ECASE'19

Leveraging Usage Data of Software Architecture Artefacts

Moon Ting Su, University of Malaya, Malaysia
John Grundy, Monash University, Australia
John Hosking, University of Auckland, New Zealand
Ewan Tempero, University of Auckland, New Zealand

PRESENTATION OUTLINE

- Introduction
- Leveraging Usage Data in SAKM
- Our KaitoroCap Tool
- Future Work

MOTIVATION

- How do I understand a complex SA?
- How do I figure out how to do XYZ with this complex SA and its implementing APIs, etc???
- Great big PDF(s), generic, for whole-of-SA knowledge, hard to find stuff, don't know which bits "useful" for own tasks, …
- What if we could use OTHERS navigation paths, recommended parts, comments, annotations, ratings of parts of the SA doc to help us???

INTRODUCTION

- Leveraging past usage data of SA artefacts could bring a number of benefits e.g.
  - most frequently accessed (could signify the most important artefact);
  - most frequently edited (could signify the most problematic artefact or centre of change).
  - other users use of document (could help others on similar tasks)

INTRODUCTION

- Types of usage data:
  - *Interaction data* - data generated from users' interaction with software artefacts such as selecting, clicking, traversing, opening, editing, …
  - *Annotation data* - additional information users attach to the software artefacts, such as ratings, comments, and tags

EXISTING WORK ON LEVERAGING USAGE DATA

Existing Work	Key Features
Computational Wear (CW) [1, 2]	Creates 'usage wear' on computational objects by making users' **interaction history** with the objects part of the objects.
Social Navigation (SN) [1, 3-10]	Uses 'usage wear' as **navigation trace/'footprint'** to help other users in navigating the same information space.
Collaborative Filtering (CF) [6, 11]	Uses annotations provided by users as **collaborative filters** to sieve the information to receive.
Social Filtering (SF) [12, 13]	Uses explicit collaboration enabled by exposed identities of a community's members, for **filtering**, and to refine SN to center on explicitly created social networks.
Wear-based Filtering (WBF) [14-16]	Combines CW and CF, and uses users' interaction history with source code elements to **filter the UI** of IDE.
Degree-Of-Interest (DOI) studies [17-22]	Emphasise the notion of task, uses a **user's own usage history** to filter the UI of IDE.

LEVERAGING USAGE DATA IN SAKM

- Additional information derived from usage data can provide assistance for:
  - **sense-making** of materials
  - **finding** information
  - **filtering** information
  - **exploring** an information space by following other users
- Providing such assistance in SAKM could enhance AK sharing and AK using.
- Usage data + users' profiles + projects' profiles
  - could be a valuable dataset for empirical research related to usage data of AK - to study how CW, SN, CF, SF, WBF can enhance AK sharing and AK application/using.

USAGE DATA CAPTURE AND USE

Interaction Data (to support CW, SN, WBF)	
Type of data	**Could be used to**
Number of visits	Deduce importance of artefact.
Low-level interaction events with artefacts (such as mouse, keyboard and window events)	Deduce time spent on artefact. Deduce importance of artefact.
Visitation sequence	Create visualization of exploration path . Show possible hidden relationships between artefacts.
Timestamp of visit	Same as previous row, and Show recently-visited artefacts. Show timeline of artefacts visit.
Number of changes (edit, delete)	Identify problematic/unstable/center of change artefact.

USAGE DATA CAPTURE AND USE

Annotation Data (to support SN, CF and SF)	
Type of data	**Could be used to**
Ratings	Capture scaled personal opinion on certain aspects of the artefacts (such as quality, relevance, importance, etc.).
Tags	Search artefacts. Group artefacts.
Comments	Capture mental note explicitly or more detailed personal opinion.
Ranking	Rank artefacts.
Vote	Gather collective preference.

USAGE DATA CAPTURE AND USE

User Profile (to support SN, CF and SF)

user ID, name, job title, experience, skills set and location.

Project Profile

project ID and name, start and end dates, members' IDs and roles in the project, application domain, and current status.

KEY REQUIREMENTS - CAPTURE

- Collect usage data of each user.
- Collect usage data for different levels of granularity of SA artefacts (for e.g. on a per element, per model, per page or per section basis).
- Separation of the content of the artefacts from the means of capturing usage data (for e.g. insert annotation fields dynamically alongside the artefacts).
- Collect usage data in machine-parsable format.

