

Rationale Management Challenges in Requirements Engineering

Peng Liang
State Key Lab of
Software Engineering
Computer School
Wuhan University, China
liangp@sklse.org

Paris Avgeriou
Department of
Computing Science
University of Groningen
The Netherlands
paris@cs.rug.nl

Keqing He
State Key Lab of
Software Engineering
Computer School
Wuhan University, China
hekeqing@sklse.org

Abstract—Rationale and rationale management have been playing an increasingly prominent role in software system development mainly due to the knowledge demand during system evaluation, maintenance, and evolution, especially for large and complex systems. The rationale management for requirements engineering, as a commencing and critical phase in software development life cycle, is still under-exploited. In this paper, we first survey briefly the state-of-the-art on rationale employment and applications in requirements engineering. Secondly, we identify the challenges in integrating rationale management in requirements engineering activities in order to promote further investigations and define a research agenda on rationale management in requirements engineering.

Keywords—rationale management; requirements rationale knowledge; requirements engineering; knowledge management

I. INTRODUCTION

Rationale and rationale management have been playing an increasingly prominent role in software system development and software engineering community mainly due to the knowledge demand during system evaluation, maintenance, and evolution, especially for large and complex systems (e.g., Ultra-Large-Scale Systems [1]). State-of-the-art research results and industrial practices on rationale usage and rationale management in software engineering have been extensively reported in [2], [3]. Rationale management for Requirements Engineering (RE), as a commencing and critical phase in software development life cycle, is still under-exploited in a perspective of integrated RE process [4]. Many research works/contributions and results on rationale applications and usage in RE have been reported in an isolated manner, and the result is a mosaic of RE activities and rationale artifacts rather than an operable RE process and a solid product. Engineering-style research is needed to investigate how to integrate requirements rationale into a coherent RE process.

Rationale¹ was historically proposed in the context of software design as means of presenting the “why” of a design (decision). In this paper, we follow the suggestions brought forward by Dutoit and Paech [5] to extend the scope of rationale in design phase to all the software development

phases, including RE, system design, implementation, testing, and maintenance. Actually, RE process has already been regarded as a decision-making process in a certain perspective [6]. In this paper, we focus on the rationale usage and rationale management in RE process from high-level business goals to features, and detailed functional and non-functional requirements. We identify the challenges of integrating rationale management in RE activities based on a brief survey in this area. We hope that this work can promote further investigations and define an research agenda on rationale management in RE.

The rest of this paper is organized as follows. A brief survey on rationale usage in RE are presented in Section II. Based on the survey results, several major challenges in integrating requirements rationale management into RE process are outlined and discussed with detailed research questions presented in Section III. The major points of this paper are concluded in Section IV with future work plan.

II. RATIONALE USAGE IN REQUIREMENTS ENGINEERING

Rationale technique has been employed in RE for years. Burge *et al.* made a thorough literature survey on how rationale can support RE in [7]. Their survey is organized according to specific RE activities and problems (e.g., obtaining requirements and requirements change), but not the integrated RE process [4]. In this section, we make a brief survey on the usage of rationale in RE based on the survey by Burge *et al.* [7]². We categorize these work according to the rationale support to various RE phases in RE process from requirements elicitation to validation, and also to RE process improvement. Note that obtaining requirements is one of the survey areas by Burge *et al.* [7], and it includes requirements elicitation phase in RE process. The major difference between Burge *et al.*'s work and our survey is that we try to identify rationale usage from a RE process perspective so that rationale can be supported in an

²The related work surveyed by Burge *et al.* is not repeated in this paper. Readers who are interested in rationale usage in obtaining requirements, requirements traceability, goal-based RE, and requirements change support can refer to [7].

¹The term *design rationale* is most often used in the literature.

integrated way with RE process. Consequently, the objective of this section is to get a panorama view on rationale usage in RE process.

