). The number of chunks which are 'Very Good' still remains at 0, the number of 'Good' chunks increases to 1 but the number of 'Average' chunks decreases by 2 to 16. The number of 'Poor' chunks decreases by 7 to 26, whereas the number of 'Very Poor' increases by 8 to 22.

The changes of the boundary values did cause some fluctuations in the numbers of chunks at the four lower-level of goodness measures. However, the overall impact of the changes of the boundary values on the numbers of chunks with goodness measures of average and above, and those below average are very small. In Study 2, the number of chunks with goodness measures of average and above changes from 51 to 49, and those of below average changes from 42 to 44. In Study 3, the number of chunks with goodness measures of average and above changes from 18 to 17, and those of below average changes from 47 to 48. **This does not affect our findings that Factors R3 and AveR3 are generally more suitable than other factors for finding better chunks (i.e. those with goodness measures of average and above)**

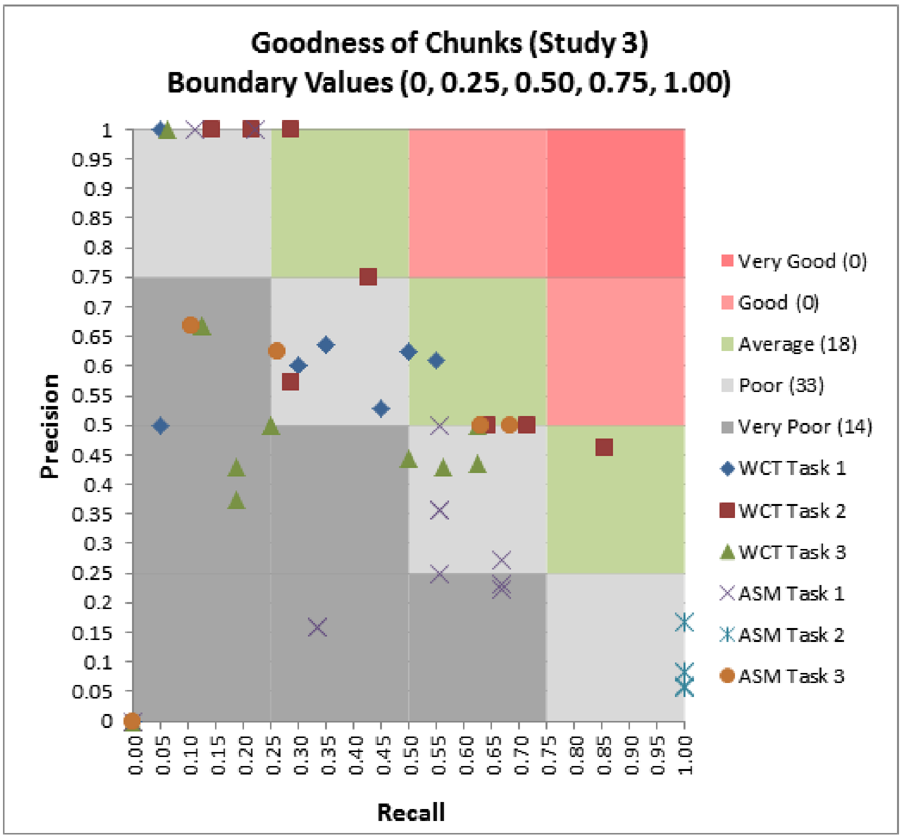


Figure 13.3: Goodness of Chunks (Study 3) – Boundary Values (0, 0.25, 0.50, 0.75, 1.00)

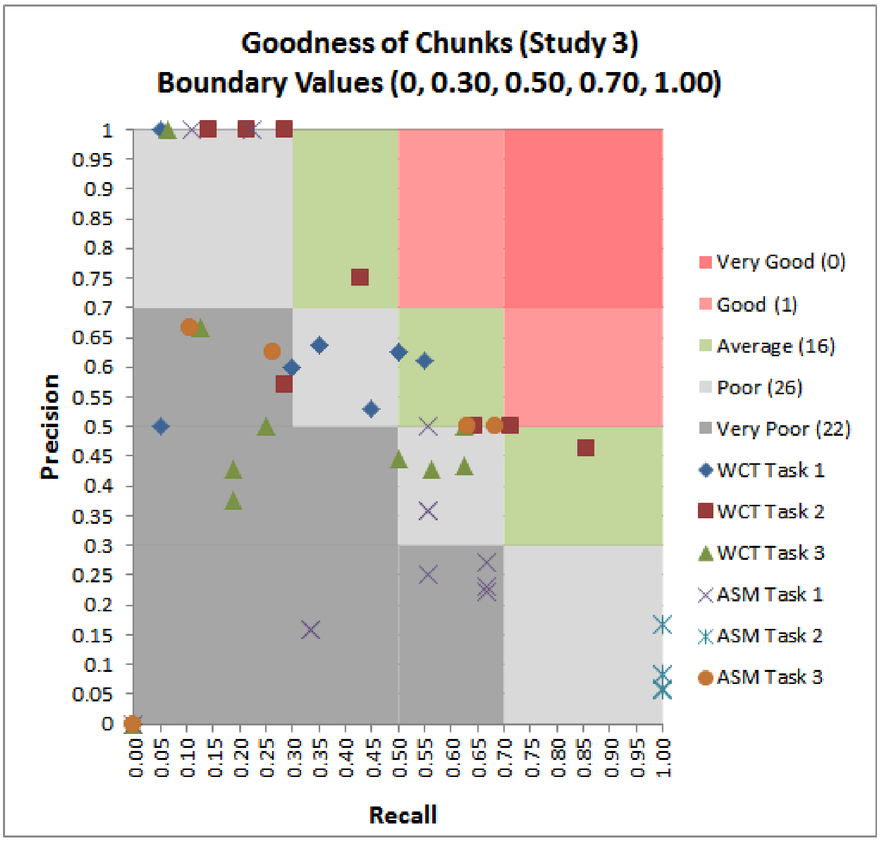


Figure 13.4: Goodness of Chunks (Study 3) – Boundary Values (0, 0.30, 0.50, 0.70, 1.00)

13.7.2 Correlations between Size of Chunk, Recall and Precision Measures

Comparison of chunks found using the same factor for different groups of participants (i.e. Groups I and A in Study 2, and Groups I, A and S in Study 3) show that, recall measure of chunk has a positive correlation with the size of chunk.

For Study 2, a chunk with larger size is generally associated with higher recall measure, and vice versa. There are only 2 instances of recall measures remaining constant with the increase of chunks' sizes (i.e. between I and A Groups' chunks found using Factor R3, and between I and A Groups' chunks found using Factor AveR3, for ASM Task 2). There is no instance of recall measure decreased with the increase of chunk's size. Similar observation is made for Study 3, which is, a chunk with larger size is generally associated with higher recall measure, and vice versa. The larger a chunk's size, the higher the recall measure of the chunk or the recall remained constant. There is only one exception when the recall measure decreased with the increase of the chunk's size (i.e. between A and S Groups' chunks found using Factor AveR3 for ASM Task 1).

We do not observe any correlation between the precision measure of chunk and the size of chunk, for Study 2. For Study 2, there are 10 instances of precision measures decreased with the increase of the chunks' sizes. There is 1 instance of precision measure remained constant with the increase of chunk's size. There are 11 instances of precision measures increased with the increase of the chunks' sizes.

We observe negative correlation between the precision measure of chunk and the size of chunk, for Study 3. In other words, a chunk with larger size is generally associated with lower recall measure, and vice versa. For Study 3, there are 27 instances of precision measures decreased with the increase of the chunks' sizes.

There are 2 instances of precision measures remained constant with the increase of chunks' sizes. There are 12 instances of precision measures increased with the increase of the chunks' sizes.

Our studies show that recall is a non-decreasing function of the size of a chunk. This is in tandem with "Recall is a non-decreasing function of the number of documents retrieved" (Manning et al., 2008) in Information Retrieval. Our studies seem to suggest that recall is dependent on the chunk's size regardless of the particular group of participants involved in producing the chunk (i.e. regardless of whether it is I Group or A Group for Study 2, and whether it is I Group or A Group or S Group for Study 3).

In Information Retrieval, empirical studies have shown a tendency for precision to decline as recall increases (Buckland & Gey, 1994). The two measures are inversely related (Rajaraman et al., 2013) or they trade-off against each other (Manning et al., 2008). Applying this finding to our studies would mean that the precision has the tendency to decline as the recall increases with the increase of the size of chunk, and vice versa. In other words, the precision decreases with the increase of the size of chunk, and vice versa (i.e. a negative correlation between precision measure and the size of chunk).

If the recall measures of chunks found using the same factor are totally dependent on the sizes of the chunks regardless of the groups of participants involved, then we should observe a negative correlation between the precision measures and the sizes of chunks in our studies. The fact that we did not in Study 2 led us to believe that **when participants with strong SA background (i.e. the experts) are involved (as in Study 2), the recall measures of chunks found are not totally dependent on the sizes of chunks but also depend on the groups of participants involved.**

In Study 3, we did observe a negative correlation between the precision measures and the sizes of chunks. This suggests that the involvement of SA novices (together with SA experts) in Study 3 and the possibility of on line reading behaviour (Study 3) promoting a more superficial approach to understanding, could have reduce the influence the groups of participants involved could have on the recall measures of chunks, making the recall measures to be dependent on the sizes of chunks.

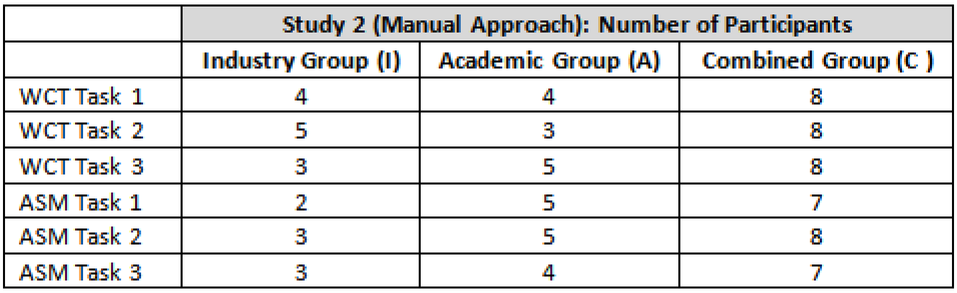
13.8 **Comparison between Different Groups of Participants**

For each information-seeking task, we compare between the different sub-groups of participants (i.e. I and A Groups in Study 2, and I, A and S Groups in Study 3) in terms of the chunks found and the information needed by the groups. Prior to that, we summarise the number of participants in each sub-group.

13.8.1 Number of Participants

Table 13.6 and Table 13.7 show the number of participants whose responses were analysed to find chunks, for each information-seeking task in Studies 2 and 3, respectively.

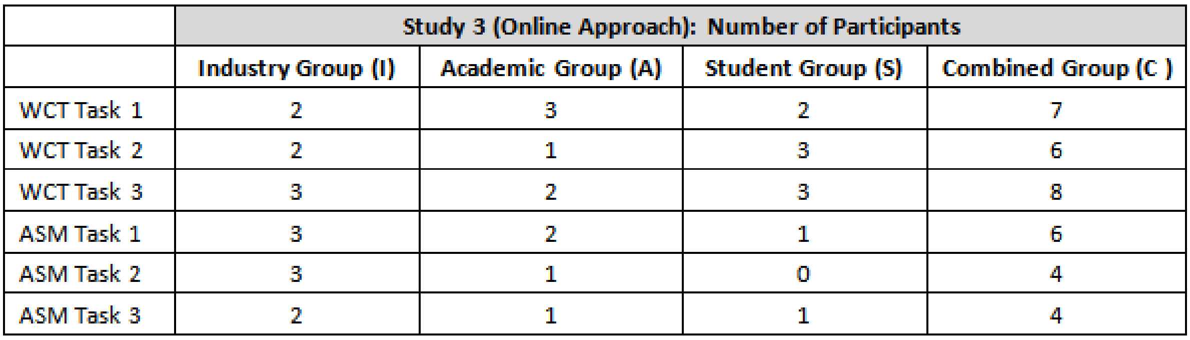
In Study 2, the number of participants in I and A Groups are equal for WCT Task 1, but not for the other 5 tasks. Consequently for these 5 tasks, we are careful when we compare between I and A Groups. The reasons for the imbalance number of participants in I and A Groups can be found in the chapters written for these tasks (Chapter 7 to 11).



	Study 2 (Manual Approach): Number of Participants		
	Industry Group (I)	Academic Group (A)	Combined Group (C)
WCT Task 1	4	4	8
WCT Task 2	5	3	8
WCT Task 3	3	5	8
ASM Task 1	2	5	7
ASM Task 2	3	5	8
ASM Task 3	3	4	7

Table 13.6: Number of Participants (Study 2)

In Study 3, the number of participants in I, A and S Groups of WCT Task 1 and WCT Task 3 are quite balance. However, for the other information-seeking tasks, some of the groups comprise only one participant: A Group for WCT Task 2 and ASM Task 2; S Group for ASM Task 1; A and S Groups for ASM Task 3. Consequently, we are cautious with the observations made for these tasks.



	Study 3 (Online Approach): Number of Participants			
	Industry Group (I)	Academic Group (A)	Student Group (S)	Combined Group (C)
WCT Task 1	2	3	2	7
WCT Task 2	2	1	3	6
WCT Task 3	3	2	3	8
ASM Task 1	3	2	1	6
ASM Task 2	3	1	0	4
ASM Task 3	2	1	1	4

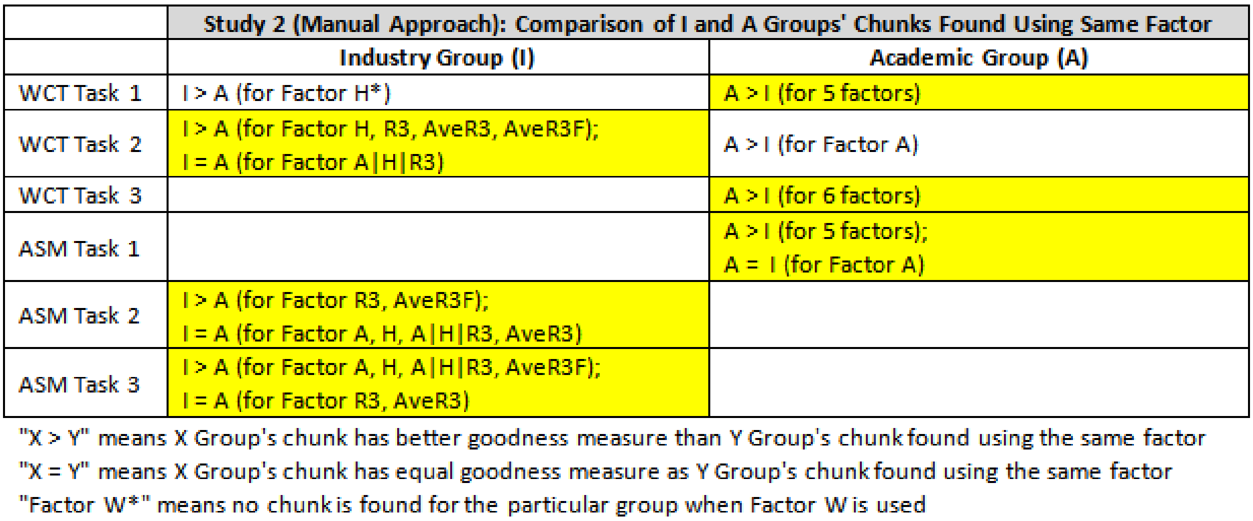
Table 13.7: Number of Participants (Study 3)

13.8.2 Comparison of Chunks

The comparison of the composition of the chunks found for each sub-group using the same factor can be found in the section entitled “Identification and Interpretation of Chunks” of Chapter 6 to Chapter 11 (for Study 2) and Chapter 12 (for Study 3). In the following, we compare the goodness of chunks to see which sub-group’s chunks are generally better for the respective task.

Study 2: Table 13.8 compares the goodness measures of I and A Groups’ chunks found using the same factor, for each information-seeking task in Study 2. In terms of which sub-groups chunks are generally better, similar tasks show similar results despite that they are performed on two different ADs. We observed this for the first two types of information-seeking tasks but not the third.

For both tasks related to getting an overview of the SA of a system (i.e. WCT Task 1 and ASM Task 1), chunk found for A Group has better goodness measure compared to I Group’s chunk found using the same factor. In other words, A Group’s chunk have closer match with the respective oracle set. The reason that academic group’s chunks fare better when benchmarked against the oracle set could be, the oracle set was constructed by judges who are also academics.



	Study 2 (Manual Approach): Comparison of I and A Groups' Chunks Found Using Same Factor	
	Industry Group (I)	Academic Group (A)
WCT Task 1	I > A (for Factor H*)	A > I (for 5 factors)

WCT Task 2	I > A (for Factor H, R3, AveR3, AveR3F);
	I = A (for Factor A|H|R3)
		A > I (for Factor A)

WCT Task 3		A > I (for 6 factors)

ASM Task 1		A > I (for 5 factors);
		A = I (for Factor A)

ASM Task 2	I > A (for Factor R3, AveR3F);
	I = A (for Factor A, H, A|H|R3, AveR3)	

ASM Task 3	I > A (for Factor A, H, A|H|R3, AveR3F);
	I = A (for Factor R3, AveR3)	

"X > Y" means X Group's chunk has better goodness measure than Y Group's chunk found using the same factor
"X = Y" means X Group's chunk has equal goodness measure as Y Group's chunk found using the same factor
"Factor W*" means no chunk is found for the particular group when Factor W is used

Table 13.8: Comparison of Goodness of Chunks found Using Same Factor (Study 2)

The opposite situation occurs for both tasks related to making some changes to systems and assessing the possible impact of changes (i.e. WCT Task 2 and ASM Task 2). For these two tasks, chunk found for I Group generally has better if not equal goodness measure compared to A Group's chunk found using the same factor. It is interesting that for more specific tasks of making some changes to systems, the industry group's chunks fare better when benchmarked against the oracle set which was constructed by academics.

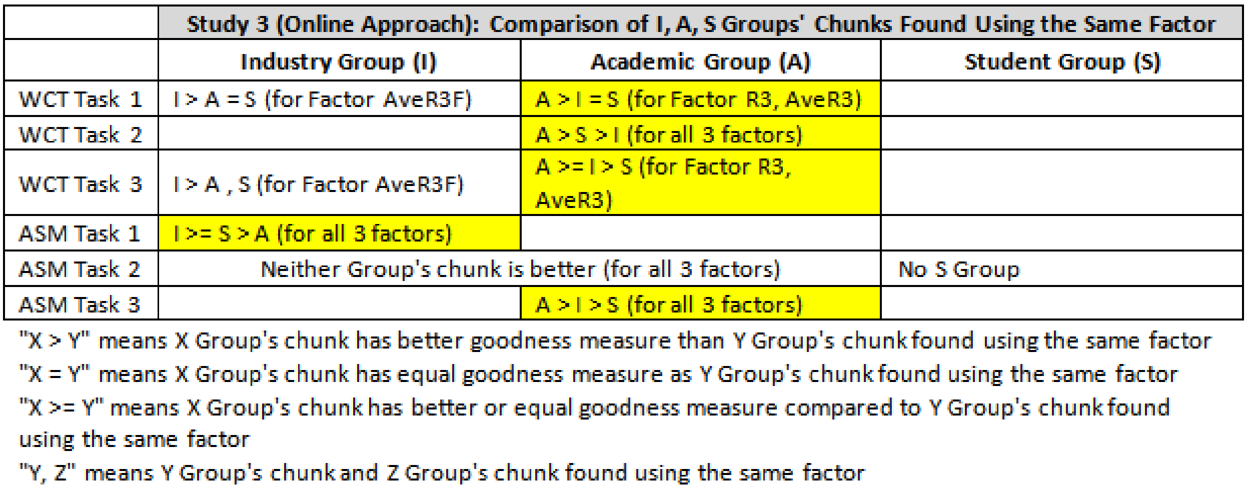
As for the third task, contradicting situation is observed. For WCT Task 3, chunk found for A Group has better goodness measure compared to I Group's chunk found using the same factor. Contrarily for ASM Task 3, chunk found for I Group generally has better if not equal goodness measure compared to A Group's chunk found using the same factor. The contradicting situation could be due to though both tasks are addressing cross-cutting concerns, they are focusing on different issues. WCT Task 3 focuses on security and ASM Task 3 on modifiability. It seems that academic group's chunks fare better on security issue whereas industry group's chunks fare better on modifiability issue, when the groups' chunks are benchmarked against oracle sets constructed by academics.

The above mixture of results shows that our oracle sets did not favour the academics though the oracle sets were constructed by academics. The findings seem to suggest the use of academic professionals to find chunks tor the tasks of getting an overview of the described SA, and security; but to use industry practitioners to find chunks for tasks related to system changes (changing a part of a system and accessing possible impact of change, and modifiability). The latter is properly due to generally industry practitioners have more exposure than academic professionals to system changes, which are frequent in industry settings.

Study 3: Table 13.9 compares the goodness measures of I, A and S Groups' chunks found using the same factor, for each information-seeking task in Study 3.

For WCT Task 1, the type of chunk-identification factor used dictates which sub-group's chunks are generally better. When factors based on the preference of the majority of those who rated (i.e. Factors R3 and AveR3) are used, the chunks found for A Group have better goodness measures than those found for I and S Groups. I and S Groups' chunks have the same goodness measures. When factor based on the ratings' preference of the majority of the participants in a group (i.e. Factor AveR3F) is used, I Group's chunk is better than A and S Groups' chunks which have the same goodness measures. The numbers of participants in I, A and S Groups are quite balance, which are 2, 3 and 2, respectively.

For ASM Task 1, I Group's chunk found using each of the 3 factors is better than or equal to S Group's chunk, which is better than A Group's chunk, found using the same factor. We are cautious with this observation because there is only 1 participant in S Group. There are 3 participants in I Group and 2 in A Group.



	Study 3 (Online Approach): Comparison of I, A, S Groups' Chunks Found Using the Same Factor		
	Industry Group (I)	Academic Group (A)	Student Group (S)
WCT Task 1	I > A = S (for Factor AveR3F)	A > I = S (for Factor R3, AveR3)	
WCT Task 2		A > S > I (for all 3 factors)	
WCT Task 3	I > A , S (for Factor AveR3F)	A >= I > S (for Factor R3, AveR3)	
ASM Task 1	I >= S > A (for all 3 factors)		
ASM Task 2	Neither Group's chunk is better (for all 3 factors)		No S Group
ASM Task 3		A > I > S (for all 3 factors)	

"X > Y" means X Group's chunk has better goodness measure than Y Group's chunk found using the same factor
"X = Y" means X Group's chunk has equal goodness measure as Y Group's chunk found using the same factor
"X >= Y" means X Group's chunk has better or equal goodness measure compared to Y Group's chunk found using the same factor
"Y, Z" means Y Group's chunk and Z Group's chunk found using the same factor

Table 13.9: Comparison of Goodness of Chunks found Using Same Factor (Study 3)

For WCT Task 2, A Group's chunk found using each of the 3 factors is better than S Group's chunk, which is better than I Group's chunk, found using the same factor. We are cautious with this observation because there is only 1 participant in A Group. The numbers of participants in I and S Groups are quite balance (i.e. 2 and 3, respectively) and it is interesting that chunks found from student group (S) are better than those found from industry group (I).

For ASM Task 2, neither I nor A Group's chunks fare better when they are compared. We are cautious with this observation because there is only one participant in A Group. There are 3 participants in I Group. There is no S Group for ASM Task 2.

For WCT Task 3, the type of chunk-identification factor used dictates which sub-group's chunks are generally better. This is similar to WCT Task 1. When Factors R3 and AveR3 are used, the chunks found for A Group have equal or better goodness measures compared to those found for I Group, which in turn have better goodness measures compared to those found for S Group. When Factor AveR3F is used, the goodness measure for I Group's chunk is better than A and S Groups' chunks. The numbers of participants in I, A and S Groups are quite balance, which are 3, 2 and 3, respectively.

