Journal of Systems and Software, Volume 122, December 2016, Pages 215–238 (c) Elsevier 2016.

Usage-based Chunking of Software Architecture Information to Assist Information Finding

Moon Ting Su[1]
Department of Computer Science,
University of Auckland,
Auckland, New Zealand
smting@um.edu.my

John Hosking
Faculty of Science,
University of Auckland,
Auckland, New Zealand
j.hosking@auckland.ac.nz

John Grundy
Faculty of Science, Engineering and Technology,
Swinburne University of Technology,
Victoria, Australia
jgrundy@swin.edu.au

Ewan Tempero
Department of Computer Science,
University of Auckland,
Auckland, New Zealand
e.tempero@auckland.ac.nz

Abstract
One of the key problems with Software Architecture Documents (ADs)[2] is the difficulty of finding information required from them. Most existing studies focus on the production of ADs or Architectural Knowledge (AK)[3], to allow them to support information finding. However, there has been little focus placed on the consumption of ADs. To address this, we postulate the existence of a concept of "usage-based chunks" of architectural information discoverable from consumers' usage of ADs when they engage in information-seeking tasks. In a set of user studies, we have found evidence that such usage-based chunks exist and that useful chunks can be identified from one type of usage data, namely, consumer's ratings of sections of ADs. This has implications for tool design to support the effective reuse of AK.

Keywords: usage-based chunking; software architecture document; information finding; task

1 Introduction

Finding useful information in large amounts of software documentation is not easy. This is a key problem in addition to the perennial problems of out of date (but sometimes still useful), poorly written and untrustworthy documents that have a high creation cost (Lethbridge, Singer, & Forward, 2003). The difficulty of finding information also applies more specifically to Software Architecture Documents (ADs) (Koning & van Vliet, 2006; Rost, Naab, Lima, & von Flach Chavez, 2013).

[1] Corresponding Author. Present Address: Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, Lembah Pantai, 50603 Kuala Lumpur, Malaysia, e-mail: smting@um.edu.my, Tel: +603 79676369, Fax: +603 79579249
[2] Software Architecture Document (AD)
[3] Architectural Knowledge (AK)

ADs hold many benefits for Architectural Knowledge (AK) sharing but as documentation increases with size and complexity of the software system, many challenges await current Software Architecture (SA) documentation approaches (Jansen, Avgeriou, & van der Ven, 2009). One of these challenges is locating relevant AK (Avgeriou, Kruchten, Lago, Grisham, & Perry, 2007; Jansen et al., 2009) either across multiple documents or within these documents (Jansen et al., 2009). Knowledge retrieval features in existing AK management tools are simple and reactive (Tang, Avgeriou, Jansen, Capilla, & Babar, 2009).

The problem of finding information from ADs is further worsened by the various stakeholders' having only partial interest in the total content of the documents. Many stakeholders' concerns are addressed by a small fraction (sometimes as little as 25%) of an AD (Koning & van Vliet, 2006). Consequently, the readers of ADs complain of having to wade through too much irrelevant information. Information needed to solve a specific task may be spread throughout the document and be organised in a linear fashion not matching user needs for a specific AK information-seeking task.

Thus, despite the wealth of AK that ADs contain, they may not be used, or not used most effectively, because of the difficulty of finding information in them. To support finding information in an AD, we argue that architectural information in it needs to be structured into or presented as *chunks* (Su, 2010; Su, Hosking, & Grundy, 2011a, 2011b). A *chunk* is *a collection of related pieces of architectural information* (Su, 2010; Su et al., 2011a, 2011b). We posit that identifying and reusing chunks simplifies finding of information, by enabling related architectural information, which may be dispersed in a document, to be retrieved collectively as a unit. We propose to identify chunks by finding 'commonality' in consumers' usage of the information in ADs when engaged with certain information-seeking tasks.

We investigated this idea by carrying out studies that acquired AD usage data when consumers performed certain information-seeking tasks. We collected both explicit data, where consumers were asked to provide information about their AD usage, and implicit data, where the usage data was gathered by KaitoroCap (Su, 2014; Su et al., 2011b), the tool we developed to track consumers' interaction with ADs. We analysed the collected usage data to identify chunks for these tasks. Our work is a preliminary study of the concept of 'usage-based chunks' in ADs. Our work aims to show that usage-based chunks exist and that they are likely to vary across different information-seeking tasks in an AD. We chose three representative use cases (or information-seeking tasks) for SA documentation to illustrate this.

This paper is organised as below: Section 2 explains the concepts of chunking and information chunk. Section 3 presents the methodology. Section 4 discusses the chunking results. Section 5 details the threats to validity of our findings. Section 6 compares our work with existing work. Section 7 presents our key findings, conclusions and possible future work.

2 Chunks to Support Finding of Information

In this section we present the concepts of "chunking" and "information chunk" introduced in other research areas and how we have adopted these concepts in our work. We also review existing work in the field of SA that supports different forms of chunking, and define the concept of a "chunk" as used in our work.

2.1 Chunking and Information Chunk

The idea of chunking in this research draws upon a number of areas that involve human processing of information. These include human cognition, human learning, perception, and the study of chess. In these areas, *chunking* generally refers to the grouping of related items into a single unit or *chunk*. 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 terms are also used in the study of expertise, and acquisition of language and education, all of which are related to learning.

Miller, a cognitive psychologist suggested that our short-term or working memory can only hold 'seven plus-or-minus two' (i.e. between five to nine) items (Miller, 1956). While this finding may not be universally true, there is nevertheless some limitation on how much information we can process and recall. However, the capacity of the working memory 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, work in cognitive science has established chunking as one of the key mechanisms of human cognition (Gobet et al., 2001).

Chunking can be goal-oriented, involving a deliberate conscious process (Gobet & Lane, 2012). An example is Miller's re-coding of specific 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 has been used to explain the ability of chess experts to recall briefly-presented positions with high precision.

We adopt similar notions for these terms in our work: *chunking* here refers to the *grouping of related pieces of information* and a *chunk* is *a collection of related pieces of architectural information*. We observe that the principle underlying all the above areas in human processing of information is: the users or consumers of information construct information chunks during their usage of the information, and use the chunks in later recall or retrieval of the information. Our work builds upon this principle of how humans process information and takes it further in two aspects. Firstly, by making the derivation of information chunks explicit. Secondly, the derivation of the information chunks is based on the 'commonality' found in the consumers' usage of information. The 'commonality' serves as possible means to group information into a chunk.

All the areas above focus on the consumption of information. *Chunking* also exists in structured writing (Horn, 1997), which focuses on the production of information. In structured writing, chunking refers to grouping of pieces of information into manageable units, called information blocks and information maps. An information block is the basic unit of subject matter. An information map is a collection of information blocks. The notion of information map in structured writing resembles the notion of a chunk in the areas that focus on the consumption of information mentioned earlier. We use the term *chunk* instead of *information map*, since our work focuses on the consumption instead of the production of information, and, the chunking principle used in structured writing originates from Miller's work in human cognition (Miller, 1956).

2.2 **Chunking in the Field of Software Architecture**

In the field of SA, there is no general consensus on what the chunks of architectural information Software Architecture Documents or Architectural Descriptions should comprise (Greefhorst, Koning, & van Vliet, 2006). In addition, the term *chunking* is not established in SA although the following forms of *chunking* seem to be supported:

a) *Chunking supported by architecture documentation constructs such as architecture framework, view, view packet, and template*. These constructs provide guidance on grouping of architectural information. Architecture frameworks such as Zachman's Framework (Zachman, 1987), 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, Clements, & Kazman, 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, respectively, by providing guidance on what should be documented for them and the organisation of their constituents. Using templates such as these place together pieces of information that are related, by following the standard groupings suggested by the templates.

b) *Chunking supported by searching facilities*. A search using the searching facilities of the documentation environment returns pieces of information that 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 the semantic structure of documents based on words in the documents (de Boer, 2006; de Boer & van Vliet, 2008), 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 (de Boer & van Vliet, 2011; de Graaf, Tang, Liang, & van Vliet, 2012; Jansen et al., 2009; Su, Hirsch, & Hosking, 2009; Tang, Liang, & van Vliet, 2011), architectural elements or knowledge instances retrieved are related because of the pre-defined relations in the underlying models.

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

The work above shows that the onus of identifying chunks has always been on the producers instead of consumers of ADs. Producers apply the architecture documentation constructs, or models of AK, in grouping architectural information. The role of the consumers in this aspect is mostly confined to the choice of the terms they provide to the searching facilities, which return a set of results (or chunk) based on the terms supplied. To the best of our knowledge there has been no previous study on chunking architectural information based on the actual consumption of ADs, to support information finding.

2.3 Definition of a Usage-based Chunk

We define a software architecture document chunk as follows:

A chunk is a collection of related pieces of architectural information in a Software Architecture Document that are needed to carry out an information-seeking task by a group of users.

A Software Architecture Document contains sections that might be or might not be part of the chunk needed for an information-seeking task – thus not all architecture document elements may be part of a chunk and different elements will be parts of different chunks needed for different information-seeking tasks.

The assumption is that these pieces of information are related because a group's usage of the information in a document shows that they are needed by the majority of the group to perform the task. Since the chunk is derived based on usage of information in the architecture document, we term this chunk a '**usage-based chunk**'.

A section of a document can be either paragraph(s) of text, table(s), image(s), hyperlink(s), or combinations of these. For example, a chunk may 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. Both of these sections contain information required by the majority of a group for a particular information-seeking task.

We use 'document section' as the lowest level of granularity for chunk elements. This level of granularity was also used in existing work that studied the relevance of the elements of ADs to perceived stakeholders and their concerns (Koning & van Vliet, 2006).

2.4 Research Questions

In this work we identify as our two research questions (RQs):

RQ1: In the set of cases that we have studied, do usage-based chunks exist?

RQ2: If so, how useful are usage-based chunks in supporting specific information finding tasks?

As mentioned earlier, chunking of architectural information in ADs has always been done by producers of ADs. We wanted to find out if chunking can be based on consumers' usage of information in ADs when they engage with information-seeking tasks. Therefore, our first research question is to find out whether usage-based chunks might actually exist in software architecture documents. If they do, then they may provide an alternative chunking of architectural information in ADs based on consumers' actual usage of these documents.

A usage-based chunk can assist information finding as it groups related pieces of architectural information together and enables them to be retrieved collectively as a unit, rather than spread throughout a document with non-useful material in between. Following on from this, if usage-based chunks do indeed exist, it is important to know how useful they are in supporting information finding tasks. This is the rationale for our RQ2.

As our work is a preliminary study of this concept of 'usage-based chunks' in ADs, we proposed to assess the usefulness of usage-based chunks found for representative information-seeking tasks by benchmarking them against oracle sets constructed by SA professionals for these tasks. In our work, an oracle set for an information-seeking task is the most useful chunk for the task. This is because it was carefully constructed following a rigorous process (Section 3.6). Recall and precision measures of usage-based chunks were calculated. Recall and precision are standard measures used in the evaluation of Information Retrieval (IR) systems (Manning, Raghavan, & Schütze, 2008). In our work, the recall measure shows how complete a chunk is for the specific task and the precision measure shows how precise a chunk is for the task. The two measures are inversely related (Manning et al., 2008) and we define criteria that take this into consideration to determine the usefulness of usage-based chunks.

3 Methodology

This section details the methodology we adopted to provide a preliminary set of answers to the key research questions of whether usage-based chunks exist in software architecture documents and if so, how useful are they for representative information-seeking tasks. We describe the design of the two studies we conducted to answer our research questions, participant selection, choices of ADs and experimental information-seeking tasks, how we identified chunks using chunk-identification factors we developed, definition of oracle sets and benchmarking of chunks.

3.1 Study Design

Table 1 summarises the two studies we conducted. For both studies we recruited participants with SA backgrounds to explore ADs to find answers to representative information-seeking tasks (or questions) and we collected their usage data. The designs of Study 1 (Manual Exploration Study) and Study 2 (Online Exploration Study) are similar. They differ mainly in the collection of data, namely, manually off-line versus online by using KaitoroCap (Su, 2014; Su et al., 2011b). KaitoroCap is an online tool we developed for creating ADs as wiki pages and for collecting usage data. Its main features are exploration path capture, retrieval and analysis; hierarchical tree-view visualisation of paths; path searching; section rating, tagging and commenting; expanding or collapsing of sections, and page model generation to enable dynamic restructuring of documents.

Details of these studies including instruments, full results analysis and the KaitaroCap toolset can be found in the first author's PhD (Su, 2014), available from http://hdl.handle.net/2292/22565

Prior to Studies 1 and 2, we conducted a user evaluation of KaitoroCap where its features were assessed in terms of their usefulness, effectiveness and ease of use (Su, 2014; Su et al., 2011a). Feedback from the user evaluation study was used to improve KaitoroCap. The improved KaitoroCap (version 2) was used in Study 2. The user evaluation study also revealed that some of the participants viewed the 'role' or the 'perspective' undertaken by them to be very critical in driving their decisions as to what information is needed. Examples of roles are as an

evaluator, a developer, and so on. Based on this, the specification of the information-seeking tasks in Studies 1 and 2 was improved to also include the roles to be assumed when performing the tasks.