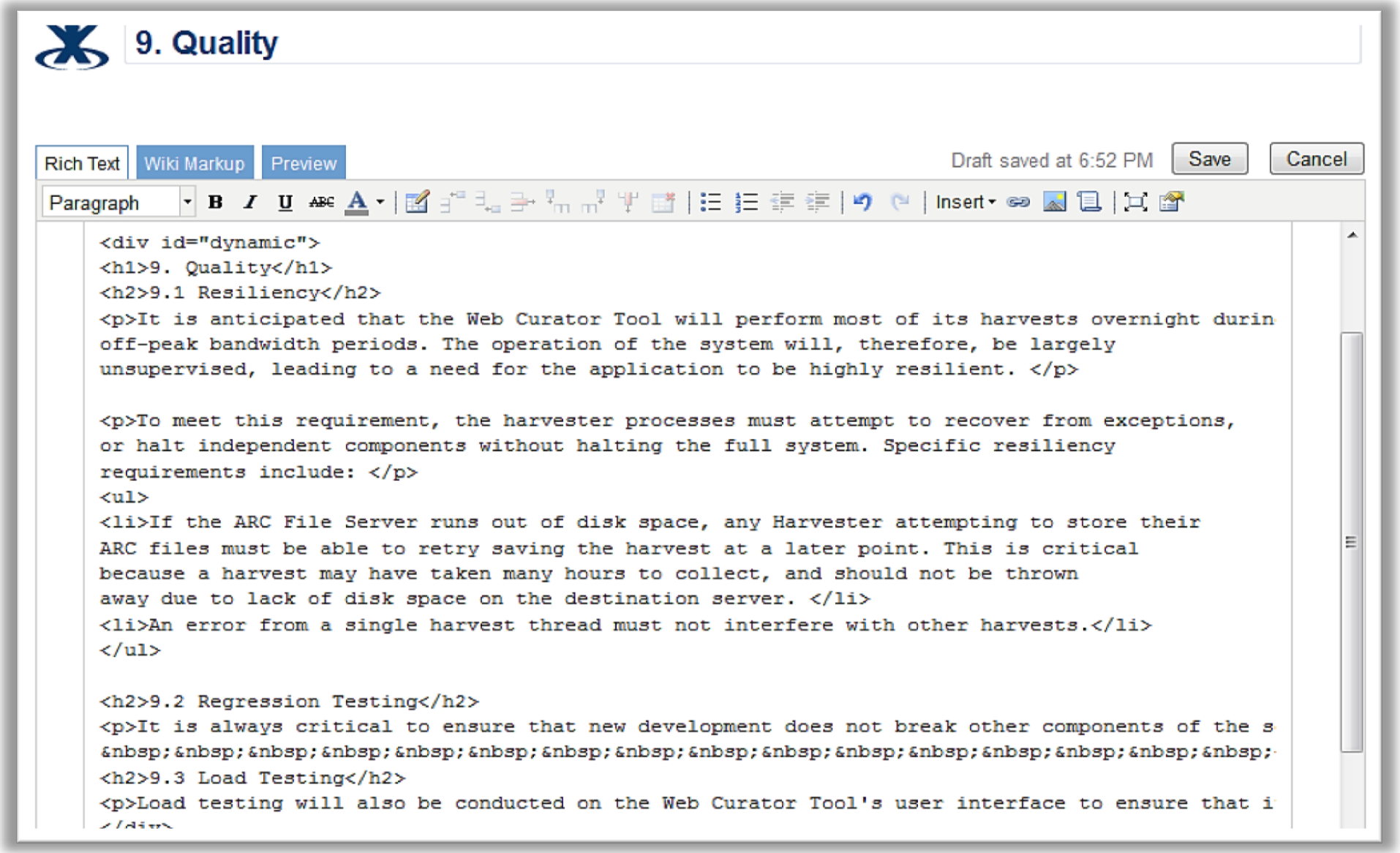
KEY REQUIREMENTS -VISUALIZATION

- Display usage data alongside artefacts (show number of visits or ratings directly).
- Create visualization that provides an overview of usage history of a group of artefacts (show number of visits as single graph, time line of visits, exploration path).
- Display raw usage data (number of visits and ratings) or in aggregated form (average ratings, % of users who visited).
- Automate analysis to produce aggregated usage data.
- Create alternative linkage structures between SA artefacts based on usage data (for e.g. linking SA artefacts based on exploration paths).