Requirements elicitation is the first step to discover early requirements, and rich rationale knowledge exists in this RE phase on “what” and “why” of a system, including source of requirements, business trade-offs, etc. Rooksby *et al.* propose a hybrid approach to elicit early requirements and assist the requirements negotiation by combining the knowledge management (cognitive mapping) and design rationale (IBIS [8]) techniques [9]. The advantage of their approach is that it can capture the rationale during the elicitation and negotiation since they argue that eliciting rather than managing rationale is most-wanted in requirements elicitation phase. Nguyen and Swatman regard the RE process as a cognitive problem solving process, and argue that requirements are actually not “elicited” but “created” by incremental refinement of the problem space [10]. Their suggested approach supports creating requirements by using design rationale technique (IBIS [8] and QOC [11]).

Requirements negotiation takes place when stakeholders propose conflicting requirements, and it tries to reconcile conflicting views and generate a consistent set of requirements in the end. Boehm and Kitapci proposed the WinWin model (a rationale-based requirements negotiation model) for capturing requirements rationale knowledge during requirements negotiation, and in turn converge on a mutually satisfactory or win-win set of requirements using captured rationale knowledge [12]. Grunbacher proposed EasyWinwin approach that extended the WinWin approach with some simplifications in the process [13]. WinWin and EasyWinwin approaches have been successfully applied in more than 100 real-world projects in various domains. The WinWin negotiation model is also extended from IBIS design rationale model by including stakeholders’ win-win relationship as the success criterion.

Requirements analysis is to understand/communicate the requirements, detect their overlaps and conflicts. Rationale knowledge can be used in requirements analysis either manually (e.g., for requirements communication, understanding) or automatically (e.g., for requirements reasoning, prioritization). Breitling *et al.* introduced Exemplary Business Process Modeling (EBPM) approach in requirements analysis in order to improve the requirements communication between requirements analysts and designers by gathering rationale in both business process and design level [14]. This approach can smoothly connect the rationale knowledge from requirements to design in an integrated rationale framework (i.e., EBPM). Laurent *et al.* develop a probabilistic traceability model (i.e., a kind of rationale model) for requirements combined with a standard hierarchical clustering algorithm to cluster and prioritize requirements automatically [15]. This approach is especially useful for large and complex projects when traditional requirements prioritization approaches do

not scale well for managing huge amount of requirements. We also propose an integrated approach of basing distributed requirements analysis through requirements reasoning with underlying requirements rationale model (RRM) and reasoning rules [16].

Requirements specification is to write down the requirements in a way that stakeholders and software developers can understand (e.g., use case specification, traceability specification, etc). Traditional requirements specification only specify “what” the requirements are, but not “why” these requirements are specified. This situation leads to problems when requirements get evolved when stakeholders change their mind or system is being updated. For example, some requirements may depend on other high-level requirements, or get selected with a trade-off about several business goals, and the rationale knowledge is important for changing/evolving the requirements. Dutoit and Paech propose to integrate rationale element (based on QOC [11]) into use case specification approach with tool support [17]. Burge and Brown developed the Software Engineering Using Rationale system (SEURAT) as an Eclipse plugin³ to support requirements traceability specification by relating functional and non-functional requirements to design artifacts (and code modules) as the argumentation [18]. Their approach improve requirements rationale recovery and ensure the consistency between requirements and implementations. Heindl and Biffel propose to enhance requirements traceability specification by including requirements rationale, hence facilitate project risk management with requirements clarification in highly distributed projects [19]. Their approach is in line with value-based RE [20] since the requirements rationale is driven by *business objectives* (values).

Requirements validation is to go back to the system stakeholders and check if the requirements are what they really need. The requirements validation should be performed by stakeholders manually. Rationale knowledge can be captured and (re)used in this phase similar to the manual usage of requirements rationale in requirements analysis.

