Keeping Requirements on Track via Visual Analytics

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

***Abstract*—For many software projects, keeping requirements on track needs an effective and efficient path from data to decision. Visual analytics creates such a path that enables the human to extract insights by interacting with the relevant information. While various requirements visualization techniques exist, few have produced end-to-end values to practitioners. In this paper, we advance the literature on visual requirements analytics by characterizing its key components and relationships. This allows us to not only assess existing approaches, but also create tool enhancements in a principled manner. We evaluate our enhanced tool supports through a case study where massive, heterogeneous, and dynamic requirements are processed, visualized, and analyzed. In particular, our study illuminates how increased interactivity of requirements visualization could lead to actionable decisions.**

***Index Terms*—Requirements management, requirements engineering visualization, visual analytical reasoning.**

I. Introduction

The most data-intensive software development activities lie in requirements engineering (RE) where stakeholders are determined, problems are explored, and goals are defined [1]. Not only does RE involve the identification of the diverse stakeholder concerns and the complex environmental constraints, critical decisions are also made in RE when business objectives are transformed into technical specifications, when conflicting viewpoints are negotiated into an agreed upon action plan, and when reasonings are performed to enable revisions without incurring the serious time or budget overturns. Keeping requirements on track thus requires a disciplined approach to handling the massive, heterogeneous, and dynamic volumes of information.

The emerging field of *visual analytics* (VA) offers a solution to turn the information overload into an opportunity. VA is described as "the science of analytical reasoning facilitated by interactive visual interfaces" [2]. The basic idea is to visually represent the data so as to allow the human to directly interact with the information, to promptly gain insights, and to ultimately make optimal decisions. It is important to note that VA is not a separate field of study [3], but a key enabler of visual data analysis that can support a variety of applications. Domains benefiting from VA include physics, climate, business, health, and many others [3].

Applied to RE, VA helps create a path from data to decision. In this process, visualization plays two pivotal roles: 1) it represents the requirements information by highlighting certain constructs and relationships while ignoring others; and 2) it serves as the interaction medium to augment a requirements analyst's knowledge discovery with advanced computational capabilities.

The first issue of effectively generating requirements-centric visualizations has received increasing attention in recent years. In particular, the requirements engineering visualization (REV) workshop series, initiated in 2006, offers a stimulating forum in which the ideas of using visualization in RE are presented, discussed, and disseminated. Cooper *et al.* [4] provided an excellent survey of all but the most recent work in the field, showing the wide range of visualizations employed: tabular, relational, sequential, hierarchical, and quantitative/metaphorical. While most REV techniques are used for requirements modeling and specifications, the survey reveals the need to thoroughly understand the *tasks* within RE that are best supported with visualization [4].

Compared to requirements visualization, only modest efforts have been devoted to the second issue of creating "interactive visual interfaces" [2] to facilitate the accomplishment of an analyst's task at hand. One of the seminal papers on the topic was presented by Gandhi and Lee [5], where metaphorical visualizations were developed for requirements-driven risk assessment. The work shows the potential of VA to tackle security certification's complex decision-making process where numerous, unstructured, multi-source, and conflicting requirements must be modeled, related, and compared. The work also sheds light on the synthesis of human's background knowledge in the VA process. For example, to produce valid visualizations, an experienced certification analyst shall manually map risk-related requirements to the concepts defined in the domain-specific ontology (e.g., assets, threats, vulnerabilities, etc.) [5].

Building on prior work, we devised a tool [6] aiming for tackling a broad spectrum of visual exploration tasks in RE. Although the tool helps extract structural insights from the visual depiction of requirements, it is still limited in delivering end-to-end (from data to decision) capabilities. This shortcoming is common to state-of-the-art VA approaches in RE, as shown by our review in Section III. A fundamental challenge here is the lack of understanding about how VA best answers the requirements analyst's needs.

In this paper, we address the challenge by characterizing the VA process itself, thereby assessing the strengths and weaknesses of contemporary approaches. Guided by this novel and in-depth understanding of VA in RE, we describe the

978-1-4673-5765-4/13 © 2013 IEEE

RE 2013, Rio de Janeiro, Brasil
Research Track

Accepted for publication by IEEE. © 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

improvements made by our enhanced tool in the context of an industrial case study. The contributions of our work are threefold: (i) development of an RE-oriented framework modeling the core VA components and their interactions; (ii) evaluation of 8 existing VA techniques by applying the framework; and (iii) advancement of the literature through building VA capabilities that can produce end-to-end values to RE practitioners.

The rest of the paper is organized as follows. Section II presents our experience of creating VA tools to facilitate requirements analysis in practice. Section III unfolds the derivation of the visual requirements analytics framework and applies the framework in various contexts. Section IV describes the case study where the increased interactivity of requirements visualizations is shown to lead to actionable decisions. Section V discusses related work, and finally, Section VI concludes the paper.

II. ReCVisu as a VA Tool for RE

Motivated by visual supports for risk assessment [5] and requirements tracing [7], we built the ReCVisu (Requirements Clustering Visualization) tool for aiding the *exploratory* tasks in RE [6]. Our main design rationale was that certain characteristics of the problem domain might become apparent upon visual exploration of the requirements [4]. ReCVisu thus employs layout-based clustering [8] to uncover the requirements structure and to ease the navigation of the requirements space. Integrated with automatic labeling of clusters [9], ReCVisu is also able to identify the semantically prominent themes in the requirements.

We illustrated the key features of ReCVisu [6] via an open-source project — iTrust (agile.csc.ncsu.edu/iTrust). We further presented the results of this proof-of-concept example to several IT professionals working in the healthcare domain. The purpose was to test whether the professionals could extract from the resulting visualization any reusable requirements for their own applications. The professionals all felt that the clustering-based visual depiction of the nontrivial set of iTrust's requirements was succinct and insightful [6]. Encouraged by these beginnings, we collaborated with a medium-sized software development organization to investigate the extent to which a VA tool like ReCVisu could help keep a real-world system's requirements on track.

Our partner organization is an interdisciplinary R&D (research and development) center that specializes in formulating and implementing software-intensive, big data solutions for many constituencies and programs. In order to honor the confidentiality agreements, we will use the pseudonyms "RDC" for the organization and "SWP" for their software project which we collaborate in. SWP, initiated to serve the State of Mississippi, has its broad goal to develop, maintain, and expand longitudinal data systems from pre-kindergarten through the workforce for better decision making that can improve student outcomes.

