Views on Software Engineering from the Twin Peaks of Requirements and Architecture

Matthias Galster
University of Canterbury, New Zealand
mgalster@ieee.org

Mehdi Mirakhorli
DePaul University, USA
mehdi@cs.depaul.edu

Jane Cleland-Huang
DePaul University, USA
jhuang@depaul.edu

Janet E. Burge
Miami University, USA
burgeje@muohio.edu

Xavier Franch
Universitat Politècnica de Catalunya, Spain
franch@essi.upc.edu

Roshanak Roshandel
Seattle University, USA
roshanak@seattleu.edu

Paris Avgeriou
University of Groningen, The Netherlands
paris@cs.rug.nl

DOI: 10.1145/2507288.2507323
http://doi.acm.org/10.1145/2507288.2507323

ABSTRACT

The disciplines of requirements engineering (RE) and software architecture (SA) are fundamental to the success of software projects. Even though RE and SA are often considered in isolation, drawing a line between RE and SA is neither feasible nor reasonable as requirements and architectural design impact each other. This observation motivated the Twin Peaks model that was the subject of the Second International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks@ICSE 2013). TwinPeaks@ICSE 2013 was held in conjunction with the 35th International Conference on Software Engineering 2013 in San Francisco, CA. The workshop aimed at providing a forum for researchers, practitioners and educators from the areas of RE and SA to discuss their experiences, forge new collaborations, and explore innovative solutions that address the challenges that occur when relating RE and SA. The workshop provided participants with an opportunity to become familiar with the relationship between RE and SA in the broader context of software engineering, rather than in an isolated context of either RE or SA. The workshop featured one industrial keynote, five research paper presentations, two invited talks and four working group discussions.

Keywords

Requirements engineering, software architecture, Twin Peaks model.

1. INTRODUCTION

Drawing a line between software requirements and architecture is neither feasible nor reasonable as requirements and architectural design processes impact each other. Requirements are constrained by what is technically feasible and by time and budget restrictions. On the other hand, feedback from the architecture leads to renegotiating architecture-significant requirements with stakeholders.

The topic of bridging requirements engineering (RE) and software architecture (SA) has been discussed in both the RE and SA communities, but mostly independently. Therefore, the motivation for Second International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks@ICSE 2013) was to bring both communities together in order to identify key issues, explore the state-of-the-art in research and practice, identify emerging trends, and define challenges related to the transition and the relationship between RE and SA. The conceptual foundation for the workshop was the Twin Peaks model proposed by Nuseibeh which suggests an intertwinement of software requirements and architecture to achieve incremental development and speedy delivery [1].

TwinPeaks@ICSE 2013 (http://re.cs.depaul.edu/twinpeaks/ICSE13/) was held in conjunction with the 35th International Conference on Software Engineering (ICSE 2013) in San Francisco, CA. Around 30 participants were registered for the workshop. The workshop was a follow-up event of the First International Workshop on the Twin Peaks of Requirements and Architecture, held at the International Conference on Requirements Engineering in 2012 (http://re.cs.depaul.edu/twinpeaks/RE12/).

2. PRESENTATIONS

The workshop featured one industrial keynote ("Surveying the Twin Peaks") delivered by Rich Hilliard. Rich argued that surveying is essential in the planning and execution of nearly every form of construction. In his talk, Rich surveyed the Twin Peaks of requirements and architecture, their surroundings, geology, morphology, etc. to examine questions, such as what are the Twin Peaks made of, why do requirements and architecture intertwine (they intertwine because of concerns), what exactly intertwines, and are there only two peaks in the Twin Peaks model. For example, according to Rich, non-functional requirements, a term frequently used in the RE community, is a non-category for requirements. Furthermore, Rich argued that "architecture is architecture", i.e., there is no good reason for differentiating "types" of architectures, such as enterprise architecture, system architecture and software architecture. This is because the cognitive processes required to design any of these architectures are the same. Rich argued that only the roles involved in the design of these architectures differ. Also, the required knowledge and expertise may differ depending on the type of architecture.