For ASM Task 3, A Group's chunk found using each of the 3 factors is better than I Group's chunk, which is better than S Group's chunk, found using the same factor. We are cautious with this observation because there is only 1 participant each in A and S Groups. There are 2 two participants in I Group.

In short for Study 3, when the numbers of participants in each sub-group are quite balance (as for WCT Task 1 and WCT Task 3) the type of chunk-identification factor used dictates which sub-group's chunks are generally better. When factors based on the preference of the majority of those who rated (i.e. Factors R3 and AveR3) are used, the chunks found for A Group generally have better goodness measures than those found for I and S Groups. However, when factor based on the ratings' preference of the majority of the participants in a group (i.e. Factor AveR3F) is used, I Group's chunk is generally better than A and S Groups' chunks.

We do not make any concluding observation for the other tasks because there is only 1 participant in some of the sub-groups involved in these tasks.

In terms of which sub-groups chunks are generally better, the observation that we made for Study 2 (i.e. similar tasks show similar results despite that they are performed on two different ADs) cannot be made for Study 3.

Comparison between Studies 2 and 3: We do not compare the findings for each information-seeking task across Studies 2 and 3 because in Study 3: 1) there was only 1 participant in some of the sub-groups involved in WCT Task 2, ASM Task 1, ASM Task 2 and ASM Task 3; 2) for WCT Task 1 and WCT Task 3 which have quite balanced numbers of participants in each sub-group, the type of chunk-identification factor used seems to dictate which sub-group's chunks are generally better.

13.8.3 Comparison of Information Needed

The comparison between the information needed by each sub-group of participants for a particular information-seeking task can be found in sections entitled "Discussion of Chunk Discovery Results" in Chapter 6 to Chapter 11 (for Study 2) and Chapter 12 (for Study 3).

In the following, we compare the information needed by each sub-group and also combined (C) group, for similar tasks between the two ADs. To make the comparison easier, the information needed by each sub-group and combined group, for similar tasks is presented together (Tables 13.10 to 13.13). This information is extracted from Part 2 of tables entitled "Summary of Results for Task X (Study Y)" in Chapter 6 to Chapter 11 and Chapter 12, where X refers to either one of the information-seeking tasks (such as WCT Task 1), and Y refers to either Study 2 or Study 3. We only present the most prominent observations as we expect much dissimilarity due to the inherent differences of the two documents and the participants involved in the similar tasks of the two documents.

First information-seeking tasks: In Study 2 (Table 13.10), the most prominent observation for the first tasks of the two documents is academic groups needed process view but industry groups did not. However, in Study 3 (Table 13.11), industry group of ASM Task 1 needed process view.

The following information was needed by the participants as a whole (i.e. C Group) for the first tasks: For WCT Task 1 in Study 2, in addition to Table of Contents, categories of key quality requirements, a number of sections on use-case view, logical view and deployment view, but not process view, were needed. For ASM Task 1 in Study 2, in addition to Table of Contents, overview information (on conceptual view, process view, ASM, and platform layer) and more detailed and comprehensive information on process view were needed. In terms of architectural view, participants engaged with WCT Task 1 in Study 2 required quite comprehensive information on 3 of the architectural views but excluding process view, whereas participants engaged with ASM Task 1 were more after overview of conceptual view but required comprehensive information on process view.

In Study 3, only categories of the key quality requirements was needed by C Group of WCT Task 1, and the overview of the conceptual view and overview of the process view were needed by C Group of ASM Task 1.

Despite the differences in both studies, overview of the conceptual view and overview of the process view stood out as needed by both C Groups involved in ASM Task 1 of both studies.

Second information-seeking tasks: We cannot make a head-on comparison of the information needed by each sub-group for the second tasks of the two ADs. Though the two tasks are of similar nature, they are about changing a part of the described system which is specific to the system and not available in the system described by the other document.

Nevertheless, we observe a phenomenon when comparing the same second tasks in Studies 2 and 3 (Table 13.12): section in the respective oracle set we explain as very important section for the task stood out as needed by the industry participants of both studies. Specifically, both I Groups of WCT Task 2 in the two studies needed detailed description of ‘Submit to Archive’ use-case (one of the two very important sections for WCT Task 2), and both I Groups of ASM Task 2 in the two studies needed information on dynamic plug-in support (i.e. the only section in the oracle set).

We do not compare A Groups involved in WCT Task 2 of both studies due to only one participant in A Group involved in WCT Task 2 of Study 3. We also do not compare A Groups involved in ASM Task 2 of both studies for the same reason.

The following information was needed by the participants as a whole (i.e. C Group) for the second tasks: For WCT Task 2 in Study 2, overall SA of the system (in particular logical view in terms of textual description and diagram of main logical components), description of actors, and specific information related to the task (detailed description of ‘Submit to Archive’ use-case), were needed. For ASM Task 2 in Study 2, specific information related to the task (dynamic plug-in support), was needed.

In Study 3, the following information was needed by the participants as a whole for the second tasks: overall SA of the system (in particular textual description on main logical components, and process view diagram), and also specific information related to the task (detailed description of ‘Submit to Archive’ use-case, and packages of Java code) were needed for WCT Task 2; underlying models used (OSGi service model and Eclipse plug-in model) and also specific information related to the task (dynamic plug-in support) were needed for ASM Task 2.

Despite the differences in both studies, textual description on main logical components and detailed description of ‘Submit to Archive’ use-case stood out as needed by both C Groups involved in WCT Task 2 of both studies, and dynamic plug-in support stood out as needed by both C Groups involved in ASM Task 2 of both studies.

Third information-seeking tasks: In Study 2 (Table 13.13), the most prominent observation for the third tasks of the two documents is industry groups needed background information when attending to tasks related to cross-cutting concerns (i.e. security for WCT Task 3 and modifiability for ASM Task 3). In particular, information on security framework used was needed by I Group of WCT Task 3, and information on underlying models used (OSGi service model and Eclipse plug-in model) was needed by I Group of ASM Task 3. This is backed-up by the findings from I Group of WCT Task 3 but not I Group of ASM Task 3 in Study 3 (Table 13.13).

Comparison between A Groups of WCT Task 3 in both studies shows that logical deployment diagram and the first deployment diagram were needed by academic groups who tried to find out how WCT was designed at architectural level to achieve security. We do not make other comparison due to no chunk being found using Factor A|H|R3 for A Group of ASM Task 3 in Study 2, and there is only one participant in A Group of ASM Task 3 in Study 3.

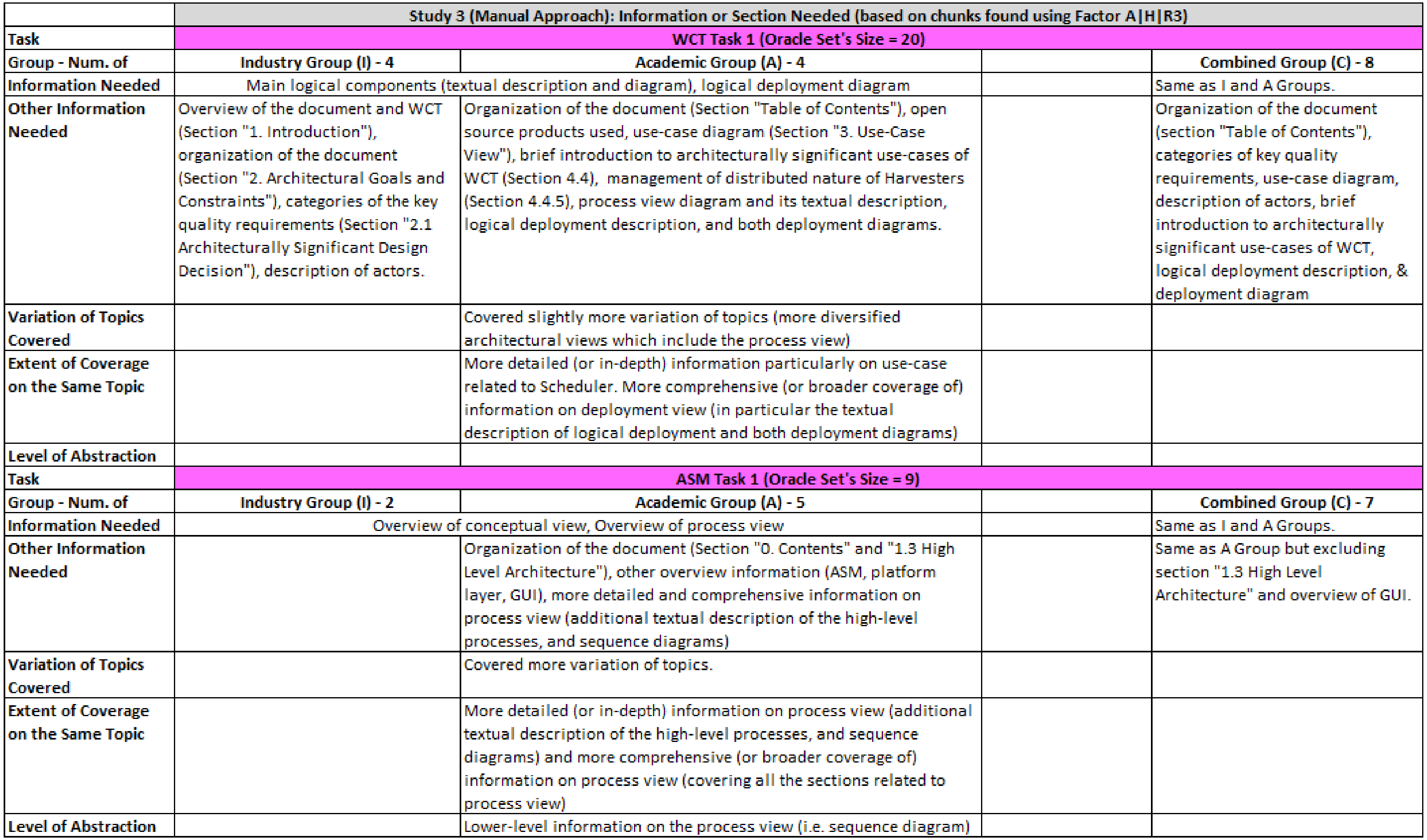
The following was needed by the participants as a whole (i.e. C Group) for the third tasks: For WCT Task 3 in Study 2, security as one of the categories of key quality requirements and background information on security framework used, were needed. For ASM Task 3 in Study 2, background information on underlying models used (overview information on OSGi service model and Eclipse plug-in model, and information on extension points), and overview information on process view, were needed. In Study 3, none of the sections (or information) of the respective AD stood out as needed by C Group who attempted WCT Task 3 and ASM Task 3, respectively. This is evidenced by no chunk being found for C Groups of these tasks.

In short, background information (such as framework or underlying models used) related to the addressed concern, stood out as needed for tasks related to cross-cutting concerns, especially by industry participants.

Summary: In terms of the information needed by each sub-group for similar tasks between the two ADs, the most prominent observations are: For the first information-seeking tasks, in Study 2, academic groups needed process view but industry groups did not. However, in Study 3, industry group of ASM Task 1 needed process view. For the second information-seeking tasks, section in the respective oracle set we regarded as very important section for the task stood out as needed by the industry participants of both studies. For the third information-seeking tasks, industry groups needed background information when attending to tasks related to cross-cutting concerns.

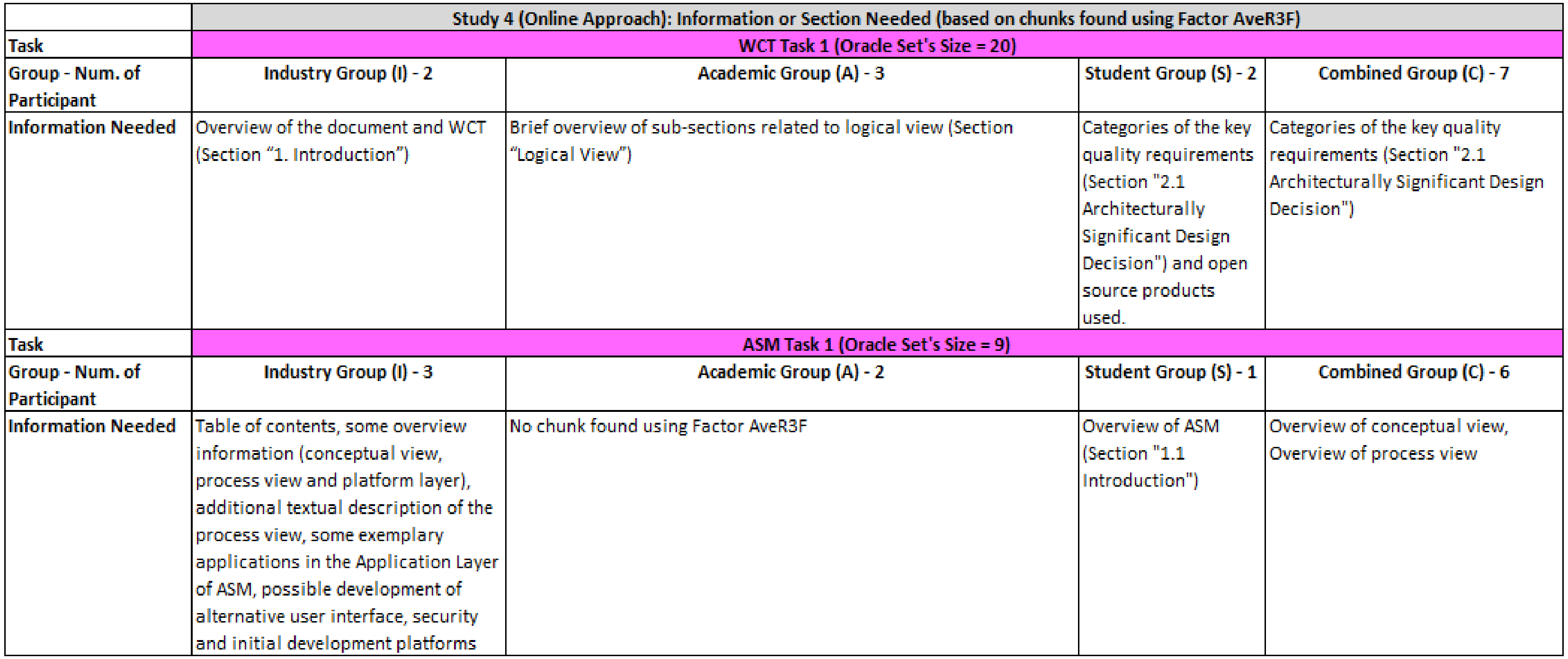
Studies have shown considerable differences between industry and academics in their perception of SA and reusable assets (Bosch, 1999). Our findings add on to that by showing that industry practitioners and academic professionals differ in their emphasis on process view in getting an overview of the SA of the described systems. For specific tasks of changing a part of a system, the industry practitioners were more able to identify the most critical section in the documents to deal

with the tasks. The industry practitioners needed background information when attending cross-cutting concerns tasks. This however, could be due to those involved did not have this background knowledge.



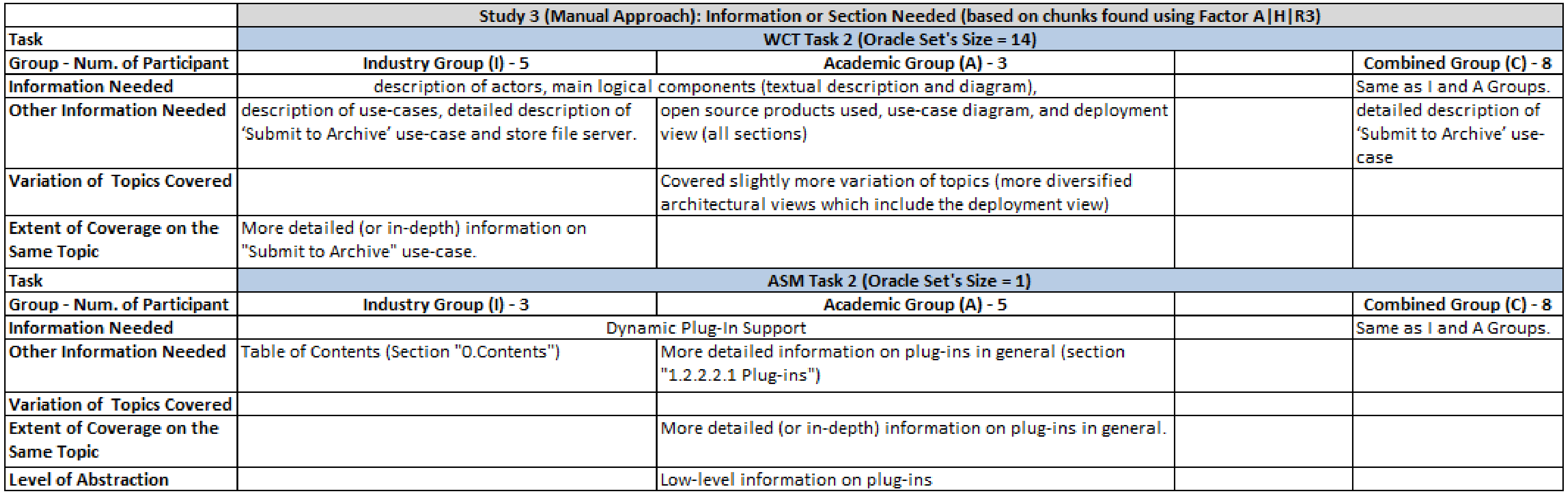
	Study 3 (Manual Approach): Information or Section Needed (based on chunks found using Factor A|H|R3)			
Task	WCT Task 1 (Oracle Set's Size = 20)			
Group - Num. of	Industry Group (I) - 4	Academic Group (A) - 4		Combined Group (C) - 8
Information Needed	Main logical components (textual description and diagram), logical deployment diagram			Same as I and A Groups.
Other Information Needed	Overview of the document and WCT (Section "1. Introduction"), organization of the document (Section "2. Architectural Goals and Constraints"), categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision"), description of actors.	Organization of the document (Section "Table of Contents"), open source products used, use-case diagram (Section "3. Use-Case View"), brief introduction to architecturally significant use-cases of WCT (Section 4.4), management of distributed nature of Harvesters (Section 4.4.5), process view diagram and its textual description, logical deployment description, and both deployment diagrams.		Organization of the document (section "Table of Contents"), categories of key quality requirements, use-case diagram, description of actors, brief introduction to architecturally significant use-cases of WCT, logical deployment description, & deployment diagram
Variation of Topics Covered		Covered slightly more variation of topics (more diversified architectural views which include the process view)		
Extent of Coverage on the Same Topic		More detailed (or in-depth) information particularly on use-case related to Scheduler. More comprehensive (or broader coverage of) information on deployment view (in particular the textual description of logical deployment and both deployment diagrams)		
Level of Abstraction				
Task	ASM Task 1 (Oracle Set's Size = 9)			
Group - Num. of	Industry Group (I) - 2	Academic Group (A) - 5		Combined Group (C) - 7
Information Needed	Overview of conceptual view, Overview of process view			Same as I and A Groups.
Other Information Needed		Organization of the document (Section "0. Contents" and "1.3 High Level Architecture"), other overview information (ASM, platform layer, GUI), more detailed and comprehensive information on process view (additional textual description of the high-level processes, and sequence diagrams)		Same as A Group but excluding section "1.3 High Level Architecture" and overview of GUI.
Variation of Topics Covered		Covered more variation of topics.		
Extent of Coverage on the Same Topic		More detailed (or in-depth) information on process view (additional textual description of the high-level processes, and sequence diagrams) and more comprehensive (or broader coverage of) information on process view (covering all the sections related to process view)		
Level of Abstraction		Lower-level information on the process view (i.e. sequence diagram)		

Table 13.10: Information Needed by Each Group of Participants for First Tasks (WCT Task 1 and ASM Task 1) (Study 2)

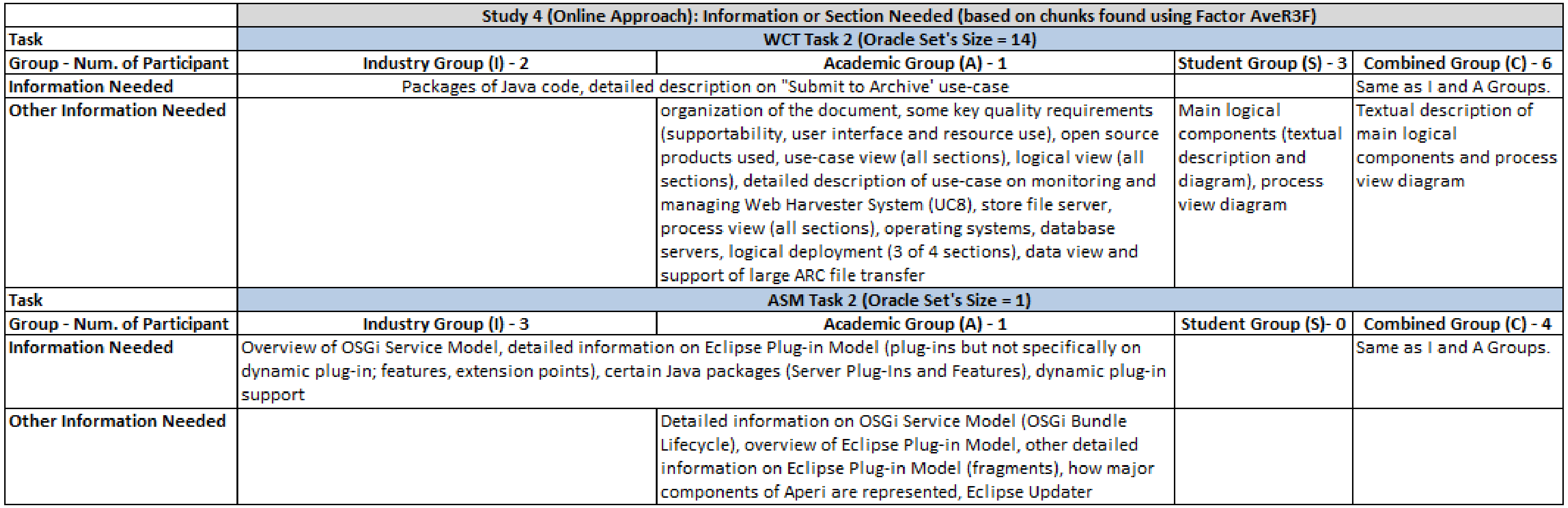


	Study 4 (Online Approach): Information or Section Needed (based on chunks found using Factor AveR3F)			
Task	**WCT Task 1 (Oracle Set's Size = 20)**			
Group - Num. of Participant	**Industry Group (I) - 2**	**Academic Group (A) - 3**	**Student Group (S) - 2**	**Combined Group (C) - 7**
Information Needed	Overview of the document and WCT (Section "1. Introduction")	Brief overview of sub-sections related to logical view (Section "Logical View")	Categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision") and open source products used.	Categories of the key quality requirements (Section "2.1 Architecturally Significant Design Decision")
Task	**ASM Task 1 (Oracle Set's Size = 9)**			
Group - Num. of Participant	**Industry Group (I) - 3**	**Academic Group (A) - 2**	**Student Group (S) - 1**	**Combined Group (C) - 6**
Information Needed	Table of contents, some overview information (conceptual view, process view and platform layer), additional textual description of the process view, some exemplary applications in the Application Layer of ASM, possible development of alternative user interface, security and initial development platforms	No chunk found using Factor AveR3F	Overview of ASM (Section "1.1 Introduction")	Overview of conceptual view, Overview of process view

Table 13.11: Information Needed by Each Group of Participants for First Tasks (WCT Task 1 and ASM Task 1) (Study 3)