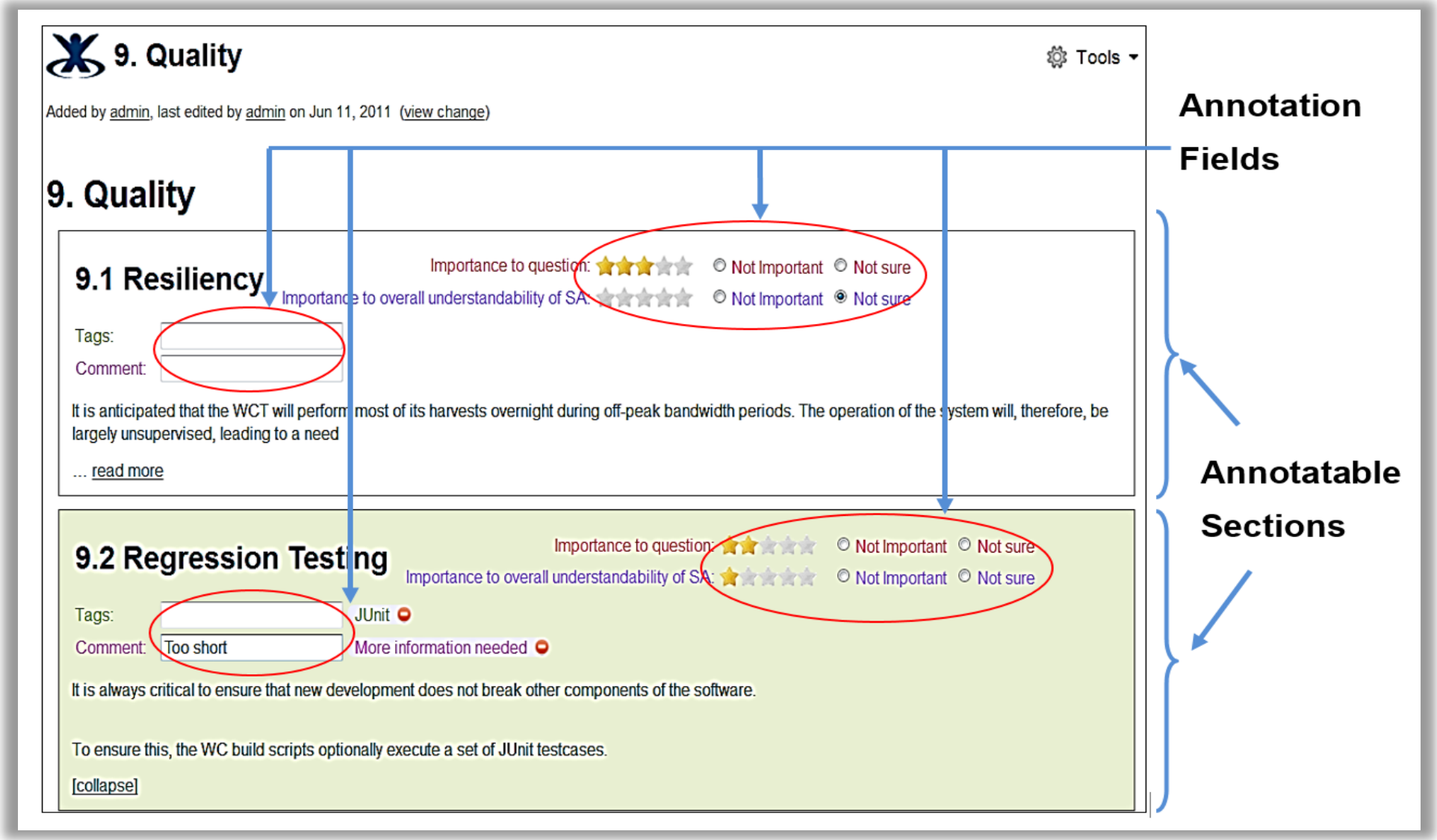
KAITOROCAP

- A plug-in for the Atlassian Confluence Enterprise Wiki that captures and provides visualization of the usage data of software architecture documents (ADs).
- Exploration paths in KaitoroCap serve as usage ‘wear’ left by previous consumers.
- Visible usage data (e.g. exploration paths, ratings, tags and comments) serve as information ‘traces’ supporting forms of social navigation.