In our work, usage data refers to data that is related to a user's uses of the information in a document when engaged with certain information-seeking task. This data can be solicited explicitly or implicitly. Usage data which was solicited explicitly include participants' responses on where the needed information (or answer) was found, their highlighting of information in the document they thought relevant to the task, their ratings of AD sections visited in terms of their importance to the respective task and to understanding the SA of the described system, and any tags or comments they provided for the sections. A tag is a keyword given by a user that reflects a section's content. A comment carries a user's more elaborated opinion on a section. It was mandatory in ours studies for participants to provide ratings for sections visited but tags and comments were optional. We term the explicit data *annotation data*. Usage data that was implicitly obtained comprises data generated by the participants' interaction with a document's pages and elements on the pages, such as clicking on a hyperlink to go to another page. We term this *interaction data*.

ADs and information-seeking tasks: In both studies, each participant was given one of two ADs (WCT and ASM – see Section 3.3) and two of three information-seeking tasks (Section 3.4) defined for the given document. The sequence of the two tasks was reversed for each alternate participant to balance-off the influence of the familiarity with the document acquired during the first task, on the second task. This resulted in 6 sets of tasks for each document.

Format of ADs: In Study 1, the participants explored ADs in Microsoft Word or printed format. In Study 2, the participants explored ADs in the form of wiki pages in an Atlassian Confluence Enterprise Wiki (Atlassian, 2013) environment, where, KaitoroCap was installed as a plug-in.

	Study 1 (Manual Exploration)	Study 2 (Online Exploration)
Purpose	Collect usage data offline without KaitoroCap	Collect usage data online using KaitoroCap
Research Method	Quasi-experiment	Same as Study 1
Study Design	Studies 1 and 2 have similar study designs and differ mainly in how data was collected	See explanation for Study 1
AD Used	WCT and ASM	Same as Study 1
AD Format	Microsoft Word or printed documents	Wiki pages
Information-seeking Tasks Defined	3 tasks for each AD	Same as Study 1
Information-seeking Tasks per Participant	One of these sets of tasks of an 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)	Same as Study 1
Usage Data Collection	Manual	Using KaitoroCap Ver. 2
Interaction Data Collected	No	Yes using KaitoroCap Ver. 2
Annotation Data Collected	Tags and comments (not mandatory), ratings on importance of AD's sections, section from which each bullet-point answer was found, highlighted information	Tags and comments (not mandatory), ratings on importance of AD's sections
Other Data collected	Start and stop time of task, answer (in bullet-point form), how each bullet-point answer was found, keywords searched for, suggested reading sequences	Metadata of path, task set and task number, answer, other ratings (own answer, satisfaction with own exploration, expertise level in terms of the role undertaken, % of information found for the task, own path)
Preparation of ADs to collect ratings, tags and comments	Fields inserted into each annotatable section of ADs automatically by a script	Fields inserted into each annotatable section of ADs automatically by KaitoroCap

Participation Duration	75 minutes	75 minutes
Participant Selection Criteria	Willingness and ability to commit the required time and effort; and either (i) at least 2 years of industry experience related to SA; or (ii) taught a SA course, or to have provided training on SA	Willingness and ability to commit the required time and effort; and some SA background (taken or taking course related to SA; research, teaching or training, or industry experience in SA) regardless of years of experience
Participant Recruitment Approach	Convenience and snowball sampling via email invitation, and recruitment advertisement on related Yahoo and FaceBook groups	Same as Study 1, and, recruitment advertisement via local acquaintances to recruit students
Participant Recruited	30 took part, 25 submitted, 23 analysed	38 took part, 32 submitted, 19 analysed
Classification of Participants	Industry, Academic	Industry, Academic, Student (students with no SA industry/teaching experience, but taken or taking course or doing research in SA)
Pre-Questionnaire	One questionnaire combining pre- and post-questionnaires of Study 2, but no questions on educational background, wiki, tool similar to KaitoroCap, navigation paths & patterns, KaitoroCap's features & suggestions to improve	Educational background, education & industry experience in SA, Wiki & AD experience, English proficiency, exposure to similar system
Post-Questionnaire	See row above	Experience in similar tool; Perceptions of KaitoroCap's features and the AD, ways AD supported and hindered understanding of the described SA, navigation characteristics, perception of the usage of textual descriptions and diagrams and their usefulness in supporting understanding, usefulness of own paths, common navigation patterns, suggestions to improve KaitoroCap
Administration of Questionnaire(s)	Offline	Online using SurveyMonkey embedded in KaitoroCap
Administration of Study	Face-to-face, email or online chat clarification.	Same as Study 1 and Help file on KaitoroCap Ver. 2
Identification of Chunks	Using 6 chunk-identification factors.	Using chunk-identification factors that involved ratings - Factors R3, AveR3, AveR3F

Table 1 : Studies Conducted

Collection of Usage Data: In Study 1, we manually collected the annotation data provided in written form. In Study 2, interaction and annotation data were collected using KaitoroCap. The interaction data were displayed as exploration paths in KaitoroCap.

Preparation of ADs: To collect some of the annotation data (namely, ratings, tags and comments), fields were inserted into each annotatable section of the ADs automatically, either by KaitoroCap in Study 2, or by a script in Study 1. The WCT document contained 47 annotatable sections, with 6 containing diagrams. One section contained two closely-related diagrams. The ASM document contained 62 annotatable sections, with 7 containing diagrams.

Participant Selection Criteria: The first participant selection criteria for both studies was willingness and ability to commit the time and effort required to participate in the studies. The second participant selection criteria for Study 1 was: either having at least 2 years of industry experience related to SA; or taught a SA course or to have provided training on SA. We had difficulty in recruiting participants that fulfilled such criteria. Therefore, we relaxed this participant selection criteria for Study 2 to: having an SA background (such as taken or taking course related to SA; research, teaching or training, or industry experience in SA) regardless of years of experience. To simplify our writings in later parts of this paper, we termed participants with at least two years of industry experience in SA, or with teaching or training experience related to SA as 'experts', and those with an SA background but who did not fulfil the criteria of 'expert' as 'novices'. We could not recruit an adequate individual

numbers of 'novices' or 'experts' and therefore included both in Study 2. Despite that, the results from Study 2 are consistent with the results from Study 1 (Section 5.2).

Participant Recruitment Approach: The requirements of the specific background in SA and the considerable amount of time (namely, 75 minutes) and effort to take part in the studies discouraged the use of random sampling to recruit the studies' participants. The reason is the targeted groups often do not respond to invitations from unfamiliar sources. Therefore, 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 (Kitchenham & Pfleeger, 2002). Snowball sampling refers to asking participants of the study to recommend other potential respondents (Kitchenham & Pfleeger, 2002). For Study 1, we invited participants via email invitation, and recruitment advertisement on related Yahoo and FaceBook groups. We did the same for Study 2, and, we also engaged local acquaintances to disseminate recruitment advertisement to recruit students.

Research Method: To collect substantial AD usage data, we needed to define a reasonable scope for the exploratory processes involved in the consumers' AD usage. This involved selecting the specific ADs and predefining the information-seeking tasks for which usage data would be collected. These requirements made the *experiment* research method suitable for the studies. The experiment research method deliberately separates the studied phenomenon from its context (Sjoberg, Dyba, & Jorgensen, 2007) to allow the control (Wohlin et al., 2012) that we needed. However, in practice, we conducted Studies 1 and 2 as quasi-experiments. A quasi-experiment is a variant of the true experiment research method (Easterbrook, Singer, Storey, & Damian, 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 our best to assign the study participants (i.e. subjects) randomly to the two ADs used and the information-seeking tasks (i.e. the treatments). However, we could not claim the allocations were fully randomised. It was difficult to recruit highly-specialised 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 AD first. Subsequent participants that we recruited were allocated to use the second AD.

Participants Recruited: For Study 1, 30 participants took part but only 25 submitted responses out of which 23 were analysed. For Study 2, 38 participants took part, 32 submitted responses out of which 19 were analysed.

Participants Classification: Studies have shown considerable differences between industry and academics in their perception of SA and reusable assets (Bosch, 1999). Consequently, we classified the participants into either industry (I) or academic (A) participants based on whether their SA experiences were from the industry, or from academic teaching or training. For those who had experience from both, the length of experience in industry and teaching decided their classifications. For Study 2 but not Study 1, we also invited students (mostly PhD) who had taken or were taking SA course or had research experience in SA, but had no industry or teaching experience in SA. These students were mainly recruited from University of Auckland (New Zealand). Auckland University of Technology (New Zealand), University of Malaya (Malaysia), University Putra Malaysia (Malaysia) and Swinburne University of Technology (Australia). We classified these participants as *student* (S) participants. Section 3.2 provides further details on the recruitment of participants, with numbers for each classification of participants.

Questionnaire: In both studies, participants were asked to complete questionnaires with regard to their background, experience in SA and AD, English proficiency, exposure to similar system, perceptions of AD, ways AD supported and hindered understanding of the described SA, navigation characteristics, perception of the usage of textual descriptions and diagrams and their usefulness in supporting understanding. For Study 2, the questionnaire was divided into pre- and post-questionnaire. The post-questionnaire included questions related to KaitoroCap.

Administration of Questionnaire: The questionnaire for Study 1 was administered offline. The questionnaire for Study 2 was administered online using SurveyMonkey (SurveyMonkey, 2015) embedded in KaitoroCap.

Administration of Study: For both studies, explanation and clarification on the tasks a participant needed to complete were done either face-to-face, through email or online chatting. For Study 2, a Help file on how to use KaitoroCap was accessible by the participants from KaitoroCap.

Chunk Identification: We used the chunk-identification factors we developed (Section 3.5) to identify chunks from the annotation data collected in Studies 1 and 2. For Study 1, all 6 factors were used. For Study 2, factors that involved ratings only (Factors R3, AveR3, and AveR3F) were used.

Ethics approval to conduct the studies was obtained from the University of Auckland Human Participants Ethics Committee (reference numbers 2010/528 and 6943). Informed consent was obtained from all participants involved in the studies.

3.2 **Participant Selection**

Table 2 shows statistics on the recruitment of participants for Studies 1 and 2. We invited 80 potential industry and academic participants for Studies 1 and 2, and 24 students for Study 2. 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 42 were included in our analysis to find chunks.

For Study 1, 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 excluded 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 in SA. Although we did not specify the minimum number of years of SA teaching experience, all the academic participants in Study 1 had at least 2 years of teaching experience. For Study 2, 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. Responses which were incomplete (such as not completing the pre- or post-questionnaire) were excluded from our analysis.

	Study 1 (S1) and Study 2 (S2)							
	Invited	Responded	Excluded	Took Part (S1)	Took Part (S2)	Dropped Out	Submitted	Analysed
Industry	42	35	1	16	18	5 (S1) 6 (S2)	25 (S1) 32 (S2)	23 (S1) 19 (S2)
Academic	38	26	1	14	11			
Student	24	11	2	0	9			
Total	104	72	4	30	38	11	57	42

Table 2 : Recruitment of Participants (Studies 1 and 2)

3.3 **Software Architecture Documents Used**

We considered the following characteristics of a document when deciding whether to use it in our studies: length, complexity, quality, mixture of textual and graphical representations, and availability (Su, Tempero, Hosking, & Grundy, 2012). These should reasonably suit the exploratory information-seeking tasks we specified for our studies. A suitable document length is critical. If the document is too short, the participants may just read the whole document without exercising selective exploration of the document. If it is too long, the participants may not be able to find the needed information within the estimated time frame. The chosen documents were of reasonable technical complexity and details. We chose documents that described systems that are not too specialised, to cater for different background of the participants. The documents also needed to have reasonable

quality in terms of language, organisation and legibility as well as a mixture of text and graphics. The chosen documents were available on the Internet and we obtained permission to use them in our work.

We chose existing ADs describing real systems in use. The first AD was a 24-page document defining the architecture of a Web Curator Tool (WCT) (National Library of New Zealand, 2006). The second AD has 21 pages and describes the SA of Aperi Storage Manager (ASM), an open source storage management platform (Slupesky & Singleton, 2006).

3.4 **Experimental Tasks**

Three experimental information-seeking tasks were specified for each AD (Table 3) and they were similar for the two ADs. Each task was phrased in terms of a scenario together with the role to be assumed. The role is to help participants to view the respective task with a similar mind set. The roles that we chose were related to the main stakeholders of ADs or AK (Kruchten, Lago, & van Vliet, 2006).

The first task was to describe in general terms the SA of the system described by the given AD by assuming the role of a Software Architect new to the software project of the system (Su et al., 2012). The chunks found from here allowed us to study the common 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 a system and to identify which parts of the system would be affected by the change (Su et al., 2012). The role to be assumed was a developer. The chunks from here allowed us to study the common architectural information needed for specific tasks (such as making a change to the software system). 'Parts' can refer to any aspect of the system, such as the system itself, subsystems and configuration. 'Affected' could be code, quality or other changes, or no change at all.

The third task was to discover how the system was designed at the architectural level to achieve a certain quality attribute, by assuming the role of a system maintainer (Su et al., 2012). The chunks found here would give some insights into the common information needed to address cross-cutting concerns which span the entire system. The third task for WCT AD focused on security. The third task for ASM AD focused on modifiability. We were not able to specify the same issue for the two tasks due to the different contents of the two ADs.