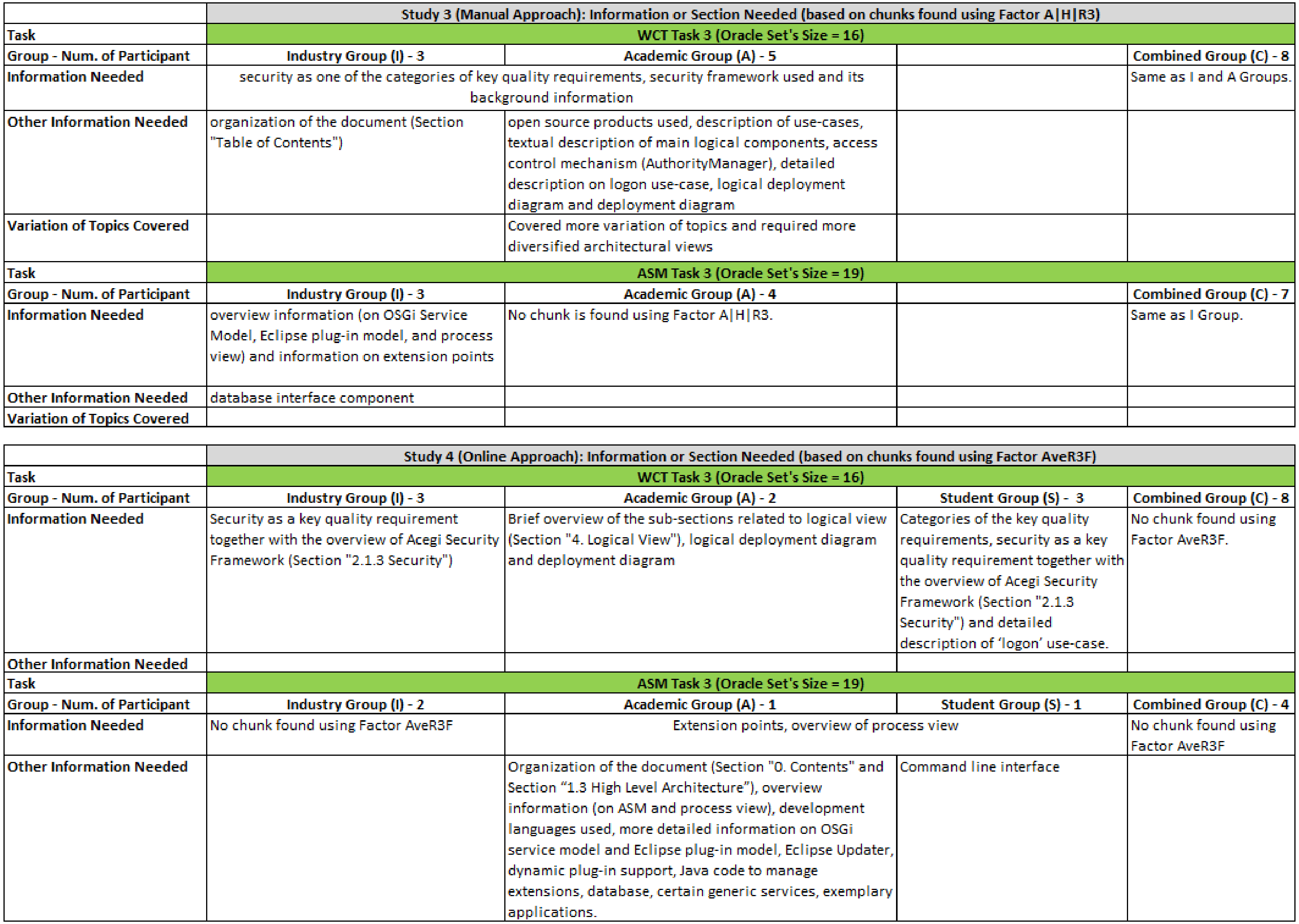


	Study 3 (Manual Approach): Information or Section Needed (based on chunks found using Factor A|H|R3)			
Task	WCT Task 2 (Oracle Set's Size = 14)			
Group - Num. of Participant	Industry Group (I) - 5	Academic Group (A) - 3		Combined Group (C) - 8
Information Needed	description of actors, main logical components (textual description and diagram),			Same as I and A Groups.
Other Information Needed	description of use-cases, detailed description of 'Submit to Archive' use-case and store file server.	open source products used, use-case diagram, and deployment view (all sections)		detailed description of 'Submit to Archive' use-case
Variation of Topics Covered		Covered slightly more variation of topics (more diversified architectural views which include the deployment view)		
Extent of Coverage on the Same Topic	More detailed (or in-depth) information on "Submit to Archive" use-case.			
Task	ASM Task 2 (Oracle Set's Size = 1)			
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 5		Combined Group (C) - 8
Information Needed	Dynamic Plug-In Support			Same as I and A Groups.
Other Information Needed	Table of Contents (Section "0.Contents")	More detailed information on plug-ins in general (section "1.2.2.2.1 Plug-ins")		
Variation of Topics Covered				
Extent of Coverage on the Same Topic		More detailed (or in-depth) information on plug-ins in general.		
Level of Abstraction		Low-level information on plug-ins		



	Study 4 (Online Approach): Information or Section Needed (based on chunks found using Factor AveR3F)			
Task	WCT Task 2 (Oracle Set's Size = 14)			
Group - Num. of Participant	Industry Group (I) - 2	Academic Group (A) - 1	Student Group (S) - 3	Combined Group (C) - 6
Information Needed	Packages of Java code, detailed description on "Submit to Archive" use-case			Same as I and A Groups.
Other Information Needed		organization of the document, some key quality requirements (supportability, user interface and resource use), open source products used, use-case view (all sections), logical view (all sections), detailed description of use-case on monitoring and managing Web Harvester System (UC8), store file server, process view (all sections), operating systems, database servers, logical deployment (3 of 4 sections), data view and support of large ARC file transfer	Main logical components (textual description and diagram), process view diagram	Textual description of main logical components and process view diagram
Task	ASM Task 2 (Oracle Set's Size = 1)			
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 1	Student Group (S)- 0	Combined Group (C) - 4
Information Needed	Overview of OSGi Service Model, detailed information on Eclipse Plug-in Model (plug-ins but not specifically on dynamic plug-in; features, extension points), certain Java packages (Server Plug-Ins and Features), dynamic plug-in support			Same as I and A Groups.
Other Information Needed		Detailed information on OSGi Service Model (OSGi Bundle Lifecycle), overview of Eclipse Plug-in Model, other detailed information on Eclipse Plug-in Model (fragments), how major components of Aperi are represented, Eclipse Updater		

Table 13.12: Information Needed by Each Group of Participants for Second Tasks (WCT Task 2 and ASM Task 2) (Studies 2 and 3)



	Study 3 (Manual Approach): Information or Section Needed (based on chunks found using Factor A|H|R3)			
Task	WCT Task 3 (Oracle Set's Size = 16)			
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 5		Combined Group (C) - 8
Information Needed	security as one of the categories of key quality requirements, security framework used and its background information			Same as I and A Groups.
Other Information Needed	organization of the document (Section "Table of Contents")	open source products used, description of use-cases, textual description of main logical components, access control mechanism (AuthorityManager), detailed description on logon use-case, logical deployment diagram and deployment diagram		
Variation of Topics Covered		Covered more variation of topics and required more diversified architectural views		
Task	ASM Task 3 (Oracle Set's Size = 19)			
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 4		Combined Group (C) - 7
Information Needed	overview information (on OSGi Service Model, Eclipse plug-in model, and process view) and information on extension points	No chunk is found using Factor A|H|R3.		Same as I Group.
Other Information Needed	database interface component			
Variation of Topics Covered				

	Study 4 (Online Approach): Information or Section Needed (based on chunks found using Factor AveR3F)			
Task	WCT Task 3 (Oracle Set's Size = 16)			
Group - Num. of Participant	Industry Group (I) - 3	Academic Group (A) - 2	Student Group (S) - 3	Combined Group (C) - 8
Information Needed	Security as a key quality requirement together with the overview of Acegi Security Framework (Section "2.1.3 Security")	Brief overview of the sub-sections related to logical view (Section "4. Logical View"), logical deployment diagram and deployment diagram	Categories of the key quality requirements, security as a key quality requirement together with the overview of Acegi Security Framework (Section "2.1.3 Security") and detailed description of 'logon' use-case.	No chunk found using Factor AveR3F.
Other Information Needed				
Task	ASM Task 3 (Oracle Set's Size = 19)			
Group - Num. of Participant	Industry Group (I) - 2	Academic Group (A) - 1	Student Group (S) - 1	Combined Group (C) - 4
Information Needed	No chunk found using Factor AveR3F	Extension points, overview of process view		No chunk found using Factor AveR3F
Other Information Needed		Organization of the document (Section "0. Contents" and Section "1.3 High Level Architecture"), overview information (on ASM and process view), development languages used, more detailed information on OSGi service model and Eclipse plug-in model, Eclipse Updater, dynamic plug-in support, Java code to manage extensions, database, certain generic services, exemplary applications.	Command line interface	

Table 13.13: Information Needed by Each Group of Participants for Third Tasks (WCT Task 3 and ASM Task 3) (Studies 2 and 3)

13.9 Comparison of Oracle Sets

Table 13.14 shows the information contained in the 6 oracle sets in our studies. We compare the oracle sets in terms of their sizes, content, and the total exclusion of oracle sets' sections and total inclusion of false sections.

13.9.1 Sizes of Oracle Sets

The oracle set for WCT Task 1 comprises 20 of 47 (42.6%) sections of the WCT software architecture document, whereas the oracle set for ASM Task 1 comprises 9 of 62 (14.5%) sections of the ASM software architecture document. The oracle sets for WCT Task 2 (14 sections, 29.8%) and ASM Task 2 (1 section, 1.6%) have the smallest sizes compared to the oracle sets for the other 2 tasks for the same document. WCT Task 2 and ASM Task 2 are very specific tasks on what to do to change a part of a system and assessing the possible impacts of change. The oracle sets for WCT Task 3 (16 sections, 34%) and ASM Task 3 (19 sections, 30.6%) comprise about 30% of the total number of sections in each document. The third information-seeking tasks of both documents are related to a particular cross-cutting concern that spans the entire system.

Though the specificity of task (such as the second information-seeking tasks of both documents) does seem to affect the sizes (i.e. the number of sections) of the oracle sets, we think that the particular AD used also impacted the sizes of the oracle sets.

13.9.2 Content of Oracle Sets

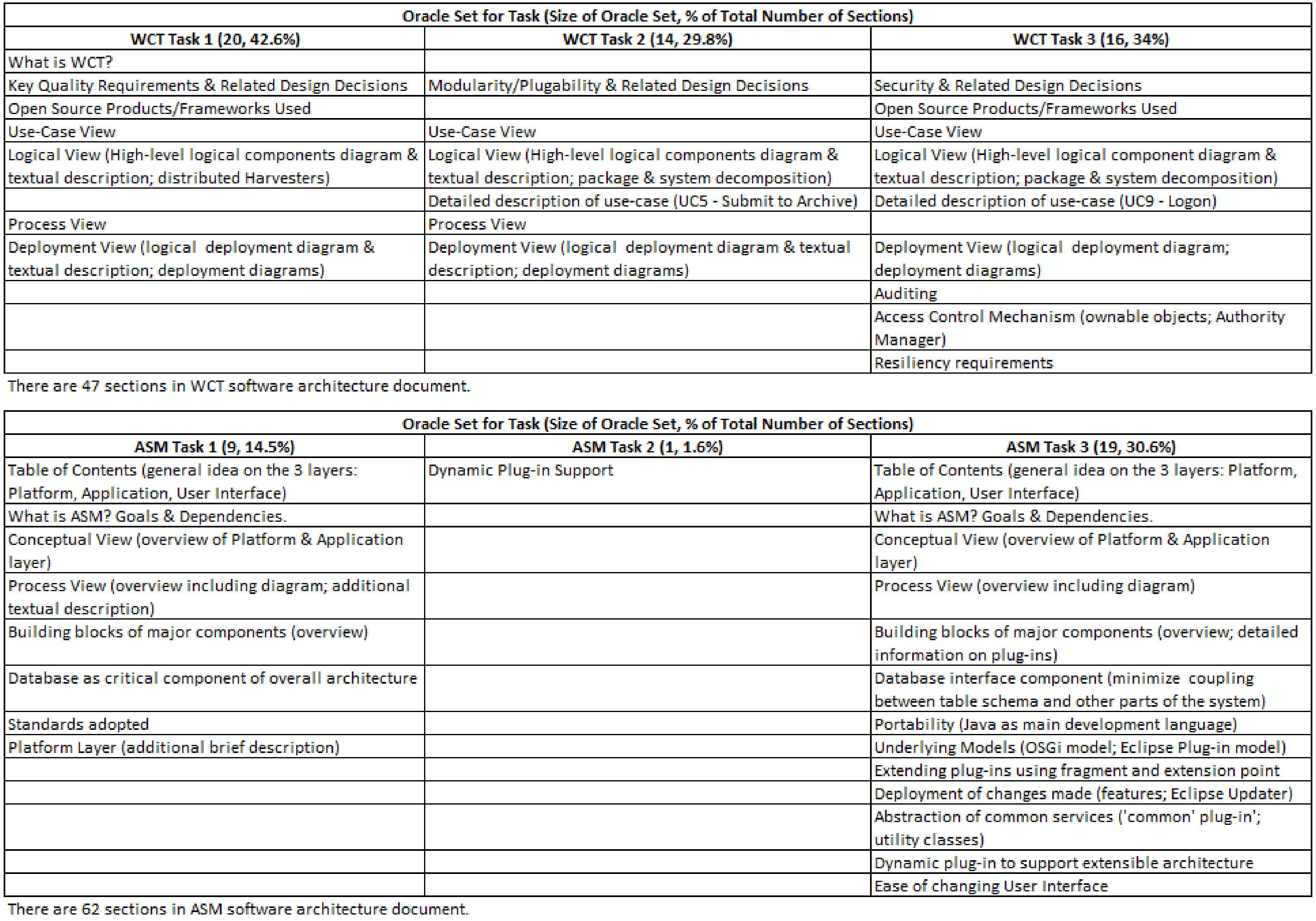
13.9.2.1 Comparison of Oracle Sets for Tasks for the Same Document

The oracle sets for the 3 information-seeking tasks of WCT software architecture document contain sections related to the architectural views (use-case, logical, process and deployment) of WCT. The oracle set for WCT Task 1 contains most of the main sections of all the 4 views, as this task is related to the SA of WCT. The oracle set for WCT Task 2 and Task 3 contain sections on architectural views which contain information specifically related to the respective task.

The oracle set for WCT Task 1 contains sections related to all the key quality requirements and the design decisions to achieve them. The oracle set for WCT Task 2 and Task 3 only contain section on the key quality requirement related to the respective task.

Other sections included in the oracle sets for the 3 tasks of WCT software architecture document are task dependent.

The oracle set for ASM Task 1 and ASM Task 3 contain sections related to Table of Contents, overview of ASM including goals and dependencies, overview of conceptual view, overview of process view, and overview of the building blocks of major components of Aperi. Other sections included in the oracle set for ASM Task 1 and ASM Task 3 are task dependent. The oracle set for ASM Task 2 contains only 1 section which contains all the information needed for the task.



Oracle Set for Task (Size of Oracle Set, % of Total Number of Sections)		
WCT Task 1 (20, 42.6%)	**WCT Task 2 (14, 29.8%)**	**WCT Task 3 (16, 34%)**
What is WCT?		
Key Quality Requirements & Related Design Decisions	Modularity/Plugability & Related Design Decisions	Security & Related Design Decisions
Open Source Products/Frameworks Used		Open Source Products/Frameworks Used
Use-Case View	Use-Case View	Use-Case View
Logical View (High-level logical components diagram & textual description; distributed Harvesters)	Logical View (High-level logical components diagram & textual description; package & system decomposition)	Logical View (High-level logical component diagram & textual description; package & system decomposition)
	Detailed description of use-case (UC5 - Submit to Archive)	Detailed description of use-case (UC9 - Logon)
Process View	Process View	
Deployment View (logical deployment diagram & textual description; deployment diagrams)	Deployment View (logical deployment diagram & textual description; deployment diagrams)	Deployment View (logical deployment diagram; deployment diagrams)
		Auditing
		Access Control Mechanism (ownable objects; Authority Manager)
		Resiliency requirements

There are 47 sections in WCT software architecture document.

Oracle Set for Task (Size of Oracle Set, % of Total Number of Sections)		
ASM Task 1 (9, 14.5%)	**ASM Task 2 (1, 1.6%)**	**ASM Task 3 (19, 30.6%)**
Table of Contents (general idea on the 3 layers: Platform, Application, User Interface)	Dynamic Plug-in Support	Table of Contents (general idea on the 3 layers: Platform, Application, User Interface)
What is ASM? Goals & Dependencies.		What is ASM? Goals & Dependencies.
Conceptual View (overview of Platform & Application layer)		Conceptual View (overview of Platform & Application layer)
Process View (overview including diagram; additional textual description)		Process View (overview including diagram)
Building blocks of major components (overview)		Building blocks of major components (overview; detailed information on plug-ins)
Database as critical component of overall architecture		Database interface component (minimize coupling between table schema and other parts of the system)
Standards adopted		Portability (Java as main development language)
Platform Layer (additional brief description)		Underlying Models (OSGi model; Eclipse Plug-in model)
		Extending plug-ins using fragment and extension point
		Deployment of changes made (features; Eclipse Updater)
		Abstraction of common services ('common' plug-in'; utility classes)
		Dynamic plug-in to support extensible architecture
		Ease of changing User Interface

There are 62 sections in ASM software architecture document.

Table 13.14: Information Contained in Oracle Sets

13.9.2.2 Comparison of Oracle Sets for Similar Tasks between the Two Documents

The oracle set for the first information-seeking task of both ADs (i.e. WCT Task 1 and ASM Task 1) contain the following sections: 1) section on the overview of the system, 2) main sections related to all the architectural views presented in the document (use-case, logical, process and deployment views for WCT software architecture document; conceptual and process views for ASM software architecture document). The oracle set for WCT Task 1 and ASM Task 1 also contain sections related to the critical component of the respective system (i.e. Harvester for WCT system and Database for ASM system).

Other sections included in the oracle set for WCT Task 1 and ASM Task 1 are dependent on the particular AD. The oracle set for WCT Task 1 also includes sections related to key quality requirements together with the related design decisions, and open source products used. The oracle set for ASM Task 1 also includes sections on the Table of Contents (which gives a general idea of the 3 layers of the conceptual view), on the overview of the building blocks of major components of Aperi, and standards adopted.

Recall that the second tasks of both ADs are very specific tasks composing of two parts: 1) what to do to change a part of the respective system, and 2) identifying the parts of the system that will be affected, both from the perspective of a developer. The difference is WCT Task 2 states that the digital archive system is to be changed, whereas ASM Task 2 states that a plug-in is to be dynamically unloaded but does not state which plug-in. Without stating the specific plug-in to be dynamically unloaded, the requirement of ASM Task 2 is more inclined towards some general information related to all plug-ins.

The oracle set for WCT Task 2 comprises the 2 sections that contain the information needed for the first part of the task. The other 12 sections of WCT Task 2's oracle set are needed for the second part of the task. These sections are the main sections related to the 4 architectural views of WCT and the section on the detailed description of the specific use-case related to the task (i.e. Section "4.4.2 UC5 – Submit to Archive"). The oracle set for ASM Task 2 comprises only 1 section of ASM software architecture document. This section contains the information needed for both parts of the task.

The third tasks of both ADs are related to a cross-cutting concern of the respective system. WCT Task 3 is about security and ASM Task 3 is about modifiability. The oracle set for WCT Task 3 and ASM Task 3 comprise sections related to architectural view (use-case, logical and deployment views for WCT Task 3, conceptual and process views for ASM Task 3) but only those that contain the information related to the task.

Other sections included in the oracle set for WCT Task 3 and ASM Task 3 are dependent on the particular AD, as well as the particular task since the two tasks are related to different cross-cutting concern. The oracle set for WCT Task 3 also includes sections related to the following: security as one of the key quality requirements and the related design decisions, security framework used, detailed description of the specific use-case (i.e. Logon) related to security, auditing, access control mechanism and resiliency requirements. The oracle set for ASM Task 3 also includes sections on Table of Contents (which gives a general idea of the 3 layers of the conceptual view), dependencies, building blocks of major components and how to extend them, database interface component, portability (Java as the main development language), underlying models used, deployment of changes made, abstraction of common services, dynamic plug-in to support extensible architecture and ease of changing the system's user interface.

As the summary, the comparison of the oracle sets for WCT Task 1 and ASM Task 1 shows that most of the main sections related to all the architectural views documented in the respective document are required for the tasks. This is not surprising for the tasks of getting an overview of the SA of the respective system. In addition, sections on the overview of the respective system and sections related to the critical component of the respective system are also needed. Other sections included in the oracle set for WCT Task 1 and ASM Task 1 are dependent on the particular AD.

The comparison of the oracle sets for WCT Task 2 and ASM Task 2 is futile because the two tasks and the documents used are different.

The comparison of the oracle sets for WCT Task 3 and ASM Task 3 shows that, in terms of architectural views, only those sections that contain information specifically related to the task, are needed for tasks related to cross-cutting concerns such as security and modifiability. Other sections included in the oracle set for WCT Task 3 and ASM Task 3 are dependent on the particular AD, as well as the particular task since the two tasks are related to different cross-cutting concern.

13.9.3 Total Exclusion of Oracle Set's Sections and Inclusion of False Sections

Commonly-missed oracle set's sections and common false sections for each group of participants of a particular information-seeking task can be found in sections entitled "Benchmarking Results" in Chapter 6 to Chapter 11 (for Study 2) and sections entitled "Benchmarking Chunks Against Oracle Set" in Chapter 12 (for Study 3). In the following, we focus on oracle set's sections totally excluded by all chunk-identification factors (i.e. the section did not appear in any chunk found, including 'no chunk' cases), and false sections included by all factors, across all groups of participants for a particular information-seeking task. This serves as a form of verification of our oracle sets.

An oracle set's section not rated by any participant in a group and excluded by all factors used to find chunks for the group, does not qualify as a totally-excluded oracle set's section. Our reason is we do not make any assumption on the value to assign to an unrated section, and 4 of 6 factors in Study 2 and all 3 factors in Study 3 made use of ratings value to identify chunks.

Before resorting to conclude that totally-excluded oracle set's sections are generally 'not needed' for the task, we inspect these sections to see if their total exclusions could be due to the sections themselves. For example, the section's title does not suggest to the participants the possibility of finding the needed information in the section, most of the information in the section can be found in other sections, and so on. For totally-included false sections (i.e. sections not from the oracle set), before resorting to conclude that these sections are generally 'needed' for the task, we also inspect these sections to see if their total inclusion could be due to the sections themselves. For example, the sections contain overview information of their sub-sections, and so on. We also point out totally-included false sections that do not contain any valuable information (such as Section "2.1 Security" of ASM document).

Study 2

Totally excluded oracle set's sections: The oracle sets' sections which are totally excluded by all factors across all groups of participants are: Section "4.4.5.1 Isolated Communication Strategy" for WCT Task 1; Sections "Logical High Level Solution Overview Diagram", "4.2 Package and system decomposition", "9.1 Resiliency" for WCT Task 3; Sections "1.2.1 Java", "1.3.8 Web based Console", "1.3.1 Conceptual View" for ASM Task 3. There is no totally-excluded oracle set's section for ASM Task 1 and ASM Task 2. As for WCT Task 2 Section "Process View Description" is excluded by all factors across all groups of participants but it does not qualify as totally-excluded oracle set's section since it was not rated by any participant who attempted the task.

Section "4.4.5.1 Isolated Communication Strategy" is related to the isolation of all the distributed communication strategy by the Harvester Manager and Agent components. For WCT Task 1, the total exclusion of Section "4.4.5.1 Isolated Communication Strategy" could be due to this section focuses mainly on just one element of WCT, which is the Harvester.

For WCT Task 3, the total exclusion of Section "4.2 Package and system decomposition" is probably because the Java package for users authentication could be too low-level to be considered as architectural for some participants. Section "9.1 Resiliency" describes resiliency requirements for the Harvester in WCT. These are related to the availability aspect of security but are on requirements and not the design. We cannot think of any plausible reason for the total exclusion of Section "Logical High Level Solution Overview Diagram", other than it was generally 'not needed' by the participants who attempted WCT Task 3.