The SWP team has elicited and gathered a great deal of requirements data and is currently faced with an enormous challenge of analyzing and eventually acting on the data, e.g., deciding to revise, reconcile, implement, or ignore a requirement. For many industrial-strength projects like SWP, the requirements to be kept on track are:

- voluminous: an example is that one of SWP's stakeholders, K-12 (kindergarten to grade 12), listed 113 requirements throughout the focus group meetings;
- heterogenous: SWP is intended to align the data and efforts of universities, community colleges, K-12 schools, and early childhood institutions among others;
- messy: the requirements are documented mostly in unstructured text with stakeholder concerns (e.g., tracking career expectations) scattered, tangled, replicated, or otherwise difficult to locate;
- dynamic: not only are requirements (e.g., reporting preferences) in constant change, but the change impact is hard to determine due to the multifaceted and often implicit interdependencies among requirements; and
- conflicting: competing needs are reflected in both functional requirements (e.g., K-12 permits parents to query their child's grades whereas higher learning institutions disallow that by default) and nonfunctional requirements (e.g., scalability versus responsiveness).

ReCVisu, as currently implemented [6], handles mainly the voluminous and messy aspects by grouping and abstracting similar requirements into a single cluster. While the tool can be improved along many dimensions, a pressing demand based on applying ReCVisu to SWP is to draw actionable decisions on top of the requirements visualizations. In other words, visualizations by themselves are not sufficient for keeping requirements on track. What is also needed is to make the visualizations truly interactive so that the analyst can directly manipulate them in real time during the decision making process. Next, we present a framework that provides the built-in analytical reasoning capabilities through interactive requirements visualizations.

III. Visual Requirements Analytics

In this section, we take a detailed look at visual analytics (VA) as it is applied to RE (Section III-A). The resulting framework allows us to assess our own ReCVisu tool [6], as well as other contemporary RE-centric VA solutions (Section III-B). Such an examination of the literature is especially useful for identifying areas for improvement which we will base to develop tool enhancements (Section IV).

A. Understanding Visual Requirements Analytics

In essence, VA is aimed at synthesizing the strengths of machines with those of humans [2]. On one hand, modern computers and automated methods, such as data mining [10] and machine learning [11], offer unprecedented computational power to facilitate knowledge discovery. On the other hand, it is indispensable for informed decision making to include humans in the data analysis process to leverage flexibility,

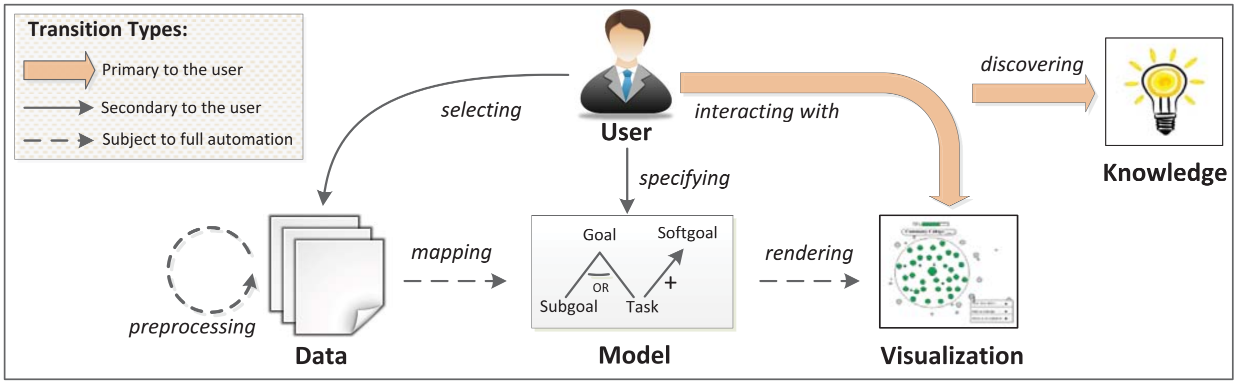

Fig. 1. A framework that characterizes the key components and their interactions in the visual requirements analytics process.

creativity, and background knowledge [3]. The specific advantage of making the human-machine synthesis in a *visual* way is that data analysts, decision makers, project managers, and other stakeholders can focus their full cognitive and perceptual attentions on the visualization-enabled analytical reasoning while taking advantage of the automatic data processing techniques [3].

Based on the VA literature, we construct a visual requirements analytics framework shown in Fig. 1. Compared with existing conceptualizations (e.g., the ones presented in [2] and [3]), our framework explicitly models the "user" to suggest that machine's computations only augment, but cannot replace, human's capabilities to perceive, relate, and conclude in the knowledge discovery and decision making process. In addition, we distinguish in Fig. 1 the degree of user involvement in the VA activities. These distinctions are made by using different transition types in Fig. 1.

It is important to note that the "user" in our framework refers to the stakeholder who uses the VA methods, techniques, and tools to carry out RE tasks. In practice, the VA "user" can be a requirements engineer, a data analyst, a business manager, a project coordinator, a developer, a tester, a customer, and/or an end user of the software system.

As mentioned earlier, the requirements for many software projects are of large scale and of different source. Therefore, the first step of VA is often to process the raw data in order to extract relevant requirements information for further visual and automatic analysis. For textual requirements, typical preprocessing includes stemming, stop word removal, and other data cleaning or normalization procedures [12, 13].

Continuing with the preprocessed data, the underlying model in Fig. 1 defines what entities and relationships will be used to support the user's RE task at hand. Goals [14], use cases [15], features [16], problem frames [17], and stakeholder social networks [18] are among the most commonly employed models. Though graphical in some cases, the model is primarily concerned with specifying the problem domain ontology [5], thereby shaping the transformation from data to visualization. In certain approaches (e.g., [6, 7]), the model is only implicit in that the natural language descriptions are extracted and treated as the main requirements constructs.

Unlike scientific visualization where the data entities are typically 3D geometries or can be explicitly referenced to time and space [19], the visualization of requirements is a type of *information visualization* (IV) [20] that deals with abstract data with hundreds of dimensions and no natural mapping to the display. Thus, novel techniques are devised by employing metaphorical [5], quantitative [6], hierarchical [21], relational [22], and other graph-based [23] visual data representations. It is well known in the IV community that, very often, there are many different ways to represent the data under consideration [20]. Searching for the best requirements visualization can be impractical and even counterproductive. It is therefore more valuable to create effective and efficient ways to *analyze* the data. In this sense, VA is more than just the visualization. It also focuses on how the user interacts with the visualization. Influenced by Shneiderman's celebrated "overview first, zoom/filter, details on demand" IV interaction mantra [24], Keim *et al.* [3] describe the VA interaction mantra to be (1) analyze first, (2) show the important, (3) zoom, filter and analyze further, and (4) details on demand.