Information-seeking tasks	Role to be assumed
AD1: Web Curator Tool (WCT)	
Task 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 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 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
AD2: Aperi Storage Manager (ASM)	
Task 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 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 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 3 : The Software Architecture Documents and the Information-Seeking Tasks

The first and the third tasks that we had chosen are typical information-seeking tasks in a software development project. These kinds of tasks are ones that a number of members involved in the same software development project are likely to be interested in, especially in collaborative software development environment. Making usage-based chunks for tasks such as these available for a software project team enables team members to reuse these chunks as alternatives to searching information using keywords or browsing the whole document when engaged with similar tasks. Using usage-based chunks leverages previous users' usage of the information in the document and can be useful especially for novice members of a team.

3.5 Identifying Chunks

We developed 6 plausible factors, described in Table 4 **Error! Reference source not found.**and Table 9, to identify chunks from the following annotation data collected in our studies:

1) Participants' responses on which section(s) of the AD they found that contained the needed information (or answer) for the respective information-seeking task. Our rationale is this is the most straightforward way of finding out sections needed by them for the task. We wanted to know if this can be used for chunking information in ADs.

2) Participants' highlighting of information in the AD they thought relevant to the respective information-seeking task. Highlighting or underlining content is one common way readers annotate documents (Asai & Yamana, 2014). It is a common activity among readers (Liu, 2005). Educational psychology literature on highlighting or underlining suggests that the act of underlining increases recall of information (Chi, Gumbrecht, & Hong, 2007). If this annotation data can be used for chunk identification, it creates an additional use for highlighting.

3) Participants' ratings of AD's sections visited in terms of their importance to the respective information-seeking task. Users' ratings have been used for collaborative filtering of information (Goldberg, Nichols, Oki, & Terry, 1992; Schafer, Frankowski, Herlocker, & Sen, 2007; Shardanand & Maes, 1995). We wanted to explore the possibility of using users' ratings for chunking information in ADs.

To determine the 'commonality' in the annotation data, we identified the preference of the majority of a group of participants for each section of an AD when the participants were engaged with the same task. The preference of the majority of a group for a section is used to decide whether to include the section in the chunk found using a chunk-identification factor. We used two aggregation mechanisms to arrive at the preference of the majority of a group, namely, frequency count (Section 3.5.1) and average (Section 3.5.2). Frequency count as used in simple majority voting (Bachrach, Graepel, Kasneci, Kosinski, & Gael, 2012), and averaging (Baker & Olaleye, 2013), have been used in aggregating opinions. We wanted to explore their uses in usage-based chunk identification. **Error! Reference source not found.**Table 4 and Table 9 show the section inclusion criteria defined to decide whether to include a section of the document in the chunk found using a particular factor. The rest of this section details the factors with examples on how they are used to find chunks.

3.5.1 Factors using Frequency Count

For Factors A, H, R3 and A|H|R3 (Table 4**Error! Reference source not found.**), we used frequency count to aggregate the participants' responses into two categories of indications, namely, 'needed' or 'not needed' for the task. We took a minimum stance on what we meant by majority, which is as long as more than half of the total number of 'involved' participants. If the 'needed' category receives at least a minimum majority (m) count or

simple majority, then the section is considered 'needed' by the group for the task and the section would be included in the respective chunk. Depending on which factor, the 'involved' participants can refer to all the participants in a group or only those in the group who provided the responses.

Factor	Section Inclusion Criteria (A section of the document is included in the chunk found using this factor if more than half of the total number of participants in the group…)
A (Answer)	stated that the section is where the answer or part of the answer was found.
H (Highlighted Information)	highlighted certain information in it.
A|H|R3	provided answer from it OR highlighted certain information in it OR rated 3 and above in terms of the importance of the section to answering the task.
Factor	**Section Inclusion Criteria (A section of the document is included in the chunk found using this factor if more than half of the total number of participants in the group who rated the section, …)**
R3 (Rating >=3)	rated it 3 and above in terms of the importance of the section to answering the task.

Table 4 : Chunk-Identification Factors Using Frequency Count

Factor A (Answer): Each participant was asked to provide an answer to the task in bullet-point form, as comprehensively as possible and to state from which section of the document each bullet-point contribution to the answer was found. A section stated by a participant is interpreted as 'needed' and a section not stated by the participant is interpreted as 'not needed' by the participant for the task. Consequently, minimum majority in Factor A refers to more than half of the total number of participants in the group and not the total number of those in the group who stated the section as where one or more of the bullet-point answers were found.

Factor A did not take into account the correctness of the answers given. Our reasons are: 1) In reality some individuals may misunderstand a task and give wrong answers. Including these individuals' responses in the identification of chunks made our findings more realistic. 2) It is difficult in actual practice to assess the correctness of tasks performed and use that to decide whether to include the respective users' usage data in chunks identification. 3) The correctness of an answer does not necessary reflect the relevance of the respective usage data for finding chunks. A low answer's score does not necessary mean that the corresponding usage data is not useful, and vice versa. For example, the usage data may suggest all the sections that are 'needed' for the task, and yet the user's answer may score low. A low score could be due to the user's misunderstanding of what should be given in an answer or inability in expressing answer.

Example: Refer to Table 5. A value of '1' under a participant column means that the participant stated that one or more of his or her bullet-point answers for the respective information-seeking task was or were found in the section. An empty cell under a participant column means that the participant did not provide such indication for the section. For example, participant E2 but not participants E1, E3 and E7 provided such indication for section with ID 1. Minimum majority for this example is 3. Three of the participants stated their answers were found in section with ID 188. This fulfils Factor A's section inclusion criteria and this section is included in the chunk found using Factor A. Fewer than 3 of the participants stated that their answers were found in sections with ID 1, 46, 54 and 64. This does not fulfil Factor A's section inclusion criteria and these sections are excluded from the chunk found using Factor A.

Section ID	Participant				Total who stated the section	Minimum Majority Count	Include in A chunk
	E1	E2	E3	E7			
1		1			1	3	No

46	1		1		2	3	No
54					0	3	No
64	1	1			2	3	No
188		1	1	1	3	3	Yes

Table 5 : Example of Chunk Identification Using Factor A

Factor H (Highlighted information): The participants were asked to highlight information in the document which is relevant to the assigned task, when they looked for the answer for the 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 the section for the task. An absence of highlighted information in a section is interpreted as the section being 'not needed' by a participant for the task. Consequently, minimum majority in Factor H refers to more than half of the total number of participants in the group and not the number of those who highlighted information in the section.

Example: Refer to Table 6. A value of '1' under a participant column means that the participant highlighted information in the section when looking for answer for the respective information-seeking task. An empty cell under a participant column means that the participant did not highlight information in the section. For example, participants E2 and E3 but not participants E1 and E7 provided such indication for section with ID 1. Minimum majority for this example is 3. Three of the participants highlighted information in section with ID 188. This fulfils Factor H's section inclusion criteria and this section is included in the chunk found using Factor H. Fewer than 3 of the participants highlighted information in sections with ID 1, 46, 54 and 64. This does not fulfil Factor H's section inclusion criteria and these sections are excluded from the chunk found using Factor H.

Section ID	Participants				Total who highlighted section	Minimum Majority Count	Include in H chunk
	E1	E2	E3	E7			
1		1	1		2	3	No
46	1		1		2	3	No
54		1			1	3	No
64	1	1			2	3	No
188	1		1	1	3	3	Yes

Table 6 : Example of Chunk Identification Using Factor H

Factor R3 (Rated 3 and above): 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. The ratings are captured on Likert-scale ranging from 1 (the lowest) to 5 (the highest). In addition, there are two other options: 'Not Important' and 'Not Sure'. 'Not Important' option is assigned a zero value. 'Not Sure' option is not assigned any value. If it is chosen, it is treated as if the participant did not rate the section.

No assumption is made about the value that should be assigned to an unrated section. Therefore, only the explicit rating values given by those who rated the sections are used for identification of chunks based on ratings. Following that, minimum majority in Factor R3 refers to more than half of the number of participants in the group who provided ratings for the section and not the total number of participants in the group.

We are interested in chunks that include sections that are of high importance for a respective 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. A rating value below 3 is interpreted as of low importance (LI) and therefore 'not needed'. As a result, chunks found using Factor R3 include only those sections rated 3 and above, by more than half of the participants who rated the sections. The six values of importance (i.e. 0 for the 'Not Important' option, and values 1 to 5) were divided equally into HI category and LI category, using 3 as the dividing value.

Chunks found using Factor R3 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, by using the explicit ratings the discovery of any preference is based on clear indications from the participants on the importance of the section to the task and not affected by any speculation that we could have made if we assigned value to a section not rated by a participant.

Example: Refer to Table 7. A value under a participant column is the participant's rating of the section in terms of its importance to the respective information-seeking task and an empty cell means that the participant did not provide rating for the section. For example, participants E1, E2 and E3 gave section with ID 1 a rating of 1, 3 and 5, respectively, and E7 did not rate it. For Factor R3, minimum majority count for a section is based on the number of participants that provided ratings for the section. For example, 3 participants rated section with ID 1 and the minimum majority count for this section is 2. Two of the 3 participants who rated this section, gave it ratings of 3 and above. This fulfils Factor R3's section inclusion criteria and this section is included in the chunk found using this factor. All 4 participants rated the section with ID 46 and the minimum majority count for this section is 3. Only two of the 4 participants rated this section 3 and above. This does not fulfil Factor R3's section inclusion criteria and this section is excluded from the chunk found using this factor.

Section ID	Participants				Rated >=3 (HI)	Rated < 3 (LI)	Did not rate	Total who rated	Minimum Majority Count	Include in R3 chunk
	E1	E2	E3	E7						
1	1	*3*	*5*		2	1	1	3	2	Yes
46	2	*3*	*3*	1	2	2	0	4	3	No
54	*4*			*4*	2	0	2	2	2	Yes
64		*4*	2	*5*	2	1	1	3	2	Yes
188	2	*4*	*5*	*5*	3	1	0	4	3	Yes

Table 7 : Example of Chunk Identification Using Factor R3

Factor A|H|R3: A participant could have provided different forms of indications of his or her needs of different sections of an AD. For example, he or she might have provided indication as required in Factor A for Section 1, but provided indication as required in Factor H for Section 2, and provided indications as required in both Factor H and Factor R3 for Section 3, and so on. To cater for this kind of situation, we proposed Factor A|H|R3. This factor takes into consideration the 3 different forms of indications in a 'OR' ('|') relationship. In other words, if a participant specified that a section was where one of his or her bullet-point answers was found (A), or the participant highlighted certain content in the section (H), or the participant gave a rating of 3 and above for the section (R3), the section is interpreted as 'needed' by the participant for the assigned task. The minimum majority in Factor A|H|R3 refers to more than half of the total number of participants in the group. Even though our stand was to exclude those who did not rate a section, if the section was indeed needed by a participant who did not rate it, he or she could have provided either one or both of the other two forms of indications.

We did not propose combinations such as A|H, A|R3, H|R3, because they do not take all the three types of indications into consideration simultaneously. We also did not propose to combine A, H, R3, with AveR3 or

AveR3F (Section 3.5.2) because the two groups of factors made use of different aggregation mechanism to arrive at the preference of the majority of a group, namely, frequency count versus average.

Example: Refer to Table 8. A value of '1' under a participant column means that the participant provided the indication as required in either Factor A, H or R3, or any combination of them. An empty cell under a participant column means that the participant did not provide any of the indication as required in Factor A, H and R3 for the section. For example, with reference to Table 5, Table 6 and Table 7, for section with ID 64, participant E1 provided two types of indications (as in Factors A and H), E2 provided all three types of indications, E7 provided one type of indication (as in Factor R3). All these participants are given a value of '1' for section with ID 64 in Table 8. E3 provided one type of indication, namely, a rating of below 3 for the section and this is not considered as providing the indication required in Factor R3. Therefore, the cell remains empty under E3 for section with ID 64 in Table 8. The minimum majority for this example is 3. Three participants provided indications as required in either Factor A, H or R3, or any combination of them, for section with ID 64. This fulfils Factor A|H|R3's section inclusion criteria and this section is included in the chunk found using this factor. Fewer than 3 of the participants provided indications as required in either Factor A, H or R3, or any combination of them, for the section with ID 1. This does not fulfil Factor A|H|R3's section inclusion criteria and these sections are excluded from the chunk found using this factor.

Section ID	Participants				Total who provided indication as required in Factor A or H or R3	Minimum Majority Count	Include in A|H|R3 chunk
	E1	E2	E3	E7			
1		1	1		2	3	No
46	1	1	1		3	3	Yes
54	1	1		1	3	3	Yes
64	1	1		1	3	3	Yes
188	1	1	1	1	4	3	Yes

Table 8 : Example of Chunk Identification Using Factor A|H|R3

3.5.2 Factors using Average

For Factors AveR3 and AveR3F (Table 9), we used the average of the ratings of the importance of a section to the task, as the preference of the majority of a group for the section. A section with an average rating of 3 and above is taken as 'needed' and included in a chunk found based on average rating. A section with an average below 3 is taken as 'not needed' and excluded from a chunk found based on average rating. The reason for choosing 3 as the dividing value is the same as for Factor R3.

Factor AveR3 (Average Rating of 3 and above): An average rating is calculated by dividing the total sum of the rating values with the number of participants in the group who rated the section. In other words, participants who did not rate the section are not included in the divisor when calculating the average rating of a section. This is done since we made no assumption about the value that should be assigned to a section not rated by a participant.

