

Investigating the Use of Clues for Scaling Document-Level Concern Graphs

Elisa Baniassad, Siobhan Clarke

Distributed Systems Group,
Department of Computer Science,
Trinity College Dublin
+353 (1) 608-2354

{Elisa.Baniassad, Siobhan.Clarke}@cs.tcd.ie

ABSTRACT

As research into early-lifecycle aspect-orientation deepens, it makes sense that requirements documentation should be examined to help identify concerns. Among others [AORE, BM-EA02, Sutton-EA03], Theme/Doc sought to exploit the text of requirements documentation to help a designer identify concerns and the relationships between concerns [Theme-ICSE]. The Theme/Doc tool created graphs of the relationships between concerns and the requirements that mentioned those concerns. The Theme/Doc approach worked on the very simple principle that if two concerns were mentioned together in a requirement, the designer should either separate them, or formalize their relationship. The only *clue* the tool exploited was that concern-terms appeared together in a requirement. It provided no summarization or interpretation of the text whatsoever. Despite the simplicity of the technique, it provided help and direction to developers attempting to identify a relevant set of concerns in their requirements as preparation for design [Theme-ICSE].

However, Theme/Doc provides no special support for large sets of requirements. As the number of concerns and requirements increases, so does the size of the view. Here we discuss initial investigations into using several latent clues in requirements documentation to help scale document concern-views. This paper does not provide any generalizable evidence to support its conjectures. Instead, it is intended to highlight open questions.

1. INTRODUCTION

As research into aspect-orientation moves into the “early” realm, it makes sense that requirement documentation should be examined to help identify concerns. Among others [Sutton-EA02, BM-EA02, AORE], Theme/Doc sought to exploit the language used in requirements documentation to help a designer identify concerns and their relationships [Theme-ICSE]. Theme/Doc provided designers with a way to see the relationships between concerns and the requirements that described those concerns.

One of the main troubles with Theme/Doc is that it seems to lack scalability. Theme/Doc provides no special support for large sets of requirements. As the number of concerns and requirements increases, so does the size of the view.

The reason we say that it “seems to” lack scalability is because it’s debatable whether the so-called “action view” of Theme/Doc really needs to scale. As you can see in Figure 1, a Theme/Doc action view is made up of two types of nodes: concerns, and requirements. Each requirement is linked to all the concerns that it mentions. Using the spring-style layout shown here, it is possible to see which concerns are related. Relationships indicate *tangling* between concerns.

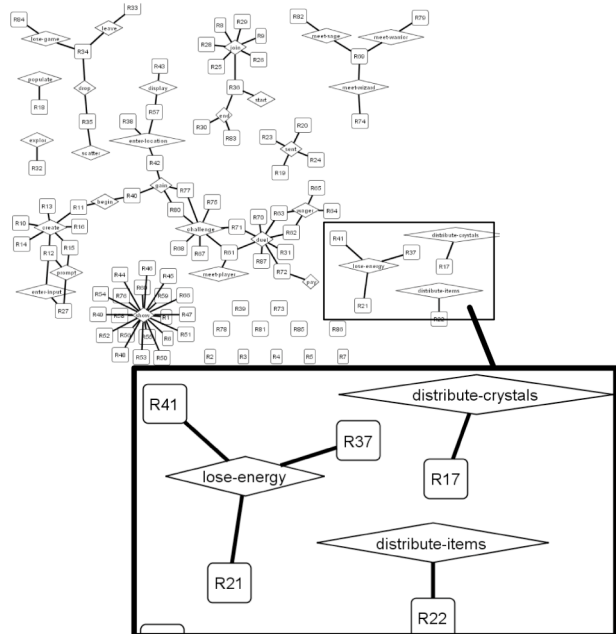


Figure 1 Theme/Doc Action view. *Concerns are diamonds, Requirements-sentences are ellipses, links indicates the mention of a concern in a requirement.*

Since a designer is, ultimately, trying to separate concerns, this tangling should be avoided or contained. If two concerns are shown in the action view to be related, it is up to a designer to determine if they *should* be related; perhaps the requirement is badly written, and could be re-written so that the two concerns are no longer linked. However, if the requirement is well-written and there’s no way to disentangle the concerns, then the designer needs to make a decision about how to formalize the relationship: Is one an aspect of the other? Is one a sub-concern of the other?