For ASM Task 3, the total exclusion of Sections "1.2.1 Java" and "1.3.8 Web based Console" coincides with the suggestion that these two parts (i.e. platform and user interface) of a system are quite distinct and subject to change, and changes to them are considered separately (Bass et al., 2003) from other genres of modifiability of a system. As for the total exclusion of "1.3.1 Conceptual View", probable reason is this section shows some aspects of the SA but does not contain specific information for ASM Task 3.

Totally-included false sections: There is no totally-included false section for any of the task.

Study 3

Totally excluded oracle set's sections: The oracle sets' sections which are totally excluded by all factors across all groups of participants are: Sections "3. Use Case View" and "3.1 Actors" for WCT Task 1; Section "Deployment Diagram" for WCT Task 2; Section "1.3.1 Conceptual View" (same as Study 2) and Section "1.3.4.5.1 Utility Classes" for ASM Task 3. There is no totally-excluded oracle set's section for WCT Task 3, ASM Task 1 and ASM Task 2.

The following oracle sets' sections are excluded by all factors across all groups of participants but do not qualify as totally-excluded oracle set's sections due to not being rated: For WCT Task 1, Sections "4.4.2 Scheduler" and "4.4.5.1 Isolated Communication Strategy" (not rated by any participants who attempted the task), and Sections "Logical Deployment Description", "Deployment Diagram" and "Alternative Deployment Diagram" (not rated by any participant in I Group); For WCT Task 2, Section "2.1.1 Modularity" (not rated by any participant in S Group); For WCT Task 3, Sections “3.1 Actors” and “3.2 Use Cases” (not rated by any participant in I and S Groups), and Section “4.3.1 Auditing” (not rated by any participant in any group); For ASM Task 1, Sections “1.3.3 Mapping Aperi Function to the OSGi or Eclipse Component Model”, “1.3.3.4 Standards” and “1.3.4.7 Database (org.eclipse.aperi.database plug-in)” (not rated by any participant in I Group).

For WCT Task 1 and WCT Task 2, we cannot think of any plausible reason for the total exclusion of the identified oracle set’s sections, other than they were generally ‘not needed’ by the participants involved in the tasks. For ASM Task 3, the probable reason for the total exclusion of “1.3.1 Conceptual View” is the same as what is given earlier for Study 2. Section “1.3.4.5.1 Utility Classes” describes the utility classes which served as helper classes used by the rest of the platform. It is related to the tactic of abstracting common services to achieve modifiability. We cannot think of any plausible reason for its total exclusion, other than it was generally ‘not needed’ by the participants who attempted ASM Task 3.

Totally-included false sections: Totally-included false sections are found only for ASM Task 2 (Sections “1.2.2 OSGi (Service Model)”, “1.2.2.2.1 Plug-ins”, “1.2.2.2.3 Features”, “1.2.2.2.4 Extension Points” and “1.3.3.1 Server Plug-ins and Features”).

Section “1.2.2 OSGi (Service Model)” contains the overview of OSGi Service Model, what are bundles and deployment via bundles. Section “1.2.2.2.1 Plug-ins” contains low-level detailed information on plug-ins but not particularly on dynamic plug-in. The information in these two sections helps to understand the basic nature of the components in ASM. This could be important for some participants to find out how to dynamically unload a plug-in. Section “1.3.3.1 Server Plug-in and Features” contains Java packages related to server plug-ins and features, and probably needed by some participants to assess possible impact of dynamically unloading a plug-in.

Section “1.2.2.2.3 Features” contains description of feature as a group of plug-ins that define a logical product feature. Section “1.2.2.2.4 Extension Points” contains detailed explanation on how plug-in could be extended through ‘extension points’ with some example code. These two sections contain detailed information related to Eclipse plug-in model. We cannot think of any plausible reason other than this information was generally ‘needed’, by participants who attempted ASM Task 2.

Summary: In Study 2, the number of totally-excluded oracle set’s sections found for each task are: one (out of 20 oracle set’s sections or 5%) for WCT Task 1, zero (out of 14 or 0%) for WCT Task 2, 3 (out of 16 or 18.8%) for WCT Task 3, zero (out of 9 oracle set’s sections or 0%) for ASM Task 1, zero (out of 1 or 0%) for ASM Task 2, 3 (out of 19 or 15.8%) for ASM Task 3. There is no totally-included false section. In Study 3, the number of totally-excluded oracle set’s sections found for each task are: two (out of 20 oracle set’s sections or 10%) for WCT Task 1, one (out of 14 or 7.1%) for WCT Task 2, zero (out of 16 or 0%) for WCT Task 3, zero (out of 9 oracle set’s sections or 0%) for ASM Task 1, zero (out of 1 or 0%) for ASM Task 2, 2 (out of 19 or 10.5%) for ASM Task 3. Totally-included false sections (5) are found only for ASM Task 2.

Our analysis of totally-excluded oracle set’s sections and totally-included false sections shows some disagreement between the sections needed by the participants for a particular task and the sections

the judges involved in building the oracle set thought are needed for the task. Nevertheless, regardless of the possible reasons we thought contributed to the occurrence of each of these sections, totally-excluded oracle sets' sections are found for only 3 tasks in Studies 2 and 3 respectively, and the numbers of these sections found for these tasks are small (with at most 18.8% for WCT Task 3 in Study 2). As for totally-included false sections, they (5) are found only for ASM Task 2 in Study 3. It is very unlikely to have an oracle set that matches perfectly with chunks discovered from the data on the usage of a AD by a group of people engaged with a particular information-seeking task. Consequently, we think that our oracle sets are reasonable for the purpose of benchmarking the chunks found in our studies.

13.10 **Threats to Validity**

In this section, we discuss the threats that might have affected the validity of the findings of Study 2 (Manual Approach Study) and Study 3 (Online Approach Study). Most of these threats are also applicable to Study 1. For each threat, we indicate the studies the threat was applicable to.

1) The ADs Used

The AD used has some impact on the finding of chunks in all our studies. We mitigated this risk in Studies 2 and 3 by using two different documents. The documents are from the industry describing real systems in use. We believe the documents are suitable for our studies. However, as discussed below, generally the participants have problem with the language the ASM document was written and the domain concepts in the document. Nevertheless, the problem with the domain concepts could be due to the unfamiliarity of the participants with the domain concepts.

In Study 2, on a 5-point Likert scale (1 – strongly disagree, 2 – disagree, 3 – undecided, 4 – agree and 5 – strongly agree), 3 of 12 participants involved with WCT software architecture document strongly agreed, 7 agreed, 1 disagreed and 1 strongly disagreed that the language the WCT document was written was easy to understand. Two participants strongly agreed, 8 agreed, 1 undecided and 1 disagreed that they had no problem in understanding the domain concepts in the WCT document. In Study 1 (User Evaluation Study), 3 of 20 participants strongly agreed and 14 agreed that the language the WCT document was written was easy to understand. Three were undecided on this aspect. One participant strongly agreed, 16 agreed, 2 undecided and 1 disagreed that they had no problem in understanding the domain concepts in the WCT document. In short, the SA 'experts' in Study 2 and the SA 'novices' in Study 1, generally have no problem with the language the WCT document was written and the domain concepts in the document.

In Study 2, 5 of 12 participants involved with ASM software architecture document agreed that the language of the document was easy to understand, and 1 strongly agreed and 3 agreed that they had no problem in understanding the domain concepts in the document. The rest of the participants disagreed with the two aspects of ASM document or were undecided. Recall that ASM document was not used in Study 1. It seems that the SA 'experts' in Study 2 generally have problem with the language used in ASM document and its domain concepts.

We exclude Study 3 from the discussion here due to the participation of both SA ‘experts’ and ‘novices’ in the study. It is unconvincing to ignore the effects the different levels of SA background of the two groups of participants, have on the overall participants’ perceptions of the two aspects of the ADs.

One possible future work is to study whether identification of chunks could be more obvious if the quality of the ADs used are better.

2) The Information-Seeking Tasks

The information-seeking tasks and the participants’ perceptions about them could have affected our results in all 3 studies, in the following ways:

a. Different interpretations of the information-seeking task.

There is no doubt that different people attempting the same task could have interpreted the task differently. For example for ASM Task 2, a participant could have focused on how to unload a plug-in, whereas another participant focused on how to 'dynamically' unload a plug-in. Having a different focus on the task would affect the information and subsequently the respective sections of the document the participants thought were needed for the task.

We mitigate this risk in all the 3 studies by explaining the tasks to the participants and asking them to seek clarifications from us on any question. In addition, we have refined the specification of the information-seeking tasks in Studies 2 and 3 based on the feedback from the Study 1 (User Evaluation Study).

b. Different strategies to answering the information-seeking task.

Different people could have adopted different strategies to answering the same information-seeking task. For example for ASM Task 2, without knowing which plug-in to dynamically unload, different people could have different strategies to look for the required information. Some could have gone through the information on all plug-ins to look for some specific information to dynamically unload a plug-in, whereas other could have focused on general information related to dynamically unload any plug-in.

Our way to mitigate this risk is the same as given for a) above. We do not impose any constraint on the participants on what information to look for and how to look for the information in the documents.

With regards to the example given for a) and b) above, the chunks found for ASM Task 2 in fact shows that, despite that the participants might have had different interpretations of the information-seeking tasks or might adopted different strategies or both, this does not affect the discovery of chunks. In addition, all the chunks found for ASM Task 2 (16 chunks in Study 2 and 9 chunks in Study 3) include the most important section of the document for ASM Task 2. This section is the one and only one section in the respective oracle set.

c. Different ideas of the concepts involved or things required.

The concept involved in WCT Task 1 and ASM Task 1 (i.e. SA) has many definitions. The cross-cutting concerns involved in WCT Task 3 (i.e. security) and ASM Task 3 (i.e. modifiability) encompass many things. Different people could have different ideas of the concepts involved in or things required for the first and the third information-seeking tasks for both ADs. For example for WCT Task 1 and ASM Task 1, different people could have different cutting points to decide which levels of design are high-level design and which are low-level design, even though the given documents are ADs that describe the SA (i.e. high-level design) of the systems involved.

Despite the possibility of different people having different ideas of the concepts involved in or things required for the tasks, we believe that it is especially these kinds of tasks, which new users (or novices in the field) will be curious of what common sections of the document previous users needed for such tasks. In other words, the chunks found for such tasks would be interesting especially for new users of the ADs.

3) Annotatable Sections

In all 3 studies, we inserted annotation fields to the beginning of the sections of the ADs to make it easier for participants to provide ratings, tags, comments and so on for the sections. Each section that could be annotated was also enclosed by a border to visually distinguish it from others. Sub-sections that contained a substantial amount of information or contained distinct information by themselves were also inserted the annotation fields and the border (but with no change to order). To gather participants' perceptions of different types of representation of architectural information, diagrams were instrumented with the annotation fields separately from surrounding text. This might have affected the participants in finding the needed information. However, all the participants involved were given the same instrumented document (be it WCT or ASM document).

4) Duration of Participation Session

The long duration of the participation session might affect the participants' focuses on the tasks and their perceptions of certain aspects of the 3 studies (such as the document used, the features of KaitoroCap, and so on), as they became tired.

We tried to mitigate this by giving two information-seeking tasks instead of three to the participants in later studies (i.e. Studies 2 and 3) and shortening the duration of participation session from 90 minutes for Study 1 to 75 minutes for Studies 2 and 3. However, due the exploratory nature of the tasks, significant amount of time (75 minutes) was still required of the participants involved.

5) Participants Recruitment and Number of Participants

Our studies sought participants with SA background. In addition, the exploratory nature of the information-seeking tasks entailed a considerable amount of time and effort (90 minutes for Study 1 and 75 minutes for

Studies 2 and 3) to complete the tasks. To be realistic in the recruitment of the participants, we employed non-probabilistic sampling techniques in recruiting the participants. This rendered the results not generalisable to the target population (Barbara and Shari Lawrence 2002).

Despite using non-probabilistic sampling techniques, we encountered much difficulty in getting participants for all 3 studies, especially for Studies 2 and 3. None of the participant took part in more than one of our studies.

For Study 1, we invited potential participants who had some background in SA from three different institutions, namely University of Auckland (New Zealand), University of Malaya (Malaysia) and Swinburne University of Technology (Australia). We did so by emailing invitations using our professional contacts and their referrals, disseminating the recruitment advertisement with the help of our local acquaintances, and giving brief talks in lectures attended by students with SA background. Despite that, we only managed to recruit 21 participants. They were mostly students at graduate levels. Only 18 of them made it to the analysis stage. The small number of participants affected the significance of the results we obtained regarding the features of KaitoroCap. In addition to rating the features of KaitoroCap, participants were asked to suggest improvements to KaitoroCap. Thus, we believe that we gained sufficient feedback on the features of KaitoroCap, which was the main purpose of the user evaluation study.

The recruitment of participants for subsequent studies was even more challenging due to the more stringent participant selection criteria in Study 2 and the shrinking of our pool of potential participants when it came to Study 3. We personally invited 80 potential industry and academic participants for Studies 2 and 3, and 24 students for Study 3. Out of the 104 invited, 72 responded, out of which 4 were excluded, 30 took part in Study 2, and 38 took part in Study 3. Out of the 30 who took part in Study 2, 5 dropped out half-way through, 25 submitted responses out of which 24 were analysed. Out of the 38 participants who took part in Study 3, 6 dropped out, 32 submitted responses out of which only 19 were of adequate quality to be analysed. Refer to Section "5.3 Participant Recruitment" of Chapter 5 and Section "12.3 Participant Recruitment" of Chapter 12 for further details.

The numbers of participants whose responses made it into the analysis stage for Study 2 was 24, and for Study 3 was 19. These small numbers of participants affected the significance of the results obtained. Nevertheless, participants in Study 2 who made it to the analysis stage were industry practitioners or academic professionals who have strong SA background with at least two years of work experience (as described next). This led us to believe that our findings are useful for providing early insights on whether chunks can be identified based on consumers' usage of ADs. For Study 3, in addition to small number of participants, the participants were a mixture of SA 'experts' and 'novices'. Consequently, we are cautious with any observations made in Study 3.

The academic participants in Study 2 had on average 9.15 (minimum 2 and maximum 20) years of experience in teaching or providing training on SA. The industry participants had on average 10.9 (minimum 2 and maximum 24) years of SA-related industry experience. Twenty-one of 23 participants had some experience with ADs, one had exposure to this type of document from course taken and another one had no prior exposure to this type of document.

In Study 2, 17 participants always and 6 participants seldom read ADs. Thirteen participants always and 10 participants seldom read and made use of ADs. Ten always, 11 seldom and 2 never, wrote ADs. Eight always, 11 seldom and 4 never, updated ADs. In short, most of the participants always read and more than half of them always read and made use of ADs. However, less than half of the participants always wrote and only about one-third of the participants always updated this type of document. There was generally more engagement in the consumption than in the production of ADs among the participants. This is not a concern since our focus is on the usage of ADs.

Note that in terms of the frequencies of consuming and producing ADs, the ratings scale used was 1 (Never) to 5 (Always) and in-between were not specified. We interpreted 1 as ‘never’ or ‘none’, 2 and 3 as ‘seldom’, and 4 and 5 as ‘always’ involved in these activities.

Most of the participants had experience with the type of the software system described by the given AD. Eleven had read about similar types of system, two each had documented, architected or maintained similar types of system. Only 5 had no prior knowledge or experience with similar types of system.

For Study 3, we do not discuss the participants’ overall SA background since the participants were made up of both SA ‘experts’ and ‘novices’.

6) Seemingly Lack of Responses

At first glance of Study 2, it seemed that there was a lack of responses from certain participants in indicating the sections of the document required by them. For example, some of the participants did not provide ratings for any section of the AD while attempting a particular information-seeking task. For WCT Task 1 one of four academic participants, for WCT Task 2 one of three academic participants, for WCT Task 3 one of three industry participants did not provide any ratings. For ASM Task 1 one of two industry participants and one of five academic participants, for ASM Task 2 one of three industry participants and two of five academic participants did not provide any ratings. All participants provided ratings for ASM Task 3.

However, a closer look at the data shows that all the participants provided some responses. They did so either by 1) specifying the sections where the needed information (answer) was found, 2) highlighting of information in the section they thought relevant to the task, or 3) rating sections visited in terms of their importance to the assigned task, or by providing a combination of the three forms of indications. All participants gave some responses in terms of the sections where the needed information (answer) was found. Those who did not provide any ratings provided the other two types of indications. Those who did not highlight information in any section, either specified the sections where their answers were found or provided ratings for sections, or both.

In Study 3, there was only one way for the participants to indicate sections of the document needed by them for the assigned task. It was by rating the sections in terms of their importance to the task. For each of the information-seeking task in Study 3, all the participants provided ratings for some sections of the AD.

In Study 1, only one of 20 participants did not provide ratings for any section of WCT software architecture document when attempting WCT Task 1 and WCT Task 3, and 3 participants did not rate any section when attempting WCT Task 2.

7) Benchmarking Against Oracle Sets

In our evaluation of the chunks and the chunk-identification factors in our studies, we benchmarked the chunks discovered for each information-seeking task against the respective oracle set. Some may argue that an oracle set is subjected to the people involved in its construction and a different oracle set would probably change our results.

The instability of relevance judgments (Wallis & Thom, 1996) (which is the basis of our oracle sets) is nothing new. However, they could still be used to compare the 'relative' effectiveness of information retrieval systems (Lesk & Salton, 1968). It has been shown that regardless of which person's relevance judgments was used to compare the 'relative' effectiveness of information retrieval systems, a technique for information retrieval that performed well on one set of judgments would perform well on other sets of judgments.

To cater for the instability of judgments, our oracle sets were constructed using a vigorous process and two professionals in SA were involved. The use of oracle sets in our studies is similar to the use of relevance judgments in information retrieval systems evaluation, in the sense that our oracle sets were used to compare the "relative' goodness of chunks and the 'relative' effectiveness of the chunk-identification factors. Following that, changing to different oracle sets would most likely have very minimal effect on our overall findings, which are, usage-based chunks exist and factors based on the preference of the majority of those who rated (be it Factor AveR3 or Factor R3) show potentials in finding chunks of average goodness. Nevertheless, verification of this can be part of our future work.

8) Experience in Accessing Materials in Confluence Wiki

A higher level of participant experience in accessing Wiki materials has a tendency to give more positive user evaluation on the features of KaitoroCap in Studies 1 and 3. In addition, the level of experience with Confluence Wiki environment might affect the exploration paths captured using KaitoroCap in both studies. In terms of the frequency of reading and navigating in wiki environment, in Study 3 one participant has no experience, 3 did so on several occasions, 6 did so on many occasions, 5 did so several days per week, and 4 did so almost every day. In Study 1, one participant did so on several occasions, 15 did so on many occasions, 3 did so several days per week, and 1 did so almost every day. Nevertheless, only one participant each in Studies 1 and 3 have experience in accessing materials in Confluence Wiki where KaitoroCap was installed as a plug-in.

13.11 **Summary of Findings**

All in all, our research shows that factors based on the preference of the majority of those who rated (be it Factor AveR3 or Factor R3) show potentials in finding chunks of average goodness for the respective task, with the performance of these factors being affected by the participants' background in SA and the different reading behaviours in on line, versus local electronic or printed environments.

13.12 **Related Work**

In this section we compare our studies on chunking with the related works discussed in our literature review (Chapter 2).

13.12.1 Comparison with Existing Works on Supporting Finding of Architectural Information

Instead of a set of documents, Latent Semantic Analysis (LSA) can also be applied to sections within a document to produce a reading guide for reading the particular document based on a term or keyword a reader wants to explore. The application of LSA approach on a single document would encounter the same limitations (though to a lesser extent) as when it was applied to a set of documents in the work of de Boer and colleague (R. C. de Boer & van Vliet, 2008): results dependent on the selection of the initial query terms, and human interpretation is needed to select suggested documents to read. The users of the chunks found in our research can decide whether to read all the sections included in a chunk or they can rely on the rating assessments done by previous consumers. Only sections accessed as 'needed' for the particular information-seeking task based on the collective perceptions of their previous consumers, made it into a chunk found in our research.

Model-based approaches (Su et al., 2009) (R. C. de Boer, 2011) (Jansen et al., 2009) (Tang et al.) (de Graaf et al., 2012) (Eloranta et al., 2012) (Rost, 2012) support better knowledge retrieval but at the cost of rigidity, steeper learning curve and less support for unstructured or semi-structured knowledge (R. C. de Boer, 2011).

Comparing to the studies that automatically generates specialized ADs (Nicoletti et al., 2012) (Diaz-Pace et al., 2013) (Eloranta et al., 2012) (Rost, 2012) as a means to help stakeholders in finding architectural information that they need, our approach (i.e. exploration paths in our KaitoroCap) automatically extracts contents needed and dynamically re-structures an existing AD, based on consumers' actual usages of the document. One of these studies ((Nicoletti et al., 2012) (Diaz-Pace et al., 2013)) made use of individual stakeholder's perception (in the form of comments given) of an AD, to personalise future version of the document for the particular stakeholder. A chunk found in our work represents a subset of the sections of a document needed for a particular information-seeking task, found by leveraging the collective actual usages of the sections by a group of consumers performing the same task. In the study of Nicoletti et al., the sections for an AD are predefined. Ours are not.

The work of Nicoletti et al. (Nicoletti et al., 2012) (Diaz-Pace et al., 2013) identified the sections of the document relevant to a stakeholder based on the semantic information in the content of the sections. The works of Eloranta et al. (Eloranta et al., 2012) and Rost (Rost, 2012) identified relevant architectural elements to include in the specialized ADs, based on the meta-models used to capture the architectural elements. Our approach uses more abstract quality of the content of the sections (for example users' ratings of the importance of the sections to the assigned task) to determine relevance.

13.12.2 Comparison with Existing Works on Leveraging Usage Data

We collected two types of consumers' usage data in our research: annotation data in Study 2, and both interaction data and annotation data in Study 3.

In our online approach study, the interaction history of a consumer (or reader) with the sections in an AD while performing an information-seeking task, was captured as an exploration path by using KaitoroCap. The meta-data (such as reason of exploration, task involved, and so on) that provides contextual information about the exploration was also captured.

An exploration path in KaitoroCap shows the consumer's sequence of interactions with a document (in the forms of visiting pages, visiting sections on a page, following hyperlinks, providing ratings, tag and comment). An exploration path is a re-structuring of the document explored. The contents of the sections visited are embedded automatically within the path. By having the information on sections visited and the sequence of their visitation, an exploration path of a consumer can be regarded as a manifestation of the 'wear' (or usage) of a document by the consumer. A consumer's annotations (ratings, tags and comments) on the document's sections are also displayed in the exploration path.

In both Studies 2 and 3, we aggregate the annotations data of different consumers performing the same information-seeking task in our finding of chunks, making the aggregation task-specific. In addition, the aggregation is based on the consumers' conscientious judgement (in the forms of ratings and so on) of a document's sections as to whether the sections are needed for the task. This is different from aggregation at a more superficial level using the approximation of read time based on lower-level events (such as a line should be visible in the editor window, no screen-dimming due to user not attending to the screen, and so on) as in read wear (Hill et al., 1992).

By showing the information traces (exploration paths, ratings, comments and tags) left by other consumers, KaitoroCap supports some forms of social navigation. The information traces are dynamically grown, and personalised for specific task. We plan to make visible the aggregated behaviour of a community of consumers with regards to specific information-seeking task in the form of *chunks* found for the task.

We compare KaitoroCap to two social navigation systems which also record exploration paths: Comparing to IBM's WBI (Maglio & Barrett, 2000), KaitoroCap records a user's sequence of visiting sections within a wiki page in addition to recording his or her sequence of visiting pages. Footprints (Wexelblat & Maes, 1999) aggregates the interaction history of all users, in the visualisations. This removes the opportunity to see each other's paths. Except for comments data which is saved to the database immediately by Footprints, other data are persisted to database overnight. KaitoroCap does not combine exploration path of each user. Recall that what we aggregated was the annotation data in our attempt to find chunks. KaitoroCap persists all interaction and annotation data to the database immediately.