Factor	Section Inclusion Criteria (A section of the document is included in the chunk found using this factor...)
AveR3 (Average Rating >=3)	if the group's average rating for the section is 3 and above in terms of its importance to answering the task.
AveR3F (Average Rating >=3)	if the group's average rating for the section is 3 and above in terms of its importance to answering the task and it is rated by more than half of the total number of participants in the group.

Table 9 : Chunk-Identification Factors Using Average

Factor AveR3F (Average Rating of 3 and above, excluding sections not rated by majority): Factor AveR3's chunks 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, its average rating would be 5 and it would be included in the chunk. Factor AveR3F eliminates the bias by excluding sections not rated by more than half of the participants from Factor AveR3's chunks.

Example: Refer to Table 10. The rating values in this table is the same as in Table 7 (Factor R3). Participants E1, E2 and E3 rated section with ID 1 and the average rating for this section is 3.0. This meets Factor AveR3's section inclusion criteria and this section is included in the chunk found using this factor. The section with ID 46 was rated by all 4 participants and its average is 2.3. This does not meet Factor AveR3's section inclusion criteria and this section is excluded from the chunk found using this factor. Looking at the data in Table 10, the chunk found using Factor AveR3 includes sections with ID 1, 54, 64 and 188 but not 46. When Factor AveR3's chunk is filtered to include only those sections that were rated by more than half of the total number of participants in the group (Factor AveR3F), the chunk now includes sections with ID 1, 64 and 188 but not 54.

Section ID	Participants				Total who rated	Average	Include in AveR3 chunk	Include in AveR3F chunk
	E1	E2	E3	E7				
1	1	3	5		3	3.0	Yes	Yes
46	2	3	3	1	4	2.3	No	No
54	4			4	2	4.0	Yes	No
64		4	2	5	3	3.7	Yes	Yes
188	2	4	5	5	4	4.0	Yes	Yes

Table 10 : Example of Chunk Identification Using Factors AveR3 and AveR3F

3.5.3 Other Factors Considered

We have also considered several other possible factors but discarded them as we found that they were not suitable for chunk identification (Su, 2014). For example, consider Factor R1. Factor R1 is similar to Factor R3. The difference is that Factor R1 involves sections rated 1 and above. We dropped Factor R1 because it allows many sections with low ratings to be included in a chunk. Feedback from our oracle set definition experts indicated such sections are most likely not be needed for the respective task. Further details of factors discarded can be found in (Su, 2014).

3.6 **Oracle Set Definition**

To evaluate the chunks and the chunk-identification factors, we adopted the approach used in the evaluation of Information Retrieval systems: recall and precision measures based on a set of relevance judgments for each query-document pair (Manning et al., 2008). We extended the idea of the set of relevance judgments to create an 'oracle set'. Our set of relevance judgments contains the judgments of not merely the relevance of each section of a document to the information-seeking task (i.e. the query), but also whether the sections are mandatorily, optionally or not required. In addition, in making the relevance judgments, consideration was taken on whether together these sections provide all the information needed for the task. We defined the oracle set for an information-seeking task **as a set constructed using a rigorous process that contains all the sections of a document which are compulsory for the task, and where together these sections provide all the information needed for the task.**

The oracle set for each task was built following a rigorous process. Two SA professionals served as judges in constructing the oracle set for each task. The main criteria in the selection of the judges 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 of constructing the oracle set for a task.

Each judge constructed a separate set of relevance judgments for a task. The sets of relevance judgments from both judges were later reconciled into one oracle set for the task. Prior to building the set of relevance judgments, a judge went through a preparation session. This session served to make sure the judge understood that the ultimate purpose of constructing the set of relevance judgements was to build the oracle set, what needed to be done and the proper way of filling up the relevance judgment template we defined to ensure relevance judgments were captured consistently. The judge was also given the AD and the tasks and was asked to seek clarification where needed.

During the construction of a relevance judgments set, the judge could seek further clarification from the researcher if needed. The completed template was later reviewed by the researcher where clarification from the judge was sought for any ambiguous responses. This was followed by a reconciliation session. The purpose of the reconciliation session was for the two judges to discuss and reach a consensus on any discrepancy in the two sets of relevance judgments in order to reconcile them into one final oracle set for the task. Only sections agreed by both of them as mandatory for the task made it into the oracle set. We did not measure inter-rater reliability since there was a reconciliation session in the process of constructing the oracle set for a task.

We did not track the time taken to build the oracle set for a task because we did not impose any time constraint on a judge in producing his or her relevance judgments set and on the reconciliation session to produce the oracle set for the task. This is because the purpose was to build 'the oracle' which was used as benchmark for calculating the precision and recall measures of chunks found for the respective task.

The rigorous construction process above gave us confidence in the accuracy of the oracle sets.

One of the researchers and another professional built the oracle sets for the 3 tasks for WCT AD. The same researcher and a different professional built the oracle sets for the 3 tasks for ASM AD. All the judges involved in building the oracle sets were academic professionals, with an average of 9.67 years of experience in teaching SA (minimum 4 and maximum 20 years). We could not find industry professionals who could devote the time and effort required to build the oracle sets. The researcher who defined the tasks was one of the judges who constructed the relevance judgments sets for the tasks. This follows the use of the person who formulated the query to build the relevance judgments set in Information Retrieval (Wallis & Thom, 1996). The oracle set for each task can be found in the table that shows the composition of chunks found for the respective task. For example, the oracle set for WCT Task 1 consists of all those sections with their sections' IDs marked with * and their rows in yellow in Table 12 and Table 13. The oracle sets for other tasks can be found in the tables in the supplementary online material of this paper.

3.7 **Benchmarking of Chunks**

We evaluated the chunks against the respective oracle set by using their recall and precision measures. These standard measures (Manning et al., 2008) are defined as:

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

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

In many situations, high recalls rather than high precisions are needed (Wallis & Thom, 1996). For example, when finding materials to lodge a new patent or when searching for precedence cases in legal work. The information-seeking tasks in our work also require high recalls (i.e. finding of a fairly good amount of all the relevant information, or 'needed' information in our case). However, an acceptable chunk for each task should at the same time not have precision which is too low (i.e. containing too many 'not needed' sections). A chunk with higher recall and precision measures is considered relatively better for the particular task, than a chunk with lower recall and precision measures. Nevertheless, recall and precision measures are inversely related or they trade-off against each other (Manning et al., 2008), where one increases at the decline of the other. In our case, a chunk could have a higher recall but a lower precision, and vice versa.

To assess how 'useful' the chunks are for the particular task, we defined criteria (Table 11) which take into account the trade-off between recall and precision measures. We used 4 levels of recall and precision and their boundary values were chosen arbitrarily. We emphasised the recall of a chunk but at the same time wanted it to have a certain level of precision. The usefulness measure assigned to each combination of the recall and precision levels was based on our intuition of how 'useful' a chunk is for the respective task, as a result of the trade-off nature between recall and precision. The five levels of usefulness measures are 'Very Good', 'Good', 'Satisfactory', 'Poor' and 'Very Poor'. We think the criteria are reasonable since our purpose was to perform a consistent assessment of the usefulness of chunks found using different chunk-identification factors.

For example, a chunk with recall measure of 0.75 and above (Level 4 of recall measure) contains many (i.e. at least 75%) of the sections compulsory for the respective task, with reference to the respective oracle set. If at the same time more than or equal to 75% of its sections are compulsory for the task (Level 4 of precision measure), this chunk is regarded as 'Very Good' for the task comparing to reading the whole document. With the high recall of 0.75 and above, the chunk is regarded as 'Good' for the 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 'Satisfactory' 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).

A reduction in usefulness for each given recall level is then taken with each reduction in precision level until finally level 1 precision and level 1 recall lead to a "Very Poor" usefulness rating.

		Recall Measure (R)			
		Level 4 (0.75 <= R <= 1.00)	Level 3 (0.50 <= R < 0.75)	Level 2 (0.25 <= R < 0.50)	Level 1 (0.00 <= R < 0.25)
Precision (P)	Level 4 (0.75 <= P <= 1.00)	Very Good	Good	Satisfactory	Poor
	Level 3 (0.50 <= P < 0.75)	Good	Satisfactory	Poor	Very Poor

Level 2 (0.25 <= P < 0.50)	Satisfactory	Poor	Very Poor	Very Poor
Level 1 (0.00 <= P < 0.25)	Poor	Very Poor	Very Poor	Very Poor

Table 11 : Criteria to Assess the Usefulness of Chunks

4 Results and Discussion

4.1 Identification and Benchmarking of Chunks

We used the 6 chunk-identification factors to identify chunks for the information-seeking tasks in Study 1. This was done for 3 groups of participants: the industry practitioner (I) group, the academic (A) group, and the combined (C) group of the previous two groups. We used Factors R3, AveR3 and AveR3F that involved ratings data only to identify chunks in Study 2 because we did not solicit the data required by the other three factors due to the use of KaitoroCap. For Study 2, chunks were identified for 4 groups of participants: I and A groups, the student (S) group where applicable, and the combined (C) group of the previous groups.

Table 12 and Table 13 show the chunks discovered for WCT Task 1 in Study 1 and the compositions of these chunks. 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 I Group, the cell corresponding to Section "4.1 Overview" is green and marked with an 'X'. This is interpreted as Section "4.1 Overview" is included in the chunk found for I Group using Factor A. There is only one section in this chunk and its size is 1.

The sections in the oracle set for this task are denoted by yellow rows and their section IDs are marked with asterisks. The tables also show the attributes of a chunk (namely, numbers of oracle set's sections matched and not matched, number of false sections, recall, precision and usefulness measures) related to the benchmarking of the chunk against the oracle set. A *false section* of a chunk is a section of the AD which is included in the chunk but not in the oracle set.

WCT (Task 1, Manual Exploration)			I group (4 participants)						A group (4 participants)					
No.	Section	Factor/ ID	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*				X								
3	2. Architectural Goals and Constraints	54			X	X	X							
4	2.1 Archit. Significant Design Decisions	64			X	X	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	
8	2.1.4 User Interface	88*			X		X				X		X	
9	2.1.5 Resource Use	94*			X		X				X		X	
10	2.1.6 Other Non-Functional Requirements	100												
11	2.2 Archit. Significant Open Source Products	108*							X			X		
12	3. Use-Case View	354*									X	X	X	
13	3.1 Actors	360*			X	X	X	X			X		X	
14	3.2 Use Cases	366*			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
18	Logical High Level Solution Overview Diagram	194*			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 Archit. Significant Design Packages	292							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 WebHarvesterSys	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 Strategy	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
34	Process View Description	386*									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	
38	Logical Deployment Description	412*			X		X		X		X	X	X	
39	Deployment Diagram	418*							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
	Oracle Sections Matched		1	1	**11**	5	11	3	7	0	**17**	11	**17**	2
	Oracle Sections Not Matched		19	19	**9**	15	9	17	13	20	**3**	9	**3**	18
	False Sections in Chunk		0	0	**4**	2	6	2	1	0	**5**	2	**5**	0
	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
	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
	Usefulness Measure		P	P	**S**	P	S	VP	S	/	**VG**	GD	**VG**	P

I Group's chunk found by Factor A comprises Section "4.1 Overview" (green cell with an "X').

I Group's best chunk (attributes in bold & blue cell) is found by Factor R3.

Chunk found by Factor R3 for A Group comprises 22 sections.

Usefulness of Chunk: VG-Very Good; GD-Good; S-Satisfactory; P-Poor; VP-Very Poor; Oracle set– Sections' IDs with *

Table 12 : The I and A Groups' Chunks for WCT Task 1 (Study 1)

WCT (Task 1, Manual Exploration)			C group (8 participants)					
No.	Section	Factor / ID	A	H	R3	A|H|R3	AveR3	AveR3F
1	Table of Contents	1			X	X	X	X
2	1. Introduction	46*						
3	2. Architectural Goals and Constraints	54			X		X	
4	2.1 Archit. Significant Design Decisions	64				X	X	
5	2.1.1 Modularity/Plugability	70*			X		X	
6	2.1.2 Supportability	76*			X		X	
7	2.1.3 Security	82*			X		X	
8	2.1.4 User Interface	88*			X		X	
9	2.1.5 Resource Use	94*			X		X	
10	2.1.6 Other Non-Functional Requirements	100						
11	2.2 Archit. Significant Open Source Products	108*						
12	3. Use-Case View	354*			X	X	X	
13	3.1 Actors	360*			X	X	X	
14	3.2 Use Cases	366*			X		X	
15	3.3 Use-Case Realizations	372						
16	4. Logical View	176						
17	4.1 Overview	188*	X		X	X	X	X
18	Logical High Level Solution Overview Diagram	194*			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 Archit. Significant Design Packages	292				X	X	
24	4.4.1 UC4 - Quality Review	298						
25	4.4.2 UC5 - Submit to Archive	304						
26	4.4.3 UC8-Monitor&Manage WebHarvesterSys	310						
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						
33	5. Process View	380*			X		X	X
34	Process View Description	386*			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	
38	Logical Deployment Description	412*			X	X	X	
39	Deployment Diagram	418*			X	X	X	
40	Alternative Deployment Diagram	424*						
41	7. Data View	432						
0	8. Size and Performance	0						
42	8.1 Performance Requirements	440						
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						
	Size of Chunk		1	0	**17**	10	19	4
	Oracle Sections Matched		1	0	**15**	7	15	3
	Oracle Sections Not Matched		19	20	**5**	13	5	17
	False Sections in Chunk		0	0	**2**	3	4	1
	Recall		0.05	0.00	**0.75**	0.35	0.75	0.15
	Precision		1.00	0.00	**0.88**	0.70	0.79	0.75
	Usefulness Measure		P	/	**VG**	P	VG	P

Usefulness Measure of Chunk: VG - Very Good; GD - Good; S - Satisfactory; P - Poor; VP - Very Poor

Note: Yellow rows are the sections in the oracle set for this task. Their sections' IDs are marked with *.