The interactions with the requirements visualizations shall augment the user's knowledge discovery and lead to actionable decisions; otherwise, they become wasted interactions. We highlight in Fig. 1 that it is through the interactive visualizations that important insights are gained, efficient reasonings are performed, defensible assessments are made, and optimal analysis results are arrived at. It is worth pointing out that the VA path from data to decision is not strictly linear but highly iterative and incremental with feedback loops between and within the stages. For example, a visual comparison may generate new hypotheses to test, which in turn triggers the user to scrutinize certain preprocessing procedures and to refine the underlying data model.

B. Assessing Visual Requirements Analytics Approaches

The main objective of the proposed framework is to assess existing VA approaches in RE. This not only substantiates the value of the framework, but also suggests potential tool integration and guides further tool development in a principled manner. The five components presented in Fig. 1 represent the key areas and thus the *conceptual goals* that a visual require-

TABLE I

FIVE CONCEPTUAL GOALS AND THEIR OPERATIONAL QUESTIONS TO BE ADDRESSED BY A VISUAL REQUIREMENTS ANALYTICS APPROACH

User		Data		Model		Visualization		Knowledge	
U1	Multiple stakeholder roles	D1	Large-scale inputs	M1	Explicit model representation	V1	Multiple views	K1	Anomaly detection
U2	Usage without heavy training	D2	Heterogeneous input types	M2	Automatic model construction	V2	Inter-view navigation	K2	Detailed explanation
U3	Real-time performance	D3	Automatic preprocessing	M3	Model extension and customization	V3	Browsing	K3	Hypothesis-based reasoning
U4	Integration into existing software development environment			M4	Model traceability	V4	Searching	K4	Scenario-based reasoning
U5	Practitioner-oriented guidelines					V5	Query-drilling	K5	Actionable decision
						V6	Filtering		
						V7	Annotation		

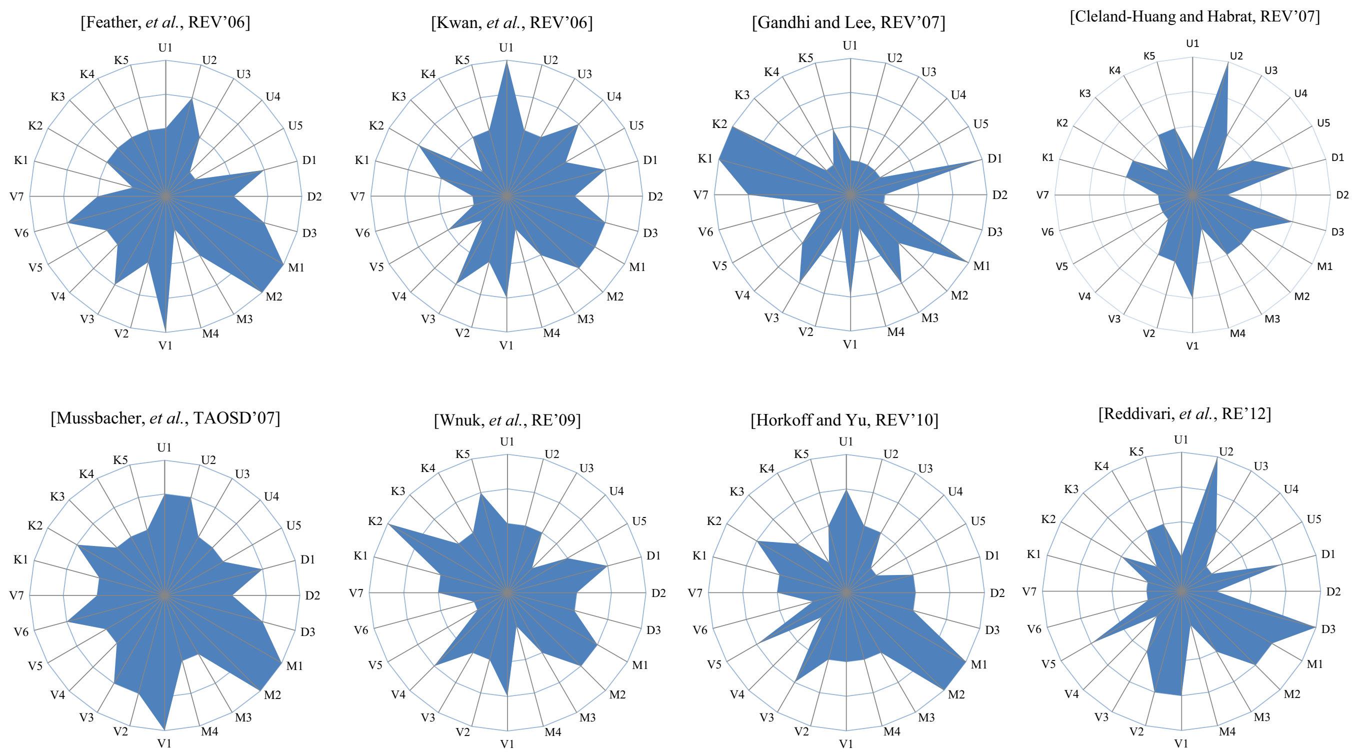

Fig. 2. Starplots of assessed visual requirements analytics approaches.

ments analytics approach shall satisfy. It is this straightforward mapping that motivates the application of the Goal Question Metric (GQM) [25] paradigm in our work. The top row of Table I lists the conceptual goals.

In GQM, a goal needs a purpose, issue, object, and viewpoint [25]. Take the "user" goal as an example; here the need is to assess (the purpose) the adequacy (the issue) of user satisfaction (the object) from the VA tool provider's perspective (the viewpoint). In order to derive the *operational questions* associated with each goal, we performed an extensive analysis of the literature in the area of requirements engineering visualization with special emphasis on analytical solutions. When reviewing Gandhi and Lee's seminal work [5], for instance, we noted that a real-world security certification and accreditation scenario could involve over 500 requirements. Thus, the question "Does the VA approach support large-scale inputs?" (D1 in Table I) was posed. Continuing in a like manner yielded all the questions for use in GQM. Table I groups and labels each goal's operational questions.