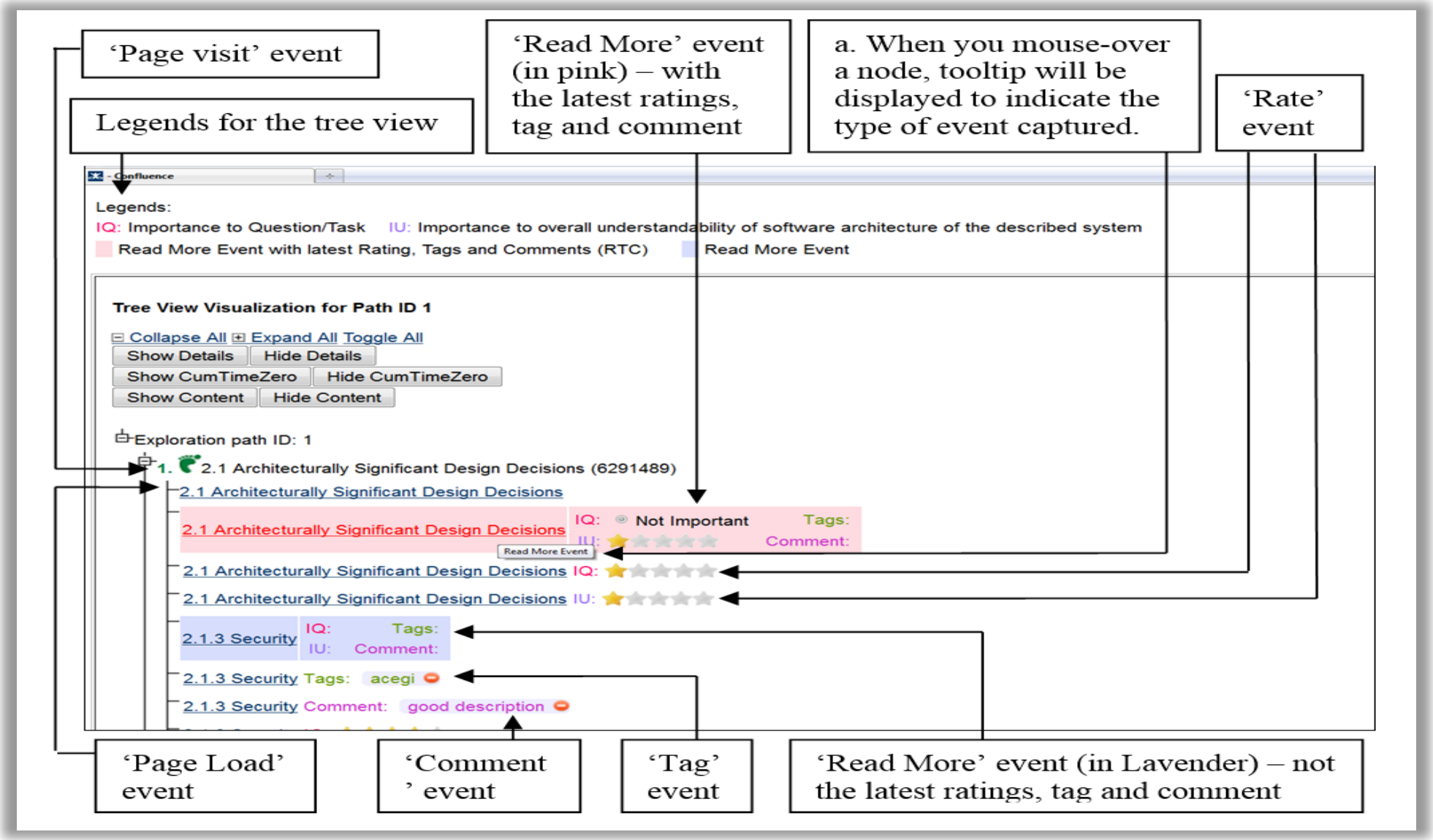
CREATING A NEW PAGE FOR AN ARCHITECTURE DESIGN