Table 13 : The C Group's Chunks for WCT Task 1 (Study 1)

For each group of participants, the compositions of the chunks that are at the best level of usefulness measure are compared to decide on the *best chunk* for the group. This is done by assessing the criticality of the inclusion and the omission of the different sections in the chunks, for the task. The factor that produces the best chunk for a group is the *best chunk-identification factor* for the group, with regard to the respective task. In Table 12 and Table 13, the best chunk for a group of participants has its attributes highlighted in blue and formatted in bold.

The tables that show the chunks found for the other five information-seeking tasks in Studies 1 and 2, and the compositions of these chunks can be found in the supplementary online material. The detailed description and interpretation of these chunks are in (Su, 2014).

4.2 Research Question Answers

RQ1: In the set of cases that we have studied, do usage-based chunks exist?

We used chunk-identification factors which discover chunks based on usage data of ADs to find chunks. Table 14 shows that chunks were found when these factors were used, except for those cases denoted by '/' in the table. I, A, S and C refer to industry, academic, student, and combined groups, respectively. For example, no chunk was found for A and C Groups when Factor H was used to identify chunk for WCT Task 1 in Study 1.

For each task, the number of cases where no chunk was found is low. For Study 1, the most is 5 of 18 cases or 27.78% for ASM Task 3. For Study 2, the most is 2 of 12 cases or 16.67% for ASM Task 3. Regardless of that, **chunks were found for each group of participants involved in the three information-seeking tasks of the two ADs, in both studies. So the answer to RQ1 is 'yes'.**

RQ2: If so, how useful are usage-based chunks in supporting specific information finding tasks?

Table 15 shows the usefulness of the chunks found in Studies 1 and 2, and the best chunk for each group of participant for each task. The usefulness measures of these best chunks are presented in 'italics' and shaded with a 'light blue' background. If the table shows more than one best chunk for a group undertaking a particular task, this means these chunks are identical but produced by different factors.

To answer RQ2, we looked at the overall usefulness of chunks produced by each factor. This is because different factors might produce chunks of different level of usefulness. Table 16 shows the number of chunks at each usefulness level produced by each factor. It also shows the percentage of chunks with usefulness of 'Satisfactory' and above which were produced by each factor. Two-third of the chunks produced by Factors R3 and AveR3 in Study 1 have usefulness of 'Satisfactory' and above. Although almost two-third (64.7%) of chunks produced by Factor A|H|R3 also have usefulness of 'Satisfactory' and above, this factor produced one 'no chunk' case. Factors A, H and AveR3F have more tendency to produce below 'Satisfactory' chunks or no chunk. For Study 2, about one-third of the chunks produced by Factors R3 and AveR3 are 'Satisfactory'. About one-fifth of the chunks produced by Factor AveR3F are 'Satisfactory' and all four 'no chunk' cases were produced by this factor. In Study 2, all three factors have more tendency to produce below 'Satisfactory' chunks.

Study 1 (Manual Exploration)

Task (^)	WCT Task 1 (20)			WCT Task 2 (14)			WCT Task 3 (16)			ASM Task 1 (9)			ASM Task 2 (1)			ASM 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)	RP	P1(1)				/			R1(2)	/	/	P1(2)	/	P1(2)	Total
Total 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%)

Study 2 (Online Exploration)

Task (^)	WCT Task 1 (20)				WCT Task 2 (14)				WCT Task 3 (16)				ASM Task 1 (9)				ASM Task 2 (1)			ASM 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)	RP		RP	P1(2)		P1(3)	P1(4)	P1(1)			/	/		P1(1)	P1(2)	R1(6)	R1(12)	R1(6)	/			/	Total
Total 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%)

^ Oracle's Size; '/' - No chunk found; R0(x) - Chunk with recall measure 0.0 & size x; P0(x) - Chunk with precision measure 0.0 & size x; RP - R0(1), P0(1); R1(x) - Chunk with recall measure 1.0 & size x; P1(x) - Chunk with precision measure 1.0 & size x; PE - The perfect chunk with recall & precision 1.0;

Table 14 : No Chunk Cases and Special Chunks

Study 1 (Manual Exploration)

Task (^)	WCT Task 1 (20)			WCT Task 2 (14)			WCT Task 3 (16)			ASM Task 1 (9)			ASM Task 2 (1)			ASM Task 3 (19)			
Group	I	A	C	I	A	C	I	A	C	I	A	C	I	A	C	I	A	C	
Factor A	P	S	P	/	S	/	P	S	P	*P*	P	P	*VG**	*VG**	*VG**	P	/	P	
Factor H	P	/	/	S	/*	/	VP	S	VP	/*	S	P	*VG**	*VG**	*VG**	P	/*	/	
Factor R3	*S*	*VG*	*VG*	*S*	*P*	S	*P*	*VG*	GD	VP	S	P	S	P	P	*S*	S	S	
Factor A|H|R3	P	GD	P	S	S	S	VP	GD	VP	*P*	S	S	GD	GD	*VG**	S	/	P	
Factor AveR3	S	*VG*	VG	*S*	*P*	*S*	*P*	GD	*GD*	VP	*S*	*S*	P	P	P	*S*	*S*	*S*	
Factor AveR3F	VP	P	P	P	VP	P	VP	S	VP	/*	S	S	GD	/	/	P	/	P	Total
Total 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 S	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%)

Study 2 (Online Exploration)

Task (^)	WCT Task 1 (20)				WCT Task 2 (14)				WCT Task 3 (16)				ASM Task 1 (9)				ASM Task 2 (1)				ASM Task 3 (19)				
Group	I	A	S	C	I	A*	S	C	I	A	S	C	I	A	S*	C	I	A*	S	C	I	A*	S*	C	
Factor R3	*P*	*S*	*P*	*S*	*P*	*S*	*S*	S	*P*	P	*VP*	P	P	*VP*	*P*	VP	P	*P*	-	P	*P*	*S*	*VP*	*S*	
Factor AveR3	*P*	*S*	*P*	P	*P*	*S*	*S*	*S*	*P*	*S*	VP	*P*	P	*VP*	*P*	VP	P	*P*	-	P	*P*	*S*	*VP*	*S*	
Factor AveR3F	P	VP	VP	VP	P	*S*	P	S	P	VP	VP	/	*S*	/	*P*	*P*	*P*	*P*	-	*P*	/	*S*	*VP*	/	Total
Total 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 S	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%)

Usefulness of Chunk: VG - Very Good; GD - Good; S - Satisfactory; P - Poor ; VP - Very Poor X* - X Group has one participant only ^ Oracle's Size

The best chunk for the group *VG* - Very Good (Perfect Chunk)* '/' - No chunk was found '/*' - No chunk was found due to lack of responses

Table 15 : Usefulness of Chunks and Best Chunks

Factor	Study 1							Study 2						
	'/'	VG	GD	S	P	VP	>= S (%)	'/'	VG	GD	S	P	VP	>= S (%)
A	3	3	0	3	9	0	40							
H	7	3	0	3	3	2	54.5							
R3	0	3	1	8	5	1	66.7*	0	0	0	7	12	4	30.4*
A|H|R3	1	1	4	6	4	2	64.7							
AveR3	0	2	2	8	5	1	66.7*	0	0	0	7	12	4	30.4*
AveR3F	4	0	1	3	6	4	28.6~	4	0	0	4	9	6	21.1~

Usefulness of Chunk: VG - Very Good; GD - Good; S - Satisfactory; P - Poor; VP - Very Poor
'/' - No chunk was found * Best Factor ~ Worst Factor

Table 16 : Overall Usefulness of Chunks for Each Factor

4.3 Performance of Chunk-Identification Factors

The results in both studies show that across different tasks and participants, the number of better chunks (i.e. those with usefulness measures of 'Satisfactory' and above) produced by Factors R3 and AveR3 (and also Factor A|H|R3 in Study 1) are relatively more than other factors (Table 16). At the same time, none of the 'no chunk' cases were produced by Factors R3 and AveR3. This shows that **Factors R3 and AveR3 are generally more suitable than other factors for finding better chunks**. Factor A|H|R3 in Study 1 also shows the potential of finding better chunks but it uses two other forms of participant response and this required more effort from the participants.

Study 2 produced poorer chunk discovery results compared to Study 1: significantly fewer chunks with usefulness of 'Satisfactory' and above, and lower tendencies of Factors R3 and AveR3 in producing chunks with these usefulness measures. These could be due to both SA 'novices' and 'experts' participated in Study 2, and only SA 'experts' participated in Study 1. Another reason could be that reading behaviour is different in an on-line (Study 2) compared to an off-line environment (Study 1), with on-line reading promoting a more superficial approach to understanding, and therefore reducing the usefulness of chunks found in Study 2. We do not discuss the difference of the results between the two document formats further because whatever the differences might be, it could be due to 'novices' and 'experts' participated in Study 2 and only 'experts' participated in Study 1, and not necessarily due to the different document format in the two studies. Despite the differences in terms of the participants and the document format in the two studies, their results are consistent: the number of chunks with usefulness measure of 'Satisfactory' and above produced by Factors R3 and AveR3 are relatively more than for other factors in both studies (previous paragraph), and Factor AveR3 followed by Factor R3 have the highest frequencies of producing best chunks in both studies (Section 4.4).

The findings here 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 behaviour in on-line versus off-line environments.

Even though there were conflicts between different factors in terms of the composition of the chunks they produced and the usefulness of these chunks, our purpose was to find which of these factors are more suitable to be used in finding chunks. By benchmarking the chunks against an oracle set, we were able to perform a consistent assessment of the usefulness of chunks found using the different factors. Based on that, we were able to identify which factors are more suitable for finding chunks. We found that Factors R3 and AveR3 are more suitable for finding chunks.

4.4 Best Chunks and Best Chunk-Identification Factor

Table 17 shows the number of best chunks produced by each factor and the distribution of their usefulness measures. For example, Factor R3 produced 8 best chunks in Study 1, and 3, 3, 2 of them are 'Very Good', 'Satisfactory' and 'Poor', respectively.

Factor	Study 1						Study 2					
	VG	GD	S	P	VP	Grand Total	VG	GD	S	P	VP	Grand Total
A	3	0	0	1	0	4						
H	3	0	0	0	0	3						
R3	3	0	3	2	0	8	0	0	6	7	3	16
A|H|R3	1	0	0	1	0	2						
AveR3	1	1	7	2	0	11*	0	0	7	8	2	17*
AveR3F	0	0	0	0	0	0~	0	0	3	5	1	9~

Usefulness of Chunk: VG - Very Good; GD - Good; S - Satisfactory; P - Poor; VP - Very Poor
* Best Factor ~ Worst Factor

Table 17 : Number of Best Chunks Produced by Each Factor

Factor AveR3 has the highest frequency of producing best chunks in both Studies 1 and 2. This is followed by Factor R3. In both studies, Factors R3 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 1). Consequently, **we cannot distinguish 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 more suitable for finding chunks**. Nevertheless, when the oracle set for an information-seeking task is very small (such as ASM Task 2), factors based on the preference of the majority of the participants in a group (such as Factor A, H or A|H|R3 in Study 1), are more suitable for finding better chunks (Table 15).

4.5 Comparison between Different Groups of Participants

The following discussion is based on Study 1 only because there was only one participant in some of the groups involved in the tasks in Study 2. Table 18 compares the usefulness of I and A Groups' chunks found using the same factor for each information-seeking task in Study 1.

For both tasks related to obtaining an overview of the SA of a system (i.e. WCT Task 1 and ASM Task 1), A Group's chunk has better usefulness measure compared to I Group's chunk found using the same factor. In other words, academic group's chunk has a closer match to the relevant oracle set. The opposite situation occurred 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 tasks, I Group's chunk has better if not equal usefulness 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 academic professionals.

For WCT Task 3, A Group's chunk has better usefulness measure compared to I Group's chunk found using the same factor. For ASM Task 3, I Group's chunk generally has better if not equal usefulness measure compared to A Group's chunk found using the same factor. This contradictory situation could be because, although both tasks address cross-cutting concerns, they focus on different issues. The results show that academic group's chunks fare

better on the security issue (WCT Task 3) whereas the industry group's chunks fare better on the modifiability issue (ASM Task 3), when the chunks were benchmarked against oracle sets constructed by academic professionals.

	Study 1 (Manual Exploration)	
	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 usefulness measure than Y Group's chunk found using the same factor.
'X = Y' means X Group's chunk has equal usefulness measure as Y Group's chunk found using the same factor.
'X >= Y' means X Group's chunk has better or equal usefulness measure compared to Y Group's chunk found using the same factor.
'Factor W*' means no chunk was found for the particular group when Factor W was used.
Italic text means there is a greater number of factors producing chunks with better or equal usefulness measure for the particular group.