RE process improvement. Similar to software development process improvement (e.g., CMMI), RE process improvement ensures that software requirements quality is continually being improved. Palyagar and Richards attempt to standardize a method for RE process improvement by capturing and reusing rationale to justify the improvement actions (e.g., understanding the root causes of the problem) [21]. Fricker and Stoiber use design rationale to discover the context/variability of software product line to customize and improve the RE process [22]. Unlike the other rationale usage in RE activities, which focus on capturing requirements rationale directly, these works stress upon the RE process improvement using rationale knowledge, which indirectly affects the quality of requirements.

³<http://www.users.muohio.edu/burgeje/SEURAT/>

There are many other works on using rationale in general RE. Thurimella addresses variability management of software product line systems using issue-based approach (a kind of rationale representation) in product line RE [23]. Stoiber and Glinz address similar problems in product line RE by using rationale management to document tacit product line requirements knowledge [24]. Maalej and Thurimella propose to employ recommendation systems in RE [25], in which requirements rationale plays a key role for the recommendation and decision-making in RE process.

III. RESEARCH CHALLENGES

Those work on applying rationale and rationale management in RE presented in the previous section is rather isolated, which targets specific RE problems and lacks of a systematic support to the integrated RE process from requirements elicitation to validation [4]. A number of research challenges on rationale management in RE remain, which are discussed in details in this section. Note that we do not argue that the identified challenges below are comprehensive and complete, but these challenges are those that need to be addressed in a high priority according to our experiences and survey results.

What content should a shared body of requirements rationale knowledge include? Various models and methodologies for capturing rationale knowledge for information systems development have been developed (e.g., IBIS [8], DRL [26], and QOC [11]), especially in the field of *design rationale*. As mentioned in Section I, Dutoit and Paech propose to extend the application of *design rationale* in design phase to all the software development phases, including RE [5]. In rationale management, participants aim at recording every *decision* (to a *decision topic*), its *alternative decisions*, and the underlying *arguments* leading to that *decision* (e.g., influencing factors, criteria, and negotiations). Actually, the requirements negotiation model used in WinWin approach discussed in Section II is a kind of rationale model for capturing requirements rationale knowledge [12], in which *issues* are *decision topics* to be decided or needing actions, *options* are *alternatives* for resolving *issues*, and *agreements* are the *decisions* that have been taken for requirements (*win conditions*). Requirements rationale knowledge is deemed to be beneficial to system development and evolution [5], however, the exhaustive forms of rationale management require substantial investments of resources before participants can readily use and benefit from the rationale knowledge. Consequently, there is always a trade-off between the cost and benefit for capturing and using rationale knowledge. In RE context, the cost is mainly caused by how much rationale content should be captured in RE activities. In [16], we propose a simple requirements rationale model to partially address the issue of automatic requirements analysis by requirements reasoning. This requirements rationale model can be extended according to different contexts of RE activities.

This challenge can be further detailed as: (1) is it possible to define a shared body of knowledge on requirements rationale, which can acts as a core model to manage the requirements rationale knowledge in an integrated repository and to be easily extended in various RE activities? (2) how to extend this core model to meet different usage scenarios in RE activities.

How to manage requirements rationale knowledge in the context of large and complex projects and evolving systems? RE process is a basically a cognitive problem solving process to a target system by stakeholders. The granularity of requirements is decomposed from business goals, features, high-level use cases to detailed functional and non-functional requirements along with this cognition process. From one perspective, Requirements rationale knowledge across different levels of requirements is interweaved and mostly implicitly existing in stakeholders' head during this cognition process. Current knowledge management techniques provide little support for managing such implicit and complex rationale knowledge, and furthermore do not scale well for managing requirements knowledge in large and complex projects with a huge number of stakeholders and requirements. From another perspective, due to time pressure and lack of resources, especially in large and complex projects, it is not practical to capture rationale knowledge for all requirements. There is a need to identify a subset of requirements to capture rationale with a balance of cost and benefit [23]. The evolution of system makes the management of requirements rationale knowledge (e.g., updating knowledge, checking knowledge consistency) even challenging. In [27], we propose to reuse the research outcomes for architectural knowledge management in RE. We target on specific requirements management problems (i.e., requirements traceability and integration of RE activities in RE process), and present knowledge-based ideas to tackle them by introducing emerging and concrete results from the architectural knowledge community. This challenge can be further refined into: (1) how to capture implicit requirements rationale knowledge into explicit knowledge which is cost-beneficial for RE activities? (2) how to link requirements rationale knowledge in different levels effectively to all the requirement artifacts generated from different RE phases? (3) how to effectively manage (represent, operate, share, update, and maintain) requirements rationale knowledge to support its usage in RE activities?