TABLE II

METRICS OF QUALITATIVE RESPONSES TO QUESTIONS IN TABLE I

Response	Meaning
++	Full support
+	Mainly supported
−	Mainly not supported
−−	No support or unable to determine

Table II gives rise to the metric for the GQM used in our assessment. The metric definition is qualitative in nature and is in line with how softgoal contributions are characterized in goal-oriented requirements analysis [26, 27]. The metric has four ordered values which can be reasonably treated as a

Likert scale. This choice of values makes the application of our framework not overcomplicated. Meanwhile the responses can be readily distinguished with intrinsic meanings.

Having instantiated the constructs of Goal Question Metric, we applied our framework to examine the state-of-the-art VA supports for RE. Through the literature review, we identified a set of representative papers among which eight were thoroughly evaluated. Specifically, two researchers independently rated the degree of support of all the eight approaches by following the GQM paradigm. Their ratings were shared and discussed in a meeting with the engagement of a third researcher. Over the joint effort, a few different responses were reconciled and the final consensus was achieved.

Fig. 2 shows the starplot representation of the evaluation of the eight visual requirements analytics approaches. In Fig. 2, each axis in the starplot denotes an operational question in Table I. The axis is scaled according to the metric values defined in Table II: a "++" response reaches the outer rim whereas a "−−" rating stays at the center of the starplot. While the list of evaluated approaches is by no means exhaustive, it contains work published by distinct research groups, on various topics, and in different years. Next is a brief summary of each approach.

- Feather *et al.* [28] shared their experiences of using the Defect Detection and Prevention (DDP) model to support risk mitigation in the early project planning phase. DDP adopts relatively simple visualizations such as bar charts, treemaps, and tabular formats. The majority of DDP applications have been in the area of technology infusion where the model shows the capabilities in scrutinizing a candidate's status, exploring the solution space, and comparing the alternatives [28].
- Kwan *et al.* [18] presented the requirements-centred social network (RCSN) as a means to communicate and improve awareness within development teams. Nodes and edges of the social network are used as the major visualizations in RCSN to represent various project information related to developers' activities. RCSN can be generated by mining a project plan and early requirements documents [18].
- Gandhi and Lee [5] proposed an ontology-based active RE (Onto-ActRE) framework to support risk assessment. Two main visualizations are used: cohesive bar graph for identifying all the potential risks and cohesive arc graph for visualizing the risk impact. The authors have applied Onto-ActRE to handle over 500 requirements used in a security certification process [5].
- Cleland-Huang and Habrat [7] introduced a tool for visualizing candidate traceability link similarity and term contributions. The tool also uses tree-like visualizations to model the link context. Applied to the Ice Breaker System with 202 requirements and 75 classes, the tool has shown to be useful for quickly and accurately determining the traceability information [7].
- Mussbacher *et al.* [15] showed how use case maps (UCMs) could model crosscutting concerns at the requirements level where early aspects and the base model are visualized together. Their work shows that aspect-oriented UCMs need no new visual notations and that the aspect-incorporated UCMs can help detect requirements interactions [15].
- Wnuk *et al.* [16] described Feature Survival Charts to help stakeholders visualize scope changes. Five scope tracking measurements are defined for understanding change rationale. The technique is evaluated in three large industrial projects, and the visualizations have been acknowledged by the case company practitioners to be useful for adjusting scoping decisions [16].
- Horkoff and Yu [14] developed a visualization tool, called OpenOME, to mitigate user's difficulty in analyzing goal models. Finding starting points and understanding trade-offs represent two major challenges, which OpenOME addresses by detecting and highlighting roots, leaves, and conflicts in a goal model. The results from 5 studies confirm the usefulness of OpenOME's visual highlighting feature [14].
- We recently devised the ReCVisu tool to facilitate visual exploration tasks in RE [6]. The tool integrates automatic labeling [9] into graph-based clustering visualizations [8]. An initial evaluation with 4 IT professionals on using ReCVisu to explore requirements reuse opportunities receives positive feedback [6].

It is worth mentioning here that even though the usefulness of our framework is demonstrated by the wide applicability of evaluating 8 approaches, a framework of this nature may not and indeed may never be complete. As our understanding evolves, new concerns will likely emerge and current ones will necessarily change. Nevertheless, we believe the framework, along with its applications, makes a timely contribution to the VA for RE literature.

The starplots shown in Fig. 2 can be used to analyze the literature from multiple perspectives. For example, the work by Feather *et al.* [28] and that by Gandhi and Lee [5] both deal with risk management in RE. Their starplots appear to be meeting the conceptual goals complementarily, which makes these approaches potential candidates for tool integration. An interesting perspective is the identification of a tool "family". For instance, the starplot of our own ReCVisu [6] tool is very similar to that of Cleland-Huang and Habrat's work on visual support in automated tracing [7]. This reveals ReCVisu's "heritage" and also depicts how the "child" (ReCVisu) inherits, extends, and deviates from its "parent".

The most useful analysis, in our opinion, is through the inspection of the starplots to identify the strengths and weaknesses of existing tools. As shown in Fig. 2, most tools do reasonably well in supporting explicit model representation (M1) and automatic model construction (M2). Also the automation level of data preprocessing (D3) seems satisfactory. However, visualization (V1-V7) and knowledge (K1-K5) are noticeable and surprising areas that existing tools fail to cover in a consistent and complete fashion. This guides our development of enhanced supports in order to fill the gaps and to increase user satisfaction.

IV. Case Study

We report in this section an exploratory case study [29] by collaborating with the RDC organization on the SWP project within the project's real-life context. Our overall goal is to gain operational insights into how a VA tool can help keep requirements on track in practice. To that end, we first describe the case study design (Section IV-A). We then present the findings (Section IV-B) and discuss the threats to validity of our study (Section IV-C).

A. Rationale, Objective, and Procedure

The main reason that we adopt case study as the basis for our research design is that the investigation of a contemporary phenomenon is suitable for addressing the 'how' and 'why' questions that can otherwise be difficult to answer through controlled experiments [30]. Essentially, the benefits and obstacles of using VA in RE are only likely to be evident for the ongoing real-world project, under conditions that cannot be replicated in the lab. In particular, the study of applying VA in RE cannot be separated from the organizational context and the effects may take weeks or months to appear.