Table 18 : Comparison of Different Groups' Chunks Found Using Same Factor

The above mixture of results shows that our oracle sets did not totally favour the academic group although they were developed by academic professionals. The findings suggest the use of academic professionals to find chunks for the task of getting an overview of the SA of a system, and the task related to architectural design on security; but to use industry practitioners to find chunks for tasks related to system changes (such as changing a part of a system and accessing possible impact of change, and modifiability). The latter could be because generally industry practitioners have more exposure to system changes, which are frequent in industry settings.

4.6 **Assessment of our Approach**

In finding chunks, we used the consumers' annotation data which carries their conscientious judgements on the relevance of the documents' sections to the assigned task. We reduced the bias of individual consumer by looking for 'commonality' in the annotation data of a group of consumers.

Our approach requires ADs to be consumed in order to produce chunks. The variety of task-specific chunks that can be found is therefore dependent on the uses of the ADs for different tasks. Both our approach and model-based approaches entail significant efforts from the partakers. With proper tool support, interaction data 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 usage of the documents, than requiring producers to produce ADs that conform to underlying models.

We experimented with 'document section' as the level of granularity for chunk elements. Unless each section describes one architectural element, our current approach would miss insights into the usage relationships between

the architectural elements located within the same section of a document. We assumed all architectural elements located in a document's section were being used or rated equally by a consumer attempting the specified task. Our approach could be adapted to a model-based approach to study whether a set of architectural elements are actually used in the way they are chained by formal models.

5 Threats to Validity

In this section, we discuss the threats that might have affected the validity of the findings of our studies following the threat classification schemes in (Wohlin et al., 2012).

5.1 Internal Validity

Annotatable sections (Instrumentation) - We inserted annotation fields to the beginning of each section of the ADs to collect participants' ratings, tags, comments and so on for the section. Each annotatable section was enclosed by a border to distinguish it from others. We did the same to diagrams and subsections that contained a substantial amount of information or distinct information by themselves, but with no change to the order. The annotation fields and borders might have affected the participants in finding the needed information. However, all the participants were given the same instrumented document, be it WCT or ASM document.

Duration of participation session (Maturation) - The 75-minute participation session might have affected the participants' focus on the tasks especially when they worked on the second task as they became tired. The effect of this on the identification of chunks was reduced through the reversal of the sequence of tasks for each alternate participant and by using the preference of the majority of a group in finding chunks.

Familiarity with document when performing the second information-seeking task (History) – Familiarity of a participant with the given document when performing the second information-seeking task might have affected his or her process of finding the information needed for the second task. The effect of this on the identification of chunks was reduced through the reversal of the sequence of tasks for each alternate participant and by using the preference of the majority of a group in finding chunks.

5.2 External Validity

*Participants recruitment and number of participants (*Interaction of selection and treatment*)* - Our studies sought participants with very specific background (namely, SA) and the exploratory nature of the information-seeking tasks entailed a considerable amount of time and effort to complete the tasks. Therefore, to be realistic in participants' recruitment, we employed non-probabilistic sampling techniques (Section 3.1). This rendered the results not generalisable to the target population (Kitchenham & Pfleeger, 2002).

Despite using non-probabilistic sampling techniques, we encountered much difficulty in obtaining participants for the studies. Nevertheless, none of the participant took part in more than one of our 3 studies. We invited 80 potential industry and academic participants for Studies 1 and 2, and 24 students for Study 2 (Section 3.2). Seventy-two responded, 4 of them were excluded, 30 took part in Study 1 and 38 took part in Study 2.

For Study 1, only 23 participants' responses were analysed to identify chunks. These participants were industry or academic professionals who have strong SA background. The industry participants had on average 10.9 (with minimum 2 and maximum 24) years of SA-related industry experience. The academic participants had on average 9.15 (with minimum 2 and maximum 20) years of experience in SA teaching or training. Most participants (21) had some experience with ADs, one had exposure to ADs from course taken and only one had no prior exposure. In terms of experience in the consumption and production of ADs, most participants (17) always read, and more than half (13) always read and made use of ADs. However, less than half (10) always wrote and only about one-third of the participants (8) always updated ADs. This shows that there was generally more involvement in the

consumption than the production of ADs among the participants. This is not a concern since our focus is on the usage of ADs. In terms of experience with the type of the software system described by the given AD, most of the participants (17) had experience. All in all, the participants in Study 1 had strong background in the aspects that we sought. This led us to believe that our findings from Study 1 are useful for providing early insights into whether chunks can be identified based on consumers' usage of ADs.

For Study 2, only 19 participants' responses were analysed to identify chunks. These participants were a mixture of SA 'experts' and 'novices' as we could not recruit an adequate individual numbers of 'novices' or 'experts'. Therefore, we are not discussing the participants' background like what we did above for Study 1, and we are very cautious in stating any findings from Study 2. Nevertheless, the results from Study 2 are consistent with the results from Study 1. In terms of factors producing better chunks, the number of chunks with usefulness measure of 'Satisfactory' and above produced by Factors R3 and AveR3 are relatively more than other factors, in both studies. In terms of factors producing best chunks, Factor AveR3 followed by Factor R3 have the highest frequencies of producing best chunks in both studies.

Choices of ADs and information-seeking tasks (Interaction of setting and treatment) – To make the object of exploration as realistic as possible, we used WCT and ASM ADs that are existing ADs describing real systems in use (Section 3.3). Two of the information-seeking tasks in our studies are typical information-seeking tasks in a software development project. They are finding out about the SA of a system, and finding out how a system was designed at the architectural level to achieve certain quality attribute, namely, security and modifiability (Section 3.4). These kinds of tasks are ones that a number of members involved in the same software development project are likely to be interested in, especially in collaborative software development environment. Making usage-based chunks for tasks of similar genres available for a software project team enables team members to reuse these chunks as alternatives to searching information using keywords or browsing the whole document when engaged with similar tasks. Using usage-based chunks leverages previous users' usage of the information in the document and can be useful especially for novice members of a team.

5.3 **Construct Validity**

The ADs Used (Mono-operation bias) - The ADs used might have affected the results of the studies. We mitigated this risk by using two ADs, and by selecting them carefully (Section 3.3).

The Information-Seeking Tasks (Mono-operation bias) - The information-seeking tasks could have affected our results. We mitigated this by using 3 tasks and refined the specification of these tasks based on the feedback from the user evaluation study.

5.4 **Conclusion Validity**

Personal bias of participants (Random heterogeneity of subjects) - Different participants attempting the same task could have interpreted the task differently. They could have adopted different strategies to answering the same task. They could also have different ideas of the concepts (SA, security and modifiability) involved in the first and third tasks of both ADs. We mitigated these risks partially by asking participants to seek clarifications from us. In addition, the effect of the inherent differences of the participants on the identification of chunks was reduced by using the preference of the majority of a group in finding chunks.

Benchmarking Against Oracle Sets (Reliability of measures) - 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. Relevance judgments (Wallis & Thom, 1996) is the basis of our oracle sets. The instability of relevance judgments (Wallis & Thom, 1996) is nothing new. However, they could still be used to compare the 'relative' effectiveness of IR systems (Lesk & Salton, 1968). It has been shown that regardless of which person's set of relevance judgments is used to compare the 'relative' effectiveness of IR systems, a technique for IR that performs well on one set of judgments would perform well on other sets of judgments (Lesk & Salton, 1968; Wallis & Thom, 1996).

The use of oracle sets in our studies is similar to the use of relevance judgments in IR systems evaluation. Our oracle sets were used to compare the 'relative' usefulness of chunks and the 'relative' performance of the chunk-identification factors. Following that, changing to different oracle sets would most likely has 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 R3) show potential in finding chunks of 'Satisfactory' usefulness.

As a form of verification of our oracle sets, we identified the oracle set's sections which were totally excluded by all chunk-identification factors (i.e. the section did not appear in any chunk found, including the 'no chunk' cases), and false sections included by all factors, across all groups of participants for a particular information-seeking task. We found some disagreement between the sections needed by the participants for a particular task and the sections in the oracle set. Nevertheless, totally-excluded oracle sets' sections were found for only 3 tasks in Studies 1 and 2 respectively, and their numbers were small (with at most 18.8% or 3 sections for WCT Task 3 in Study 1). As for totally-included false sections, they (5) were found only for ASM Task 2 in Study 2.

No involvement of judges from industry in the construction of the oracle sets could be a threat to the validity of our results. However, as in Section 4.5, we found that our oracle sets did not totally favour the academic groups although they were developed by academic professionals. We found a mixture of results for Study 1 (Table 18): For WCT Task 1, WCT Task 3 and ASM Task 1, Academic (A) Group's chunk has better or equal usefulness measure compared to Industry (I) Group's chunk found using the same factor, except for when no chunk was found for A Group when Factor H was used for WCT Task 1. For the other 3 information-seeking tasks, I Group's chunks has better or equal usefulness measure compared to A Group's chunk found using the same factor, except when Factor A was used for WCT Task 2. We excluded Study 2 from our discussion here because there was only one participant in some of the groups involved in the tasks in Study 2.

It seems that which group's chunks are of better usefulness measure is affected by the information-seeking task. For each task (except ASM Task 2), regardless of whether A or I Group is the winning group, chunk found using Factor AveR3 or R3 for the winning group is always the best chunk among the chunks found using the different factors (Study 1 in Table 15). Therefore, our overall findings still hold, namely, usage-based chunks exist and factors based on the preference of the majority of those who rated (be it Factor AveR3 or R3) show potential in finding chunks.

6 Related Work

6.1 Existing Approaches in Finding of Architectural Information

To assist finding of information, a reading guide for a document can be produced by applying Latent Semantic Analysis (LSA) to the document's sections. Although to a lesser extent, this will encounter the same limitations as when LSA was applied to a set of documents, which are, results are dependent on the initial query terms and human interpretation is needed to select suggested documents (or sections) to read (de Boer & van Vliet, 2008). Usage-based chunks in our work are found based on the consumers' actual usage of the content of the documents whereas reading guides produced by LSA are based on the documents' content as described by the documents' producers. These two types of reading guides complement each other.

Some studies have been using formal models to capture AK as a prelude to support finding of AK (Avgeriou et al., 2007; de Boer & van Vliet, 2011; de Graaf et al., 2012; Jansen et al., 2009; Su et al., 2009; Tang et al., 2011). Approaches that use formal models assist knowledge retrieval by enabling automated reasoning and querying. Nevertheless, model-based approaches entail rigidity. The producers need to follow the underlying 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 (de Boer & van Vliet, 2011).

A number of studies have been focusing on automatically generating stakeholder-specific ADs as a means to help stakeholders in finding architectural information. These studies are described in the next three paragraphs.

Nicoletti et al. personalised ADs' content to suit the information needs of stakeholders by calculating the similarity measures between the stakeholders' profiles and the profiles of sections of an AD (Diaz-Pace et al., 2013; Nicoletti et al., 2012). They also used a stakeholder's perception (in the form of comments given) of an AD to personalise future version of the AD for him or her. They implemented their approach in a recommendation tool with promising preliminary results (Nicoletti, Diaz-Pace, Schiaffino, Tommasel, & Godoy, 2014). Both the work of Nicoletti et al. and ours make use of data of users' usage of documents. They use a user's interaction data (such as number of visits, mouse clicks and so on) to infer his or her interests to enrich the basic user profile (Nicoletti et al., 2014). Basic user profiles were derived from View & Beyond characterisation of stakeholders (Clements et al., 2010). Our work differs by using annotation data (such as ratings) provided by multiple users to identify chunks that comprises sections of an AD needed for an information-seeking task.

TopDocs (Eloranta et al., 2012) dynamically generates topical ADs using information in an Architectural Knowledge Base (AKB). A topical document is an information package tailored to a specific task or concern at hand. The information in AKB is organised based on a meta-model. The approach involves looking at the meta-model from the viewpoint of a particular stakeholder, finding the meta-model elements that identify a major concern for the stakeholder and retrieving all connected elements to include them in the topical document.

Rost et al. proposed to generate a task-specific architecture documentation for each individual developer from general documentation (Rost, 2012). A Software Architect creates a specification for a developer's task, which is used to produce architecture documentation based on pre-defined identification and representation models. Identification model contains rules that specify which architectural elements can be classified as relevant and will be included, and representation model specifies how the relevant elements will be represented in the generated documentation. They focus on architecture documentation in the form of architecture models.

Compared to the studies above that automatically generate specialised ADs, the exploration paths in our KaitoroCap automatically extract contents needed and dynamically restructures an existing AD based on consumers' actual usage of the AD. TopDocs (Eloranta et al., 2012) and Rost et al.'s work on task-specific documents (Rost, 2012) are fundamentally model-based. Model-based approaches as mentioned earlier suffer in terms of rigidity, steeper learning curve and less support for unstructured or semi-structured knowledge. The work of Nicoletti et al. (Diaz-Pace et al., 2013; Nicoletti et al., 2012) uses the semantic information in the content of the sections of a document, to identify sections relevant to a stakeholder. Eloranta et al. (Eloranta et al., 2012) and Rost et al. (Rost, 2012) employ models to capture the architectural elements, for identifying the relevant architectural elements to include in the specialised ADs. Our approach uses more abstract quality of the content of the sections to determine relevance. For example, users' ratings of the sections' importance to the assigned task.

