ReCVisu: A Tool for Clustering-Based Visual Exploration of Requirements

Sandeep Reddivari, Zhangji Chen, and Nan Niu
Department of Computer Science and Engineering, Mississippi State University, USA
{srr159, zc140}@msstate.edu, niu@cse.msstate.edu

***Abstract*—Clustering is of great practical value in discovering natural groupings of large numbers of requirements artifacts. Clustering-based visualization has shown promise in supporting requirements tracing. In this paper, we transform the success to a wider range of clustering-based visual exploration tasks in requirements engineering. We describe ReCVisu, a requirements exploration tool based on quantitative visualizations. We discuss the key features of ReCVisu and its potential improvements over previous work.**

Keywords-requirements clustering; layout-based clustering; requirements visualization; visual exploration;automatic labeling

I. INTRODUCTION

Studies of requirements engineering (RE) practice show that the majority of requirements are written in natural language [1]. However, the textual artifacts can be cumbersome to comprehend and analyze. Similar challenges exist when the stakeholders have to review and understand requirements for large and complex systems. In addition, certain characteristics of requirements or the problem domain may only become apparent upon visual exploration of the elicited information or metrics [2].

Basic information visualizations, such as bar graphs, pie charts, and hierarchical structures, have long been employed in RE. These visualizations are routinely used to augment textual requirements with summarization that aggregates large amounts of information into a single representation for shared understanding and quick absorption by stakeholders. An inherent challenge in RE visualization is to tackle the typical lack of structure in the requirements artifacts [2].

The need for discovering the natural structure of a set of requirements artifacts has given rise to the research area of *requirements clustering* – the automatic grouping of requirements into clusters that are in some way characterized by an internal coherence and/or an external isolation. Researchers have applied clustering to support a variety of RE activities, including feature identification, system modularization, and automated tracing. A particular area that visualization has emerged to offer promising support is clustering-based traceability, where Cleland-Huang and Habrat proposed the novel ideas for visualizing the link context as well as filtering incorrect link clusters [3]. Building on the prior work, we recently presented TraCter aimed at enhancing the search user interfaces of the tool support for clustering-based traceability [4].

In this paper, we expand the clustering-based visual support beyond traceability towards a wider spectrum of exploration tasks in RE, such as uncovering requirements interaction and dependency. Visual exploration is the process of extracting insight from data via interaction with visual depictions of that data [5]. Applied in RE, visual exploration can help to tackle many issues, such as uncover the requirements structure, navigate around the requirements space, modularize crosscutting concerns, and understand requirements interactions and evolution. The next section provides the context of our research. We then present the ReCVisu (**Re**quirements **C**lustering **Visu**alization) tool by discussing its key features and its potential improvements over previous work.

II. CLUSTERING-BASED RE VISUALIZATION

In a state-of-the-art survey, Cooper Jr. et al. offered a classification scheme through which the RE visualization approaches could be categorized [2]. The survey showed that hierarchical and relational visualizations became more available in RE, whereas quantitative visualizations received relatively limited attention. Another key finding was that, compared to requirements modeling, validation of requirements lacked sufficient visualization support.

In an attempt to provide visual support for validating and tracing requirements, Cleland-Huang and Habrat leveraged clustering to organize the requirements artifacts in a hierarchical manner [3]. Fig. 1 shows a view of such support, in which each requirements artifact is represented as a leaf node and the requirements headings are portrayed by larger nodes than lower-level (leaf-level) requirements. The edges in Fig. 1 imply the hierarchical information, e.g., a leaf-level node belongs to a certain heading. In this way, clusters are depicted as trees and are connected by their root nodes. The visualization layout is computed based primarily on the principle of minimizing the overlap of nodes and edges [3].

Fig. 1 offers a birds-eye-view of the requirements artifacts and their distribution across the information space. A key design to enable the effective birds-eye-view is the explicit labeling of each cluster, which uses a succinct term extracted from the headings to summarize the requirements cluster [3]. Other features in Fig. 1 relate specifically to automated tracing. For example, the color of the node indicates the degree to which each requirement is linked to the trace query: darker nodes are more likely to be linked than paler ones and unshaded nodes are very unlikely to represent correct traceability links. The hierarchical visualization of requirements clusters has successfully addressed several traceability issues [3]. Inspired by this seminal work, we extend the use of clustering to support visual exploration of requirements.

III. RECVISU

The design intent of ReCVisu is to support the exploration of requirements via quantitative visualizations.

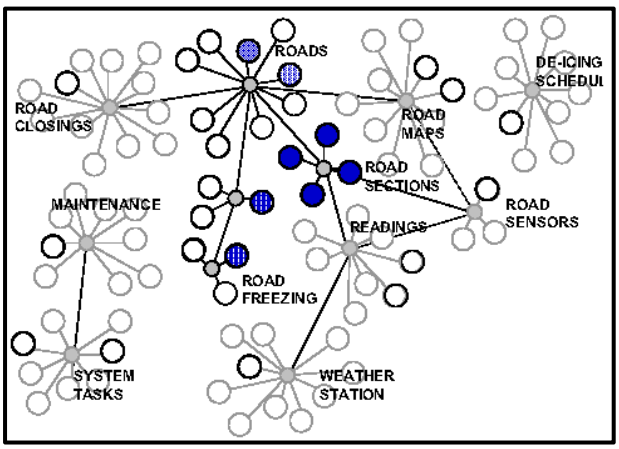

Figure 1. Clustering-based visual support for requirements tracing [3]