In terms of scalability of the views, it makes sense that a developer might want to see a less hairy view, but they are going to need to deal with each shared requirement anyway, regardless of how they are presented with the task.

The instinct, though, is to theorize that there are some “higher-level” (perhaps system-wide, or more-abstract) decisions that could be made were a more summarized view available; that perhaps by providing some scalability in these views that higher-level implementation aspects could be captured before needing to deal with lower and more local ones.

Theme/Doc worked on the very simple principle that if two concerns were described together in a requirement, the designer should either separate them, or formalize their relationship. It provided no summarization or interpretation of the text whatsoever. The only textual *clue* it exploited was that

is specially coloured to allow at-a-glance differentiation of concern-clusters.

Because the edge-labels on the concern-clusters are difficult to read, this view has been annotated to clarify the identity of three of the clusters: “display”, “meet-wizard” and “join”. Display is highly connected to two other concerns, whereas join is isolated: its requirements sentences each describe only one concern.

This view is likely not helpful for associating requirements with particular concerns. For one thing, just as a coincidence of layout, it is not always clear which requirements describe more than one concern. If the requirements are close to the edge of a concern-cluster then it is easy to spot, however, if the nodes are positioned like those in the circled area in the middle of the screen (labelled with question marks) it is more difficult to see.

However, this view is helpful in terms of seeing how widely a concern is described. Display is mentioned in many places, whereas meet-wizard is just described in three sentences. This information could reflect the relative importance of these concerns as early concerns. Perhaps less-described concerns should be targets for grouping with other such concerns. Or maybe concerns that are disproportionately large (such as the display concern) should be targeted for a split.

Additionally, since display affects more sentences than meet-wizard, it could be inferred to be more crosscutting. Since crosscutting concerns are aspects, perhaps display is a better potential aspect than meet-wizard. The cautionary note with this line of reasoning, however, is that detailed or extensive description in requirements doesn't necessarily translate to more complex or extensive design. Display might *look* like a good aspect since it crosscuts the requirements documentation, but in reality it might work much better using another kind of module altogether. More investigation into how complexity in the requirements translates to design and implementation is needed.

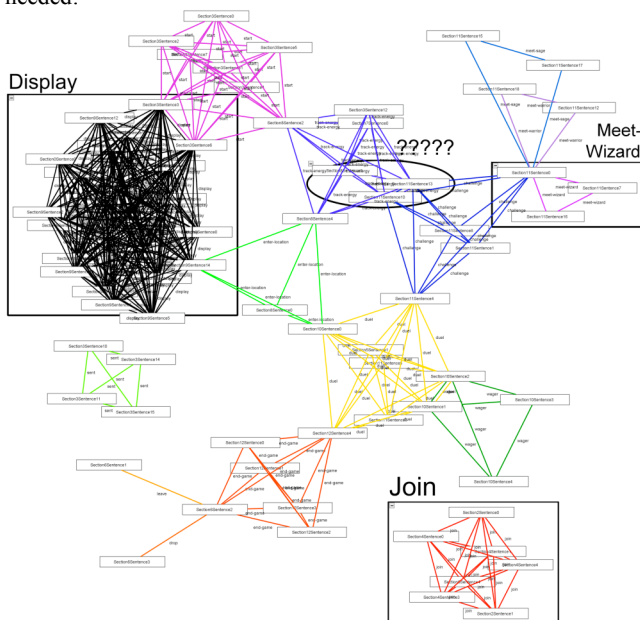


Figure 3 Degree of Inter-Requirement Crosscutting: Sentences related by Concerns. Nodes are sentences, edges concerns. Edge colours and labels differentiate concerns

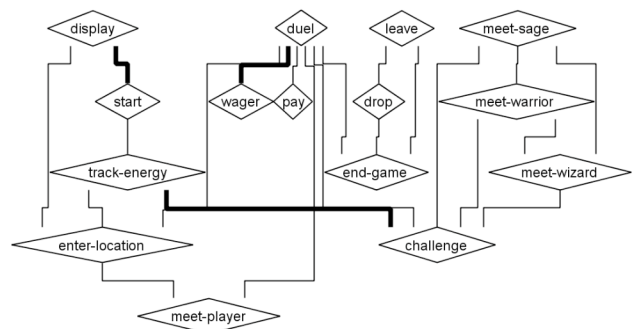


Figure 4 Concern Relatedness Graph. Concerns are linked by and weighted based on the number of shared requirements sentences. Concerns that are isolated (such as the join concern) are not shown.