6.2 **Leveraging Usage Data**

Our basic idea in identifying chunks is by finding 'commonality' in the consumers' usage of the information in ADs when engaged with certain information-seeking task. In relation to that, this section presents the related works in leveraging the data of previous consumers' usage of some artefacts (such as documents, source code and so on) to assist information finding by other consumers. These related works span a number of areas that are inter-related but with different focuses. These areas are computational wear (Hill & Hollan, 1994; Hill, Hollan, Wroblewski, & McCandless, 1992), social navigation (Dieberger, Dourish, Höök, Resnick, & Wexelblat, 2000; Dourish & Chalmers, 1994; Munro, Hook, & Benyon, 1999), collaborative filtering (CF) (Goldberg et al., 1992; Schafer et al., 2007; Shardanand & Maes, 1995), social filtering (Lerman, 2007), wear-based filtering (DeLine, Khella, Czerwinski, & Robertson, 2005), and the set of studies on Degree-Of-Interest (DOI) model (Elves, 2014; Kersten & Gail, 2005, 2006; Tasktop Technologies Inc., 2013) and Degree-Of-Knowledge (DOK) (Thomas et al. 2010).

Table 19 compare these works and ours. They are compared in terms of task specificity, type of usage data being leveraged, source of usage data, type of artefact of which usage data is being leveraged, and the type of collaboration involved.

In terms of task specificity, most of the existing works aggregate usage data collected for multiple tasks instead of for individual task. One exception is the later study on DOI model (Kersten & Gail, 2006). In fact, there is no clear emphasis on the notion of task in existing works, except for the studies on DOI and DOK. We think that task is an important notion in information finding. We also think that a particular task drives the information exploration process and scopes the set of information or sections from a document needed for the task. Therefore, the aggregation of usage data in our work is task-specific instead of spanning across multiple tasks.

The type of usage data leveraged is either interaction, annotation data, or both. Interaction data used includes: approximation of time spent on an artefact based on lower-level interaction events with editor as in Zmacs Editor (Hill et al., 1992); sequence of visiting artefacts or path as in IBM's WBI (Maglio & Barrett, 2000) and Footprints (Wexelblat & Maes, 1999); frequencies of visiting artefacts as in Team Tracks (DeLine, Czerwinski, & Robertson, 2005; DeLine, Khella, et al., 2005); frequencies of visiting artefacts and recentness of interactions as in Mylar (Kersten & Gail, 2006) or Mylyn (The Eclipse Foundation, 2013), in Tasktop (Elves, 2014) or Tasktop Dev (Tasktop Technologies Inc., 2013), and in DOK model (Thomas, Jingwen, Gail, & Emerson, 2010). Annotation data used includes ratings, tag, comment in Tapestry (Goldberg et al., 1992); vote and tagging in Diggs (Lerman, 2007), Reddit (Reddit Inc., 2014), and Flickr (Yahoo! Inc., 2013); and percentage of users who followed a hyperlink as in Footprints (Wexelblat & Maes, 1999). In summary, computational wear (in particular Zmacs Editor (Hill et al., 1992)), wear-based filtering and studies on DOI and DOK leverage interaction data. Collaborative and social filtering leverage annotation data. Social navigation leverages both data types. Our current work on chunks identification leverages annotation data. Comparing to interaction data (such as frequency of visit) which is at a more superficial level, annotation data carries the consumers' conscientious judgments (in the forms of ratings and so on) on whether a document's sections are needed for a particular task.

In terms of the source of usage data being leveraged, most of the existing works leverage the aggregated usage data of multiple users instead of leveraging the usage data of a single user. One exception is the studies on DOI and DOK that leverage a user's own instead of others' usage data. Our aggregation involves usage data of multiple users instead of single user, but differs from existing works since ours is task-specific.

The types of artefact of which usage data are being leveraged range from unstructured to structured data. The former include document, with a line of text as in Zmacs Editor (Hill et al., 1992) or a page as in IBM's WBI (Maglio & Barrett, 2000), Footprints (Wexelblat & Maes, 1999) and CoWEB (Dieberger & Guzdial, 2003), Tasktop (Elves, 2014) or Tasktop Dev (Tasktop Technologies Inc., 2013), as the level of granularity. The more structured data include news as in Tapestry (Goldberg et al., 1992), Diggs (Lerman, 2007), and Reddit (Reddit Inc., 2014); media, such as photos as in Flickr (Yahoo! Inc., 2013); and source code as in Team Tracks (DeLine, Czerwinski, et al., 2005; DeLine, Khella, et al., 2005) and Mylar (Kersten & Gail, 2006) or Mylyn (The Eclipse Foundation, 2013). The type of artefact involved is dependent on the specific purpose of the particular study. The type of artefact in our work is semi-structured ADs. Our purpose is to find chunks to assist information finding in these documents. We focused on per section basis of a document. To us, individual lines as in Zmacs Editor are too low a level for chunking of architectural information.

The type of collaboration involved is either implicit, explicit or both. Users' identities are unknown in implicit collaboration, but exposed and might affect the users in explicit collaboration. Generally, implicit collaboration is involved in existing works, except for areas that overtly emphasise collaboration in information finding, such as social and collaborative filtering. Our work involves implicit collaboration. We aggregated the usage data of all users performing the same task and they did not know the identities of each other when performing the task.

	Key Features	Task Specificity	Type of Usage Data	Source of Usage Data	Type of Artefact	Type of Collaboration
Computational Wear (Hill & Hollan, 1994; Hill et al., 1992)	Makes users' interaction history with computational objects part of the objects to create 'usage wear'.	Multiple Tasks - Zmacs Editor (Hill et al., 1992)	Interaction (approximate time spent) - read wear in Zmacs Editor (Hill et al., 1992)	Multiple Users	Document (line of text) - Zmacs Editor (Hill et al., 1992)	Implicit
Social Navigation (SN) (Dieberger et al., 2000; Dourish & Chalmers, 1994; Munro et al., 1999)	Information navigation mechanism based on the behaviour of other people manifested as navigation traces or 'footprints'; Navigation traces are dynamically grown; Personalised to the user; Aggregated behaviour of a community (optional).	Multiple Tasks	Interaction (path) - IBM's WBI (Maglio & Barrett, 2000); Interaction (traffic through pages of web site, aggregated path) - Footprints (Wexelblat & Maes, 1999); Annotation (% of users who followed link; comment) – Footprints (Wexelblat & Maes, 1999)	Single and Multiple Users	Document (web pages) - IBM's WBI (Maglio & Barrett, 2000), Footprint (Wexelblat & Maes, 1999), CoWEB (Dieberger & Guzdial, 2003); Document (line of text) - Zmacs Editor Hill et al., 1992)	Implicit
Collaborative Filtering (CF) /Social Information Filtering (Goldberg et al., 1992; Schafer et al., 2007; Shardanand & Maes, 1995)	People collaborate by recording their opinions as annotations on information read. The annotations are used as filters to sieve the information to receive.	Multiple Tasks	Annotation (ratings, tag, comment) – Tapestry (Goldberg et al., 1992)	Multiple Users	News in moderated newsgroups- Tapestry (Goldberg et al., 1992);	Implicit; Explicit (filter for items annotated by certain user – Tapestry (Goldberg et al., 1992))
Social Filtering / Social Recommendation / Social Information processing (Lerman, 2007)	Users choose people with similar interests to form explicit social networks to find items-of-interest; Includes social media & media sharing sites; Extends CF to use social networks for filtering; Refines SN to centre on	Multiple Tasks	Annotation (tag, voting) – Diggs (Lerman, 2007), Reddit (Reddit Inc., 2014), Flickr (Yahoo! Inc., 2013)	Multiple Users	News – Diggs (Lerman, 2007), Reddit (Reddit Inc., 2014); Photo – Flickr (Yahoo! Inc., 2013)	Explicit (identities of community's members are exposed)

	explicit social networks.					
Wear-based Filtering (DeLine, Khella, et al., 2005)	Combines computational wear and CF, and uses users' interaction history with code elements to filter interface of IDE.	Multiple Tasks - Team Tracks (DeLine, Czerwinski, et al., 2005; DeLine, Khella, et al., 2005)	Interaction (frequency of visit) - Team Tracks (DeLine, Czerwinski, et al., 2005; DeLine, Khella, et al., 2005)	Multiple Users	Source Code - Team Tracks (DeLine, Czerwinski, et al., 2005; DeLine, Khella, et al., 2005)	Implicit
Degree-Of-Interest (DOI) Model (Elves, 2014; Kersten & Gail, 2005, 2006; Tasktop Technologies Inc., 2013)	DOI model (relevance of code elements to current task, constructed from a programmer owns interaction history with code) to filter interface of IDE; Combines DOI with degree-of-authorship to form degree-of-knowledge (DOK) to find who knows what about the code.	Multiple Tasks - 1st version of DOI (Kersten & Gail, 2005) and DOK (Thomas et al. 2010); Task Specific - 2nd version of DOI (Kersten & Gail, 2006);	Interaction (frequency of visit; recentness of interaction) – Mylar (Kersten & Gail, 2006) /Mylyn,(The Eclipse Foundation, 2013), Tasktop (Elves, 2014) /Tasktop Dev (Tasktop Technologies Inc., 2013), DOK model (Thomas et al. 2010)	Own Usage Data	Source Code - Mylar (Kersten & Gail, 2006) /Mylyn,(The Eclipse Foundation, 2013); Document (pages, web pages) – Tasktop (Elves, 2014) /Tasktop Dev (Tasktop Technologies Inc., 2013)	Not Applicable – DOI (Kersten & Gail, 2005, 2006); Explicit – DOK (Thomas et al. 2010)
Our work	Supports read wear (paths in KaitoroCap), social navigation (paths, ratings, tags and comments), and collaborative filtering.	Task Specific	Annotation (ratings, specification of from which sections answer was found, highlighted content)	Multiple Users	Document (pages/wiki pages, sections on pages/wiki pages)	Implicit

Table 19 : Summary of Existing Works on Leveraging Usage Data

In summary, our work supports computational wear, social navigation and CF. Exploration paths in KaitoroCap serve as usage 'wear' left by previous consumers. Visible usage data (such as exploration paths, ratings, tags and comments) serve as information traces which support some forms of social navigation. Chunks serve as collaborative filters for information needed for specific tasks.

7 Conclusions and Future Work

The key findings from our studies are:

- usage-based chunks exist (Section 0).

- usage-based chunks of 'Satisfactory' usefulness can be found using factors that make use of the preference of the majority of those who rated (Factor AveR3 or R3), with the performance of these factors being affected by the participants' background in SA and the different reading behaviour in on- line versus off-line environments (Sections 0, 4.3, 4.4)
- the use of different groups of professionals to find chunks for different types of tasks: the use of academic professionals to find chunks for the task of getting an overview of the described SA, and the task related to architectural design on security; and the use of industry practitioners to find chunks for tasks related to system changes (Section 4.5).

The usefulness of usage-based chunks give some insight on their support of information searching in ADs when similar information-seeking tasks are undertaken. The usefulness of chunks were determined using criteria which trade-off the recall and precision measures of the chunks. These two measures tell us how complete and precise a chunk is for the specific information-seeking task. As a collection of related pieces of architectural information needed for a particular task, a chunk simplifies finding of information by consumers engage with similar tasks, by enabling related architectural information which may be dispersed in an AD to be retrieved collectively as a unit.

To support information searching in ADs, most existing studies focus on the production aspect of ADs or AK. We explore the other side of the coin, by proposing usage-based chunks found from consumers' usage of the information in ADs when they engage with information-seeking tasks. These chunks can be used to explore ADs when performing similar information-seeking task. Our usage-based chunking approach shows potential for collaborative construction of architectural chunks. It seems to be able to identify chunks relatively well even from a relatively small number of people wanting to use an AD for a given task. We have a proof-of-concept tool (KaitoroCap) that shows how identified chunks can be presented as restructured documents. However, there are still many things that need to be addressed. For example:

- how useful do end users find the identified chunks in practice?
- what is the best way of presenting the chunks to users - as we have done in KaitoroCap or is there a better way?
- how is it best to identify the task that a person has and match that to a previously identified chunk?

Our work can also be extended in a number of ways, such as, to use interaction data for identifying chunks and to study usage-based chunking in other types of documentation.

In conclusion, the novelty of our work lies in chunking of architectural information based on usage of ADs. Our work provides a new starting point for future tool builders of AK management or AD. It also provides a new direction for a collaborative construction of AK at a higher level than individual architectural elements.

8 Acknowledgements

We thank Thiam Kian Chiew for constructive feedback on the work, all the participants in our studies, and those who helped us in recruiting the participants.

9 Funding

This work was supported by University of Malaya; Ministry of Higher Education, Malaysia; PReSS, University of Auckland; and FRST Software Process and Product Improvement project. The sponsors had no involvement in the study design; in the collection, analysis, and interpretation of data; in the writing of the report; and in the decision to submit the paper for publication.