We borrowed from collaborative filtering the idea of using annotations (in particular ratings) provided by a group of collaborating users to filter information for new user. We did not use users' profiles to find users of similar taste to recommend items-of-interests, as the chunks of information that we were after are for specific tasks. Refer to Section "5.6 Chunking of Architectural Information" in Chapter 5, on how we made use of the commonality in the consumers' explicit indications (given in the forms of where answer can be found, highlighting of relevant information and sections' ratings) of which sections of the document they needed for the task, to group architectural information into a *chunk* for the task. We focus on per section basis as we believe that individual lines are too low a level for chunking of architectural information.

The set of studies on DOI (Mylar or Mylyn, and Tasktop) made use of a user's own usage history. This is different from the rest of the studies reviewed and our work. However, studies on DOI are related to our work because of their emphasis on the notion of task as in our work. Earlier version of Mylar aggregates the interaction data across all of a developer's workday (Kersten & Murphy, 2005) (Thomas et al., 2010). Subsequent implementation (Mik & Gail, 2006) is similar to our approach where the aggregation of data is task-specific (i.e. scoped by task). In the subsequent work on degree-of-knowledge model to identify expertise in or familiarity with source code, the aggregation of interaction data again involves all of a developer's work (Thomas et al., 2010). Different from our approach, Tasktop (Elves, 2010) (Tasktop Technologies Inc., 2013) does not track interactions with sections in documents or web pages.

Aggregation of data in our approach is task-specific. Our aggregation involves usage data (in particular annotation data) of multiple users instead of per user basis as in the works based on DOI. Wear-based filtering (Team Tracks) (DeLine, Khella, et al., 2005) also aggregates usage data of multiple users (specifically interaction data of multiple developers), but across multiple tasks.

The works on DOI (Mylyn or Mylar and to a lesser extent Tasktop) and wear-based filtering by Deline and his colleagues, focus on source code elements which are structured data with explicit relationships between elements. Our approach focuses on semi-structured ADs. Another difference is we make use of annotation data to filter the information instead of the frequencies of interactions (with the code elements) and recentness of interactions (in the case of DOI).

13.13 **Advantages and Limitations of Our Approach**

In this section, we discuss our approach in term of its advantages and limitations.

1) Noise in usage data

A disadvantage of deducing user's actions by capture and analysis of usage data (in particular interaction data) is the difficulty of determining portions of usage data generated by non-intentional actions. For example a user may navigate absent-mindedly, but interactions pertaining to this are captured together with intentional interactions. These false steps in navigation can taint the interaction data (DeLine, Khella, et al., 2005), and any derivatives of it. In our case tainted interaction data exhibited in exploration paths in KaitoroCap, can compromise the usefulness of the paths in guiding other consumers in exploring the same AD.

We mitigated this difficulty by using conscientious judgements of users. We solicited from the consumers explicit annotation data (ratings, tags and comments and so on) as indication of the relevance of sections to the task. Although it is inevitable that different consumers may provide contradictory ratings, tags and comments for the same sections visited during similar information-seeking task, these annotations provide an extra dimension to interpret the relevance of the sections in an exploration path. Tags and comments also afford other searching modalities.

To reduce the bias of individual consumer and to take advantage of collective opinions, we look for “commonality” in the consumers' annotation data to identify chunks. Chunks serve as guides for exploring architectural information in ADs when performing certain information-seeking tasks. In addition, these usage-based chunks provide alternative suggestions on how architectural information in ADs could be chunked or grouped for easier finding of related pieces of information.

2) Subjective goodness of chunks

In terms of how 'good' the discovered chunks are for the respective task, the goodness of the chunks are affected by the SA background of the consumers. The number of consumers of whose annotation data were used in finding chunks also has effect on the goodness of chunks. Model-based approaches retrieve more objective 'chunks' of architectural information. The set of information retrieved is as intended by what the producers captured using the underlying meta-models.

3) Consumption needed to produce chunks

The variety of task-specific chunks that can be found is dependent on the consumption of the ADs by consumers in performing different tasks. In supporting information finding, model-based approaches enables diversified querying of information, producing assortments of 'chunks'.

4) Effort required

Both our approach and model-based approaches entail significant efforts from the partakers. The tension is between asking consumers to share usage knowledge voluntarily as a by-product of actual usages of the documents, and asking producers to formalize knowledge in architecture documentations by coding the knowledge based on underlying meta-models to support possible future usage of the knowledge. In many circumstances, producers just want a medium to record their architectural designs and design decisions, to facilitate communication between stakeholders during architecting, instead of creating formal models of these artefacts for possible future uses enabled by automated reasoning and retrieval.

With proper tool support, interaction data (such as exploration paths) can be captured in the background without much user interventions. Annotation data (such as ratings, tags and comments) is easier to solicit from consumers as a by-product of actual usages of the documents, than requiring producers to produce ADs conforming to underlying meta-models.

With substantial interaction data, pattern mining tools can be used to mine common patterns in the data. These common patterns are also chunks that can be made available to subsequent consumers of ADs.

5) Granularity of architectural information

We experimented with 'document section' as the level of granularity for chunk elements. This approach is similar to existing work that studied the relevance of the elements of ADs to perceived stakeholders and their concerns (Koning & van Vliet, 2006). Unless each section describes one architectural element, otherwise our current approach missed the insights on the usage relationships between the architectural elements from a section of a document. We assumed all architectural elements from a section of a document were being used or rated equally by a consumer attempting the specified task.

Our approach can be adapted to the model-based approach to study whether a set of architectural elements are actually used in the way they are pre-chained by formal models.

13.14 **Summary**

This chapter presents the overall results of our investigation into the chunking of architectural information. It presents the answering of our research questions, and all relevant results. This chapter also presents our own assessment of our approach, the threats to validity of our findings in all three studies, comparison with the related work in the literature, and the advantages and limitations of our approach.

The next chapter presents our investigation into architectural information foraging in ADs, while trying to get an overview of the described SA (in particular WCT Task 1).

---- Blank ----

14 A Study of Architectural Information Foraging

This chapter presents our investigation into architectural information foraging in ADs. We are particularly interested in what the consumers of ADs would look for when attempting to learn about the SA of the described system. We think that it is important for them to obtain an overall picture of the described SA, in addition to more specific architectural issues. Despite the subtle issue of what constitutes SA, we wanted to see what the most commonly-sought information in ADs is.

This is an additional investigation on top of our studies on chunking of architectural information. Therefore, we only did this for the data collected for WCT Task 1 in Study 2 (Manual Approach Study). Recall that WCT Task 1 was:

> "You are a software architect new to the Web Curator Tool project. You would like to know what the software architecture of the Web Curator Tool is."

This chapter starts by introducing the idea that ADs' consumers or users typically "forage" for information and how this relates to Information Foraging theory (Pirolli, 2007). The chapter continues by highlighting key related work, followed by analysis of data, the results and discussion of results, threats to validity, and concludes with insights found and possible future work.

14.1 Introduction

Creating useful software documentation is difficult, but if it is not useful, the effort required is wasted. As Lethbridge et al. comment "Finding useful content in documentation can be so challenging that people might not try to do so" (Lethbridge et al., 2003). This difficulty of finding needed information also applies to ADs (Koning & van Vliet, 2006), the main purpose of which, after all, is to clearly inform.

ADs may have inherent limitations for finding information in them, for example depending on the search capabilities available. However, the behaviour of those seeking information can also impact their usefulness.

Our default assumption is that the users of AD forage, which our investigation seeks to confirm. This is consistent with Information Foraging theory (Pirolli, 2007) which assumes that humans are informavores (Miller, 1983), and so try to maximize the value of knowledge gained per unit cost of interaction (Pirolli, 2007). This is similar to the way animals forage in a patchy environment for food maximizing the rate of energy gained per effort expended (Stephens & Krebs, 1987). This maximization tendency for architectural information could be due to limited resources (time or budget) that the forager has to spend on finding information. To better structure and utilise ADs, we need to better understand the foraging behaviour of their users.

We investigated issues - forages, foraging sequence and styles - related to architectural information foraging in ADs when engaged in the task of discovering the SA of described systems. The keywords, answers and highlighted information provided by the participants were analysed to investigate what they forage for. We also analysed other annotations (tags and comments) of the documents provided by the participants during the foraging processes to see what insight we could gain from this extra meta-information. In terms of the foraging sequence, we focused on the order of reading the information to support better understanding of the described SA. We were interested if there is ‘common sequence’ of foraging that supports better understanding. We also compared the suggested sequences of reading the sections with the actual order of the sections as written in the documents. We also investigated the features of ADs that supported or hindered understanding.

In addition to the total combined group (C) of participants, we studied two different sub-groups of foragers: industry (I Group) and academic professionals (A Group). The rationale for having the two sub-groups is given in Section “5.5 Industry Practitioners Versus Academic Professionals” of Chapter 5: studies show considerable differences between academics and industry in their perception of SA and reusable assets (Bosch, 1999). Our focus is ADs and our interest is whether there is any different emphasis on architectural information in ADs between the SA academics and practitioners. By looking at architectural information in ADs commonly foraged by the two groups, we can gain some insight on this. Any differences between the two groups can serve as motivation for further study to reduce the gaps between academic research and industrial practice in terms of SA documentation.

14.2 **Related Work**

Information foraging relates to “activities involved with assessing, seeking, and handling information sources” (Pirolli & Card, 1995). Related studies mainly focus on understanding the navigation or exploration behaviour of the foragers to improve navigation tools. Some studies had been conducted on the navigation of program code during software maintenance (Ko, Myers, Coblenz, & Aung, 2006) (Niu, Mahmoud, & Bradshaw, 2011) and debugging (Lawrance et al., 2010). These studies build upon the underlying concepts of information foraging theory (Pirolli, 2007) to explain code navigation behaviour. The theory suggests that people will adapt themselves to their information environments or modify the latter, to maximize their rate of gaining valuable information. Central concepts include patchy structure of the task environment (information patch), selective consumption of the forage (information diet) and the use of information cues (information scent) to decide paths to take. For example, the concept of information patch could be used to explain developers’ tendency to visit files in clusters (Niu et al., 2011). The information scent concept supports developers’ uses of cues in their tools to support the code foraging process (Ko et al., 2006).

As far as we know no previous study has investigated the foraging of architectural information in ADs and how that supports understanding of the described SA. Our focus is on (SA) document navigation specifically the forages or *information diet* (Pirolli, 2007) resulting from the tendency to maximize benefit per cost. In particular is there any ‘commonly foraged information’ in the information diets across different foragers and if so what this is. We are also interested if there are ‘common sequences’ of foraging supporting better understanding. These can give insight to support the usage of ADs.

Other studies have investigated ways to produce understandable ADs and to support finding relevant AK in them. One proposal was to enrich ADs by annotating them with formal AK (Jansen et al., 2009). To improve the retrieval of the needed information, another suggestion was to index ADs with an ontology (Tang et al.). The foci of these studies were on the production of ADs or AK to support SA understanding and retrieval of information.

Relevant to AD usage, a survey of how software engineers perceive the usage of software documentation in general (Lethbridge et al., 2003) found software engineers often do not maintain documentation. Despite that, out-of-date documentation remains useful. In addition, finding useful content in documentation can be very difficult. This is further affirmed by the findings of a recent survey focused specifically on SA documentation from the perspective of developers (Rost et al., 2013). Three of the five main findings of the survey are related to the problem of finding the needed information in SA documentation (Rost et al., 2013). Another study investigated the perceptions of AD producers on the relevance of the different parts of an AD to its perceived stakeholders and their concerns (Koning & van Vliet, 2006). The study showed a scattered pattern of stakeholders' interests in an AD's content. The insights gained from perceived usage of ADs (Lethbridge et al., 2003) (Rost et al., 2013) (Koning & van Vliet, 2006) are useful but need to be verified by actual AD usage.

Most existing related work focuses on perceived usage of ADs and how ADs can be produced to make them better. Our study focuses on actual usage of ADs and the behaviour of their users. The closest study to our work is a study of AD consumers' usage of the two types of media (text and diagrams) used in ADs (Heijstek, Kuhne, & Chaudron). One main finding was that those who predominantly use text show better understanding. Our study differs in that we investigate the understanding issue from a wider perceptive and by investigating foraging.

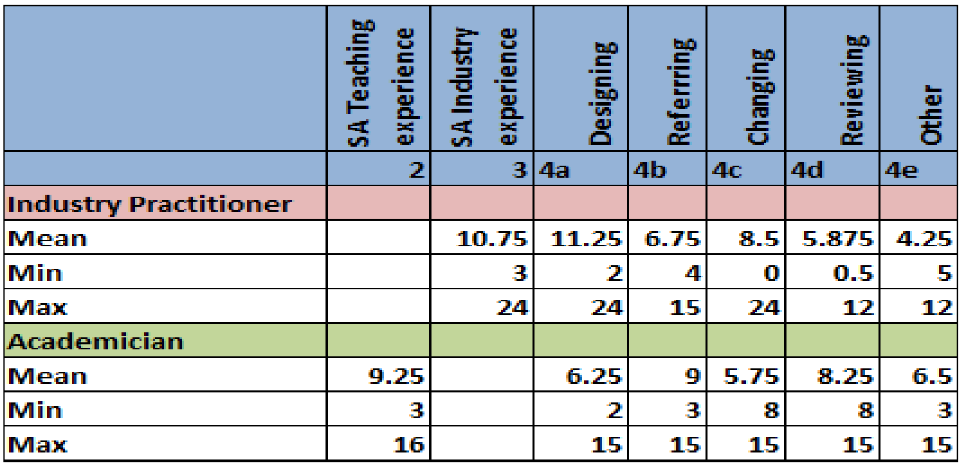
14.3 **Results**

We employed descriptive statistics and qualitative analysis on the data collected. For qualitative analysis the data was coded and their number of occurrences counted to find the main theme(s) or concept(s). The codes were words extracted from the participants' responses or representative words derived by the researchers based on the responses.

The findings reported in this chapter were based on the responses of eight participants who were involved with the information-seeking task of finding out what is the SA of the system described by the WCT software architecture document (WCT Task 1). We do not verify the "success" of participants in carrying out the tasks. This is because our assumption is that informavores try to maximize the rate of valuable information gained per unit cost and this comes at the expenses of the quality (in terms of completeness) of the information gained.

14.3.1 SA-Related Experience

The industry participants' had on average 10.75 (minimum 3 and maximum 24, Table 14.1) years of SA-related industry experience. The averages for more specific SA experience were 11.25 (SA design), 6.7 (referring to SA of software systems to perform tasks), 8.5 (changing SA), 5.88 (reviewing SA) and 4.25 (other SA experience) years, respectively. 'Other' experiences specified by the participants involved architecting specialized systems. The domains of systems designed were different between the 3 industry participants who responded on this. These were embedded, smart devices, autonomous, transactional (finance and banking), and engineering systems. Systems referred were real-time measurement and control, resource constraints and embedded systems.



	SA Teaching experience	SA Industry experience	Designing	Referrring	Changing	Reviewing	Other
	2	3	4a	4b	4c	4d	4e
Industry Practitioner							
Mean		10.75	11.25	6.75	8.5	5.875	4.25
Min		3	2	4	0	0.5	5
Max		24	24	15	24	12	12
Academician							
Mean	9.25		6.25	9	5.75	8.25	6.5
Min	3		2	3	8	8	3
Max	16		15	15	15	15	15

Table 14.1: Participants' SA Experience

On average, the academic participants had 9.25 years (minimum 3 and maximum 16) of experience in teaching or providing training on SA. Their average years of more specific SA experience are 6.25 (SA design), 9 (referring to SA), 5.75 (changing SA), 8.25 (reviewing SA) and 6.5 (other SA experience), respectively. The domains of the systems designed were different between the 2 academic respondents and included web-based or fat-client, point-of-sale, telecommunication, automotive, data processing and software tools. Systems referred are similar to those designed plus case studies in books and architecture of web browsers. 'Other' SA experiences include undertaking SA research.

All participants had prior exposure to ADs in general. In terms of how often they read AD, on a 5-point Likert scale (where 1 is never, 5 is always and the in-between were not specified), 2 out of 4 of the participants from both categories chose 5 indicating that they always read ADs (Figure 14.1). One of the industry participants chose 4 and the other chose 3. The remaining 2 academic participants chose 4. For reading and making use of ADs in their tasks, 1 of the industry participants chose 5 and 2 of them indicated that they did that on a less frequent basis by choosing 4. Most of the academic participants did that less frequently with 3 out of 4 of them choosing 4. The remaining 1 from each group chose 2.

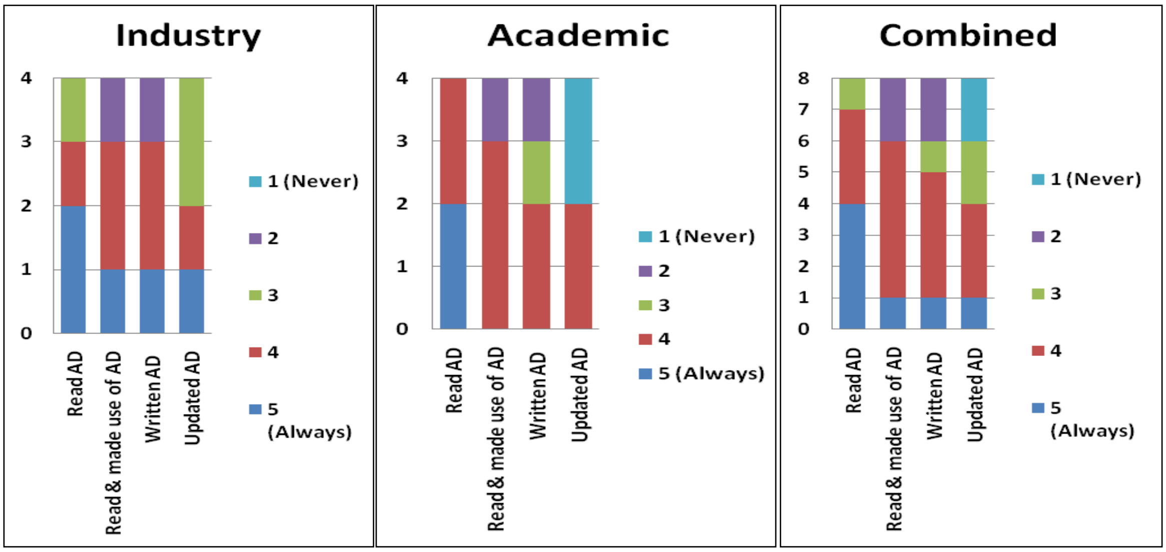


Figure 14.1: Exposure to ADs

In terms of writing ADs, 2 out 4 of the participants from each group chose 4 and 1 from each group chose 2. The difference was that the remaining industry participant chose 5 (wrote ADs always) whereas the remaining academic participant chose 3. With regards to updating ADs, 1 industry participant chose 5 and 4 respectively and 2 of them chose 3. No academic participants chose 5 for this aspect. Two of them chose 4 and the other 2 chose 1 showing that they have never updated an AD.

For the industry participants, the distribution of responses for the aspect of ADs writing was the same as for reading and making use of ADs. When no differentiation is made between the industry and academic participants, 7(87.5%), 6(75%), 5(62.5%) and 4 (50%) out of 8 participants were quite often (chose 4 and above) engaged in reading AD, reading and making use of ADs in their tasks, writing ADs and updating ADs, respectively.

All participants had some prior background with WCT style systems. Half of the industry participants had read ADs related to these. The rest either had architected or documented a similar type of system. All academic participants had read about similar types of system.

The average time spent by the industry participants on the task was 46.3 minutes (minimum 23 and maximum 85), and the academic participants spent on average 40 minutes (minimum 12 and maximum 80). It is worth mentioning that half of the academic participants attempted this task as the second task on their list. Therefore these participants could have spent less time for this task due to the familiarity with the AD they acquired during their earlier round of exploration of the AD for the first task. It was also intended for half of the industry participants to attempt the task as the second task. However not all participants followed the sequence of the tasks as instructed. In addition, after analysing the experiences of some participants, they were put into a different group from initially anticipated. For those who had both industry and teaching experience in SA, the length of experience decided their group.

14.3.2 Perceptions of the Given AD

On a 5-point Likert scale (1 - strongly disagree, 2 - disagree, 3 - undecided, 4 - agree and 5 - strongly agree), all the academic participants agreed (4) that the language the given AD was written was easy to understand and they had no problem in understanding the domain concepts in the AD. Two out of 4 of the industry participants strongly agreed and 1 of them agreed on the ease of understanding the language. The remaining 1 strongly disagreed on this. However none of the industry participants strongly disagreed that they had no problem in understanding the domain concepts, their responses were evenly spread across the remaining choices (with 1 participant each choosing 2, 3, 4 and 5).

When both groups are combined, 7 out of 8 of the total participants were in agreement (chose 4 and above) with the ease of understanding the language the AD was written and 6 out of 8 agreed or strongly agreed that they had no problem in understanding the domain concepts in the AD.

We did not further validate the participants' responses on these aspects. Our participants were professionals and the ADs that we chose describe the SA of systems of reasonable complexity. These questions were asked as an added check on the suitability of the participants and the ADs for the study.

14.3.3 Architectural Information Forages

To understand what industry and academic professionals with SA background forage for when trying to find out about the SA of a system by exploring an AD, the keywords, answers and highlighted information provided by the participants were analysed. Answers and the highlighted information contained cues about what information was searched for. We identified the main themes in the answers and the highlighted content provided by the participants to gain insight on this.

Three out of four industry participants provided keywords: architecture goal, constraint, design decisions, framework, interface, overview, pattern, purpose, quality, SA and views. Only one academic participant provided keywords. These were modules, processes and system architecture.

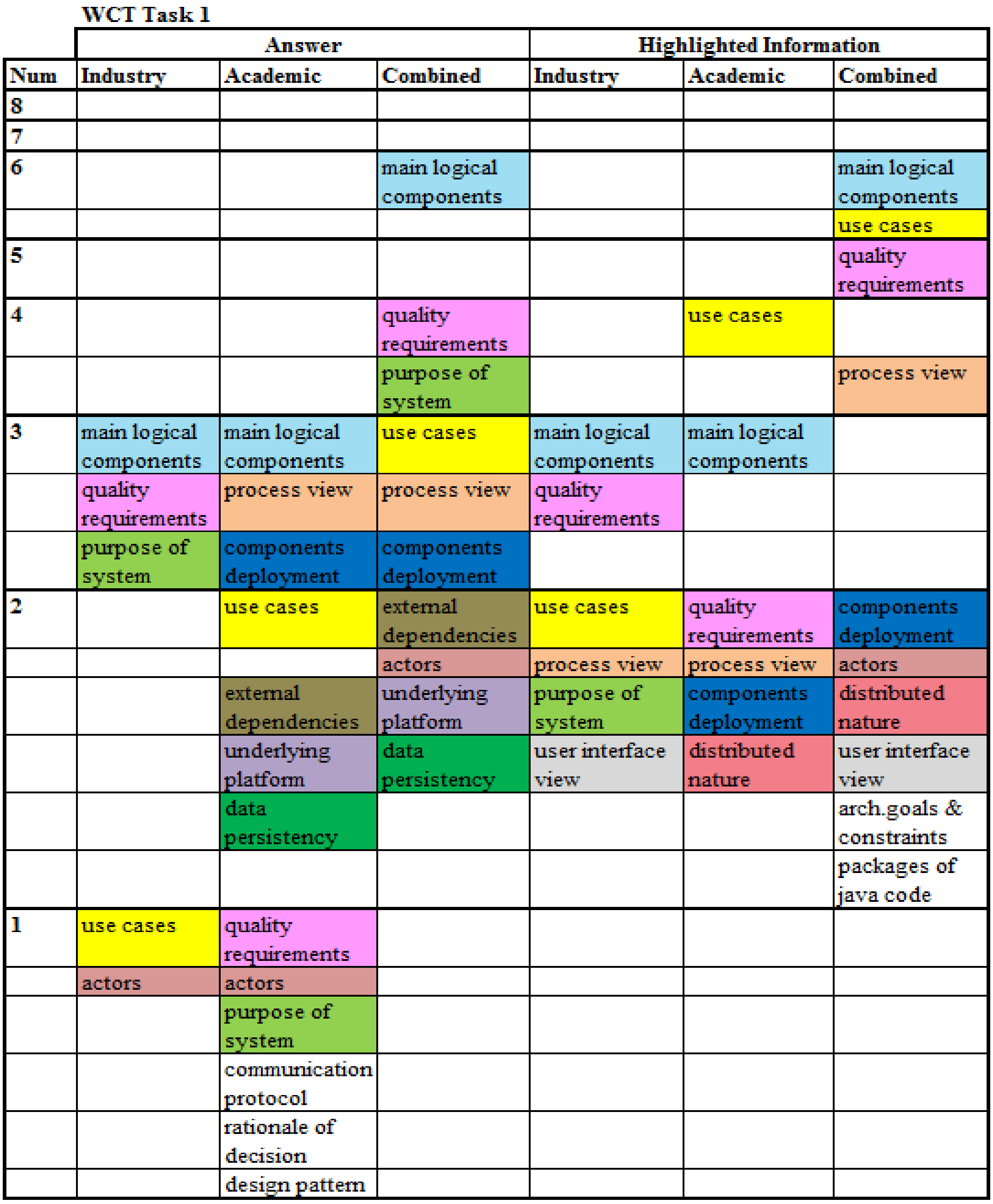
The answers given by the participants were either generic or specific. Generic answers contained general terms that reflected the content of the sections (for example ‘use cases’) whereas specific answers pertained specifically to the described system (for example ‘use case UC1’). Table 14.2 shows the main themes discovered in the participants’ answers, together with the total number of participants (column ‘Num’ in the table) in whose answers the themes were discovered. The table also contains the same information discovered in the highlighted content provided by the participants.