We therefore designed an *exploratory* case study in collaborating with RDC's SWP team. According to Yin [29], an exploratory case study is appropriate for preliminary inquiries in which it is not yet clear which phenomena are important, or how to measure these phenomena. In our case, we were particularly interested in understanding the practical impacts of VA on the RE tasks. The current literature, unfortunately, provides little insight into what RE tasks are best supported with visualization and how a visual approach is best deployed [4]. For these reasons, it would be premature to try to measure the cost/benefit trade-off and the statistical significance of certain variables. For our exploratory study, we set out to answer the following questions: i) what RE tasks are in need of VA support; ii) how VA supports these tasks; and iii) what benefits can be expected.

To achieve the research objective, we decide to make the ReCVisu tool [6] more interactive by implementing the visualization features, such as inter-view navigation (V2) and annotation (V7), which are currently less supported (cf. Fig. 2). We name the enhanced tool "ReCVisu+" to advocate that the tool should strive for producing end-to-end, from-data-to-decision values to its users. The development of ReCVisu+ has been tightly coupled with the SWP project. Table III provides the basic information about the development efforts of ReCVisu+.

Over the past few months, we held 4 meetings in RDC's workplace. Each meeting engaged one or more SWP project members; the requirements analyst participated in all the meetings. We regarded these collaborative efforts as being similar to joint application development (JAD) [31] workshops where "knowledge workers and IT specialists meet to define and review the requirements for the system" [32]. In our study, reviewing and analyzing the requirements for SWP simultaneously helped define and clarify the requirements for ReCVisu+. This allowed us to implement the most desired features between the meetings and to assess how ReCVisu+ supported the RE tasks in short cycles.

TABLE III
Joint application development for the ReCVisu+ tool

Preparation by the research team	Meeting date & duration	SWP participant(s)*	Main Activities
Demo ReCVisu on iTrust	Aug 8, 2012 & 1 hour	PM, SA, RA	• Present ReCVisu • Collect feedback • Gather SWP requirements
Launch ReCVisu+ & improve "overview"	Sept 21, 2012 & 2.5 hours	RA	• Detect & act on extremity • Elicit RE tasks
Implement "anom -aly" handling	Nov 30, 2012 & 1.5 hours	PM, RA	• Diagnose & handle outliers • Elicit RE tasks further
Implement "heteroge -neity" utilization	Jan 14, 2013 & 0.5 hour	SA, RA	• Relate multiple artifacts • Refine ReCVisu+ design

* PM: Project Manager, SA: Software Architect, RA: Requirements Analyst

B. Findings

The data in our study were collected mainly through observations and interviews during the 4 meetings (cf. Table III). E-mails were occasionally exchanged with the SWP requirements analyst for clarifying the issues and concerns. We took extensive notes and transcribed all the interviews. We then collectively applied qualitative data analysis [33] to code and categorize the data. For coding, we segmented and allocated units of meaning to the data collected. For categorizing, we interpreted and assigned these units for answering our research questions. The qualitative data analysis was performed by two researchers manually in a collaborative fashion.

Table IV details the main results of our exploratory study. We focus the result discussion more on how VA best shows its value in establishing the path from data to decision, and highlight only the necessary technical aspects of ReCVisu+. As noted by Shneiderman [24] in characterizing information visualization interactions, the SWP project members invariably confirm that an instant visual *overview* of the requirements space is crucial for determining the starting points (regions of interest) of analyses. Fig. 3 shows the overview of SWP's requirements, each of which is depicted as a green node (clustered) or a grey node (unclustered).

In our study, the SWP analyst specifies the desired number of clusters to be 5 as SWP services 5 major agencies: early childhood, K-12, community college, university, and workforce. This is an example of integrating human's background knowledge into the VA process. ReCVisu+ then exploits layout-based clustering [8] to automatically position the nodes in the two-dimensional space; here the similarity between nodes is determined by the requirements' textual descriptions [6]. For each cluster, ReCVisu+ displays its labels [9] and signals its cohesiveness via a percentage bar. Currently, cohesiveness is computed as the average pairwise similarities of a given cluster's items. The percentage bar is shown in red if a cluster's cohesiveness is below a user-adjustable threshold

TABLE IV
MAIN RESULTS OF THE CASE STUDY

RE task in need of VA support	Support provided by ReCVisu+	Sample actionable decision
Overview – Summarize a large requirements space – Identify regions of interest	● Present flexible labels for each visual cluster ● Show a cohesiveness bar to suggest interesting regions	The user decides to split the less cohesive cluster in the visual overview (cf. Fig. 3) and further decides to elicit more requirements for SPED (cf. Fig. 4).
Anomaly – Locate peculiar requirements – Deal with the deviations	● Annotate an outlier as a bridge between clusters ● Adopt an outlier as an orphan to a cluster	The user decides to mark the “Transfer students” as a bridge (cf. Fig. 5) and further informs the developers to implement “Transfer student” interfaces to both “Community College” and “University” classes.
Heterogeneity – Relate requirements to other artifacts – Compare multi-stakeholder concerns	*(partially supported in the current ReCVisu+ implementation)* ● Use an artifact’s tags to link requirements in different clusters	By comparing a design mockup’s tags, the users uncovers a hidden link and decides to support Community College’s career counseling with Workforce’s job market trend report (cf. Fig. 6).
Causality – Perform semantic analysis – Make multivariate comparison – Support exploratory reasoning	*(to be implemented in the future releases of ReCVisu+)* ● Transform a hypothesis into a visual representation and test the hypothesis interactively	A hypothetical decision-making scenario on requirements prioritization that can be supported via interactive visualizations is: Requirements cluster A cannot be implemented unless cluster B is implemented, but certain requirements in B are in conflict with part of A.

Fig. 3. Visual overview leading the action of splitting the low-quality requirements cluster into sub-clusters.