10 References

Asai, H., & Yamana, H. (2014). Intelligent Ink Annotation Framework that uses User's Intention in Electronic Document Annotation. *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces* (pp. 333-338). 2669542: ACM. doi:10.1145/2669485.2669542

Atlassian. (2013). *Confluence*. Retrieved Dec 1, 2013, from http://www.atlassian.com/software/confluence/

Avgeriou, P., Kruchten, P., Lago, P., Grisham, P., & Perry, D. (2007). Architectural knowledge and rationale: issues, trends, challenges. *ACM SIGSOFT Software Engineering Notes, 32*(4), 41-46. doi: 10.1145/1281421.1281443

Bachrach, Y., Graepel, T., Kasneci, G., Kosinski, M., & Gael, J. V. (2012). Crowd IQ: aggregating opinions to boost performance. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1* (pp. 535-542). 2343653: International Foundation for Autonomous Agents and Multiagent Systems

Baker, E., & Olaleye, O. (2013). Combining Experts: Decomposition and Aggregation Order. *Risk Analysis, 33*(6), 1116-1127. doi: 10.1111/j.1539-6924.2012.01937.x

Bass, L., Clements, P., & Kazman, R. (2003). *Software Architecture in practice* (2nd ed.). Boston, MA: Addison-Wesley Professional.

Bosch, J. (1999). Evolution and composition of reusable assets in product-line architectures: A case study. In P. Donohoe (Ed.), *Proceedings of the TC2 First Working IFIP Conference on Software Architecture,* 321-339. doi:10.1007/978-0-387-35563-4_18

Chi, E. H., Gumbrecht, M., & Hong, L. (2007). Visual Foraging of Highlighted Text: An Eye-Tracking Study. In J. A. Jacko (Ed.), *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments: 12th International Conference, HCI International 2007, Beijing, China, July 22-27, 2007, Proceedings, Part III* (pp. 589-598). Berlin, Heidelberg: Springer Berlin Heidelberg.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., . . . Stafford, J. (2010). *Documenting software architectures: Views and beyond* (2nd ed.). Boston, MA: Addison-Wesley Professional.

Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. (2003). *Documenting software architectures: Views and beyond*. Boston, MA: Pearson Education.

Curtis, B. (1984). Fifteen years of psychology in software engineering: Individual differences and cognitive science. *Proceedings of the 7th International Conference on Software Engineering* (pp. 97-106). Piscataway, NJ: IEEE Press

de Boer, R. C. (2006). Architectural knowledge discovery: why and how? *ACM SIGSOFT Software Engineering Notes, 31*(5), 1. doi: 10.1145/1163514.1178641

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. doi: 10.1016/j.jss.2007.12.815

de Boer, R. C., & van Vliet, H. (2011). Experiences with semantic wikis for architectural knowledge management. *Proceedings of the 2011 9th Working IEEE/IFIP Conference on Software Architecture (WICSA),* 32-41. doi:10.1109/WICSA.2011.14

de Graaf, K. A., Tang, A., Liang, P., & van Vliet, H. (2012). Ontology-based Software Architecture documentation. *Proceedings of the 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture,* 121-130. doi:10.1109/WICSA-ECSA.212.20

DeLine, R., Czerwinski, M., & Robertson, G. (2005). Easing program comprehension by sharing navigation data. *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing,* 241-248. doi:10.1109/VLHCC.2005.32

DeLine, R., Khella, A., Czerwinski, M., & Robertson, G. (2005). Towards understanding programs through wear-based filtering. *Proceedings of the 2005 ACM Symposium on Software Visualization,* 183-192. doi:10.1145/1056018.1056044

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*, *Lecture Notes in Computer Science* (Vol. 8216, pp. 104-117). doi: 10.1007/978-3-642-41366-7_9

Dieberger, A., Dourish, P., Höök, K., Resnick, P., & Wexelblat, A. (2000). Social navigation: techniques for building more usable systems. *Interactions, 7*(6), 36-45. doi: 10.1145/352580.352587

Dieberger, A., & Guzdial, M. (2003). CoWeb - experiences with collaborative web spaces. In C. Lueg & D. Fisher (Eds.), *From Usenet to CoWebs* (pp. 155-166). doi: 10.1007/978-1-4471-0057-7_8

Dourish, P., & Chalmers, M. (1994). Running out of space: Models of information navigation. *Proceedings of the Human Computer Interaction '94,* Retrieved Jan 1, 2014, from http://fields.eca.ac.uk/deaua/wp-content/uploads/2008/2010/hci2094-navigation.pdf.

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). doi: 10.1007/978-1-84800-044-5_11

Eloranta, V.-P., Hylli, O., Vepsalainen, T., & Koskimies, K. (2012). TopDocs: Using Software Architecture knowledge base for generating topical documents. *Proceedings of the 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture,* 191-195. doi:10.1109/WICSA-ECSA.212.27

Elves, R. (2014). *Tasktop for Eclipse - Get more out of Mylyn*. Retrieved Jan 1, 2014, from http://dc-35717-989588872.us-east-1.elb.amazonaws.com/about/resources/MoreOutOfMylyn.php

Gobet, F., & Lane, P. C. R. (2012). Chunking mechanisms and learning. In N. M. Seel (Ed.), *Encyclopedia of the sciences of learning* (pp. 541-544). doi: 10.1007/978-1-4419-1428-6_1731

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

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61-70. doi: 10.1145/138859.138867

Greefhorst, D., Koning, H., & van Vliet, H. (2006). The many faces of architectural descriptions. *Inf Syst Frontiers, 8*(2), 103-113. doi: 10.1007/s10796-006-7975-x

Hill, W. C., & Hollan, J. D. (1994). History - enriched digital objects: Prototypes and policy issues. *The Inf. Society, 10*(2), 139-145.

Hill, W. C., Hollan, J. D., Wroblewski, D., & McCandless, T. (1992). Edit wear and read wear. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 3-9. doi:10.1145/142750.142751

Horn, R. E. (1997). Structured Writing as a paradigm. In C. R. Dills & A. J. Romiszowski (Eds.), *Instructional development paradigms* (pp. 697-714). Englewood Cliffs, NJ: Educational Technology Publications.

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

Kersten, M., & Gail, C. M. (2005). Mylar: a degree-of-interest model for IDEs. *Proceedings of the 4th International Conference on Aspect-oriented Software Development,* 159-168. doi:10.1145/1052898.1052912

Kersten, M., & Gail, C. M. (2006). Using task context to improve programmer productivity. *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering,* 1-11. doi:10.1145/1181775.1181777

Kitchenham, B., & Pfleeger, S. L. (2002). Principles of survey research: Part 5: Populations and samples. *ACM SIGSOFT Software Engineering Notes, 27*(5), 17-20. doi: 10.1145/571681.571686

Koning, H., & van Vliet, H. (2006). Real-life IT architecture design reports and their relation to IEEE Std 1471 stakeholders and concerns. *Automated Softw. Engineering, 13*(2), 201-223. doi: 10.1007/s10515-006-7736-6

Kruchten, P., Lago, P., & van Vliet, H. (2006). Building up and reasoning about architectural knowledge. In C. Hofmeister, I. Crnkovic & R. Reussner (Eds.), *Quality of software architectures*, *Lecture Notes in Computer Science* (Vol. 4214, pp. 43-58). doi: 10.1007/11921998_8

Lerman, K. (2007). Social information processing in news aggregation. *IEEE Internet Computing, 11*(6), 16-28. doi: 10.1109/MIC.2007.136

Lesk, M. E., & Salton, G. (1968). Relevance assessments and retrieval system evaluation. *Inf. Storage and Retr., 4*(4), 343-359. doi: 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 Softw, 20*(6), 35-39. doi: 10.1109/MS.2003.1241364

Liu, Z. (2005). Reading behavior in the digital environment: Changes in reading behavior over the past ten years. *Journal of Documentation, 61*(6), 700-712. doi: doi:10.1108/00220410510632040

Maglio, P., & Barrett, R. (2000). Intermediaries personalize information streams. *Communications of the ACM, 43*(8), 96-101. doi: 10.1145/345124.345158

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Evaluation in information retrieval. *Introduction to Information Retrieval* (pp. 151-175). Retrieved from http://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp45430.

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.

Munro, A. J., Hook, K., & Benyon, D. (1999). Footprints in the snow. In A. J. Munro, K. Hook & D. Benyon (Eds.), *Social navigation of information space* (pp. 1-14). London, UK: Springer.

National Library of New Zealand. (2006). *Web Curator Tool Software Architecture document*. Retrieved June 1, 2009, from http://webcurator.sourceforge.net/docs/1.0/wct-system-architecture.pdf

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*, *Lecture Notes in Computer Science* (Vol. 7547, pp. 176-185). doi: 10.1007/978-3-642-34010-9_17

Nicoletti, M., Diaz-Pace, J. A., Schiaffino, S., Tommasel, A., & Godoy, D. (2014). Personalized architectural documentation based on stakeholders' information needs. *Journal of Softw. Engineering Res. and Development, 2*(1), 1-26. doi: 10.1186/s40411-014-0009-3

Reddit Inc. (2014). *Reddit*. Retrieved Jan 1, 2014, from http://www.reddit.com/

Rost, D. (2012). Generation of task-specific architecture documentation for developers. *Proceedings of the 17th International Doctoral Symposium on Components and Architecture,* 1-6. doi:10.1145/2304676.2304678

Rost, D., Naab, M., Lima, C., & von Flach Chavez, C. (2013). *Architecture documentation for developers : A survey (IESE-Report No. 028.13/E)*. Retrieved Jan 1, 2014, from http://www.iese.fraunhofer.de/content/dam/iese/de/dokumente/oeffentliche_studien/Fraunhofer-IESE_Software_Architecture_Documentation_for_Developers-A_Survey.pdf

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*, *Lecture Notes in Computer Science* (Vol. 4321, pp. 291-324). doi: 10.1007/978-3-540-72079-9_9

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 210-217. doi:10.1145/223904.223931

Sjoberg, D. I. K., Dyba, T., & Jorgensen, M. (2007). The future of empirical methods in Software Engineering research. *Proceedings of the 2007 Future of Software Engineering,* 358-378. doi:10.1109/FOSE.2007.30

Slupesky, T., & Singleton, T. (2006). *Aperi architecture document*. Retrieved June 1, 2009, from http://wiki.eclipse.org/Aperi_Architecture_Document

Su, M. T. (2010). Capturing exploration to improve Software Architecture documentation. *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume,* 17-21. doi:10.1145/1842752.1842758

Su, M. T. (2014). *Supporting Information Searching in Software Architecture Documents (PhD Thesis in Computer Science)*. University of Auckland, Auckland, New Zealand. Retrieved from http://hdl.handle.net/2292/22565

Su, M. T., Hirsch, C., & Hosking, J. (2009). KaitoroBase: Visual exploration of Software Architecture documents. *Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering,* 657-659. doi:10.1109/ASE.2009.26

Su, M. T., Hosking, J., & Grundy, J. (2011a). Capturing architecture documentation navigation trails for content chunking and sharing. *Proceedings of the 2011 9th Working IEEE/IFIP Conference on Software Architecture (WICSA),* 256-259. doi:10.1109/wicsa.2011.41

Su, M. T., Hosking, J., & Grundy, J. (2011b). KaitoroCap: A document navigation capture and visualisation tool. *Proceedings of the 2011 9th Working IEEE/IFIP Conference on Software Architecture (WICSA),* 359-362. doi:10.1109/wicsa.2011.58

Su, M. T., Tempero, E., Hosking, J., & Grundy, J. (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,* 141-150. doi:10.1109/wicsa-ecsa.212.22

SurveyMonkey. (2015). *SurveyMonkey*. Retrieved 15 April, 2015, from http://www.surveymonkey.com/

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. doi: 10.1016/j.jss.2009.08.032

Tang, A., Liang, P., & van Vliet, H. (2011). Software Architecture documentation: The road ahead. *Proceedings of the 2011 9th IEEE/IFIP Working Conference on Software Architecture,* 252–255. doi:10.1109/WICSA.2011.40

Tasktop Technologies Inc. (2013). *Tasktop Dev for Eclipse*. Retrieved Sept 1, 2013, from http://tasktop.com/eclipse#benefits

The Eclipse Foundation. (2013). *Mylyn*. Retrieved Sept 15, 2013, from http://www.eclipse.org/mylyn/

Thomas, F., Jingwen, O., Gail, C. M., & Emerson, M.-H. (2010). A degree-of-knowledge model to capture source code familiarity. *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering,* 385-394. doi:10.1145/1806799.1806856

Tyree, J., & Akerman, A. (2005). Architecture decisions: demystifying architecture. *IEEE Softw., 22*(2), 19-27. doi: 10.1109/MS.2005.27

Wallis, P., & Thom, J. A. (1996). Relevance judgments for assessing recall. *Inf. Processing and Management, 32*(3), 273-286. doi: 10.1016/0306-4573(95)00061-5

Wexelblat, A., & Maes, P. (1999). Footprints: history-rich tools for information foraging. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems,* 270-277. doi:10.1145/302979.303060

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., & Wessln, A. (2012). *Experimentation in Software Engineering*: Springer Publishing Company, Incorporated.

Yahoo! Inc. (2013). *Flickr*. Retrieved Dec 1, 2013, from www.flickr.com

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Syst. J, 26*(3), 276-292. doi: 10.1147/sj.263.0276