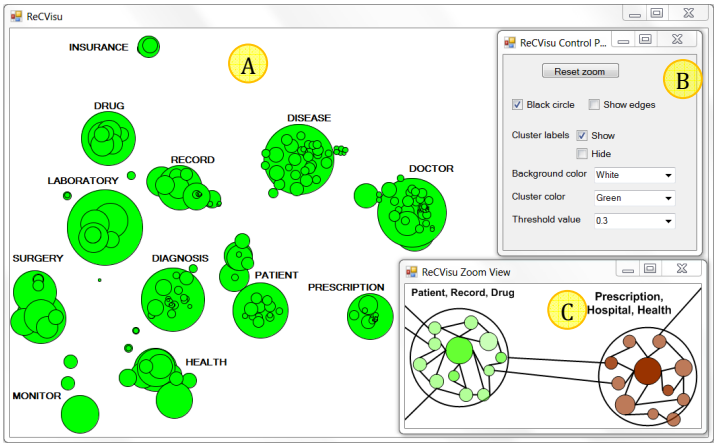

Figure 2. ReCVisu (Requirements Clustering Visualization)

To that end, ReCVisu advances the prior work by adopting two principles: layout-based clustering [6] and automatic labeling [7]. Fig. 2 illustrates the ReCVisu tool.

Layout-based clustering introduces rigor into the depiction scheme by attempting to ensure the drawn layout fulfills the desired clustering properties [6]. In ReCVisu, the dependence graph consists of requirements artifacts as nodes and the textual similarities (i.e., tf-idf cosine values) as edges. ReCVisu then maps each node to a position in the two-dimensional space, and computes the layout by force-directed graph drawing where an energy model encodes the layout goal. For clustering, this means to produce layouts that provide separation of cohesive subgraphs and interpretable distances. ReCVisu uses the LinLog energy model that most naturally reveals software structures [6]. As shown in Fig. 2-A (area A), the iTrust project's requirements (http://agile.csc.ncsu.edu/iTrust) are indeed grouped naturally into 12 clusters. Compared to Fig. 1, the birds-eye-view of Fig. 2-A offers additional and remarkable insights into the requirements space, e.g., the size of the node indicates the dependence fan-in and fan-out of the requirement, the distance between nodes reflects the similarity of requirements, etc.

Similar to [3], ReCVisu summarizes requirements clusters by presenting labels for stakeholders to grasp the essential themes. Different from [3], ReCVisu takes full advantage of automatic methods to generate cluster labels. In our recent work [7], we identified 3 categories of automated labeling techniques: cluster-internal, differential, and hybrid. Our empirical evaluations showed that differential labeling outperformed the other two categories. Thus, ReCVisu adopts chi-square selection to produce cluster labels in a differential way [7]. Other key features of ReCVisu include:

- *Filtering* specifies a threshold to ignore the edges with weak associations. For example, using ReCVisu's current vector space model similarity calculation, a user can choose the cutoff value of 0.3 from the control panel (cf. Fig. 2-B) to discard low-quality (outlier) edges, which allows a more sensible layout to be produced.
- *Zooming-in* enables a fish-eye-view (a set of clusters as shown in Fig. 2-C) or even a bulls-eye-view (a targeted examination of a particular cluster). In Fig. 2-C, not only more automatically generated cluster labels are shown, but the intra- and inter-cluster edges are displayed as well. These edges can facilitate the traversals within a cluster and across requirements clusters.
- *Coloring* is a feature inspired by [3]. When a certain cluster is zoomed in (cf. Fig. 2-C), the node shading indicates the degree to which the requirement is associated with the cluster labels (themes): the darker the node, the more similar the requirement is to the cluster labels. This feature can help visually assess the cluster's cohesiveness, as well as identify cluster centroids and barriers from a semantic perspective.

IV. SUMMARY

The visual support based on cluster hierarchies has shown promise in requirements tracing [3]. This paper presents the ReCVisu tool aimed at supporting a wide variety of visual exploration tasks in RE. As an ongoing evaluation of ReCVisu, we have worked with four professional requirements/business analysts in the healthcare domain on a set of requirements reuse problems. The initial feedback of using ReCVisu's birds-eye-view (cf. Fig. 2-A) and fish-eye-view (cf. Fig. 2-C) to explore and discover reusable requirements was positive, but several enhancements were also identified, including keyword search and metrics calculation, which we plan to address in our future work.

REFERENCES

[1] M. Luisa, et al. Market research for requirements analysis using linguistic tools. *Req.s Eng. J.*, 9(1): 40-56, 2004.

[2] J. R. Cooper Jr., et al. Requirements engineering visualization: a survey on the state-of-the-art. In *REV*, pages 46-55, 2009.

[3] J. Cleland-Huang and R. Habrat. Visual support in automated tracing. In *REV*, article no. 4, 2007.

[4] A. Mahmoud and N. Niu. TraCter: a tool for candidate traceability link clustering. In *RE*, pages 335-336, 2011.

[5] T. J. Jankun-Kelly, et al. A model and framework for visualization exploration. *IEEE TVCG*, 13(2): 357-369, 2007.

[6] A. Noack. Energy models for graph clustering. *J. Graph Alg.s and App.s*, 11(2): 453-480, 2007.

[7] N. Niu, et al. Automatic Labeling of Software Requirements Clusters. In *SUITE*, pages 17-20, 2012.