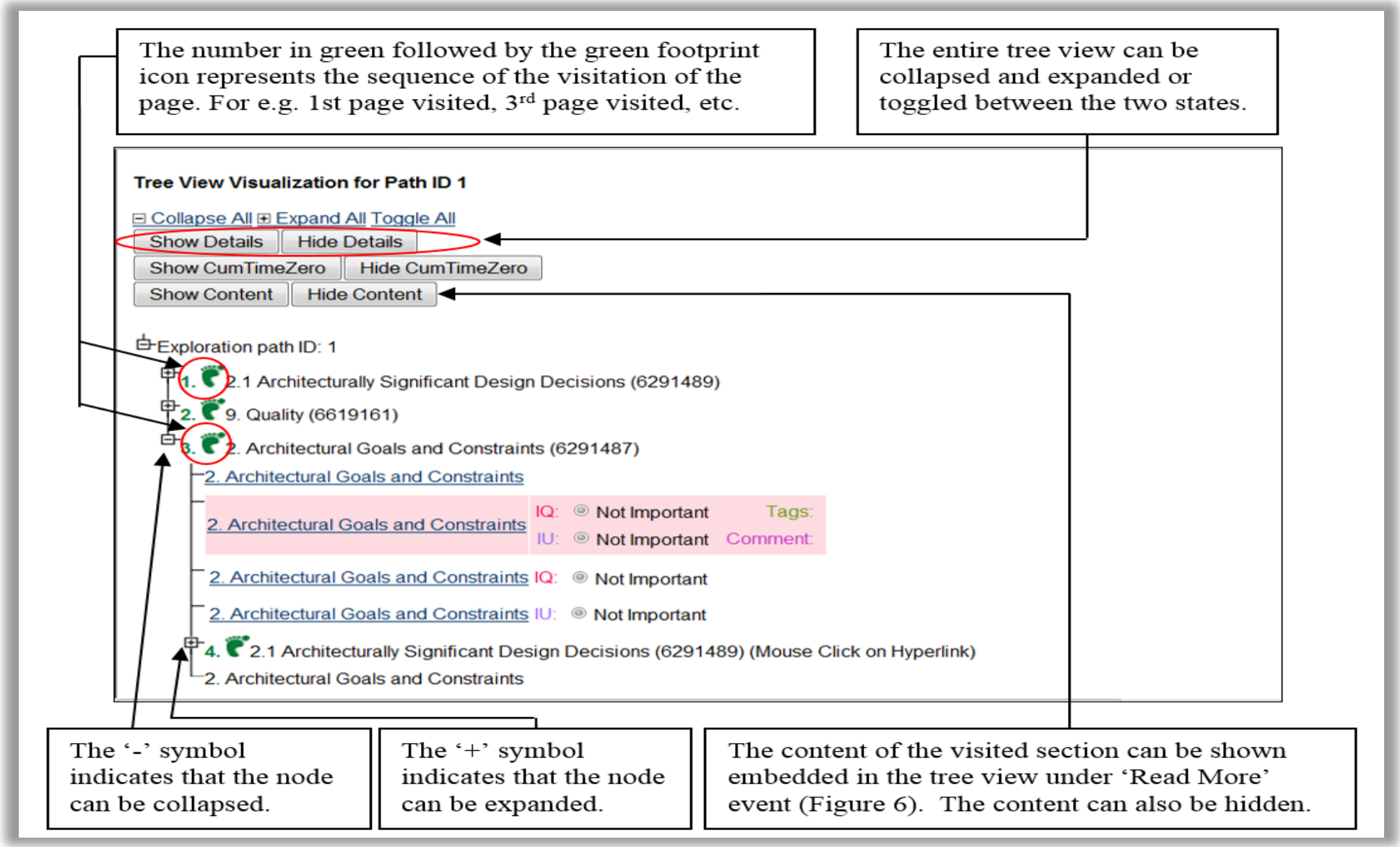
RESULTING PAGE WITH ANNOTATABLE SECTIONS CREATED DYNAMICALLY



TREE VIEW OF EXPLORATION PATH - PART A

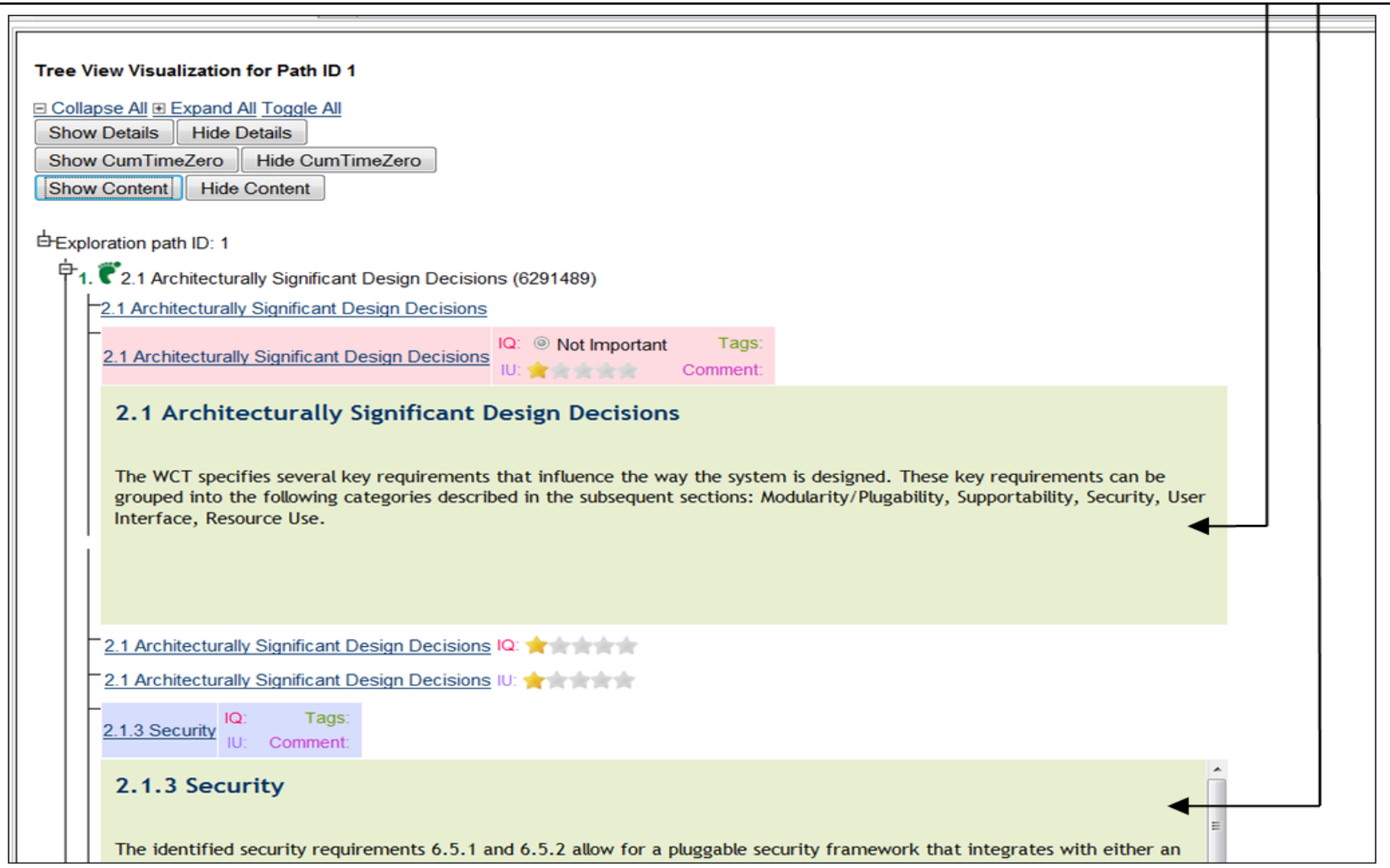


TREE VIEW OF EXPLORATION PATH - PART B