How to manage requirements rationale knowledge in distributed development? More and more projects are running in geographically distributed environments, and Global Software Development (GSD) is becoming a norm in the software industry. This trend makes a great impact in RE as well, and RE practice has been a key challenge in GSD [28]. Meanwhile, wiki tools have been extensively employed in distributed software development to collaboratively document software artifacts (e.g., source code, documentation,

project work plans, bug reports, and so on) for decades. We surveyed wiki and semantic wiki techniques and related tools for their potential capabilities and usability in RE process [29]⁴. Based on this survey, we argue that semantic wiki, as a lightweight and semantic/social Web based knowledge management tool, is promising for effective management of requirements rationale knowledge in a distributed development context, and is a good candidate to address the cost-benefit issue since it can capture rationale knowledge collaboratively and share/use the knowledge as soon as it is captured. Semantic wiki provides an open platform for better communication, collaboration, and documentation of requirements rationale knowledge with the support of underlying ontology (e.g., requirements rationale ontology). The challenge can be further presented as: (1) how can semantic wiki (or other collaboration tools) support requirements rationale knowledge management in a distributed context? (2) how can semantic wiki (or other collaboration tools) be accommodated/integrated into existing RE tools (or development environment)?

How to (re)use requirements rationale knowledge in RE activities? The ultimate goal of requirements rationale management is to (re)use it in various RE activities (e.g., understanding requirements, detecting conflicting requirements, prioritizing requirements, and refining requirements etc). Various part of requirements rationale knowledge can be (re)used according to the usage scenario in different RE activities. For example, requirements refinement patterns is the key rationale knowledge in requirements refinement activity for elaborating high-level goals to detailed (non)-functional requirements [30]. In [27], we propose to employ the research outcomes for architectural knowledge reuse in RE for requirements rationale knowledge management. This challenge can be further extended into: (1) what RE activities can be mostly supported by (re)using requirements rationale knowledge, and which part of requirements rationale knowledge is employed in these reusing activities (this issue is also concerned with the first challenge on defining a shared body of requirements rationale knowledge)? (2) what is the benefit of (re)using requirements rationale knowledge in order to trade off the cost and benefit of capturing and managing requirements rationale knowledge.

How to visualize requirements rationale knowledge to assist the understanding and communication of requirement artifacts? Traditional RE techniques are heavily based on the use of unstructured text and lists (e.g., Office Word and Excel). With the increasing complexity of software requirements, understanding and communicating requirements become more difficult, and the intercultural misunderstanding is a major cause to project failure or extended lead

⁴Other techniques and tools besides wiki/semantic wiki should also be examined for collaborative management of requirements rationale knowledge, and we stress upon semantic wiki in this paper as one of promising solutions in this area.

time in a distributed development context [31]. Visualization techniques have already been employed to overcome the communication problems in RE fields from requirements elicitation to modeling [32]. Rationale knowledge visualization, in the similar way, can assist stakeholders in understanding the reasoning process underlying the decisions on requirements. Some rationale visualization tools (e.g., Compendium⁵) based on IBIS or QOC rationale notations have been used for early requirements elicitation [9]. We also employed Compendium to improve the understandability of architecture design through the visualization of architectural design rationale [33], [34], and that experience can be beneficial in requirements rationale visualization as well. The challenge in this topic include sub-topics: (1) which part of requirements rationale knowledge needs to and can be visualized in RE activities, especially for requirements of large and complex systems? (2) how rationale knowledge visualization can assist various rationale usage scenarios in RE activities?

