Feature-oriented Modeling for Collaborative Virtual Environment Construction

Xin Zhao
Department of Computer Science
School of Engineering
University of Alabama
Tuscaloosa, Alabama 35487
Email: xzhao24@crimson.ua.edu

***Abstract*—Collaborative Virtual Environments (CVEs) introduce significant improvements to communication and interaction by offering a simulated visualization for data representation and synchronized exchange. Participants can share their context and information in a consistent, integrated environment even if they are geographically distributed. These characteristics greatly enhance the virtualization of collaborative applications such as combat training, distance education, and a broad range of domain-specific needs. However, there are many accidental complexities involved in the creation of such environments that make it very challenging for end-users who do not have software development expertise. Our proposed work seeks to simplify and customize CVEs through feature-oriented modeling. The objective of this Ph.D. investigation is to provide a step-wise collaborative environment construction approach for end-users, thus allowing non-experts to create CVEs simply by choosing system features. Specific research challenges in using feature models for this specific application approach will be introduced along with our vision for investigating these issues as a doctoral dissertation.**

I. Introduction and Problem

Virtual Reality (VR) is a computer-based technique that simulates a physical presence wherein users are able to interact with a virtual world and gain sensory experiences (e.g., visual, haptic, aural). Although an exact definition of VR is hard to formulate (the earliest existence of this concept comes from a science fiction novel written by Weinbaum [8]), the great benefit of this technique is easy to see —"going nowhere but experiencing anywhere" with equipment (such as headsets and remote controllers). There are many areas that VR could apply, such as education (distance learning), military (combat training), health (surgery simulation), and civil engineering (urban planning). A Collaborative Virtual Environment (CVE) is an extension of a networked virtual environment that provides a context for cooperation combining both the participants and their information into a common display space with integration and persistence.

The construction of CVEs is challenging, both from a hardware aspect ([17], [16], [3]) and software aspect. Software challenges, which are the focus of my research, mainly involve how to facilitate the CVE construction process. Although there are several development tools (such as Unity 3D[1], Blender[2], JMonkeyEngine[3]) supporting CVE construction, these tools are highly expertized and developers need programming skills to implement systems using these tools. Such activities always introduce many accidental complexities [5] to the environment construction.

Model-Driven Engineering focuses more on problem solving from domain aspects rather than a focus on the underlying technology space. The promise of MDE — improving system portability, reuseability and adaptability [19] [15] [24] — makes it an acceptable development methodology both in academic and industrial fields. Feature-oriented software development (FOSD), a MDE-based software development paradigm, involves product synthesis and customization based on system features. Feature models, the core assets in FOSD, encapsulate the commonalities and variabilities of software products. The set of these products builds up a Software Product Line (SPL) [21]. Feature modeling allows developers to concentrate on a particular domain at an abstract level without knowing implementation details. Though FOSD has been applied to several application domains successfully [18], employing feature-oriented modeling to virtual environment construction is still new.

In summary, my Ph.D. research aims to answer the following questions:

RQ1 — Methodology: Can we design a novel step-wise feature-oriented approach based on feature priority for software development?

RQ2 — Design: Can we design collaborative virtual environments through step-wise feature modeling?

RQ3 — Evaluation: To what extent do end-users benefit from feature-oriented CVEs?

II. Related Work

Applying FOSD to CVE construction is still a new research field. Few works attempt to provide CVE construction through feature-oriented approaches. In this section, we review some existing literature related to two research fields: CVE construction and FOSD.

CVEs aim at the synchronization of communication among different participants in one virtual environment. Greenhalgh

[1]https://unity3d.com/
[2]https://www.blender.org/
[3]https://jmonkeyengine.org/

Fig. 1. Framework for our proposed solution to configure CVEs

et al. proposed a collaborative virtual teleconferencing system [9]. Though this is one of the earliest papers discussing CVEs, the focus of this work is how to process interactions between a human and a computer. For research aiming at the construction of virtual environments, Hernández et al. [10] proposed a 3D real-time CVE for GUI sketching. In this work, the authors focus on collaborative GUI sketching and implemented a prototype called WeSketch. A similar tool is Teleplace[4], a 3D application for virtual on-line meetings. In the industrial context, there are some tools (e.g., Unity 3D, Blender) that allow users to develop a customized virtual environment. However, these tools are highly-expertized and developers need some programming skills to accomplish specific tasks. None of these industrial tools provide support for real-time collaboration.