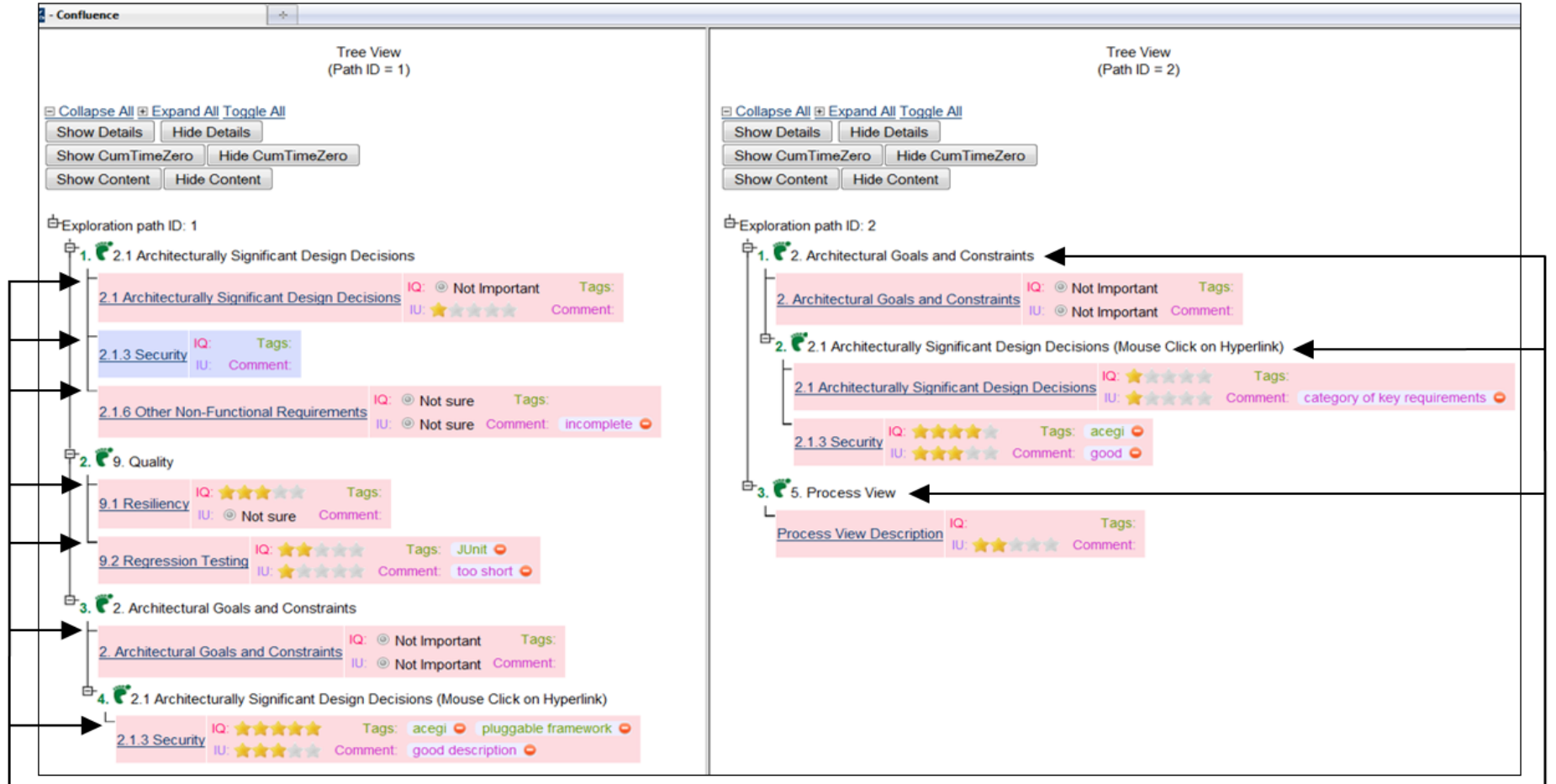


CONTENT OF THE VISITED SECTIONS EMBEDDED IN THE TREE VIEW

By clicking on ‘Show Content’, the content of the visited sections are shown under ‘Read More’ event.