Analysis based on the participant answers shows that 3 out of 4 industry participants looked for main logical components, quality requirements or the purpose of the system. One looked for the actors or use cases. Three out of four academic participants looked for main logical components, process view or components deployment. Two looked for use cases, external dependencies (open source products used), underlying platform or data persistency. Only one looked for other things such as quality requirements, actors, purpose of the system, communication protocol, rationale of decision, design pattern and so on.

When the answers from both groups were combined and analysed, the top three things searched for were main logical components (6 out of 8 participants), purpose of system (4 participants) and quality requirements (4 participants). Among other things searched for were: use cases, process view, deployment of components (3 participants); external dependencies, actors, underlying platform and data persistency (2 participants).

The participants’ highlighted information in the document was also either generic or specific. Three out of four industry participants highlighted information that was related to main logical components or quality requirements (Table 14.2). Half of them highlighted information related to use cases, process view, purpose of the system or user interface view of the system. The most highlighted information by the academic participants was related to use cases (all the 4 participants). This is followed by information related to main logical components (3 participants), quality requirements, process view, components deployment and distributed nature (2 participants).

For the combined group, the following concepts (based on the highlighted information) emerged at the top of the list: main logical components and use cases (6 out of 8 participants); quality requirements (5 participants); process view (4 participants); components deployment, actors, distributed nature, user interface view of the system, architectural goals and constraints, and packages of java code (2 participants).



WCT Task 1

	Answer			Highlighted Information		
Num	Industry	Academic	Combined	Industry	Academic	Combined
8						
7						
6			main logical components			main logical components
						use cases
5						quality requirements
4			quality requirements		use cases	
			purpose of system			process view
3	main logical components	main logical components	use cases	main logical components	main logical components	
	quality requirements	process view	process view	quality requirements		
	purpose of system	components deployment	components deployment			
2		use cases	external dependencies	use cases	quality requirements	components deployment
			actors	process view	process view	actors
		external dependencies	underlying platform	purpose of system	components deployment	distributed nature
		underlying platform	data persistency	user interface view	distributed nature	user interface view
		data persistency				arch.goals & constraints
						packages of java code
1	use cases	quality requirements				
	actors	actors				
		purpose of system				
		communication protocol				
		rationale of decision				
		design pattern				

Note: The numbers under the 'Num' column refer to the total number of participants in whose answers or highlighted information the themes were discovered.

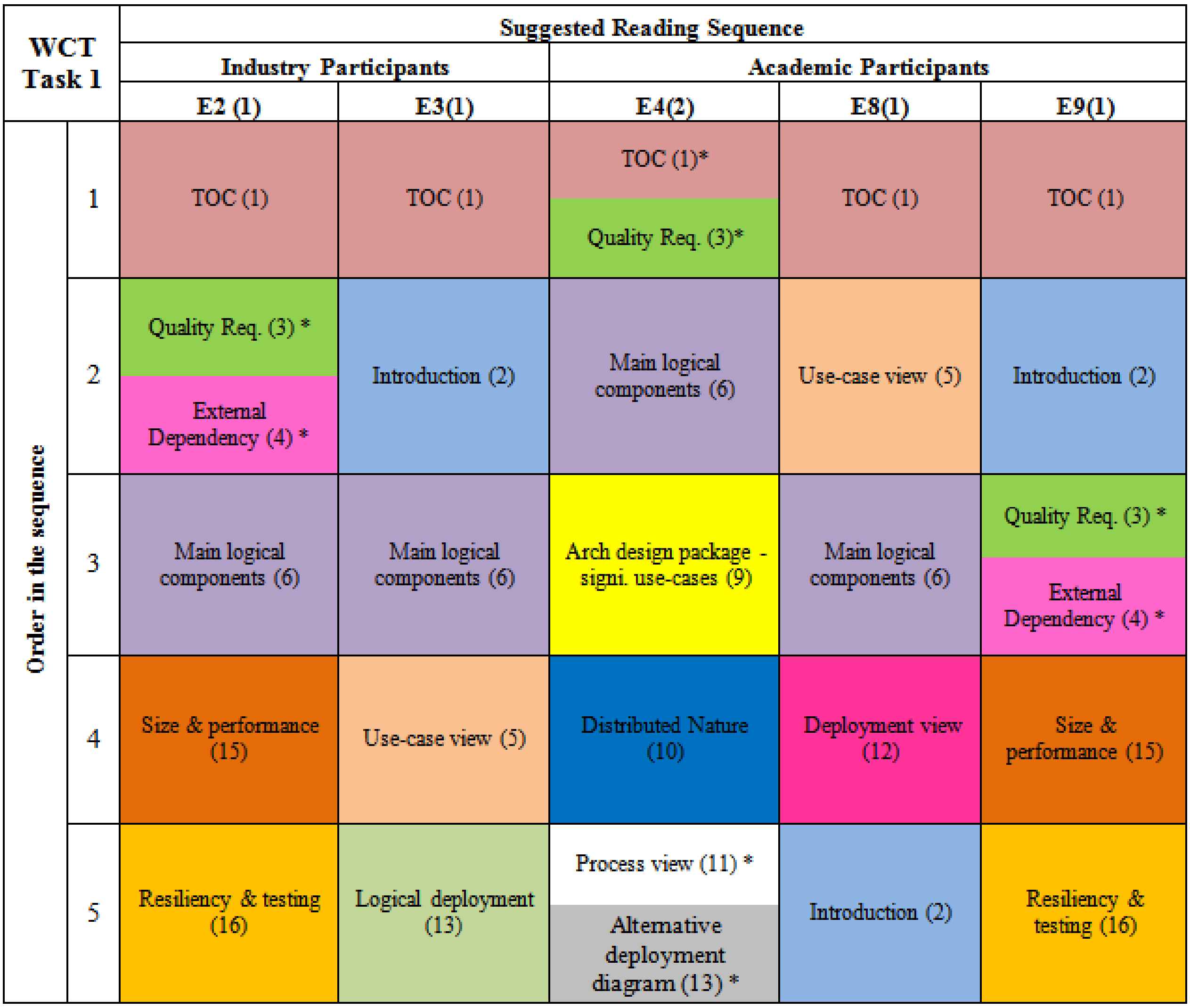
Table 14.2: Main Concepts in Answers and Highlighted information (WCT Task 1)

14.3.4 Foraging Sequences to Support Better Understanding of SA

Apart from the information foraged for, the foraging sequence is also of interest, specifically in terms of the order of reading the information to support better understanding of the described SA. We were interested if there is 'common sequence' of foraging that supports better understanding.

The participants were asked to suggest the sequences of reading the sections (relevant to the task) which would help to better understand the SA of WCT. We analysed the commonality and differences between the reading sequences suggested by the participants. We also compared the suggested sequences of reading the sections with the actual order of the sections as written in the documents.

Table 14.3 shows the participants' suggestions on the first five types of information or sections to be read to support better understanding of the SA of WCT. The number in bracket behind a participant ID refers to the order the participant attempted the task. The rest of the numbers in the bracket represent the actual order the sections were written in the document. For example, participant E3 attempted this task as his or her first information-seeking task. This participant suggested to start by reading Table of Contents (TOC), followed by sections related to Introduction, main logical components, use-case view and logical deployment, where the actual order of these sections in the document are 1st, 2nd, 6th, 5th and 13th respectively.



WCT Task 1		Suggested Reading Sequence				
		Industry Participants		Academic Participants		
		E2 (1)	E3(1)	E4(2)	E8(1)	E9(1)
Order in the sequence	1	TOC (1)	TOC (1)	TOC (1)* Quality Req. (3)*	TOC (1)	TOC (1)
	2	Quality Req. (3) * External Dependency (4) *	Introduction (2)	Main logical components (6)	Use-case view (5)	Introduction (2)
	3	Main logical components (6)	Main logical components (6)	Arch design package - signi. use-cases (9)	Main logical components (6)	Quality Req. (3) * External Dependency (4) *
	4	Size & performance (15)	Use-case view (5)	Distributed Nature (10)	Deployment view (12)	Size & performance (15)
	5	Resiliency & testing (16)	Logical deployment (13)	Process view (11) * Alternative deployment diagram (13) *	Introduction (2)	Resiliency & testing (16)

* Those in the same cell have the same order

Table 14.3: First Five Sections in Suggested Reading Sequences (WCT Task 1)

Two industry and three academic participants suggested reading sequences. Those suggested by the industry participants have Table of Contents (TOC) and main logical components as the first and third section to be read. The rest of the first-five sections differed between the two sequences. Reading sequences suggested by the three academic participants also have ‘Table of Contents’ as the first-to-read section. The sections related to main logical components and Introduction had two academic participants suggesting them in the first-five sections to read but at a different order in the respective sequence. The rest of the first-five sections varied between the academic participants.

Due to the highly-variable reading sequences suggested in each group, we focused on the suggested sequences from both groups as a whole. Of all the five participants who suggested reading sequences, Table of Contents was proposed as the first to be read. Three suggested the section on main logical component as the third section in the reading sequences after reading either Introduction, architectural goals and constraints (which include quality requirements and external dependencies), or use case view. The introduction section was suggested as the second-to-read section by two of the participants. Sections on ‘size & performance’ and ‘resiliency & testing’ were suggested as the fourth and the fifth section to be read by two of the participants. Other sections (such as use-case view and quality requirements), which has two or more participants suggesting them as the first-five sections to be read, are in different order in the suggested sequences.

The first 3 sections in the reading sequences suggested by the different participants revolve around sections related to Table of Contents, quality requirements, external dependencies, Introduction, main logical components and use cases. Subsequent sections (4th and 5th) in the reading sequences tend to be more variable across the participants.

In comparison with the actual order of the sections as written in the document, some of the participants suggested reading sequences that followed the written order of these sections, with different participants recommending different sections to be read while skipping some sections. Some participants (E2 and E8) suggested reading sequences which were slightly different from the stipulated order of sections in the document. For example, participant E8 suggested to read Introduction section after reading sections related to use-case view, main logical components and deployment view, but the Introduction section came first in the document before these sections. When asked about whether they backtracked to previous sections most of the time during the exploration of the document, one disagreed and the other was undecided. This shows that their suggestions of out-of-order reading sequences were more inclined to their perceived usefulness of these sequences in supporting understanding, rather than due to their individual exploration styles.

14.3.5 Foraging Styles

Do the foragers explore based on Table of Contents, titles and subtitles? What about sections skipping, backtracking and forward-browsing? These foraging styles were also investigated. On a 5-point Likert scale (where 1 is strongly disagree, 2 disagree, 3 is undecided, 4 is agree and 5 is strongly agree), all the industry participants agreed or strongly agreed that they frequently referred to the Table of Contents (Figure 14.2).

Three out of four of them also explored by looking at the titles and subtitles on a page most of the time. Half of them tended to skip some sections whereas the remaining was either undecided or disagreed on this. Frequent backtracking to previous section is not popular among them (1 out of 4). Half of them tended to forward-browse long sections (i.e. look ahead of the section to see where it ended before focusing on its content) most of the time.

Only one academic participant referred to the Table of Contents most of the time. Three out of four frequently explored based on the titles and subtitles on a page, often skipped sections and engaged in forward-browsing. Half of them backtracked to previous section most of the time.

When the participants from both groups are combined, their collective responses show that some foraging styles are quite popular among them (5 or more participants): referencing of Table of Contents, exploration based on titles and subtitles on pages, skipping sections and forward-browsing long section (Figure 14.3). Of less popularity is backtracking to previous section (3 out of 8).

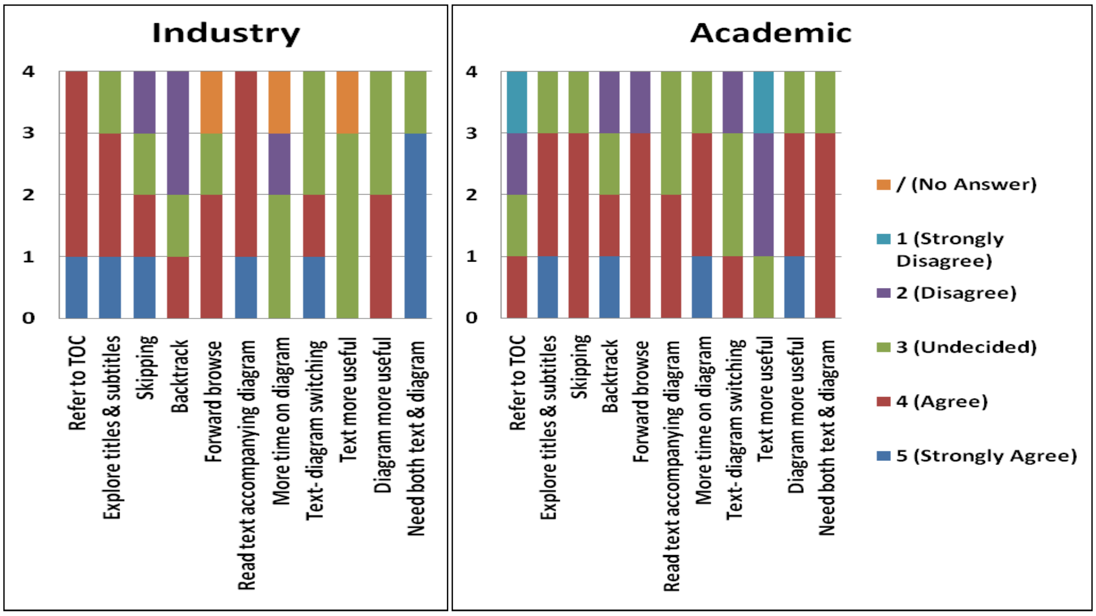


Figure 14.2: Foraging Styles and Perception on Text and Diagram (Industry and Academic Participants)

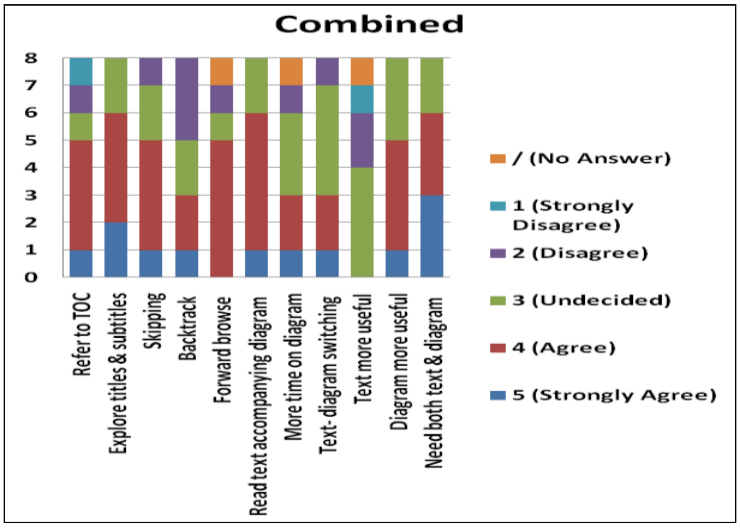


Figure 14.3: Foraging Styles and Perception on Text and Diagram (All Participants)

14.3.6 Understanding Support and Hindrance

When asked about the ways this AD supported their understanding of the SA of WCT, industry participants stated the following: use of terminologies, styles, patterns, standard protocols, notation, the structure of the document, charts and diagrams, and the presence of information that is generally included in ADs such as views (mentioned twice by different participant) and design decisions. For the academic participants, the following supported their understanding: diagrams (mentioned twice) and the related explanation, use case description, design considerations and decisions to achieve functional and non-functional requirements and having the expected information which was generally easy to find.

On what hindered their understanding of the WCT software architecture, the responses from the industry participants were descriptions without diagrams or examples, separation of design and execution time quality constraints, and the AD not being very precise or concise. The academic participants viewed too much textual description (mentioned twice), no mapping of modules to components, some out-of-order sections, inconsistency in naming sections and grouping contents, as factors that hindered their understanding.

14.3.7 Other Annotations of Document

We also analysed other annotations (in the forms of tags and comments) of the documents provided by the participants during the foraging processes to see what insight we could gain from this extra meta-information.

Only one participant from each group provided tags for some of the sections they visited. The industry participant annotated two sections with tags which were quite similar to the keywords searched for at the start of the exploration task. The academic participant annotated 15 sections with words that represented the participant's interpretation of the main gist of the content (for e.g. persistency) as well as the specific terms related to it (for e.g. Hibernate).

The same two participants provided comments for some of the sections they visited. The comments ranged from 'Logical High Level Solution Overview Diagram' as the most important section for an architect to the general type of information the sections contained (for e.g. background information, explanation of diagram, and so on), the type of understanding (understand architecture, code layout, or rationale) the sections supported, plus other comments.

14.4 Discussion

Our interpretation of the data is presented here following the same organisation as in the previous section (Section 15.3) of this chapter.

14.4.1 SA-Related Experience

The two groups have comparable average years of SA experience in their occupations, making them suitable for the purpose of this study. The industry participants have longer experience in designing and changing SA of software systems whereas the academic participants have longer experience in referring to and reviewing the SA of software systems. The 'other' SA experiences for both groups were different and therefore not compared.

All of the participants had prior exposure to AD in general. Collectively there is little difference between the two groups in reading ADs, reading and making use of ADs in their tasks and writing ADs. An obvious difference is seen in terms of AD updating. All the industry participants had experience in that while half of the academic participants did not.

Collectively the industry participants were quite equally engaged in consuming (reading and making use of ADs in their tasks) and producing ADs (in terms of writing ADs) though half of them did not update ADs that often. The academic participants, collectively, were more consumers than producers of ADs, especially having lesser involvement in updating of ADs. This is in line with the job nature of the two groups.

The combined group shows a diminishing trend in terms of reading AD (7 out of 8 or 87.5%), reading and making use of ADs in their tasks (75%), writing ADs (62.5%) and updating ADs (50%). This shows that generally there was more engagement in AD consumption than in production among the participants. This is not a concern for this study as its main focus is on the usage of ADs.

Some industry participants had experience in architecting or documenting systems similar to the WCT system, while the rest (especially all the academic participants) had only read about similar types of system. The participants were also asked how each of their answer points was found (i.e. either by looking at the section title, reading the section or came from his or her past experience and knowledge). Only one participant from each group indicated that one of his or her answer point was not obvious from the AD but derived from past experience and knowledge. As such, the lack of experience in architecting or documenting systems similar to WCT among the participants is not a concern for this study.

14.4.2 Perceptions on the Given AD

The industry participants are slightly less in agreement on the ease of understanding the language the AD was written and the domain concepts, when compared to the academic participants where all agreed on these aspects. Nevertheless, generally the participants found the language the AD was written was not a barrier to understanding its content and the domain concepts were comprehensible. This again reinforced the suitability of the participants and the AD for this study.

14.4.3 Architectural Information Forages

To discover what information was foraged, the keywords, answers and highlighted information provided by the participants were analysed.

The two groups of participants were not compared based on the keywords because of the imbalance in the number of participants from the two groups who provided them. Nevertheless one observation is that there is no repetition of keywords across different participants. One possible explanation is these participants have different pre-conceived ideas on what to look for with regards to the SA of a system prior to the exploration task.

However analysis on the participants’ answers shows there are some similarities among the participants in each group in terms of what they were looking for when trying to find out about the SA of WCT. Comparing the two groups, it is interesting to note that one of the most common thing industry participants specified in their answers, namely quality requirements of the system, was not common in the answers given by the academic participants. Nevertheless they seemed to place equal emphasis on main logical components with a majority of the participants (three out of four) from the two groups having it in their answers. Other than that, the majority of the industry participants also focused on the purpose of the system, whereas the majority of the academic participants focused on components deployment and process view. When the groups are combined, the information most searched for relates to main logical components.

The observation made from the analysis of the answers is also evident in the analysis of the highlighted information, which is though the majority (three out of four) of the two groups emphasized the main logical components of the system, they were also after other types of information. The highlighted content showed that information related to use cases was most sought after among the academic participants whereas the industry participants were more after quality requirements.

Similarly to what was found from the analysis of the answers of the combined group, the most sought after information was related to main logical components. In addition, analysis on the highlighted information uncovered another main information searched for, which is use cases.

All in all, the variation of keywords searched for during the information foraging suggest that individual participants had different pre-conceived ideas on what to look for with regards to the SA of a system prior to the exploration task. However, the analysis on the participants' answers and the highlighted information shed some insights on the commonly foraged information with regards to the SA of a system. Combining the findings from both the analysis on the answer and the highlighted information, it seems that the majority of the industry participants (75% or more) were foraging for information related to main logical components, quality requirements and purpose of the system. The majority of the academic participants were seeking after information related to main logical components, components deployment, process view and use cases. This shows that apart from the information related to the main logical components, both groups have different emphasis on information related to quality requirements, purpose of the system, component deployment, process views and use cases, when looking for the SA of the system.

This perhaps points to a bias for academics to be more concerned with logical structure and function with industry participants being also concerned about structure but more concerned about quality attributes impacting on that structure.

14.4.4 Foraging Sequences to Support Better Understanding of SA

An analysis of suggested foraging sequences reveals that at the outset certain information is essential to be read first (Table of Contents, quality requirements, external dependencies, 'Introduction', main logical components and use cases) to gain understanding of the SA of WCT. Following that, subsequent sections tend to vary among the participants indicating their requirements of different types of other information and of different sequences in reading them, to better support their understanding of the SA.

Comparison with the actual order of the sections in the document shows that to support better understanding of the SA of WCT, the participants suggest foraging sequences that follow the sequencing of information as dictated by the document producers, but skipping some of the sections and deviating intermittently from the flow. This could possibly mean that the sections in this AD have been ordered in a way that supports the understanding of the SA of WCT or the written order of the sections has some influences on the participants' perceptions on the reading sequence that would best support the understanding of the described SA, or both. In either of case, it reinforces the critical responsibility of ADs producers to structure the architectural information conducively in these documents for understanding purposes.

14.4.5 Foraging Styles

In terms of foraging styles, the most apparent difference between the two groups is that the industry participants frequently referred to 'Table of Contents' whereas only a minority of academic participants did that. This could be due to our coincidental recruitment of participants with the aforementioned-foraging styles for both groups.

Some foraging styles are quite popular among the combined participants. They are referencing of Table of Contents, exploration based on titles and subtitles on pages, skipping sections and forward-browsing long section, with the exception of backtracking to previous sections. This phenomenon could be because most of the participants are sequential readers or the information in the AD is well-sequenced with each section sufficiently self-contained.