How to get users to contribute/use requirements rationale knowledge in practice? Rationale management approaches have been proposed for decades in information systems development, such as IBIS (Issue-Based Information System) [8] and QOC (Questions, Options, and Criteria) [11]. It is further systematically introduced in software engineering, and the state-of-the-art results are included in the book edited by Dutoit *et al.* [2], but the actual use of rationale management in practical software development is rather limited. Similar issue has been identified in architectural knowledge management as well [35]. The obstacles that prevent users from using rationale knowledge in RE are mainly due to two points: the cost-benefit (ROI) of rationale knowledge management effort and the concern of losing (personal) intellectual properties through explicit rationale knowledge (e.g., only business analysts know the rationale about why a business requirement gets selected). The ROI of rationale knowledge management can be improved by a guided process on using rationale knowledge in RE process, similar to the unified development process for the quality improvement of software products. The ROI also depends on the use of tools that support rationale knowledge management in RE. For the concern of losing intellectual properties, bonus for contributing and recording requirements rationale knowledge is an alternative solution, and personalized knowledge management strategy (knowledge is kept with its creator and shared by personal communication) [36] is another alternative solution for capturing, sharing, and using implicit requirements rationale knowledge in practice. This challenge includes (1) the development of a process for using rationale knowledge in RE process, and (2) the selection of appropriate knowledge strategies for contributing/using requirements rationale knowledge in practice.

⁵<http://compendium.open.ac.uk/institute/>

IV. CONCLUSIONS AND FUTURE WORK

This work makes a first step in investigating how the rationale and rationale management can be systematically integrated into RE process. The major contributions of this work are two-fold: it presents a brief survey on major outcomes on applying rationale in RE activities, and based on the survey, it further outlines the challenges in integrating rationale knowledge management into RE process. The research challenges identified in Section III are not isolated, but interrelated with each other, for example, the shared body of requirements rationale knowledge can act as the underlying domain model for rationale knowledge management, and requirements rationale visualization can improve the (re)use of requirements rationale knowledge.

Knowledge management is beneficial but also costly, and considerable resources, time, and effort are required before benefits become visible [37]. This situation also holds for requirements rationale knowledge management since project stakeholders (e.g., customers, requirements engineers) who focus on completing the requirements specification of current project on time, but not helping the next project succeed, often considers rationale management (capturing, recording, and maintaining knowledge) a burden. It is crucial for the acceptance of rationale management in RE practice to leverage the benefit and cost. In this sense, appropriate tool support for rationale knowledge management in RE is one more challenge.

The challenges on rationale management in RE identified in this paper will be approached in the next step by following an action research cycle that is iterative and incremental from the problem (i.e., originated from the challenges) to the solution space through the verification and validation of industrial cases [38]. Our research will give a particular focus on large and complex projects running in a distributed development context.

ACKNOWLEDGMENTS.

This research has been partially sponsored by the Natural Science Foundation of China (NSFC) under Grant No. 60950110352, STAND: Semantic-enabled collaboration Towards Analysis, Negotiation and Documentation on distributed requirements engineering, the National Basic Research Program of China (973) under Grant No. 2007CB310801, RE4ULS: Requirements Engineering: Fundamentals for Ultra-Large-Scale Systems. Special thanks to all the reviewers for their constructive comments.