COMPACT TREE-VIEW OF EXPANDED EXPLORATION PATHS



'Read More' events (pink or lavender in colour) with the ratings, tags and comment.

'Page Visit' events (number in green followed by the green footprint icon and page title)

KEY CHALLENGES

- Difficulty in identifying noise (generated by non-intentional actions) in interaction data.
  - Can verify some findings from interaction data with findings from annotation data (explicit conscientious judgements of users).
- Success is dependent on users of SAKM tools
  - Have to convince AK producers and consumers of the benefits leveraging usage data could bring and motivate them to use usage data capturing features.
- A robust framework
  - To reduce implementation overlap in different SAKM tools, needs to develop a robust framework containing generic usage data capturing and visualization functionalities that can be customized and extended by different SAKM tools.
  - Such a framework can be a part of the community-wide infrastructure for architecture-based software engineering.

CONCLUSION

- We propose including usage data capture and visualization as essential features of SAKM tools.
- Usage data can serve as a valuable unbiased dataset for empirical research in AK usage and support sharing of "usage" knowledge among team members.
- Future work:
  - Explore how leveraging further SA artefact usage data could provide further insights to improve SAKM tools.
  - Repository of usage / annotation data for set of SA docs

ACKNOWLEDGMENT

- University of Malaya and Ministry of Higher Education, Malaysia
- PReSS (No. 3624232), University of Auckland
- Software Process and Product Improvement for New Zealand Software Industry (UOAX0710)

Thank you
Q & A

REFERENCES

[1] W.C. Hill, J.D. Hollan, D. Wroblewski, and T. McCandless, "Edit wear and read wear", Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM, 1992, pp. 3- 9, doi:10.1145/142750.142751.

[2] W.C. Hill, and J.D. Hollan, " History-enriched digital objects: Prototypes and policy issues", The Inf. Society, vol. 10, no. 2, 1994, pp. 139-145.

[3] P. Dourish, and M. Chalmers, "Running out of space: Models of information navigation", Proc. Human Computer Interaction '94, ACM Press, 1994.

[4] A. Dieberger, P. Dourish, K. Höök, P. Resnick, and A. Wexelblat, "Social navigation: techniques for building more usable systems", interactions, vol. 7, no. 6, 2000, pp. 36-45, doi:10.1145/352580.352587.

[5] A.J. Munro, K. Hook, and D. Benyon, "Footprints in the snow", in Social navigation of information space, A.J. Munro, K. Hook, and D. Benyon, Eds. Springer, 1999, pp. 1-14.

REFERENCES

[6] J.B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems", in The adaptive web, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer-Verlag, 2007, pp. 291-324.

[7] M. Svensson, K. Höök, J. Laaksolahti, and A. Waern, "Social navigation of food recipes", Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM, 2001, pp. 341-348.

[8] A. Dieberger, and M. Guzdial, "CoWeb - experiences with collaborative web spaces", in From Usenet to CoWebs, C. Lueg, and D. Fisher, Eds. Springer, 2003, pp. 155-166.

[9] P. Maglio, and R. Barrett, "Intermediaries personalize information streams", Communications of the ACM, vol. 43, no. 8, 2000, pp. 96-101, doi:10.1145/345124.345158.

[10] A. Wexelblat, and P. Maes, "Footprints: history-rich tools for information foraging", Proc. SIGCHI conference on Human Factors in Computing Systems, ACM, 1999, pp. 270-277.

REFERENCES

[11] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", Communications of the ACM, vol. 35, no. 12, 1992, pp. 61-70.

[12] K. Lerman, "Social information processing in news aggregation", IEEE Internet Computing, vol. 11, no. 6, 2007, pp. 16-28.

[13] SmugMug Inc., "Flickr", 2019. www.flickr.com.

[14] R. DeLine, A. Khella, M. Czerwinski, and G. Robertson, "Towards understanding programs through wear-based filtering", Proc. 2005 ACM Symposium on Software Visualization, ACM, 2005, pp. 183-192, doi:10.1145/1056018.1056044.

[15] U. Shardanand, and P. Maes, "Social information filtering: algorithms for automating "word of mouth"", Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing, 1995, pp. 210-217, doi:10.1145/223904.223931.

REFERENCES

[16] R. DeLine, M. Czerwinski, and G. Robertson, "Easing program comprehension by sharing navigation data", Proc. 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, IEEE, 2005, pp. 241-248, doi:10.1109/VLHCC.2005.32.

[17] M. Kersten, and C.M. Gail, "Mylar: a degree-of-interest model for IDEs", Proc. 4th International Conference on Aspect-oriented Software Development, ACM, 2005, pp. 159-168.