FOSD deals with external system features in a specific domain. In FOSD, features are treated separately and designers do not need to consider implementation details. A feature model is an abstraction of system features and consists of all the functional and non-functional elements in the system. Feature diagrams [11] and formal semantics [20] are widely used to represent feature models. Though FOSD decomposes the complexity of system design, it increases the complexity when integrating and organizing all the system features [4]. Lee et al. proposed a guideline for solving this problem from domain planning and feature identification [13]. Although feature prioritization and feature selection are not novel ideas [2] [23], there still lacks related work toward applying a selection process to CVE construction. FOSD has been applied to many application domains, such as reverse engineering [7], computer networks [22] and image processing [6].

III. Proposed Solution

We propose a step-wise feature-oriented modeling approach for CVE construction. Figure 1 shows the framework for our proposed solution. In the following subsections, we discuss the details in each step and possible tools used to support each step.

[4]https://telexlr8.wordpress.com/openqwaq/teleplace/

A. Feature Model Construction

In feature model construction, we will first build a prioritized feature model. A prioritized feature diagram is a feature diagram with all the features defined with priorities. For same level nodes, the priorities are labeled as 1,2,3...N. For parent-children nodes, the priorities are labeled as 1-1, 1-2, 1-3...1-N. When features have the same priorities, their labels are the same. Then, we transform this prioritized feature diagram to a **feature table**. This table is organized according to priorities and presented to end-users. In order to simplify the construction process and avoid construction problems (e.g., feature interaction [14] may violate cross-tree dependencies), we propose a step-wise construction for CVEs from a feature table. Users choose features from the feature table step-by-step. For example, if feature A excludes feature B and feature A has higher priority, then if feature A is chosen, feature B is automatically disabled in later steps. For features that have child nodes, when such features are chosen, the feature table will automatically extend to allow users to choose their child feature nodes. There are several tools that support feature modeling construction, such as Feature Modeling Plug-in (fmp) [1] and FeatureIDE [12]. We plan to implement our approach based on FeatureIDE, which supports all phases of FOSD and is fully integrated in the Eclipse IDE.

B. Product Configuration

In the system implementation phase, the input of this step is the summation of all the features selected by the user from the feature table and the output is a CVE. We plan to design two parts of a user interface — one part is the feature table allowing end-users to choose system features they want and the other part is the real-time generation of the CVE. This phase will be achieved using MDE, which means that CVE construction is generated automatically. Eclipse and FeatureIDE would be our preferred tools for this step. FeatureIDE enables code generation from feature diagram, helping users to jump from high-level abstraction to system implementation. However, FeatureIDE only supports the automated generation of static part in the system. A model for behavioral aspects for the system need to be added in this step. For the convenience of our development, Java will be the hosting language because Eclipse provides an integrated environment for Java development.

C. Collaborative Communication

In the system collaboration phase, we will connect different users who are geographically distributed after the construction of the virtual environment. In our proposed framework, collaborative here means real-time collaboration — data is shared with timely synchronization. One user could see another participant's actions with low latency. There are some tools supporting real-time CVE development, such as Photon unity networking[5] and JMonkeyEngine[6]. In this step, we plan to apply JMonkeyEngine as an implementation tool because it supports Java development, which is consistent with implementation languages in previous steps.

IV. Evaluation and Expected Contribution

In order to evaluate our proposed framework, we plan to implement an application based on a feature-oriented CVE — a collaborative virtual classroom. This prototype will implement some basic teaching tasks in a real classroom, such as real-time communication among participants. We plan to test this simple collaborative virtual classroom to real scenarios and test its performance. The analysis could include both qualitative evaluation (such as delay time and storage requirement) and quantitative evaluation (such as semi-structured interview with users and post-surveys).

The contribution of this Ph.D. project is two-fold. We expect that this research will be both beneficial to academia and industry:

- In academia, we expect the proposal of step-wise feature modeling could solve part of existing challenges in FOSD and provide related researchers a new aspect to the system construction for product lines from feature modeling.
- In industry, the contribution to the community may be the new approach for building easy and customized CVEs (software aspects) by non-experts across many domains.

V. Current Status

This dissertation work is still in a very early age. At present, we achieved several tasks. First, we performed some existing literature reviews to understand challenges in current FOSD approaches. Furthermore, we tested several mainstream virtual environment building tools. From this testing, we found some issues that prevent an easy and fast way to CVE construction for non-experts. To make the customization of CVEs more accessible to non-experts, we propose applying MDE to overcome these challenges. The next steps of our work include step-wise feature model construction design and implementation, applying step-wise feature modeling approach to build CVEs, and system testing. shows a proposed timeline for completion. The highlights of proposed timeline are shown as follows:

- 06/2017 — 11/2017: feature table construction
- 12/2017 — 05/2018: feature-based virtual environment construction
- 12/2017 — 06/2018: feature-based virtual environment testing
- 07/2018 — 10/2018: feature-based CVE construction
- 07/2018 — 11/2018: feature-based CVE testing
- 12/2018 — 04/2019: collaborative virtual classroom implementation
- 12/2018 — 05/2019: collaborative virtual classroom testing
- 06/2019 — 08/2019: empirical evaluation of feature-based CVE construction
- 09/2019 — 12/2019: dissertation and defense

[5]https://www.photonengine.com/en/Photon

[6]http://jmonkeyengine.org

References

[1] M. Antkiewicz and K. Czarnecki. Featureplugin: feature modeling plug-in for eclipse. In *Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange*, pages 67–72. ACM, 2004.