5. CLUE: CONCERN-RELATEDNESS

Another latent clue in the text is how often concerns are mentioned together. Figure 4 shows the concern-relatedness graph for the crystal game. All concerns are related to the same degree except for three pairs: “display” and “start”, “duel” and “wager”, and “track-energy” and “challenge”. The requirements associated with display and start, describe the user interface issues related to starting the game. The duel/wager requirements describe how bets are placed on duels between players, and the track-energy/challenge requirements discuss how a player’s energy level is altered depending upon whether a challenge is met or failed.

Different action could be taken when considering each of these pairs: duel and wager could be merged into just a duel concern, display and start could conceivably be split into a starting-UI concern, and no action would likely be taken for the track-energy/challenge pair. However, these three solutions might just as well be used on pairs that share only one requirement: developers may chose to merge, split-out or keep any of these concerns based on the relationships between them regardless of the strength of those relationships. It is not clear, from this case study, whether such a clue would be useful, or if it would, how.

This kind of view may, however, be more useful when used with a more formal set of requirements. The crystal game requirements were written with no real attention to the scope of any one requirement. They are very heterogeneous in terms of what they describe, and the level of abstraction at which they describe it. It is not even ensured that a requirement that is subtly referring to a concern will mention it explicitly. If provided with a more carefully constructed set of requirements, degree of coupling between the concerns could be helpful in knowing whether concerns chosen were appropriate.

6. CLUE: DOCUMENT SECTIONS

One of the most tantalizing ideas for scalability is looking at how concerns relate across document sections. By “section”, we mean any partial set of requirements blocked together. A section could be as large as one document in a set of documents, or it could be as small as a paragraph describing one portion of a use case.

The motivation for using existing document structure would be to allow a developer to consider inter-section concerns before considering intra-section concerns.

Questions obviously arise as to whether inter-section concerns are in fact higher-level concerns than intra-section concerns. If each section of a set of requirements describes functionality

related to a subsystem, then seeing that there are concerns spanning sections would be very interesting and potentially important. A developer may want to give special consideration to concerns that relate portions of a system that were either thought to be unrelated, or whose relationships are carefully monitored. However, if sections do not reflect the modularity of the implementation, or if the document is badly written, then the inter-section concerns may be an uninteresting subset of the intra-section ones.

6.1 Relationships Between Sections

A view like the one in Figure 5 could be used to assess the degree of crosscutting between document sections. The graph in Figure 5 shows each section as a node. If two sections both describe a concern, an edge extends between them. The edge is coloured and labeled to differentiate which concern forms the link. We can see, for instance, that Sections 2, 4, and 6 all mention the “join” concern.

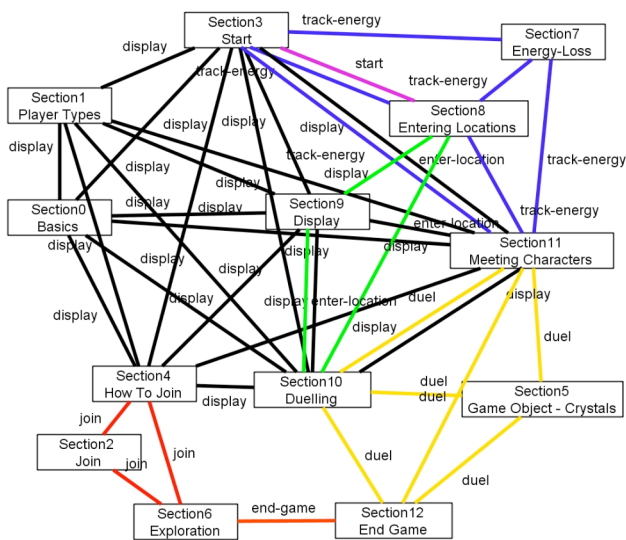


Figure 5 Degree of Inter-Section Crosscutting: Sections Related by Concerns. Concerns are shown as edges and are differentiated by labels and colours. Nodes are Sections, which are numbered and labeled by topic

inside sections: Join is split between sections, though it at the sentence-level it is completely isolated.