[18] M. Kersten, and C.M. Gail, "Using task context to improve programmer productivity", Proc. 14th ACM SIGSOFT International Symposium on Foundations of SE, ACM, 2006, pp. 1-11.

[19] The Eclipse Foundation, "Mylyn", 2019. http://www.eclipse.org/mylyn/.

[20] F. Thomas, O. Jingwen, C.M. Gail, and M.-H. Emerson, "A degree-of knowledge model to capture source code familiarity", Proc. 32nd ACM/IEEE International Conference on Software Engineering, ACM, 2010, pp. 385-394, doi:10.1145/1806799.1806856.

REFERENCES

[21] R. Elves, "Tasktop for Eclipse - Get more out of Mylyn", 2008. https://dzone.com/articles/more-productive-programming-wi.

[22] Tasktop Technologies Inc., "Tasktop Dev for Eclipse", 2013. http://tasktop.com/eclipse#benefits.

[23] M.A. Babar, I. Gorton, and R. Jeffery, "Capturing and using software architecture knowledge for architecture-based software development", Proc. Fifth International Conference on Quality Software, IEEE, 2005, pp. 169-176, doi:10.1109/QSIC.2005.17.

[24] R. Weinreich, and I. Groher, "Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review", Inf. Softw. Technol., vol. 80, 2016, pp. 265-286.

[25] Z. Li, P. Liang, and P. Avgeriou, "Application of knowledge-based approaches in software architecture: A systematic mapping study", Inf. Softw. Technol., vol. 55, no. 5, 2013, pp. 777-794.

REFERENCES

[26] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M.A. Babar, "10 years of software architecture knowledge management: Practice and future", J Syst. Softw., vol. 116, 2016, pp. 191-205, doi: 10.1016/j.jss.2015.08.054.

[27] M. Shahin, P. Liang, and M.A. Babar, "A systematic review of software architecture visualization techniques", J Syst. Softw., vol. 94, 2014, pp. 161-185, doi: 10.1016/j.jss.2014.03.071.

[28] J.F. Hoorn, R. Farenhorst, P. Lago, and H. van Vliet, "The lonesome architect", J Syst. Softw., vol. 84, no. 9, 2011, pp. 1424-1435.

[29] Sherman S., and U.-S. N., "What Do Software Architects Think They (Should) Do?", in Advanced Information Systems Engineering Workshops. CAiSE 2014. Lecture Notes in Business Information Processing, Iliadis L., Papazoglou M., and P. K., Eds. Springer, Cham, 2014, pp. 219-225.

[30] S. Sherman, and I. Hadar, "Toward Defining the Role of the Software Architect", Proc. 2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering, 2015, pp.71-76, doi:10.1109/CHASE.2015.17.

REFERENCES

[31] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, and M.A. Babar, "A comparative study of architecture knowledge management tools", J Syst. Softw., vol. 83, no. 3, 2010, pp. 352-370.

[32] F. Nickols, "The knowledge in knowledge management", in The knowledge management yearbook 2000-2001, J.A. Woods, and J. Cortada, Eds. Butterworth-Heinemann, 2000, pp. 12-21.

[33] J.R. Anderson, Cognitive Psychology and its implications, W.H. Freeman and Company, 1995.

[34] M.T. Su, J. Hosking, and J. Grundy, "Capturing architecture documentation navigation trails for content chunking and sharing", Proc. 2011 9th Working IEEE/IFIP Conference on Software Architecture (WICSA), IEEE, 2011, pp. 256-259.

[35] M.T. Su, J. Hosking, and J. Grundy, "KaitoroCap: A document navigation capture and visualisation tool", Proc. 2011 9th Working IEEE/IFIP Conference on Software Architecture (WICSA), IEEE, 2011, pp. 359-362, doi:10.1109/wicsa.2011.58.

REFERENCES

[36] M.T. Su, J. Hosking, J. Grundy, and E. Tempero, "Usage-based chunking of Software Architecture information to assist information finding", J Syst. Softw., vol. 122, 2016, pp. 215-238.

[37] M.T. Su, "Supporting Information Searching in Software Architecture Documents (PhD Thesis)", Department of Computer Science, University of Auckland, Auckland, New Zealand, 2014.

[38] Atlassian, "Atlassian Confluence", 2018. https://www.atlassian.com/software/confluence.

[39] The jQuery Foundation, "JQuery", 2019. http://jquery.com/.

[40] P. Liang, and P. Avgeriou, "Tools and technologies for architecture knowledge management", in Software Architecture knowledge management, M.A. Babar, T. Dingsøyr, P. Lago, and H. van Vliet, Eds. Springer, 2009, pp. 91-111.

[41] A. Jansen, P. Avgeriou, and J.S. van der Ven, "Enriching software architecture documentation", J Syst. Softw., vol. 82, no. 8, 2009, pp. 1232-1248, doi:10.1016/j.jss.2009.04.052.