REFERENCES

- [1] L. Northrop, P. Feiler, R. Gabriel *et al.*, *Ultra-Large-Scale Systems: The Software Challenge of the Future*. SEI, Carnegie Mellon University, 2006.
- [2] A. H. Dutoit, R. McCall, I. Mistrik, and B. Paech, *Rationale Management in Software Engineering*. Springer, 2006.
- [3] J. Burge, J. Carroll, R. McCall, and I. Mistrik, *Rationale-Based Software Engineering*. Springer, 2008.
- [4] I. Sommerville, "Integrated Requirements Engineering: a Tutorial," *IEEE software*, pp. 16–23, 2005.
- [5] A. H. Dutoit and B. Paech, "Rationale Management in Software Engineering," in *Handbook of Software Engineering and Knowledge Engineering*. World Scientific Publishing, 2001, pp. 787–815.
- [6] A. Aurum and C. Wohlin, "The Fundamental Nature of Requirements Engineering Activities as a Decision-making Process," *Information and Software Technology*, vol. 45, no. 14, pp. 945–954, 2003.
- [7] J. Burge, J. Carroll, R. McCall, and I. Mistrik, "Rationale and Requirements Engineering," in *Rationale-Based Software Engineering*. Springer, 2008, pp. 139–153.
- [8] W. Kunz and H. Rittel, *Issues as Elements of Information Systems*. Institute of Urban and Regional Development, University of California, 1970.
- [9] J. Rooksby, I. Sommerville, and M. Pidd, "A Hybrid Approach to Upstream Requirements: IBIS and Cognitive Mapping," in *Rationale Management in Software Engineering*. Springer, 2006, pp. 137–154.
- [10] L. Nguyen and P. Swatman, "Promoting and Supporting Requirements Engineering Creativity," in *Rationale Management in Software Engineering*. Springer, 2006, pp. 209–230.
- [11] A. MacLean, R. Young, V. Bellotti, and T. Moran, "Questions, Options, and Criteria: Elements of Design Space Analysis," *Human-Computer Interaction*, vol. 6, no. 3, pp. 201–250, 1991.
- [12] B. Boehm and H. Kitapci, "The WinWin Approach: Using a Requirements Negotiation Tool for Rationale Capture and Use," in *Rationale Management in Software Engineering*. Springer, 2006, pp. 173–190.
- [13] P. Gruenbacher, "Collaborative Requirements Negotiation with EasyWinWin," in *Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA)*. IEEE Computer Society, 2000, pp. 954–958.
- [14] H. Breitling, A. Kornstädt, and J. Sauer, "Design Rationale in Exemplary Business Process Modeling," in *Rationale Management in Software Engineering*. Springer, 2006, pp. 191–208.
- [15] P. Laurent, J. Cleland-Huang, and C. Duan, "Towards Automated Requirements Triage," in *Proceedings of the 15th IEEE International Requirements Engineering Conference (RE)*. IEEE Computer Society, 2007, pp. 131–140.
- [16] P. Liang, P. Avgeriou, and V. Clerc, "Requirements Reasoning for Distributed Requirements Analysis using Semantic Wiki," in *Proceedings of the International Workshop on KNOWledge engINeering in Global software development (KNOWING)*. IEEE Computer Society, 2009, pp. 388–393.