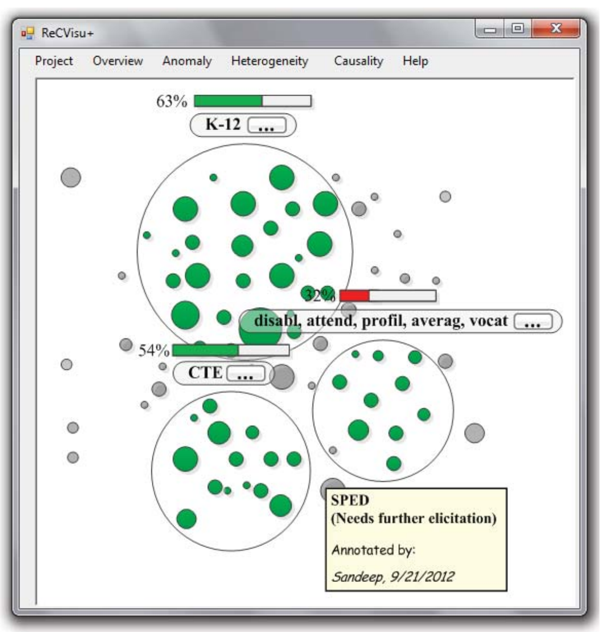

Fig. 4. Split sub-clusters leading the action of further requirements elicitation.

(e.g., 50%). As shown in Fig. 3, such a visual cue triggers the SWP analyst’s action on the low-quality cluster.

The analyst decides to split the less cohesive yet large cluster into 3 sub-clusters: K-12, CTE (Career Technical Education), and SPED (Special Education). Once again, human knowledge is integral to this decision. Meanwhile, automatic analysis offers valuable help: the automatically generated labels “career” and “special” shown in Fig. 3 confirm the analyst’s considerations of “CTE” and “SPED” respectively. Fig. 4 shows the split results. Noticeably, one sub-cluster (SPED) is flagged with the red percentage bar. This time, the analyst’s decision is to elicit further requirements for SPED and to investigate the outliers close to the cluster boundary.

Acting on the outliers is essential to *anomaly* handling, which is identified as an important RE task in need of VA support (cf. Table IV). ReCVisu+ allows the analyst to either annotate an outlier as a bridge connecting two clusters, or to adopt it as an orphan to a cluster. The latter option is inspired by Tzerpos and Holt’s work on clustering-based architecture recovery [34]. Fig. 5 illustrates the anomaly handling features implemented in ReCVisu+. “Transfer students” is recognized as an interface to both “Community College” and “University”.

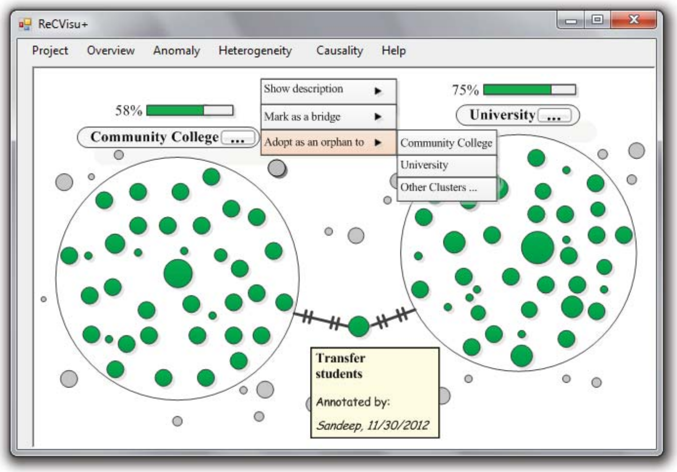

Fig. 5. Anomaly handling in ReCVisu+.

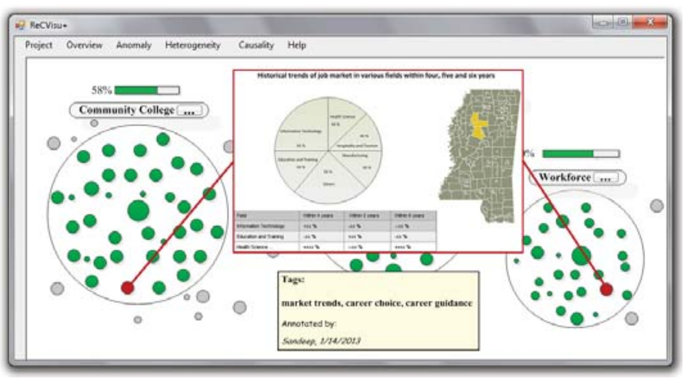

Fig. 6. Using a design mockup to link requirements in different clusters.

This leads to an important implementation decision for SWP, as shown in Table IV. Note that the cluster labels in Fig. 5 are manually adjusted by the SWP analyst; similar adjustments are made to the "K-12" and "CTE" clusters in Fig. 4. Fig. 5 also shows that after an outlier is annotated as a bridge, its color turns from grey to green. This provides an instant feedback to the analyst's action and can also be used to monitor the progress of anomaly handling.

While it is valuable to overview the requirements in groupings and investigate the outliers that are not easily classified, the SWP team is interested in discovering hidden links. The requirements visualization, though generated by a scalable data mining technique (namely clustering), does exhibit a "static" view imposed by the underlying computation. An example of ReCVisu+ is the use of textual similarity to calculate requirements' distance, which neglects other relations. This seemingly drawback, as shown in Table IV, uncovers an opportunity for VA to leverage *heterogeneity* to extend the knowledge discovery.

In our case study, the SWP software architect shares with us a set of user interface (UI) mockups that are designed at the same time the requirements are analyzed. In order to make use of these heterogeneous artifacts, we semantically tag the UI mockups with help from the SWP architect. The tags are then used as queries to search against the SWP requirements for plausible interconnections. This can be seen as an instance of the software traceability problem [35]. We therefore integrate ReCVisu+ with our latest semantically-enhanced information retrieval method [36].

Fig. 6 illustrates how heterogeneous artifacts are related to discover implicit but potentially useful information. The UI mockup, together with its tags ("market trends", "career choice", "career guidance"), is loaded in ReCVisu+. Through automated information retrieval and human-centered analysis, two requirements from different clusters are found to be connected. On one end, a requirement inside the "Workforce" cluster specifies the SWP:

> *"to provide annual historical trends of the job market in various STEM fields and to allow the statistics to be viewed across and by different years and demographics."*

On the other end, a "Community College" requirement demands SWP:

> *"to provide information for the college advisors, counsellors, and recruiters to better guide the students to choose a major and a career program."*

The identification of hidden links like this receives positive feedback from the SWP team. In our opinion, such a utilization of heterogeneity provides support for VA's vision of "turning the information overload into an opportunity" [2]. In the above scenario, the identified traceability links offer a promising opportunity to make SWP successful, and more importantly, to meet the stakeholder (Community College) needs and goals. It is interesting to note that relating requirements to other artifacts has been found useful in IBM's Collaborative Lifecycle Management (jazz.net/clm) that provides integrations across Jazz-based products to connect the work of analysts with development and test teams. This has direct impact on keeping requirements on track: One may not guarantee requirements by themselves but can connect them to other areas and cycles of the project.