Prior to the workshop, we invited workshop participants to submit one slide to be presented in one minute. The slide should cover a topic or question that participants were passionate about and interested in discussing with other workshop participants. We received nine single slides of which some posed questions (e.g., how can we make requirements architecture friendly) while others proposed potential solutions to problems related to intertwining requirements and architecture (e.g., how can we bridge the gap between requirements and architecture based on a distributed cognition theory). These short presentations triggered interesting discussions among participants.

Based on a peer reviewer process, the workshop selected five research papers for inclusion in the proceedings. The papers were presented in 20-minute presentations. The list of papers can be found in the workshop summary [2]. Furthermore, we included two invited talks. Ian Gorton from the SEI talked about tales from the (scientific software) engineering abyss. Leyna Zimdars explored a practitioner's perspective on developing requirements using a twin peaks paradigm. Furthermore, Bashar Nuseibeh, the original author of the Twin Peaks model, joined for a brief interview through Skype.

3. WORKING GROUP DISCUSSIONS

The presentations provided starting points for the discussion in four working group sessions. The following topics and questions were selected for further discussion:

1. Twin Peaks in software engineering (SE) education: How can we improve the understanding of the importance of the interplay between requirements and architecture in software engineering education?
2. Twin Peaks in software product line engineering (SPLE): What is the role of Twin Peaks when engineering systems that are part of a software product line?
3. Twin Peaks and decisions: Does the Twin Peaks paradigm affect requirements and architecture decision making, and if so, how?
4. Twin Peaks and related "spaces": What are the relationships between the requirements and architecture design spaces?

The topics were selected based on the interests of workshop participants, i.e., the selected four topics received the most votes from the participants. We formed groups that established a balance between participants from academia and industry. Thus, all groups discussed both, the industrial and academic perspectives on the topics listed above. The following sections elaborate on the results of the discussions in the working groups.

3.1 Twin Peaks in SE Education

The group explored the shortcomings of existing SE curricula to support the intertwinement of requirements and architecture. A major shortcoming was identified in that requirements and architectures are often taught independently and in a fashion that resembles a waterfall process. The discussion led to a proposed Master's curriculum that would leverage course components, capstone projects and research components. The curriculum intentionally combines teaching requirements and architecture topics in a more coordinated way.

Typical of many software engineering programs, the course component would include a course designated to cover topics related to requirements solicitation, modeling and analysis techniques and another course to cover architecture related topics. The two courses should be taught as co-requisites, be synchronized, and may use a shared project or case study. The synchronous nature of the courses along with the use of a shared case study project will allow the students to traverse the two peaks at the same time while in-depth treatment of topics in each course will offer them the thorough knowledge needed in each discipline. Ideally advanced courses in requirements analysis and software architecture would be offered as electives to provide a more comprehensive coverage of topics.

The capstone project typically offered in the final year of the program should involve industrial partners as customers. Students would have the opportunity to be exposed to a real and true experience of intertwining requirements and architecture. Similarly, the research project should involve industrial partners and explore a topic that is relevant for practitioners from a research perspective.

In addition to these three components, the Master's curriculum should also incorporate programming and development approaches to provide students with hands-on experience and allow them to experience the full intertwinement, from requirements to architecture to detailed design to implementation. This will further prepare students for an industry-sponsored capstone project.

3.2 Twin Peaks in SPLE

The group explored the extension of the Twin Peaks model for a product line context and variability-intensive system. Furthermore, the group explored challenges related to intertwining requirements and architecture in the context of product lines. The following challenges were identified:

1. Consistency: Achieving consistency between requirements and architecture appears to be more difficult in SPLE since requirements in a product line context include requirements that apply to all products of a product line (core or common requirements), and requirements that only apply to some products of the product line (variable requirements).
2. Evolution: Similar as with consistency, evolution usually happens separately for the two types of requirements (core and variable requirements).

The group found that many research prototypes and tools exist for linking requirements and features to architecture elements / artifacts. In this sense, one could argue that software product line engineering enforces the intertwinement of requirements and architectures.

Figure 1 shows an adaptation of the Twin Peaks model in the context of software product line engineering. The main characteristics of this adaptation are outlined below.