14.4.6 Understanding Support and Hindrance

The industry participants emphasized 'views' as the way the AD supported their understanding of the WCT software architecture. The academic participants emphasized the diagrams. A diagram can be part of a view but not vice versa. For example a deployment view may consist of a deployment diagram as well as textual explanation related to deployment. All in all, diagrams, views and design decisions were most frequently cited by the combined group of the participants as supporting their understanding of the SA of WCT.

The main problem stated as hindering understanding was too much text with lack of diagrams. But the availability of diagrams topped the list when asked about the ways the AD supported the understanding. Further inspection uncovered that those who stated that the availability of diagrams in the AD supported their understanding were also the ones who said that too much text with lack of diagrams hindered their understanding. It seems that diagrams played a vital role in assisting these participants to understand the SA of WCT and though were present in its AD there were still not enough for them. There are 7 diagrams in the 24-page AD namely: use-case diagram, high level logical components diagram, high level diagram on distributed nature, process diagram and 3 diagrams related to deployment. Nevertheless for graphical-oriented foragers these are still inadequate.

We posit the existence of graphical-oriented foragers, whose inherent cognitive styles (Riding & Cheema, 1991) were more inclined towards imagery as opposed to verbal. This is in tandem with the discovery of a developer group who predominantly made use of diagrams in ADs (Heijstek et al.). AD producers should take into consideration the existence of graphical-oriented foragers so as to produce ADs that support their needs as well as the needs of more textual-oriented foragers.

14.4.7 Other Annotations of Document

Tags and comments provided additional insight on the content of the annotated sections which at times cannot be found by ‘keyword search’. This extra meta-information shared by the previous consumers of the document can be valuable to newcomers but comes at the expense of additional effort. This is evidenced by the small number of the participants who took the opportunity to provide them. We need more data to be able to discuss further on these aspects.

14.5 Threats to Validity

The use of non-probabilistic sampling techniques in recruiting the participants rendered the results not generalizable to the target population (Barbara & Shari Lawrence, 2002). Also contributing to that was the small number of participants who took part in WCT Task 1, in our Study 2. However, these participants had strong experience in SA with 5.75 to 11.25 average years of experience in designing, referring, changing and reviewing SAs of systems. The domains of the systems designed were different between all the five participants who provided information on this. Collectively they also had good experiences in the production, and especially in the consumption of ADs, which is the focus of this exploratory investigation. This leads us to believe that the findings are useful for providing early insights into architectural information foraging. In addition, as in any qualitative analysis, our study is subject to the bias of the researcher when coding the data. This is mitigated by undertaking two meticulous rounds of coding. The choice of the WCT software architecture document could have some influence on participant foraging activities and therefore findings.

We instrumented the AD with annotation fields to capture tags, comments and so on for each section of the document. We also inserted borders to surround each section for easy identification of section. We also separated diagrams from the texts to allow diagrams to be annotated (i.e. tagged, commented and so on) separately from the text. These might have affected the behaviour of the foragers. However, all the participants involved were given the same instrumented AD and since we are looking for 'commonly' foraged information and foraging sequence amongst the participants, this does not affect our results.

14.6 Conclusion

In our investigation, we discovered a number of interesting insights with regards to architectural information foraging in AD. These include issues related to the forages, foraging sequence and styles, features of AD that supported or hindered understanding.

There seems to be different pre-conceived ideas on what to look for with regards to the SA of a system prior to the exploration task. However, it turns out that during the foraging process there was some commonly foraged information. The most popular among the industry participants was information related to main logical components, quality requirements and purpose of the system. The majority of the academic participants were also after information related to main logical components but not the other two. Instead they foraged for information related to components deployment, process view and use cases.

Interestingly apart from the information related to the main logical components, both groups have different emphases on information related to quality requirements, purpose of the system, components deployment, process view and use cases, when looking for the SA of the system. The focus on the main logical components shows that the most important view for the industry participants was the logical view. The academic participants seemed to prefer a more diversified views of the SA by looking at logical view (main logical components), process view and physical view (component deployment).

The focus of industry participants on the quality requirements could be due to their experience with the impact of this type of requirement on the sustainability of the systems they developed. The academic participants most likely had less experience here as the impact of quality requirements might not be fully manifested in the one-off prototype systems they usually build.

It seems that the industry participants were comfortable with knowing the purpose of the system without the use-cases to get an overview of the SA of the system. However the academic participants required detailed use-cases description to understand the functionalities of the system. It was cited as one of the ways the AD supported their understanding of the SA. This shows the two groups required different levels of information in this aspect to understand the described SA.

In terms of foraging sequences, those that started with certain information such as the overview of document ('Table of Contents' and Introduction), logical view (main logical components), quality requirements, use-cases and external dependencies (which may affect other design decisions) were suggested to better support understanding of the described SA. Subsequent sections to be read and their ordering varied based on the requirements of the individuals. Suggested foraging sequences typically followed the written order of the information as dictated by the producers of the document, albeit skipping some of the sections and deviating intermittently from the flow. This is regardless of whether the participants were first time readers or not. This shows that it is of vital importance for the producer of ADs to structure the architectural information in the documents in ways that support understanding.

In general some foraging styles were quite popular. They are referencing of Table of Contents, exploration based on titles and subtitles on pages, skipping sections and forward-browsing long section, with the exception of backtracking to previous section. One possibility of not much backtracking could be the foragers are predominantly sequential readers and therefore sections of ADs should be sufficiently self-contained.

The industry participants reported a more balanced experience in terms of consuming and producing ADs whereas the academics were more of consumers. One possibility of future work is to see if this could have influenced their foraging of architectural information in ADs.

While foraging of architectural information in ADs is quite dependent on individuals, there exists commonly foraged information and general foraging styles. These insights serve as useful considerations for ADs producers when writing these documents. The suggested foraging sequences which typically followed the written order of the information in the ADs, further reinforces the critical responsibility of producers of ADs to structure the architectural information conducively for understanding purposes.

Tags and comments provided additional insight on the content of the annotated sections of the document which at times cannot be found by 'keyword search'. This extra meta-information shared by the previous consumers of the document can be valuable to newcomers but comes at the expense of additional effort. This is evidenced by the small number of the participants who took the opportunity to provide them in our study.

One possibility of future work is to gather more data to investigate further on these aspects.

14.7 **Summary**

This chapter presents our investigation into architectural information foraging in AD, in relation to a task of finding out about the SA of the described system (WCT Task 1). In addition to the overall combined group, we analysed two different sub-groups of foragers, industry practitioner and academic AD users, to investigate issues related to architectural information foraging in AD: types of forages, foraging sequences and styles.

The next chapter concludes the whole research by summarizing the key findings and main contributions of our studies, and possible future work.

---- Blank ----

15 Conclusion and Future Work

This chapter concludes the thesis by summarising our research, highlighting the key findings and main contributions, and discussing possible future work.

15.1 Summary of Research

This research proposed to alleviate the problem of finding relevant information in Software Architecture Documents (ADs) by chunking the architectural information in the documents based on consumers' usage data. Our main approach in identifying chunk was by finding "commonality" in the consumers' usages of the architectural information in the ADs when engaged with certain information-seeking tasks. The "commonality" serves as possible means to group architectural information into a chunk for the task. If such chunks exist and are made available, they serve as alternative approach in searching for information in ADs. As a collection of related pieces of architectural information needed for a particular task, a chunk simplifies finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

We conducted three studies. Study 1 was a user evaluation study conducted to gain feedback on the features of KaitoroCap, the online prototype tool we developed to capture exploration paths and annotation data (Chapter 4). Feedback from Study 1 was used to improve KaitoroCap. The improved version of KaitoroCap was used in Study 3 or Online Approach Study (Chapter 12) to collect usage data. We also conducted Study 2 or Manual Approach Study (Chapter 5) to collect usage data (in particular annotation data) manually without the use of KaitoroCap,

We analysed the usage data (in particular annotation data) collected in Study 2 (Chapter 6 to Chapter 11) and Study 3 (Chapter 12) quantitatively to identify chunks. In doing that, we also investigated the difference between industry practitioners and academic professionals in terms of chunks found and information needed. We consolidated the results from Studies 2 and 3 and summarised them (Chapter 13). In addition, we investigated the issues of architectural information foraging in ADs, while trying to get an overview of the described SA (Chapter 14).

15.2 Key Findings

In this section, we summarise the key findings of our studies with regards to usage-based chunks, difference between industry practitioners and academic professionals in terms of chunks found and information needed, and architectural information foraging in ADs.

15.2.1 Usage-based Chunks

We found that chunks based on consumers' usage data (in particular annotation data) of ADs existed. We discovered that consumers' common preference of a document's sections, in the forms of their ratings of the sections aggregated based on frequency count or average, show potentials to produce chunks of average goodness for the respective task, when the contributing consumers have strong software architecture background and explored the documents in local electronic or printed environment.

Do usage-based chunks support information searching in ADs? The goodness measures of chunks approximate to their support of information searching in ADs for similar information-seeking task. The goodness of chunks were determined using criteria which trade-off the recall and precision measures of the chunks. The recall and precision were calculated by benchmarking the chunks against the oracle set for the task. The recall measure of a chunk tells us how many percentages of the sections needed for the particular task, are in a chunk. The precision measure of a chunk tells us how many percentages of the sections included in a chunk are needed for the particular task. These two measures tell us how complete and precise the chunk is as a collection of related pieces of architectural information needed for a particular task. As a collection of related pieces of architectural information needed for a particular task, a chunk simplifies finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

A considerable number of usage-based chunks (in particular ratings-based chunks found based on the preference of the majority of those who rated) are averagely good for the respective task, when the SA background of the set of contributing consumers are consistently strong and they explored the documents in local electronic or printed environment. So whether usage-based chunks (in particular ratings-based chunks) support information searching in ADs, is much dependent on the SA background of the consumers involved and the environment of the documents.

15.2.2 Industry Practitioners Versus Academic Professionals

We segregated consumers engaged with the same information-seeking task into industry (I) group and academics (A) group, (and student S group in Study 3) based on their background experience in SA. We analysed the responses of each sub-group in addition to the whole combined (C) group, to find chunks and to study the information needed.

From our investigation of which sub-groups' chunks are generally better for a particular task, the findings from Study 2 seem to suggest the use of academic professionals to find chunks for the tasks of getting an overview of the described SA, and tasks related to architectural design on security; but to use industry practitioners to find chunks for tasks related to system changes (changing a part of a system and accessing possible impact of change, and architectural design on modifiability). The latter is properly due to generally industry practitioners have more exposure than academic professionals to system changes, which are frequent in industry settings.

In comparing the information needed by industry practitioners and academics for similar tasks between the two ADs, the most prominent observations are: Industry practitioners and academic professionals differ in their emphasis on process view in getting an overview of the SA of the described systems; For specific tasks of changing a part of a system, the industry practitioners were more able to identify the most critical section in the documents to deal with the tasks: The industry practitioners needed background information when attending cross-cutting concerns tasks. This however could be due to those involved did not have this background knowledge.

15.2.3 Architectural Information Foraging in ADs

We investigated issues - forages, foraging sequence and styles - related to architectural information foraging in ADs, when engaged in the task of discovering the SA of described systems. This is an additional investigation on top of our studies on architectural information chunking. Therefore, we only did this for the data collected for WCT Task 1 in Study 2.

We found that there were different pre-conceived ideas of what to forage for prior to the exploration, but during foraging there was commonly foraged information. The most popular forages among the industry participants were information related to main logical components, quality requirements and purpose of the system. The majority of the academic participants were also after information related to main logical components but not the other two. Instead they foraged for information related to components deployment (physical view), process view and use cases.

In terms of foraging sequences, those that started with certain information such as the overview of document ('Table of Contents' and Introduction), logical view (main logical components), quality requirements, use-cases and external dependencies (which may affect other design decisions) were suggested to better support understanding of the described SA. Subsequent sections to be read and their ordering varied based on the requirements of the individuals. Suggested foraging sequences typically followed the written order of the information as dictated by the AD producers. This reinforces the critical responsibility of AD producers to structure the architectural information to support understanding.

Some foraging styles were quite popular: referencing of Table of Contents, exploration based on titles and subtitles on pages, skipping sections and forward-browsing long section. The exception was backtracking to previous section. The most obvious difference between the industry and academic participants is, the former frequently referred to 'Table of Contents' but only a minority of the latter did that.

Diagrams, views and design decisions were most frequently cited as supporting understanding of the SA. The main hindrance was too much text and a lack of diagrams.

15.2.3.1 Foraging Versus Chunking

Both the investigations into architectural information foraging and chunking aimed to discover insights to support information searching in ADs. The investigation on foraging was more qualitative compared to the investigation on chunking. The difference between chunks and 'common forages' is chunks comprise sections of a document, whereas 'common forages' comprise common information themes or concepts, found needed by the majority of a group of participants who attempted a particular information-seeking task. The information themes or concepts ('common forages') were discovered from 1) the keywords or terms participants searched for in relation to the task, 2) from the participants' answers for the particular information-seeking task, and 3) from the document's content the participants highlighted as relevant to the task.

'Common forages' show what was the commonly foraged information but not the location of where the information could be found. Chunks show the opposite. The themes in 'common forages' can be used to chunk (or group) information or sections of a document. On the other hand, the availability of chunks can reduce the needs of documents' future consumers to become information foragers over and over again.

Recall that when we investigated chunking, we also investigated ‘information needed’. ‘Common forages’ are more similar to ‘information needed’. Both show common information required for the task, but found using different ways. ‘Common forages’ were discovered from the main themes of keywords searched, answers given and content highlighted. ‘Information needed’ was discovered from the main theme of each section in the chunks found using Factor A|H|R3.

‘Information needed’ is more general than ‘common forages’ in the sense that the themes are based on the whole content of individual sections, whereas ‘common forages’ are based on the specific answers given or specific content highlighted in sections. ‘Information needed’ could cover more themes since sections’ ratings were also used in finding it. ‘Information needed’ could also include sections related to organisation of document (such as Table of Contents or sections that contain only the titles of their sub-sections).

Comparing the ‘common forages’ and the ‘information needed’ found for WCT Task 1, the ‘common forages’ of industry, academic and combined groups were subsets of ‘information needed’ by the respective groups.

15.3 **Research Contributions**

The main contributions of our research are:

a. Novelty of chunking based on usage of ADs

To support information searching in ADs, all previous studies except for (Nicoletti et al., 2012) (Diaz-Pace et al., 2013), focused mainly on the production aspect of ADs or Architectural Knowledge. Our research explored the other side of the coin, which is the consumption of the ADs to gain insight to improve information finding. In particular, our research focuses on consumers’ actual usages of ADs and their perceptions of the documents’ content when they were engaged with certain information-seeking tasks. These could be very different from the producers’ anticipated usages of the documents’ content or AK.

We proposed the notion of *chunk* to alleviate information searching problem. As a collection of related pieces of architectural information needed for a particular task, a chunk simplifies finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

We identified chunks by finding “commonality” in the consumers’ usages of the architectural information in the ADs when engaged with certain information-seeking task. The “commonality” serves as possible means to group architectural information into a chunk for the task.

The idea of finding chunks using the “commonality” in the usage data of a group of consumers performing similar task, raises the sense of ‘community of users’ in the otherwise lonely quest of information finding. This brings a number of other research areas into supporting information finding in architecture documentation. This ranges from computational wear, social navigation to collaborative filtering.

We captured two types of usage data: interaction data (manifested as exploration paths in KaitoroCap) and annotation data (ratings, tags, comments, specifications of from which sections answer was found, and highlighting content). Exploration paths serve as usage ‘wear’ left by previous consumers. Visible usage data (exploration paths, ratings, comments and tags) serves as information traces supporting some forms of social navigation. Chunks serve as collaborated filters for information needed for specific tasks.

Exploration paths support the transfer of procedural knowledge (Nickols, 2000). This special type of tacit knowledge (i.e. the knowledge on how people explore existing AK in ADs) had often been ignored by the SA community. Following others' exploration paths does not guarantee better effectiveness or efficiency, but retracing navigation sequences of others', a newcomer can gain insight into their exploration processes enhancing task understanding. This is particularly so if paths are provided by more experienced co-workers.

Chunks serve as guides for exploring architectural information in ADs when performing certain information-seeking tasks. In addition, these usage-based chunks provide alternative suggestions to the documents producer on how architectural information in ADs could be chunked or grouped for easier finding of related pieces of information.

b. Development of KaitoroCap

KaitoroCap was developed using a number of technologies (wiki, plug-in, voting, Web 2.0) deemed valuable for building AKM tools (Liang & Avgeriou, 2009). As a plugin for Atlassian Confluence Enterprise Wiki (Atlassian, 2013a), KaitoroCap leverages the existing functionalities provided by a wiki environment in the creation and management of documents in the form of wiki pages. In addition, KaitoroCap provides the following added advantages:

i. Support for dynamic re-structuring of a document, through automatic creation of page model for each wiki page created or edited.
ii. Capturing of users' exploration paths through document (sequence of visiting pages, interaction with sections of a page, annotation fields and hyperlinks).
iii. Capturing of users' annotation data (tags, comments and ratings) on per section basis, through dynamically inserted annotation fields on every section of a page.
iv. Visualisation of exploration paths in the form of tree-view. Content of visited sections are embedded inside the tree-view visualisation of an exploration path, making a path a dynamic re-structuring of the AD based on a consumer's usage of the document.
v. Additional searching facilities, one that is based on exploration paths.

KaitoroCap is a fully-functional tool that supports document creation and dynamic re-structuring; annotation; exploration path capturing, visualisation and searching. KaitoroCap can be used for all types of documents, and not limited to ADs.

c. Reusability of research instruments

The instruments developed in our research (such as studies designs, data collection instruments, template for the construction of oracle set, and KaitoroCap) can be adapted to study usage-based chunking of information in other types of documentations, for examples requirement documents, user manuals, API documentations, among others.

15.4 **Future Work**

There are many interesting extensions to our research that can be explored in the future:

a. Chunking using other types of usage data

We identified chunks by using annotation data in this research. One possible future work is to identify chunks by using interaction data. Frequency of visiting items (or document sections) and time spent on items can be used to determine the usage patterns in the interaction data. With substantial interaction data, pattern mining tools can also be used to mine common patterns in the interaction data of a pool of consumers. These patterns serve as potential chunks.

b. Improving KaitoroCap to automate the process of chunking

KaitoroCap can be extended to perform the analysis of the usage data to identify chunks, and to embed discovered chunks automatically in KaitoroCap to *filter* the contents (or sections) displayed to new consumers performing similar task. It can also be extended to perform real-time chunking with the arrival of new interaction data.

c. Usage-based chunking of information in other types of documentations

It will be interesting to study usage-based chunking of information in other types of documentations. Examples are requirement documents, user manuals, API documentations, and among others.

d. Evaluation of chunks

We benchmarked chunks against oracle sets. Each oracle set was constructed using a vigorous process involving two SA professionals. We think that, changing to different oracle sets would most likely have very minimal effect on our overall findings, which are, usage-based chunks exist and factors based on the preference of the majority of those who rated (be it Factor AveR3 or Factor R3) show potentials in finding chunks of average goodness for the respective task. Nevertheless, verification of this can be part of our future work.

In addition, evaluation study can be conducted to see how consumers make use of chunks in information searching.

e. Task specification

Another area of future work is related to the tasks performed by stakeholders of architectural information. This includes investigating common information-seeking tasks of different groups of stakeholders, ways to classify and organise tasks, data to include in task specifications, among others. The purpose is to enable usage data collected for tasks to be grouped together based on the similarity of tasks, to identify chunks for the particular group of task.

f. Finer granularity of architectural information

We experimented with 'document section' as the level of granularity for chunk elements. To study chunking at a finer level of granularity, the document can be organised so that each section describes only one architectural element. Our approach can also be adapted to the model-based approach to study whether a set of architectural elements are actually used in the way they are pre-chained by formal models.

g. Foraging of architectural information

The investigation into the foraging of architectural information in ADs can be extended to the other five information-seeking tasks, to see if the foraging sequences and styles change with the change of tasks.

15.5 Summary

This research investigated the consumption of ADs to gain insights to support information searching in these documents. This complements most existing studies which focused mainly on the production aspect of ADs or AK, to support information finding.

We proposed the notion of *chunk* to alleviate information searching problem. As a collection of related pieces of architectural information needed for a particular task, a chunk simplifies finding of information by consumers engaged with similar tasks, by enabling related architectural information which otherwise may be dispersed in an AD to be retrieved collectively as a unit.

Our approach in identifying chunk is by finding "commonality" in the consumers' usages of the architectural information in the ADs when engaged with certain information-seeking task. The "commonality" serves as possible means to group architectural information into a chunk for the task. If such chunks exist and are made available, they serve as alternative approach to search for information in ADs.

We collected two types of usage data: interaction data (manifested as exploration paths in KaitoroCap) and annotation data (ratings, tags, comments, specifications of from which sections answer was found, and highlighted content). We used six chunk-identification factors that made use of three types of annotation data (ratings, specifications of from which sections answer was found, and the presence of highlighted information in sections) to identify chunks.

We found that chunks based on consumers' usage data (in particular annotation data) of ADs existed. We found that consumers' common preference of a document's sections, in the forms of their ratings of the sections aggregated based on frequency count or average, show potentials to produce chunks of average goodness for the respective task, when the contributing consumers have strong software architecture background and explored the documents in local electronic or printed environment. The goodness of a chunk was determined using criteria which trade-off the recall and precision measures of the chunk, calculated by benchmarking against the oracle set for the task. The goodness of a chunk approximates to its support of information searching in the document for similar information-seeking task.

We found some differences between industry practitioners and academic professionals in terms of chunks found and information needed. In addition to architectural information chunking, we investigated architectural information foraging to study the behaviours of architectural information foragers in terms of common forages, foraging sequences, and foraging styles.

As the conclusion, our research produced some early insights on usage-based chunking of architectural information and also architectural information foraging. However, much still need to be done to gain better insights into supporting information searching in ADs based on their actual consumption by consumers, with many exciting challenges-cum-opportunities waiting to be addressed.

Appendices

---- Blank ----

Appendix 4.1: Data Collection Instruments (Study 1)

Appendix 5.1: Data Collection Instruments (Study 2)

Appendix 5.2: Oracle Set Construction Template

Appendix 12.1: Data Collection Instruments (Study 3)

https://docs.google.com/file/d/0B9mNuBce6NmMYnNUcXdFRlBkWms/edit?usp=sharing

---- Blank ----

References

Agarwal, R. C., Aggarwal, C. C., & Prasad, V. V. V. (2000). A tree projection algorithm for generation of frequent itemsets. *Journal of Parallel and Distributed Computing, 61*, 350-371.

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec., 22*(2), 207-216. doi: 10.1145/170036.170072

AKSW, Agile Knowledge Engineering and Semantic Web. (2013). OntoWiki. Retrieved 1 March, 2013, from http://aksw.org/Projects/OntoWiki.html

Anderson, J. R. (1976). *Language, Memory and Thought*. Hillsdale: Lawrence Erlbaum Associates, Inc.

Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale: Lawrence Erlbaum Associates, Inc.

Anderson, J. R. (1995). *Cognitive Psychology and Its Implications* (4th ed.). New York: W.H. Freeman and Company.

Atlassian. (2013a). Confluence. 2009, from http://www.atlassian.com/software/confluence/

Atlassian. (2013b). Set up the Atlassian Plugin SDK and Build a Project 2009, from https://developer.atlassian.com/display/DOCS/Set+up+the+Atlassian+Plugin+SDK+and+Build+a+Project

Atlassian. (2013c). XWork-WebWork Module. 2009, from http://confluence.atlassian.com/display/CONFDEV/XWork-WebWork+Module

Atlassian. (2013d). Velocity Template Overview. 2009, from http://confluence.atlassian.com/display/CONFDEV/Velocity+Template+Overview

Avgeriou, P., Kruchten, P., Lago, P., Grisham, P., & Perry, D. (2007). Architectural knowledge and rationale: issues, trends, challenges. *ACM SIGSOFT SE Notes, 32*(4), 41-46.

Babar, M. A. (2009). Supporting the Software Architecture Process with Knowledge Management. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.