All of the concerns in Figure 5 were present at the sentence-relationship level (in Figure 3), but some are missing, indicating that they are intra-section relationships. Note that there is no meet-wizard concern. It is absent from this graph since it is only described in one section.

The requirements documentation for the crystal game was split into features such as “Duelling”, and “Entering Locations”. Each of these features was enclosed in a section. This approach has revealed the track-energy concern that has been identified using other means as an aspect of the crystal game. As can be seen in Figure 5, “track-energy” crosscuts several sections: “Start”, “Entering Locations”, “Meeting Characters”, and the “Energy Loss” section that sets up the main concept of energy.

The fact that there is already an Energy Loss section suggests that the text related to energy in the other sections was, in fact, misplaced except perhaps as context for the reader. However, had all the track-energy related sentences been moved into the Energy Loss section, they would merely have dislocated other, intra-section concerns. The character meeting related concerns (such as meet-wizard) would then have been partially located in the Meeting Characters sections, and in Energy Loss. Perhaps the fact that track-energy is tangled at both the section and the requirement level indicates that it is a good candidate for “aspecthood”.

Another example of misplaced text can be seen in Figure 6. The join concern spans three sections. This is in contrast to its associations at the requirement level (Figure 2) where it is completely isolated. This isolation is also evident in Figure 6, which shows that while requirements for join are found in three sections, that they have no relationships to other requirements in those sections. In this case, this technique has highlighted a definite need for document re-structuring. Only one requirement related to join sits in the Exploration section, and, upon further inspection, it is not overly related to exploration. Furthermore, the Join and How To Join sections could and should be merged.

6.2 Concerns Related By Sections

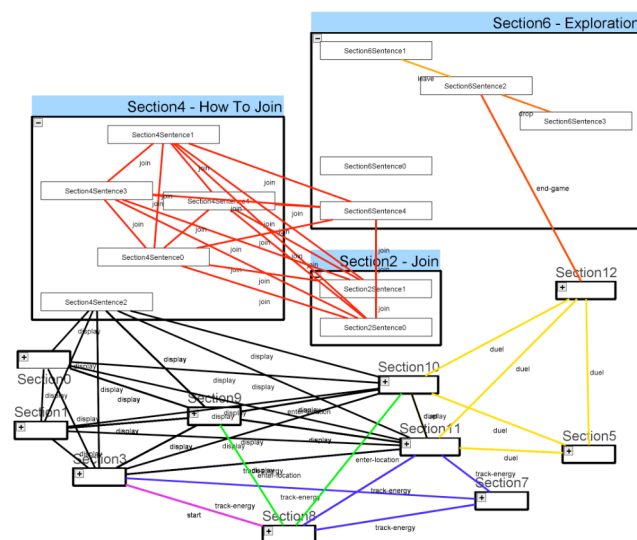


Figure 6 Sections Related by Concern: Join Concern Exposed. Three enlarged sections display a partial Sentence Level View (from Figure 3). They reveal mixing of concerns

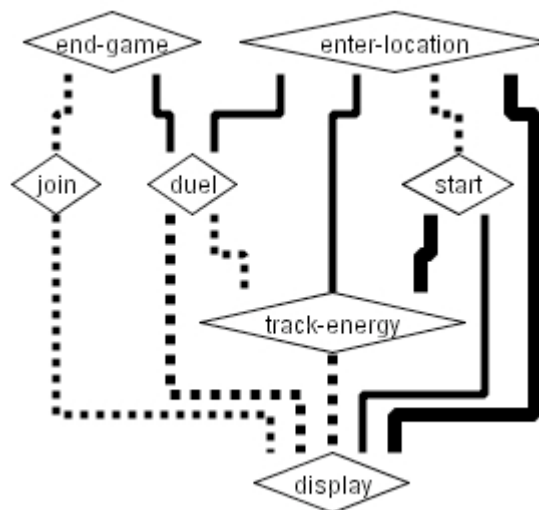


Figure 7 Concerns Related by Section Directness. Edges weighted based on shared sections, and dotted if no direct sentence relationship.