As can be noted from Table III, our case study is ongoing and some design decisions of ReCVisu+ need refinement. Among our planned revisions are making the drawing of cluster boundary more transparent and the visual exploration based on node size more explicit. In the meantime, advanced features of ReCVisu+ are being discussed and formulated. In particular, the *causality* analysis that takes into account deep semantics is considered an RE task that can benefit from the visualization-enabled multivariate comparison and exploratory reasoning. Because ReCVisu+ is currently inadequate to provide these capabilities, only a hypothetical decision-making scenario on requirements prioritization is given in the bottom row of Table IV.

C. Threats to Validity

Several factors can affect the validity of our exploratory case study. Construct validity concerns establishing correct operational measures for the concepts being studied [29]. The main constructs in our case study are 'VA supports' and 'keeping requirements on track in practice'. As for the first construct, the VA supports are embedded in the ReCVisu+ tool, which is developed with the intention of enhancing the state-of-the-art in visual requirements analytics (cf. Section III). A practical concern stems from our experience of building VA tool support for RE [6]. As shown in Fig. 2, different VA tools have different strengths and weaknesses. How other tools may support RE remains an open question. As for the second construct, our interpretation of 'keeping requirements on track' is anchored in the actionable decisions made during the VA process. Thus we do not feel this practitioner-oriented view poses a serious limitation.

Regarding internal validity [29], an important question is whether the benefits observed (e.g., actionable decisions) really are due to the VA supports. A likely confounding variable is that the software team has come up with similar decisions by using other requirements management tools. For SWP's requirements analysis, the only automated tool support is the use of a relational database management system (RDBMS) to record, update, and query the requirements. Therefore the insights produced by the VA tool (e.g., the visual overview in Fig. 3) are not only fresh but also difficult to obtain through database queries. A relevant issue hinges on the contributions of actionable decisions to keeping requirements on track. It might be the case that certain decisions resulted from VA support are only suboptimal due to technical limitations or other reasons revealed in the RE process. While this issue has not occurred in our study, it represents a potential threat.

A major limitation with our study design is that the researchers are also the ReCVisu+ tool builders. This compounds the problem of experimenter bias, because the researchers may manipulate the study to obtain the expected outcome. We mitigate such a threat in two ways. First, by using an exploratory case study, rather than an explanatory or causal study, we are able to concentrate more on reporting our experience than on trying to prove our hypotheses. Second, the pre-defined data analysis methods (coding and categorizing) are applied jointly rather than by a single researcher. Neither of these steps removes the threat of experimenter bias entirely; only replication with neutral participants can address this issue.

The results of our study may not generalize beyond RDC's organizational conditions and the SWP project's situational characteristics, a threat to external validity [29]. Nevertheless, our investigation of the contemporary project within its real-life context, together with the validation carried out in a real industry setting, provides a firm footing for applying VA in RE. Finally, in term of reliability [29], we expect that replications of our study should offer results similar to ours. Of course, the requirements under study may differ, but the underlying trends should remain unchanged.

V. Related Work

A growing body of work on requirements engineering visualization (REV) has emerged in recent years. The review by Cooper *et al.* [4] analyzes the historical trends and also organizes REV approaches in a unified framework. Referring to this framework, we believe VA tools like ReCVisu+ are especially suitable for requirements elaboration and refinement [4]. In addition, our study shows VA is helpful in supporting requirements exploration and discovery.

How to evaluate visual notations used in RE has attracted much interest lately. Moody *et al.* [37] proposed a set of principles based on the physics of notations and further applied the principles to evaluate the i^* visual notation. Similarly, Amyot *et al.* [27] developed quantitative, qualitative, and hybrid mechanisms for evaluating the goal-oriented requirements language. In contrast, our framework applies the GQM paradigm [25] to qualitatively assess the visual requirements analytics approaches.

While eight VA approaches for RE are reviewed in Section III-B, VA has also been applied to support other software engineering activities, such as release planning [38], product assessment [39], and product line engineering [40]. Nevertheless, as argued by Gotel *et al.* [1], RE is one of the most fruitful areas benefiting from the synergy of information visualization. This is because RE tasks often involve the reconciliation of multiple viewpoints, the discovery of structure in complex unstructured datasets, the fusion of data from disparate sources, and the development of agreed models [1]. Our work extends the synergy between information visualization and RE by showing how interactive visualizations can be used to facilitate comprehension, enable exploration, and communicate decisions.

Keim *et al.* [3] provided a general introduction to VA, in which the scope of VA is defined and the application scenarios of VA are presented. Disciplines that handle data amenable to scientific visualization provide rich opportunities to apply VA. In physics and astronomy, for example, VA has been used in applications like flow visualization, fluid dynamics, molecular dynamics, nuclear science and astrophysics, and the like [3]. However, it is encouraging to note that *visual software analytics* is considered as a promising application area [3]. In fact, survey papers on software visualization exist, such as REV [4], software architecture visualization [41], and visualization of the static aspects of source code [42]. Our work complements this literature by providing with the users a VA framework for RE to compare different methods and by developing the tool enhancements in a principled way.

VI. Conclusions

In this paper, we have proposed a framework to characterize and improve the state of practice in visual requirements analytics. We apply the framework to examine existing VA for RE solutions, which in turn helps identify areas for improvement. Guided by this understanding, we developed the ReCVisu+ tool with enhanced interactive visualization supports to RE practitioners. We further conducted a case study to explore how ReCVisu+ might help keep requirements on track in practice. The study uncovers four RE tasks (overview, anomaly handling, heterogeneity utilization, and causality reasoning) that could best be supported by VA, and shows how increased visual interactivity could lead to actionable decisions.

From our experience, we feel that VA has a rich value in helping requirements analysts, decision makers, and other stakeholders to rapidly extract insights from the flood of data. Our future work includes refining the design of ReCVisu+ and implementing advanced features to facilitate causality analysis. As our research collaboration with the SWP team continues, we also plan to conduct further empirical studies to quantitatively investigate the cost and benefit of the VA supports for RE. Finally, we want to study the possible usage barriers and find principled ways to overcome the barriers in order to tap the full potential of VA approaches in RE.

Acknowledgement

We thank the partner company for the generous support throughout the case study, especially for sharing their data, time, and expertise. We also thank the anonymous reviewers for their constructive comments. The work is in part supported by the U.S. NSF (National Science Foundation) Grant CCF-1238336.