- [17] A. H. Dutoit and B. Paech, "Rationale-based Use Case Specification," *Requirements Engineering*, vol. 7, no. 1, pp. 3–19, 2002.
- [18] J. Burge and D. Brown, "Supporting Requirements Traceability with Rationale," in *Proceedings of International Symposium on Grand Challenges in Traceability (GTC)*. ACM, 2007, pp. 67–75.
- [19] M. Heindl and S. Biffl, "Risk Management with Enhanced Tracing of Requirements Rationale in Highly Distributed Projects," in *Proceedings of the 1st International Workshop on Global Software Development for the Practitioner (GSD)*. ACM, 2006, pp. 20–26.
- [20] J. Gordijn and J. Akkermans, "Value-based Requirements Engineering: Exploring Innovative e-commerce Ideas," *Requirements Engineering*, vol. 8, no. 2, pp. 114–134, 2003.
- [21] B. Palyagar and D. Richards, "Capturing and Reusing Rationale Associated with Requirements Engineering Process Improvement: A Case Study," in *Rationale Management in Software Engineering*. Springer, 2006, pp. 391–408.
- [22] S. Fricker and R. Stoiber, "Relating Product Line Context to Requirements Engineering Processes Using Design Rationale," in *Proceedings of Produktlinien im Kontext: Technologie, Prozesse, Business und Organisation (PiK)*. Springer LNCS, 2008, pp. 8–19.
- [23] A. Thurimella, *Issue-based Variability Modeling: Extending Variability Management with Rationale Management*. VDM Verlag, 2009.
- [24] R. Stoiber and M. Glinz, "Modeling and Managing Tacit Product Line Requirements Knowledge," in *Proceedings of the 2nd International Workshop on Managing Requirements Knowledge (MaRK)*. IEEE Computer Society, 2009, pp. 60–64.
- [25] W. Maalej and A. Thurimella, "Towards a Research Agenda for Recommendation Systems in Requirements Engineering," in *Proceedings of the 2nd International Workshop on Managing Requirements Knowledge (MaRK)*. IEEE Computer Society, 2009, pp. 32–39.
- [26] J. Lee and K. Lai, "What's in Design Rationale?" *Human-Computer Interaction*, vol. 6, no. 3, pp. 251–280, 1991.
- [27] P. Liang and P. Avgeriou, "From Architectural Knowledge to Requirements Knowledge Management," SEARCH, University of Groningen, The Netherlands, Tech. Rep. RUG-SEARCH-09-L02, Feb. 2009. [Online]. Available: <http://www.cs.rug.nl/~liangp/download/liang2009akr.pdf>
- [28] T. Gorschek, S. Fricker, R. Felt, C. Wohlin, and M. Mattsson, "1st International Global Requirements Engineering Workshop - GREW'07," *ACM SIGSOFT Software Engineering Notes*, vol. 33, no. 2, pp. 29–32, 2008.
- [29] B. Hoenderboom and P. Liang, "Semantic Wikis Survey for Requirements Engineering," SEARCH, University of Groningen, The Netherlands, Tech. Rep. RUG-SEARCH-09-B01, Jul. 2009. [Online]. Available: <http://www.cs.rug.nl/~liangp/download/hoenderboom2009sws.pdf>
- [30] R. Darimont and A. Van Lamsweerde, "Formal Refinement Patterns for Goal-driven Requirements Elaboration," in *Proceedings of the 4th ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE)*. ACM, 1996, pp. 179–190.
- [31] F. Zarnadt, "What Did You Say? Intercultural Expectations, Misunderstandings, and Communications," in *Proceedings of the 2nd International Conference on Global Software Engineering (ICGSE)*. IEEE Computer Society, 2007, pp. 295–296.
- [32] O. Gotel, F. Marchese, and S. Morris, "On Requirements Visualization," in *Proceedings of the 2nd International Workshop on Requirements Engineering Visualization (REV)*. IEEE Computer Society, 2007, pp. 11–20.
- [33] M. Shahin, P. Liang, and M. Khayyambashi, "Rationale Visualization of Software Architectural Design Decision using Compendium," in *Proceedings of the 25th ACM Symposium on Applied Computing (SAC)*. ACM, 2010, pp. 2367–2368.
- [34] —, "Improving Understandability of Architecture Design through Visualization of Architectural Design Decision," in *Proceedings of the 5th Workshop on SHaring and Reusing architectural Knowledge (SHARK)*. ACM, 2010, pp. 88–95.
- [35] M. Babar, R. de Boer, T. Dingsoyr, and R. Farenhorst, "Architectural Knowledge Management Strategies: Approaches in Research and Industry," in *Proceedings of the 2nd Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent (SHARK/ADI)*. ACM, 2007, pp. 2–8.
- [36] T. Morten, N. Nohria, and T. Tierney, "What's Your Strategy for Managing Knowledge," *Harvard Business Review*, vol. 77, no. 2, pp. 106–116, 1999.
- [37] I. Rus and M. Lindvall, "Knowledge Management in Software Engineering," *IEEE Software*, vol. 19, no. 3, pp. 26–38, 2002.
- [38] R. Baskerville, "Investigating Information Systems with Action Research," *Communications of the AIS*, vol. 2, no. 3es, pp. 4–35, 1999.