Just as it was useful at the requirement-level to see how concerns relate, it may also be of use to see how concerns relate at the section-level. Figure 7 shows the relationships between inter-section concerns in the crystal game. If two concerns are described in a section then a link is drawn between the two.

The weight of links suggests the number of sections shared between the concerns. Enter-location and display hold the highest number of sections in common. Dashed lines indicate that while two concerns may share a section, they do not also share individual requirements. For instance, Join and end-game both appear in Section 6, but as Figure 4 shows, there is no requirement-level link between the two. Without the dashes, the summarization might be overly simplistic, and may suggest that two concerns share a relationship when in fact they do not.

The main value of this view seems to be that it is substantially more compact than the original action view. Much of the information found in that graph is obscured, but it is accessible. For instance, one can expand one or more concerns (or several concerns) to reveal the sections to which they relate. The expansion of the track-energy concern is shown in Figure 8. A particular section can then be further expanded to show the intra concerns, as Section 8 has been. It would also be straightforward to expand a link (or a collection of links) between two of the concerns to reveal the appropriate sections.

Using this high-level concerns graph as a basis, a developer could browse through the views depending on the task of interest. If identifying aspects was the goal, then expanding section by section and looking at the associations of requirements might be appropriate. Or, a developer could use this view to question the document structure, and drive document-level refactorings like that suggested for the join concern by targeting the dashed-edges. Additional validation would be needed to determine whether a developer could satisfactorily track their progress when working at multiple levels in this way, however.

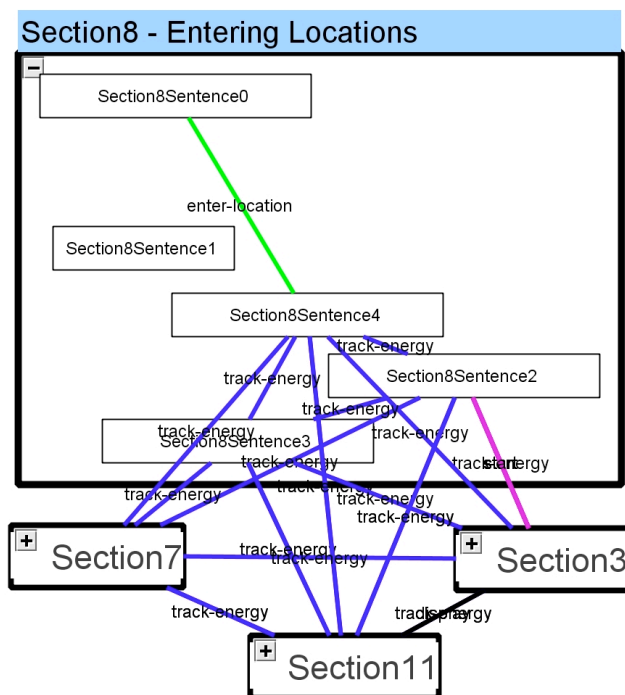


Figure 8 Zoom-in on the Track-Energy concern at the section-interaction level

7. OTHER WORK THAT MIGHT HELP

7.1.1 Concern Relatedness

[Sutton-EA02] looks at concerns in several requirements and use-cases. In that work, Sutton identifies different types of concerns than were identified for the crystal game, including classes, classifications, and properties. Here, we have mainly concentrated on “active” concerns in the crystal game, though this was not dictated by the Theme/Doc approach. It is possible that this more inclusive kind of concern would help with scalability of the Theme/Doc approach. Different concerns might be interesting in different contexts or at different scales. Sutton, for instance, finds many shared concerns between documents used in his case study. The concerns that are shared most generally seem to be mainly structural in nature. It would be interesting to know how including structural concerns for the crystal game would affect inter-section concern relationships.

Brito and Moreira [BM-EA02] introduced work on composition of aspect-oriented requirements. There, they described how to use composition semantics to essentially weave abstract requirements to concrete ones. Here, we went over how both the way in which requirements are written as well as their placement within a requirements document can cause tangling of concerns. Some concerns (like the join concern) might be easily untangled simply by moving the relevant requirements into a section by themselves. However, other concerns, such as the track-energy concern, are tangled both at the requirement and at the section level. It might be interesting to explore how using requirement-level composition could help separate concerns even when they are tangled within a requirement. Perhaps some requirements could have abstract portions that could then be concretised using an approach similar to that expressed in [BM-EA02].