References

[1] O. Gotel, F. T. Marchese, and S. J. Morris, "The potential for synergy between information visualization and software engineering visualization," in *IV*, 2008, pp. 547–552.

[2] J. J. Thomas and K. A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.

[3] D. A. Keim, F. Mansmann, J. Schneidewind, J. J. Thomas, and H. Ziegler, "Visual analytics: scope and challenges," in *Visual Data Mining — Theory, Techniques and Tools for Visual Analytics*, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Springer, 2008, pp. 76–90.

[4] J. R. Cooper Jr., S.-W. Lee, R. A. Gandhi, and O. Gotel, "Requirements engineering visualization: a survey on the state-of-the-art," in *REV*, 2009, pp. 46–55.

[5] R. A. Gandhi and S.-W. Lee, "Visual analytics for requirements-driven risk assessment," in *REV*, 2007, article no. 6.

[6] S. Reddivari, Z. Chen, and N. Niu, "ReCVisu: a tool for clustering-based visual exploration of requirements," in *RE*, 2012, pp. 327–328.

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

[8] A. Noack, "Energy models for graph clustering," *Journal of Graph Algorithms and Applications*, vol. 11, no. 2, pp. 453–480, 2007.

[9] N. Niu, S. Reddivari, A. Mahmoud, T. Bhowmik, and S. Xu, "Automatic labeling of software requirements clusters," in *SUITE*, 2012, pp. 17–20.

[10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Academic Press, 2000.

[11] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2004.

[12] J. N. och Dag, V. Gervasi, S. Brinkkemper, and B. Regnell, "A linguistic-engineering approach to large-scale requirements management," *IEEE Software*, vol. 22, no. 1, pp. 32–39, January/February 2005.

[13] R. Gacitua, P. Sawyer, and V. Gervasi, "Relevance-based abstraction identification: technique and evaluation," *Requirements Engineering*, vol. 16, no. 3, pp. 251–265, September 2011.

[14] J. Horkoff and E. Yu, "Visualizations to support interactive goal model analysis," in *REV*, 2010, pp. 1–10.

[15] G. Mussbacher, D. Amyot, and M. Weiss, "Visualizing early aspects with use case maps," *Transactions on Aspect-Oriented Software Development*, vol. 3, pp. 105–143, 2007.

[16] K. Wnuk, B. Regnell, and L. Karlsson, "What happened to our features? visualization and understanding of scope change dynamics in a large-scale industrial setting," in *RE*, 2009, pp. 89–98.

[17] R. Seater, D. Jackson, and R. Gheyi, "Requirement progression in problem frames: deriving specifications from requirements," *Requirements Engineering*, vol. 12, no. 2, pp. 77–102, April 2007.

[18] I. Kwan, D. Damian, and M.-A. Storey, "Visualizing a requirements-centred social network to maintain awareness within development teams," in *REV*, 2006, article no. 7.

[19] C. D. Hansen and C. R. Johnson, *Visualization Handbook*. Academic Press, 2004.

[20] S. K. Card, J. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[21] T. Reinhard, S. Meier, and M. Glinz, "An improved fisheye zoom algorithm for visualizing and editing hierarchical models," in *REV*, 2007, article no. 9.

[22] R. Matulevicius and P. Heymans, "Visually effective goal models using KAOS," in *RIGiM*, 2007, pp. 265–275.

[23] P. Heim, S. Lohmann, K. Lauenroth, and J. Ziegler, "Graph-based visualization of requirements relationships," in *REV*, 2008, pp. 51–55.

[24] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *VL*, 1996, pp. 336–343.

[25] V. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric paradigm," *Encyclopedia of Software Engineering*, vol. 2, pp. 528–532, 1994.

[26] N. Niu and S. Easterbrook, "So, you think you know others' goals? A repertory grid study," *IEEE Software*, vol. 24, no. 2, pp. 53–61, March/April 2007.

[27] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. S. K. Yu, "Evaluating goal models within the goal-oriented requirement language," *International Journal of Intelligent Systems*, vol. 25, no. 8, pp. 841–877, August 2010.

[28] M. S. Feather, S. L. Cornford, J. D. Kiper, and T. Menzies, "Experiences using visualization techniques to present requirements, risks to them, and options for risk mitigation," in *REV*, 2006, article no. 10.

[29] R. K. Yin, *Case Study Research: Design and Methods*. Sage Publications, 2003.

[30] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer, 2007, pp. 285–311.

[31] J. Wood and D. Silver, *Joint Application Development*. Wiley, 1995.

[32] S. Haag, M. Cummings, and D. J. McCubbrey, *Management Information Systems for the Information Age*. McGraw-Hill, 2003.

[33] L. Richards, *Handling Qualitative Data: A Practical Guide*. Sage, 2005.

[34] V. Tzerpos and R. C. Holt, "The orphan adoption problem in architecture maintenance," in *WCRE*, 1997, pp. 76–82.

[35] J. Cleland-Huang, B. Berenbach, S. Clark, R. Settimi, and E. Romanova, "Best practices for automated traceability," *IEEE Computer*, vol. 40, no. 6, pp. 27–35, June 2007.

[36] A. Mahmoud, N. Niu, and S. Xu, "A semantic relatedness approach for traceability link recovery," in *ICPC*, 2012, pp. 183–192.

[37] D. L. Moody, P. Heymans, and R. Matulevicius, "Visual syntax does matter: improving the cognitive effectiveness of the i^* visual notation," *Requirements Engineering*, vol. 15, no. 2, pp. 141–175, June 2010.

[38] K. Herrmann, "Visualization of release planning," in *REV*, 2006, article no. 8.

[39] A. Telea and L. Voinea, "Case study: visual analytics in software product assessments," in *VISSOFT*, 2009, pp. 65–72.

[40] D. Nestor, S. Thiel, G. Botterweck, C. Cawley, and P. Healy, "Applying visualisation techniques in software product lines," in *SOFTVIS*, 2008, pp. 175–184.

[41] K. Gallagher, A. Hatch, and M. Munro, "Software architecture visualization: an evaluation framework and its application," *IEEE Transactions on Software Engineering*, vol. 34, no. 2, pp. 260–270, March/April 2008.

[42] P. Caserta and O. Zendra, "Visualization of the static aspects of software: a survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 7, pp. 913–933, July 2011.