Babar, M. A., de Boer, R. C., Dingsoyr, T., & Farenhorst, R. (2007). *Architectural Knowledge Management Strategies: Approaches in Research and Industry.* Paper presented at the 2nd Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent, 2007 (SHARK/ADI '07) Minneapolis, MN, May 19 - 27, 2007.

Babar, M. A., & Gorton, I. (2007). *A Tool for Managing Software Architecture Knowledge.* Paper presented at the 2nd Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent, 2007. (SHARK/ADI '07), Minneapolis, MN, May 19 - 27, 2007.

Babar, M. A., Gorton, I., & Jeffery, R. (2005). *Capturing and using software architecture knowledge for architecture-based software development.* Paper presented at the Fifth International Conference on Quality Software, 2005. (QSIC 2005). .

Bachmann, F., & Merson, P. (2005). Experience Using the Web-Based Tool Wiki for Architecture Documentation. : SEI-2005-TN-041. September 2005.

Barbara, K., & Shari Lawrence, P. (2002). Principles of survey research: part 5: populations and samples. *SIGSOFT Softw. Eng. Notes, 27*(5), 17-20. doi: 10.1145/571681.571686

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice*. Boston: Addison-Wesley Professional.

Bass, L., & Kazman, R. (1999). Architecture-based development: CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin? *Commun. ACM, 35*(12), 29-38. doi: 10.1145/138859.138861

Bosch, J. (1999). *Evolution and Composition of Reusable Assets in Product-Line Architectures: A Case Study.* Paper presented at the TC2 First Working IFIP Conference on Software Architecture (WICSA1).

Bosch, J. (2004). Software architecture: The next step. *Lecture Notes in Computer Science*, 194-199.

Bruckman, A. (1998). Finding one's own in cyberspace. *High Wired: On the Design, Use, and Theory of Educational MOOs. Ed. Cynthia Haynes and Jan Rune Holmevik. Ann Arbor, MI: U of Michigan P*, 15-24.

Buchgeher, G., & Weinreich, R. (2008). *Integrated Software Architecture Management and Validation*.

Buckland, M., & Gey, F. (1994). The relationship between recall and precision. *JASIS, 45*(1), 12-19.

Burge, J. E., & Brown, D. C. (2008). *SEURAT: integrated rationale management*. Paper presented at the Proceedings of the 30th international conference on Software engineering, Leipzig, Germany.

Capilla, R., Nava, F., Pérez, S., & Dueñas, J. C. (2006). A web-based tool for managing architectural design decisions. *ACM SIGSOFT SE Notes, 31*(5).

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., . . . Stafford, J. (2010). *Documenting software architectures: views and beyond* (2nd Edition ed.): Addison-Wesley Professional.

Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. (2003). *Documenting software architectures: views and beyond*: Pearson Education.

Curtis, B. (1984). *Fifteen years of psychology in software engineering: Individual differences and cognitive science*.

Dag, I. K. S., Tore, D., & Magne, J. (2007). *The Future of Empirical Methods in Software Engineering Research*. Paper presented at the 2007 Future of Software Engineering.

de Boer, R. C. (2006). Architectural knowledge discovery: why and how? *SIGSOFT Softw. Eng. Notes, 31*(5), 1. doi: 10.1145/1163514.1178641

de Boer, R. C. (2011). *Experiences with Semantic Wikis for Architectural Knowledge Management*.

de Boer, R. C., & van Vliet, H. (2007a, 6-9 Jan. 2007). *Constructing a Reading Guide for Software Product Audits.* Paper presented at the Software Architecture, 2007. WICSA '07. The Working IEEE/IFIP Conference on.

de Boer, R. C., & van Vliet, H. (2008). Architectural knowledge discovery with latent semantic analysis: Constructing a reading guide for software product audits. *J. Syst. Softw., 81*(9), 1456-1469.

de Boer, R. C., & van Vliet, H. (2009). *Writing and Reading Software Documentation: How the development process may affect understanding.* Paper presented at the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering.

de Boer, R. C., & van Vliet, H. (2011). *Experiences with semantic wikis for architectural knowledge management.* Paper presented at the Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on.

de Graaf, K. A., Tang, A., Liang, P., & Van Vliet, H. (2012, 20-24 Aug. 2012). *Ontology-based Software Architecture Documentation.* Paper presented at the Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science, 41*(6), 391-407.

DeLine, R., Czerwinski, M., & Robertson, G. (2005, September 20 - 24, 2005). *Easing program comprehension by sharing navigation data.* Paper presented at the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), Dallas, Texas

DeLine, R., Khella, A., Czerwinski, M., & Robertson, G. (2005). *Towards understanding programs through wear-based filtering*. Paper presented at the Proceedings of the 2005 ACM symposium on Software visualization, St. Louis, Missouri.

Desouza, K., Awazu, Y., & Baloh, P. (2006). Managing Knowledge in Global Software Development Efforts: Issues and Practices. *Software, IEEE, 23*(5), 30-37. doi: 10.1109/MS.2006.135

Diaz-Pace, J. A., Nicoletti, M., Schiaffino, S., Villavicencio, C., & Sanchez, L. (2013). A Stakeholder-Centric Optimization Strategy for Architectural Documentation. In A. Cuzzocrea & S. Maabout (Eds.), *Model and Data Engineering* (Vol. 8216, pp. 104-117): Springer Berlin Heidelberg.

Dieberger, A., Dourish, P., Höök, K., Resnick, P., & Wexelblat, A. (2000). Social navigation: techniques for building more usable systems. *interactions, 7*(6), 36-45.

Dieberger, A., & Guzdial, M. (2003). CoWeb—experiences with collaborative web spaces *From Usenet to CoWebs* (pp. 155-166): Springer.

Dingsøyr, T. (2009). Strategies and Approaches for Managing Architectural Knowledge. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.

Dingsøyr, T., & Vliet, H. (2009). Introduction to Software Architecture and Knowledge Management. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.

Dourish, P., & Chalmers, M. (1994). *Running out of space: Models of information navigation.* Paper presented at the Proceedings of the Conference Human Computer Interaction '94.

Earl, M. (2001). Knowledge Management Strategies: Toward a Taxonomy. *J. Manage. Inf. Syst., 18*(1), 215-233.

Easterbrook, S., Singer, J., Storey, M.-a., & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer & D. I. K. Sjøberg (Eds.), *Guide to Advanced Empirical Software Engineering* (pp. 285-311). London: Springer-Verlag.

Edland, A., & Svenson, O. (1993). Judgment and Decision Making Under Time Pressure, Studies and Findings. In A. J. Maule & O. Svenson (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making* (pp. 36). New York: Plenum Press.

Eloranta, V.-P., Hylli, O., Vepsalainen, T., & Koskimies, K. (2012). *TopDocs: Using Software Architecture Knowledge Base for Generating Topical Documents.* Paper presented at the Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on.

Elves, R. (2010). Tasktop for Eclipse - Get More out of Mylyn. Retrieved July 15, 2010, 2010, from http://tasktop.com/resources/tutorials/MoreOutOfMylyn.php

Farenhorst, R., & de Boer, R. C. (2009). Knowledge Management in Software Architecture: State of the Art. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.

Farenhorst, R., Izaks, R., Lago, P., & van Vliet, H. (2008, February 18 – 22, 2008). *A Just-In-Time Architectural Knowledge Sharing Portal.* Paper presented at the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), Vancouver, BC, Canada.

Farenhorst, R., Lago, P., & van Vliet, H. (2007a). Effective Tool Support for Architectural Knowledge Sharing. *Software Architecture, LNCS, 4758*, 123-138.

Farenhorst, R., Lago, P., & van Vliet, H. (2007b). EAGLE: Effective Tool Support for Sharing Architectural Knowledge. *International Journal of Cooperative Information Systems, 16*(3-4), 413-438.

Farenhorst, R., & van Vliet, H. (2008). *Experiences with a Wiki to Support Architectural Knowledge Sharing.* Paper presented at the Wikis4SE @ WikiSym 2008, Porto, Portugal, September 8 - 10, 2008.

Farenhorst, R., & Van Vliet, H. (2009). *Understanding how to support architects in sharing knowledge.* Paper presented at the 2009 ICSE Workshop on Sharing and Reusing Architectural Knowledge, Vancouver, Canada, May 16-24, 2009.

Freebase. (2013). Freebase. Retrieved 1 April, 2009, from http://www.freebase.com/

Garlan, D., & Perry, D. E. (1995). Introduction to the special issue on software architecture. *IEEE Transactions on software engineering, 21*(4), 269-274.

Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C. H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences, 5*(6), 236-243. doi: http://dx.doi.org/10.1016/S1364-6613(00)01662-4

Gobet, F., & Lane, P. R. (2012). Chunking Mechanisms and Learning. In N. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 541-544): Springer US.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM, 35*(12), 61-70. doi: 10.1145/138859.138867

Grady, B., Rumbaugh, J., & Jacobson, I. (1999). The unified modeling language user guide. *Reading: Addison Wesley Longman, Inc*.

Greefhorst, D., Koning, H., & Vliet, H. (2006). The many faces of architectural descriptions. *Information Systems Frontiers, 8*(2), 103-113.

Hai, Z. (2004). *The Knowledge Grid*: World Scientific Publishing Company.

Hansen, M. T., Nohria, N., & Tierney, T. (1999). What is your strategy for managing knowledge? . *Harvard Business Review, 77*(2), 106 - 116.

Heijstek, W., Kuhne, T., & Chaudron, M. R. V. (22-23 Sept. 2011). *Experimental Analysis of Textual and Graphical Representations for Software Architecture Design.* Paper presented at the 2011 International Symposium on Empirical Software Engineering and Measurement (ESEM).

Hill, W. C., & Hollan, J. D. (1994). History‐enriched digital objects: Prototypes and policy issues. *The Information Society, 10*(2), 139-145.

Hill, W. C., Hollan, J. D., Wroblewski, D., & McCandless, T. (1992, May 3-7, 1992). *Edit wear and read wear.* Paper presented at the SIGCHI conference on Human factors in computing systems, Monterey, California, USA.

Hirsch, C., Hosking, J., Grundy, J., Chaffe, T., MacDonald, D., & Halytskyy, Y. (2009, January 5 - 8, 2009). *The Visual Wiki: A New Metaphor for Knowledge Access and Management.* Paper presented at the 42nd Hawaii International Conference on System Sciences, Waikoloa, Big Island, Hawaii.

Horn, R. E. (1997). Structured Writing as a Paradigm. In C. R. Dills & A. J. Romiszowski (Eds.), *Instructional Development Paradigms*. New Jersey: Educational Technology Publications.

IEEE. (2000). Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE-Std-1471-2000).

ISO/IEC. (2007). ISO/IEC 42010: Systems and software engineering - Recommended practice for architectural description of software-intensive systems (pp. c1-24).

ISO/IEC/IEEE. (2011). Systems and software engineering -- Architecture description *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)* (pp. 1-46).

Jansen, A., Avgeriou, P., & van der Ven, J. S. (2009). Enriching software architecture documentation. *J. Syst. Softw., 82*(8), 1232-1248. doi: 10.1016/j.jss.2009.04.052

Jansen, A., & Bosch, J. (2005). *Software architecture as a set of architectural design decisions.* Paper presented at the Fifth Working IEEE/IFIP Conference on Software Architecture (WICSA'05), Pittsburgh, Pennsylvania, November 6 - 10, 2005.

Jansen, A., Van der Ven, J., Avgeriou, P., & Hammer, D. K. (2007). *Tool support for architectural decisions.* Paper presented at the Working IEEE/IFIP Conference on Software Architecture (WICSA'07) Mumbai, India, January 6 - 9, 2007.

Kersten, M., & Murphy, G. C. (2005). *Mylar: a degree-of-interest model for IDEs*. Paper presented at the Proceedings of the 4th international conference on Aspect-oriented software development, Chicago, Illinois.

Ko, A. J., Myers, B. A., Coblenz, M. J., & Aung, H. H. (2006). An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks. *IEEE Transactions on software engineering, 32*(12), 971-987. doi: 10.1109/tse.2006.116

Koning, H., & van Vliet, H. (2006). Real-life IT architecture design reports and their relation to IEEE Std 1471 stakeholders and concerns. *Automated Software Engineering, 13*(2), 201-223. doi: 10.1007/s10515-006-7736-6

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM, 40*(3), 77-87. doi: 10.1145/245108.245126

Konstan, J. A., Riedl, J., Borchers, A., & Herlocker, J. L. (1998). *Recommender systems: A grouplens perspective.* Paper presented at the Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08).

Kruchten, P. (2009). Documentation of Software Architecture from a Knowledge Management Perspective –Design Representation. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.

Kruchten, P., Lago, P., & van Vliet, H. (2006). Building up and reasoning about architectural knowledge *Quality of Software Architectures* (Vol. 4214, pp. 43-58): Springer Berlin/Heidelberg.

Lago, P., & Avgeriou, P. (2006). First workshop on sharing and reusing architectural knowledge. *SIGSOFT Softw. Eng. Notes, 31*(5), 32-36. doi: 10.1145/1163514.1163526

Lago, P., Avgeriou, P., Capilla, R., & Kruchten, P. (2008, 18 – 22 February 2008). *Wishes and Boundaries for a Software Architecture Knowledge Community.* Paper presented at the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), Vancouver, BC, Canada, February 18 – 22, 2008.

Lawrance, J., Bogart, C., Burnett, M., Bellamy, R., Rector, K., & Fleming, S. (2010). How Programmers Debug, Revisited: An Information Foraging Theory Perspective. *IEEE Transactions on software engineering, PP*(99), 1-1. doi: 10.1109/tse.2010.111

Lerman, K. (2007). Social Information Processing in News Aggregation. *Internet Computing, IEEE, 11*(6), 16-28. doi: 10.1109/MIC.2007.136

Lesk, M. E., & Salton, G. (1968). Relevance assessments and retrieval system evaluation. *Information Storage and Retrieval, 4*(4), 343-359. doi: http://dx.doi.org/10.1016/0020-0271(68)90029-6

Lethbridge, T. C., Singer, J., & Forward, A. (2003). How software engineers use documentation: the state of the practice. *IEEE SOFTWARE, 20*(6), 35-39. doi: 10.1109/MS.2003.1241364

Liang, P., & Avgeriou, P. (2009). Tools and Technologies for Architecture Knowledge Management. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.

Liang, P., Jansen, A., & Avgeriou, P. (2009). Knowledge architect: a tool suite for managing software architecture knowledge. Technical Report RUG-SEARCH-09-L01: University of Groningen.

Liang, P., Jansen, A., & Avgeriou, P. (2010). Collaborative Software Architectingcollaborative software architecting Through Knowledge Sharing. In I. Mistrík, J. Grundy, A. Hoek & J. Whitehead (Eds.), *Collaborative Software Engineering* (pp. 343-367): Springer Berlin Heidelberg.

Loeb, S., & Terry, D. (1992). Information filtering. *Commun. ACM, 35*(12), 26-28. doi: 10.1145/138859.138860

Louridas, P. (2006). Using Wikis in Software Development. *IEEE SOFTWARE, 23*(2), 88-91.

Maglio, P., & Barrett, R. (2000). Intermediaries personalize information streams. *Communications of the ACM, 43*(8), 96-101.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Evaluation in information retrievalIntroduction to Information Retrieval (pp. 155-156). Retrieved from http://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp45430.

Maybury, M. T. (1995). Generating summaries from event data. *Information Processing & Management, 31*(5), 735-751.

Mik, K., & Gail, C. M. (2006, November 5-11, 2006). *Using task context to improve programmer productivity.* Paper presented at the 14th ACM SIGSOFT international symposium on Foundations of software engineering, Portland, Oregon, USA.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for information processing. *Psychological Review, 63*(2), 81-97.

Miller, G. A. (1983). Informavores. In F. Machlup & U. Mansfield (Eds.), *The Study of Information: Interdisciplinary Messages* (pp. 111-113). New York: Wiley.

Munro, A. J., Hook, K., & Benyon, D. (1999). Footprints in the Snow *Social navigation of information space* (pp. 1-14): Springer.

National Library of New Zealand. (2006). Web Curator Tool Software Architecture Document. Retrieved 1 June 2009, from http://webcurator.sourceforge.net/docs/1.0/wct-system-architecture.pdf

Nickols, F. (2000). The Knowledge in Knowledge Management. In J. A. Woods & J. Cortada (Eds.), *The Knowledge Management Yearbook 2000-2001* (pp. 12-21): Butterworth-Heinemann.

Nicoletti, M., Diaz-Pace, J. A., & Schiaffino, S. (2012). Towards Software Architecture Documents Matching Stakeholders' Interests. In F. Cipolla-Ficarra, K. Veltman, D. Verber, M. Cipolla-Ficarra & F. Kammüller (Eds.), *Advances in New Technologies, Interactive Interfaces and Communicability* (Vol. 7547, pp. 176-185): Springer Berlin Heidelberg.

Niu, N., Mahmoud, A., & Bradshaw, G. (2011). *Information foraging as a foundation for code navigation (NIER track).* Paper presented at the of the 33rd International Conference on Software Engineering, Waikiki, Honolulu, HI, USA.

Parnas, D. L. (2005). *Document driven disciplined development of software.* Paper presented at the Software Engineering Conference, 2005. Proceedings. 2005 Australian.

Parnas, D. L., & Clements, P. C. (1986). A rational design process: How and why to fake it. *IEEE Transactions on software engineering, 12*(2), 251-257.

Perry, D. E., & Wolf, A. L. (1992). Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes, 17*(4), 40.

Pirolli, P. (2007). *Information Foraging Theory: Adaptive Interaction with Information* (1st Edition ed.). New York: Oxford University Press.

Pirolli, P., & Card, S. (1995). *Information foraging in information access environments.* Paper presented at the of the SIGCHI conference on Human factors in computing systems, Denver, Colorado, United States.

Polanyi, M. (1997). Tacit Knowledge. In L. E. Prusak (Ed.), *Knowledge in Organisations*. Boston: Butterworth-Heinemann.

Rajaraman, A., Leskovec, J., & Ullman, J. D. (2013). *Mining of Massive Datasets, Chapter 6* Retrieved from http://infolab.stanford.edu/~ullman/mmds.html#latest

Reddit Inc. (2013). Reddit. Retrieved 1 March 2013, 2013, from http://www.reddit.com/

Riding, R., & Cheema, I. (1991). Cognitive Styles—an overview and integration. *Educational Psychology, 11*(3-4), 193-215. doi: 10.1080/0144341910110301

Rost, D. (2012). *Generation of task-specific architecture documentation for developers*. Paper presented at the Proceedings of the 17th international doctoral symposium on Components and Architecture, Bertinoro, Italy.

Rost, D., Naab, M., Lima, C., & von Flach Chavez, C. (2013). Architecture Documentation for Developers : A Survey (1.0 ed.).

Rozanski, N., & Woods, E. (2005). *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*: Addison-Wesley Professional.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In P. Brusilovsky, A. Kobsa & W. Nejdl (Eds.), *The Adaptive Web* (Vol. 4321, pp. 291-324): Springer Berlin Heidelberg.

Schuster, N., Zimmermann, O., & Pautasso, C. (2007). *ADkwik: Web 2.0 collaboration system for architectural decision engineering*. Paper presented at the Proceedings of the 19th International Conference on Software Engineering & Knowledge Engineering (SEKE).

Shardanand, U., & Maes, P. (1995). *Social information filtering: algorithms for automating &ldquo;word of mouth&rdquo*. Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado, USA.

Shaw, M., & Garlan, D. (1996). *Software architecture: perspectives on an emerging discipline*: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Siw Elisabeth, H., & Bente, A. (2005). *Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research*. Paper presented at the Proceedings of the 11th IEEE International Software Metrics Symposium.

Slupesky, T., & Singleton, T. (2006). Aperi Architecture Document. Retrieved 1 June 2009, from http://wiki.eclipse.org/Aperi_Architecture_Document

Smolander, K. (2002). *Four metaphors of architecture in software organisations: finding out the meaning of architecture in practice.* Paper presented at the Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n.

Smolander, K., & Paivarinta, T. (2002b). Describing and Communicating Software Architecture in Practice: Observations on Stakeholders and Rationale *Advanced Information Systems Engineering* (Vol. 2348, pp. 117-133): Springer Berlin/Heidelberg.

Smolander, K., Rossi, M., & Purao, S. (2005). *Going beyond the blueprint: Unraveling the complex reality of software architectures*.

Software Engineering Institute, C. M. U. (2012). Community Software Architecture Definitions. Retrieved 18/12/2012, from http://www.sei.cmu.edu/architecture/start/glossary/community.cfm

Stephens, D. W., & Krebs, J. R. (1987). *Foraging Theory*. NJ: Princeton University Press.

Su, M. T., Hirsch, C., & Hosking, J. (2009). *KaitoroBase: Visual Exploration of Software Architecture Documents*. Paper presented at the Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering.

SurveyMonkey. (2012). SurveyMonkey. 2010, from http://www.surveymonkey.com/

Svensson, M., Höök, K., Laaksolahti, J., & Waern, A. (2001). *Social navigation of food recipes.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Tang, A., Avgeriou, P., Jansen, A., Capilla, R., & Babar, M. A. (2009). A comparative study of architecture knowledge management tools. *J. Syst. Softw., 83*(3), 352-370.

Tang, A., & Han, J. (2005). *Architecture rationalization: a methodology for architecture verifiability, traceability and completeness*.

Tang, A., Jin, Y., & Han, J. (2007). A rationale-based architecture model for design traceability and reasoning. *The Journal of Systems & Software, 80*(6), 918-934.

Tang, A., Liang, P., & van Vliet, H. *Software Architecture Documentation: The Road Ahead.* Paper presented at the the 9th IEEE/IFIP Working Conference on Software Architecture (WICSA), Boulder, Colorado, United States.

Tasktop Technologies Inc. (2013). Tasktop Dev for Eclipse.   Retrieved Sept 2013, 2013, from http://tasktop.com/eclipse#benefits

Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2008). Software Architecture: Foundations, Theory, and Practice: John Wiley & Sons.

The Eclipse Foundation. (2013). Mylyn.   Retrieved July 15, 2010, 2010, from http://www.eclipse.org/mylyn/

The jQuery Foundation. (2013). JQuery. 2009, from http://jquery.com/

The Open Group. (2013). The Open Group Architecture Framework (TOGAF). from http://www.opengroup.org/togaf/

Thinkbase. (2009). Thinkbase. from http://thinkbase.cs.auckland.ac.nz/

Thomas, F., Jingwen, O., Gail, C. M., & Emerson, M.-H. (2010, May 2 - 8, 2010). *A degree-of-knowledge model to capture source code familiarity.* Paper presented at the 32nd ACM/IEEE International Conference on Software Engineering, Cape Town, South Africa.

Tyree, J., & Akerman, A. (2005). Architecture decisions: demystifying architecture. *IEEE SOFTWARE, 22*(2), 19-27.

van der Ven, J., Jansen, A., Avgeriou, P., & Hammer, D. (2006). *Using architectural decisions.* Paper presented at the 2nd International Conference on the Quality of Software-Architectures (QoSA).

VU University of Amsterdam. (2010). ClioPatria: the SWI-Prolog RDF toolkit. from http://cliopatria.swi-prolog.org/home

Wallis, P., & Thom, J. A. (1996). Relevance judgments for assessing recall. *Inf. Process. Manage., 32*(3), 273-286. doi: 10.1016/0306-4573(95)00061-5

Wexelblat, A., & Maes, P. (1999). *Footprints: history-rich tools for information foraging*. Paper presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems, Pittsburgh, Pennsylvania, USA.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimentation in software engineering: an introduction*. Norwell, MA, USA: Kluwer Academic Publishers

Yahoo! Inc. (2013). Flickr. 2013, from www.flickr.com

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal, 26*(3), 276-292. doi: 10.1147/sj.263.0276

Zimmermann, O., Kopp, P., & Pappe, S. (2009). Architectural Knowledge in an SOA Infrastructure Reference Architecture. In M. A. Babar, T. Dingsøyr, P. Lago & H. E. Vliet (Eds.), *Software Architecture Knowledge Management: Theory and Practice* Springer.