7.1.2 Semantic Analysis

Other work that might give insight into how to locate different levels of concerns in documentation is research into textual analysis for the sake of knowledge discovery. Loosely described, their work involves finding *topics* in text [ND-Topics, Multidoc]. Techniques such as these use data mining or statistical approaches to identify clusters of topics over certain ranges of documentation.

In Latent Semantic Analysis (LSA) [DS-90], relatedness of terms is determined based on how often they appear together in a corpus of text. Such an approach would clearly lend itself to application for generation of requirement-level concern relatedness graphs. In a graph where each concern-term is displayed individually, zooming in could show very light relationships, where as zooming out would reveal only the stronger relationships (and hence less detail).

7.1.3 Visualization Techniques

Continuous fish-eye style zooming [LB-1995, MS-1995] is a layout technique where the focus point of a display is zoomed-in more closely than the areas surrounding it; surrounding areas show decreasing zoom levels. This technique has been applied in software engineering tools [MS-1998], as a way to help developers remain oriented within a large information space, such as an extensive class diagram, or call-graph. It is likely that a display technique like continuous zooming could be integrated into the graphical layout of graphs such as those described here. Some level of fish-eye zooming is already evident in Figure 6, in which certain portions of the graph are shown in more detail than others. However, applying a simple -to-use continuous zoom, where the user could expand and contract portions of the graph through mouse-gestures may greatly enhance the usability of the technique.

There is currently an approach, Covise (Concern Visualizer) under development [JM-04] that will allow developers to view implementation-level concern relationships. Covise is intended for integration with the Concern Manipulation Environment [CME]. It depicts concerns as spheres, and shows connections between them. A developer can click on a sphere to find more information about it, and can navigate through and rotate the (also spherical) concern space. Since it is as yet unavailable for use, we are not yet aware what the relationships between the concerns represent, or what determines the placement of a concern-sphere inside the concern space. However, such an approach would be an interesting starting point for a three-dimensional depiction of requirements-level concern-relationships.

8. FINAL THOUGHTS

This paper has described the use of three latent clues found in text that might be useful when considering document-level aspects: degree of crosscutting, concern relatedness, and document structure. It described several views that could exploit those clues. In conclusion, we will highlight some of the questions that arose in writing this paper.

How dependent is this upon “Good Writing”?

Any view that exploits latent clues in natural text relies heavily on that text being well written and well structured. This was true to a small extent in Theme/Doc, since it relied on the writing of individual requirements. Part of the task of using Theme/Doc to identify concerns was checking whether each requirement was well-written. In this paper we have presented views that rely not just on requirement-text, but also on the placement of requirements within document sections. This “zooming-out” also zooms out the reliance on writing: now well-placedness of requirements becomes an issue. However, just as Theme/Doc provided a framework within which to consider the writing of requirements, these section-level views can provide a starting point for re-structuring documentation to arrive at a better concern-to-section relationship.

Are Inter-Section concerns actually “higher level” in the system sense?

As was discussed, the “high-level” nature of inter-section concerns is highly dependent upon how the requirements are structured. If requirements are organized by architectural component, for instance, then concerns that relate sections might be higher-level than those that are within a section. That is not to say that they might be more abstract, but just that they would deal with more broad systemic issues.

As was pointed out when looking at the track-energy concern’s relationship to multiple sections, there was no way to arrange requirements within sections such sections 3, 7, 8, and 11 could be unlinked. In section 11, for instance, track-energy is linked directly to the challenge concern. Moving track-energy into Section 7 (the “Energy Loss” section) would have caused challenge to become an inter-section concern. The suggestion was made that since track-energy is both tangled at the section and the requirement level that it could be a *real* inter-section concern (while those tangled due to bad placement would not). However, what to infer from that is as yet un-researched.

Another avenue of investigation would be to look at concern-relationships between use cases for a system. It is possible that, since use cases represent behavioral scenarios, the concerns that relate them might be more meaningful than those linking (perhaps) arbitrarily chosen sections.

Differing Affordances?