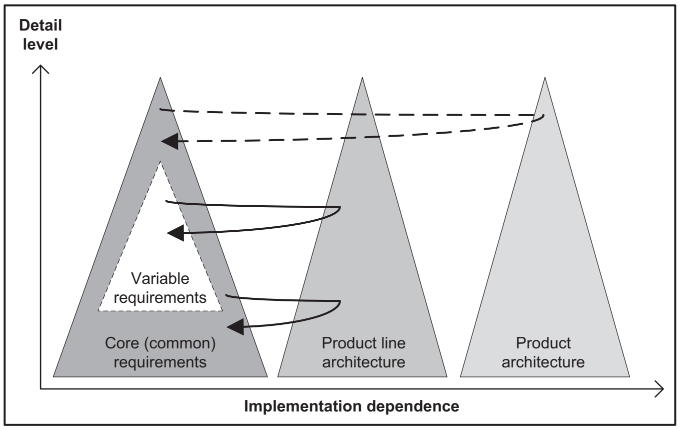
Detail level
Variable requirements
Core (common) requirements
Product line architecture
Product architecture
Implementation dependence

Figure 1. The Twin Peaks model in the context of SPLE

1. Instead of one set of requirements, the model includes one peak that covers two types of requirements. Core (or common) requirements are requirements that must be implemented in all products of a product line. Variable requirements represent variation points in requirements. These requirements may or may not be implemented in a concrete product of the product line, depending on the configuration of the concrete product.

2. Instead of one peak for the architecture, the adapted model contains two peaks related to architecture. One peak represents the product line architecture, i.e., the architecture for all products of a product line. The second peak relates to architecture represents the architecture of a concrete product of a product line.
3. In contrast to the original Twin Peaks model, the adapted model develops progressively more detailed core requirements and product line architecture, variable requirements and product line architecture, and requirements (core and variable) and product architecture. The lines in Figure 1 only show one iteration. However, as with the original Twin Peaks model, multiple iterations to achieve true intertwinement would occur in practice.

3.3 Twin Peaks and Decisions

The discussion was about the similarities and differences of decisions on requirements and architectural decisions. The group concluded that both types of decisions are fundamentally the same thing across system design. The major difference is in the people involved in making the respective decisions and the skills and knowledge required to make these two different types as decisions. However, the cognitive biases involved in the two types of decisions are the same.

3.4 Twin Peaks and Related “Spaces”

There are many “spaces” involved in requirements elicitation / elaboration and in design: requirements space, design space, problem space, and solution space. The group discussed the relationship between the requirements and the design space. Requirements constrain the design space by describing what the system has to do, and sometimes how it has to do it, particularly when the system under development interfaces with previously existing systems. Design exploration elicits requirements both through modeling and simulation and through prototyping. Design decisions constrain requirements—they may involve the reuse of prior (design/code/system) expertise, be limited by personnel availability, and be influenced significantly by politics. Which requirements are documented may depend the level of project risk and on the organizational experience with the domain and application.

One question raised by the group was if there is any way to automate design space search and if the requirements sufficiently define the boundaries of the design space to make this possible. The suspicion was that this often was not the case.

4. CONCLUSIONS

The workshop discussed the applicability of the Twin Peaks model in current software engineering practices as a conceptual approach to visualize and reason about the tight relationship between requirements engineering and software architecture. As briefly reported here, there are some emerging lines of research which call for further efforts in the community. Therefore, the third edition of the workshop will be held at the 21st IEEE International Requirements Engineering Conference (TwinPeaks@RE13, http://re.cs.depaul.edu/twinpeaks/RE13/).

5. ACKNOWLEDGMENTS

We extend our thanks to all who have participated in the organization of the workshop, particularly submitters and presenters, workshop participants, the members of the program committee, and the ICSE organizers.

6. REFERENCES

[1] B. Nuseibeh, Weaving Together Requirements and Architecture, IEEE Software, 34 (2001) 115-117.

[2] P. Avgeriou, J. Burge, J. Cleland-Huang, X. Franch, M. Galster, M. Mirakhorli, Roshanak Roshandel, 2nd International Worskhop on the Twin Peaks of Requirements and Architecture (TwinPeaks 2013), in: 35th International Conference on Software Engineering, IEEE Computer Society, San Francisco, CA, 2013, pp. 1556-1557.