A NUI Based Multiple Perspective Variability Modeling CASE Tool

R. Bashroush

School of Computing, IT and Engineering,
University of East London,
London, United Kingdom
rabih@uel.ac.uk

Abstract. With current trends towards moving variability from hardware to software, and given the increasing desire to postpone design decisions as much as is economically feasible, managing the variability from requirements elicitation to implementation is becoming a primary business requirement in the product line engineering process. One of the main challenges in variability management is the visualization and management of industry size variability models. In this demonstration, we introduce our CASE tool, MUSA. MUSA is designed around our work on multiple perspective variability modeling and is implemented using the state-of-the-art in NUI, multi-touch interfaces, giving it the power and flexibility to create and manage large-scale variability models with relative ease.

Keywords: Software Product Lines, Variability Management, Feature Modeling.

1 Introduction

Software Product-line Engineering (SPLE) has emerged as a major strategy for maximizing reuse when a family of related software systems is developed. In this approach, commonality-variability analysis [1] (Variability Management - VM) of the member products is a major phase of the process and plays an important role in its success.

One of the main challenges within VM is the handling and visualizing "industry-size" models which usually comprise a large number of variability points along with the dependency relationships that exist among them. The challenge comes from the large amount of information captured within a model (business related, dependency and relationships, etc.) as well as the current techniques and I/O devices used to visualize the model which do not inherently scale.

The MUSA CASE tool was designed to overcome these challenges. MUSA is based on our successful work on multiple-perspective based variability management which provides a rich modeling framework while using the concept of separation-of-concerns to alleviate the problem of information overloading. MUSA implements this theory using a mind-mapping modeling approach over the state-of-the-art in HCI, the multi-touch Microsoft Surface [2]. This provides a scalable solution that taps on the latest in Natural User Interface (NUI) [3] design providing an intuitive and large display for VM. In addition, the MUSA solution provides interfaces over other multi-touch platforms including Windows 7 (using its native multi-touch support).

The theory behind MUSA is highlighted in section 2. An overview of the MUSA CASE tool is then presented in section 3. Finally, section 4 ends with related work and conclusion.

2 Theoretical Background

The Four Views Model (4VM) forms the theoretical foundation upon which MUSA is designed as a Proof-of-Concept. The original version of the 4VM can be found here [4] and to appear here [5].

It is generally agreed that different stakeholders have interest in considering different views of the product line variability model [4],[6]. So, it is important for a VM mechanism to be able to extract and present relevant information about the family model in dedicated views for different groups of stakeholders (users, system analysts, developers, etc.). This could considerably contribute to alleviating the graphical overload when showing all the information in one view (as compared to using multiple views). This is one of the core concepts behind 4VM.

The 4VM proposes a four view presentation of the feature model. The views are:

- *Business View*: where the information related to the project management, cost/benefit analysis, closed/open sets of features, etc. is presented.
- *Hierarchical & Behavioral View*: where the way the different features are organized (usually presented in a tree structure) along with the behavior attached to each feature is presented.
- *Dependency & Interaction View*: where the dependency and interaction among features is presented.
- *Intermediate View*: where some design decisions are injected into the feature model to take it one step further towards the architecture domain in an attempt to bridge the gap between the feature model and the system architecture.

For further information about 4VM, please refer to [4],[5].

3 Technical Foundation

MUSA was funded as a Proof-of-Concept project to demonstrate the theoretical foundation provided in 4VM. The MUSA system provides an end-to-end variability management solution as shown in Figure 1 below. MUSA provides a rich and collaborative interface to elicit and manage requirements and variability from stakeholders while allowing for appropriate access to the variability model to different teams including: implementation, testing and deployment teams. In addition, MUSA automates model verification (with the use of SAT solvers) and maintains consistency among the different views with the help of a centralized Database (as shown in Fig. 1).

This is the first official demonstration of the toolset and will focus on the interface that is used for variability management and requirements elicitation by Software Architects/Requirements engineers. The main features of this interface are:

- Based on the Microsoft Surface platform [2], it provides a large gesture based interface for managing the variability model.
- The interface design principles followed (360-D UI and NUI [3]) support a seamless multi-user simultaneous interaction and collaborative environment.
- The variability model itself is implemented using a mind-mapping approach based on hyperbolic trees providing an unprecedented potential for scalability

MUSA is considered among the very first CASE tools to move into the NUI space in order to overcome scalability issues.