It is possible that some views have different affordances², that is to say that they naturally lend themselves to particular uses, based on the information they display, and the way in which they display it. Some views may lend themselves to tasks involving document re-organization (Figure 6, for instance), whereas others might be more perceived as a way to look at conceptual relationships between portions of a requirements set (Figure 7). It would be interesting to perform experimental analysis to compare the affordances of the graphs in terms of their uses for software engineering tasks, and for aspect-identification.

Do these views affect traceability?

In [Theme-ICSE04] we claimed that the Theme views would help traceability since requirements could be identified in Theme/Doc views, which corresponded directly to Theme/UML models, which would then relate to implementation. The views presented here do not significantly affect this level of backward-traceability. Although summarization is used (particularly in Figure 7 and Figure 8), it is still possible to access which requirements contributed to the links and nodes forming the displays. Forward-traceability would need more investigation, since in the refined Theme/Doc action view, all themes relate to Theme/UML themes that can then be implemented. The summarized view in Figure 7 will not have a 1:1 relationship between concerns and Theme/UML models. This view may need to be used in combination with more detailed views to achieve full forward-traceability.

Do these views help scalability?

This paper also looked at how certain clues might help with scalability of the Theme/Doc action view. The initial worry, expressed at the start of the paper, was that the fine granularity of the action view was needed to help guide the developer through the requirements analysis process. Further validation is needed to determine whether views such as those introduced here would help or hinder that flow. Additionally, this work has not looked at a larger set of requirements on which to test these theories. The next step will be to try out these views with a truly large set of requirements in a more experimental setting, and test the hypotheses presented here. Finally, the graphs presented in this paper will also suffer from scalability issues. If a document contained thousands of sections, then the section-level graphs would also become intractably large. More research is needed into mechanisms such as continuous zooming and other visualization techniques to achieve full-scalability of these views.

9. ACKNOWLEDGEMENTS

Thanks to Ryan van Roode for his help on strategies for lexical-scalability. Also, thanks to the anonymous reviewers, who contributed several interesting ideas found in the paper.

10. REFERENCES

- [AORE] Rashid, A., et al. Early Aspects: A Model for Aspect-Oriented Requirements Engineering. IEEE Joint International Requirements Engineering Conference.
- [LB-1995] L. Bartram, A. Ho, J. Dill, and F. Henigman. The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces. In ACM Symposium on User Interface Software and Technology, pages 207–215, 1995.

² J. J. Gibson (1977, 1979): a situation where an object's sensory characteristics intuitively imply its functionality and use.

- [BM-EA02] I. Brito, A. Moreira. Towards a composition process for aspect-oriented requirements. Early-Aspects Workshop at AOSD2002.
- [CME] <http://www.research.ibm.com/cme/>
- [DPRG-ICSE] E. Baniassad, G. Murphy, C. Schwanninger. Design Pattern Rationale Graphs: Linking Design to Source. ICSE2003.
- [JM-04] Covise: Concern Visualizer. 3-Dimensional Concern Manipulation Environment User Interface. Developed by the Fraunhofer Institute, Juri Memmert and Jan Wloka.
- [Sutton-EA02] S. Sutton Jr. Concerns in a requirements model - a small case study. Early-Aspects Workshop at AOSD2002
- [DS-90]Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6),391-407.
- [Multidoc] R. K. Ando, B. K. Boguraev, R. J. Byrd, and M. S. Neff. Multi-document summarization by visualizing topical content. ANLP/NAACL 2000 Workshop on Automatic Summarization, 2000
- [MS-1995] M. Storey and H.A. Muller. Graph layout adjustment strategies. In *Proceedings of Graph Drawing 1995*, Passau, Germany, pages 487--499. Springer Verlag, September 1995. *Lecture Notes in Computer Science*.
- [MS-1998] M. Storey (1998). *A Cognitive Framework For Describing and Evaluating Software Exploration Tools*. PhD thesis, Computing Science, Simon Fraser University, Canada.
- [ND-Topics] Charles K. Nicholas and Randall Dahlberg. Spotting topics with the singular value decomposition. PODDP1998
- [RB-1998] Rajman and Besanon, *Text Mining - Knowledge Extraction from Unstructured Textual Data*, 6th Conference of International Federation of Classification Societies (IFCS-98), Rome, 1998
- [THEME-ICSE]E. Baniassad, S. Clarke. Theme: an Approach for Aspect-Oriented Analysis and Design. ICSE2004