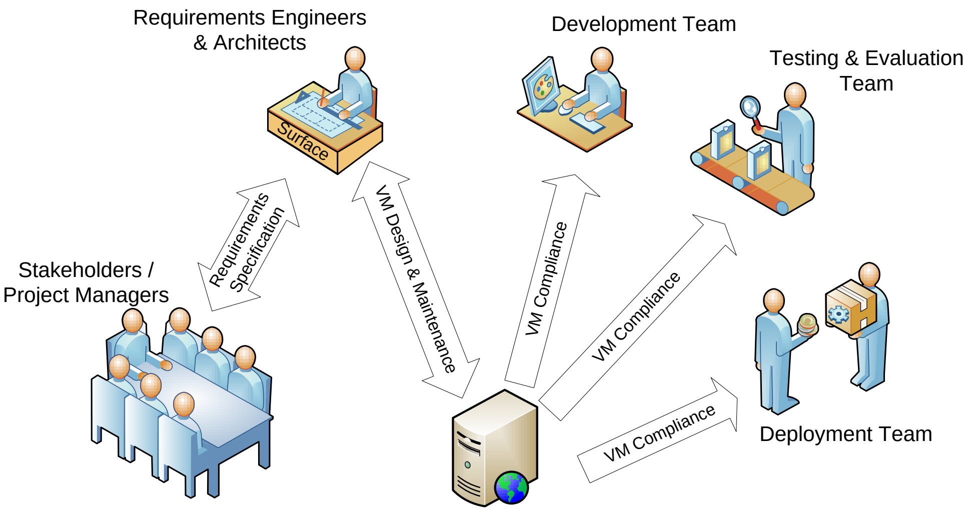

Fig. 1. The end-to-end MUSA System overview

4 Conclusion and Related Work

Over the past few years, a number of VM approaches have been developed ranging from research techniques [7],[8],[9] to commercial products [10],[11],[12].

The major challenge for most research techniques is scalability. The scalability issue arises from the graphical modeling techniques traditionally adopted (e.g. trees) and the I/O devices used (standard keyboard, mouse, and monitors). Although virtual reality technologies have been recently reported as being explored as a potential approach for VM, it is hard to see how such techniques could make their way to commercial environments due to the difficulty involved in integrating such approaches within existing industrial development settings.

Commercial products on the other hand have managed scalability by largely moving away from graphical representation of models. File system tree like structures and even text listings (e.g. using MS Excel sheets) have been seen in use. Although such approaches scale and are in industrial use, adopting NUI interfaces such as the one we implemented in MUSA will increase productivity, time-to-market and allow for the creation and management of larger and more complex product families.

Acknowledgments. The work on the MUSA project has been funded by the European RD Fund through INI under the Proof of Concept funding scheme [2008-2010]. It has just received further funding under the Challenge Fund scheme at the University of East London [2010-2011]. We thank all the postgraduate students at the CITE school at UEL who contributed to some of the testing and development of the MUSA toolset as part of their thesis work.

References

1. K. C. Kang, J. Lee, and P. Donohoe, "Feature-Oriented Product Line Engineering," IEEE Software, vol. 19, pp. 58-65, (2002)
2. Microsoft Surface, http://www.microsoft.com/surface/
3. Natural User Interfaces, http://en.wikipedia.org/wiki/Natural_user_interface
4. R. Bashroush, I. Spence, P. Kilpatrick, TJ Brown, and C. Gillan. "A Multiple Views Model for Variability Management in Software Product Lines," Proceedings of the Second International Workshop on Variability Modelling of Software-intensive Systems. Essen, Germany, (2008)
5. US Patent Application No 12/349,797, Inventor: Rabih Bashroush, Title: “Multiple Perspective Feature-based Variability Management”, (Patent Pending)
6. B. Nuseibeh, J. Kramer, and A. Finkelstein, "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification," IEEE Transactions on Software Engineering, vol. 20(10), pp. 760-773, (1994)
7. M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch, "COVAMOF: A Framework for Modeling Variability in Software Product Families." In proceedings of Third Software Product Line Conference 2004, Boston, (2004)
8. M. Antkiewicz and K. Czarnecki, "FeaturePlugin: feature modeling plug-in for Eclipse." In proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange, (2004)
9. T. Asikainen, T. Männistö, and T. Soininen, "Kumbang: A domain ontology for modelling variability in software product families," Advanced Engineering Informatics, Elsevier Science Publishers B. V., vol. 21, pp. 23-40, (2007)
10. D. Beuche, "Variant Management with pure::variants," pure-systems GmbH (2003)
11. Pure-Systems Pure::Variants, http://www.pure-systems.com/Variant_Management.49.0.html
12. "BigLever Software Gears," http://www.biglever.com/solution/product.html