[2] E. Bagheri, M. Asadi, D. Gasevic, and S. Soltani. Stratified analytic hierarchy process: Prioritization and selection of software features. *Software Product Lines: Going Beyond*, pages 300–315, 2010.

[3] D. R. Begault and L. J. Trejo. 3-d sound for virtual reality and multimedia. 2000.

[4] C. Bocovich and J. M. Atlee. Feature-oriented modelling in bip: A case study. In *ModComp@ MoDELS*, pages 6–11, 2016.

[5] F. Brooks. No silver bullet : essence and accidents of software engineering, 1987.

[6] X. Deng, Y. Lan, T. Hong, and J. Chen. Citrus greening detection using visible spectrum imaging and c-svc. *Computers and Electronics in Agriculture*, 130:177–183, 2016.

[7] S. Ferber, J. Haag, and J. Savolainen. Feature interaction and dependencies: Modeling features for reengineering a legacy product line. *Software product lines*, pages 37–60, 2002.

[8] S. Grauman Weinbaum. Pygmalion's spectacles, 1935.

[9] C. Greenhalgh and S. Benford. Massive: a collaborative virtual environment for teleconferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):239–261, 1995.

[10] H. A. Hernández and H. Trefftz. Wesketch: A 3d real time collaborative virtual environment that improves the gui sketching task. In *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*, pages 163–168. IEEE, 2011.

[11] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[12] C. Kastner, T. Thum, G. Saake, J. Feigenspan, T. Leich, F. Wielgorz, and S. Apel. Featureide: A tool framework for feature-oriented software development. In *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*, pages 611–614. IEEE, 2009.

[13] K. Lee, K. C. Kang, and J. Lee. Concepts and guidelines of feature modeling for product line software engineering. In *International Conference on Software Reuse*, pages 62–77. Springer, 2002.

[14] J. Liu, D. S. Batory, and S. Nedunuri. Modeling interactions in feature oriented software designs. In *FIW*, pages 178–197, 2005.

[15] S. J. Mellor, T. Clark, and T. Futagami. Model-driven development: guest editors' introduction. *IEEE software*, 20(5):14–18, 2003.

[16] T. Ni, G. S. Schmidt, O. G. Staadt, M. A. Livingston, R. Ball, and R. May. A survey of large high-resolution display technologies, techniques, and applications. In *Virtual Reality Conference, 2006*, pages 223–236. IEEE, 2006.

[17] E. Peters, B. Heijligers, J. de Kievith, X. Razafindrakoto, R. van Oosterhout, C. Santos, I. Mayer, and M. Louwerse. Design for collaboration in mixed reality: Technical challenges and solutions. In *Games and Virtual Worlds for Serious Applications (VS-Games), 2016 8th International Conference on*, pages 1–7. IEEE, 2016.

[18] D. Rabiser, H. Prähofer, P. Grünbacher, M. Petruzelka, K. Eder, F. Angerer, M. Kromoser, and A. Grimmer. Multi-purpose, multi-level feature modeling of large-scale industrial software systems. *Software & Systems Modeling*, pages 1–26, 2016.

[19] B. Selic. The pragmatics of model-driven development. *IEEE software*, 20(5):19–25, 2003.

[20] J. Sun, H. Zhang, Y. Fang, and L. H. Wang. Formal semantics and verification for feature modeling. In *Engineering of Complex Computer Systems, 2005. ICECCS 2005. Proceedings. 10th IEEE International Conference on*, pages 303–312. IEEE, 2005.

[21] D. M. Weiss and C. T. R. Lai. *Software product-line engineering: a family-based software development process*, volume 12. Addison-Wesley Reading, 1999.

[22] L. Wu, T. G. Barker, and V. S. Desai. Computer-implemented modeling systems and methods for analyzing and predicting computer network intrusions, Sept. 6 2011. US Patent 8,015,133.

[23] E. Yamany, A. Eid, M. Shaheen, and A. S. Sayyad. Opti-select: An interactive tool for user-in-the-loop feature selection in software product lines. In *Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools-Volume 2*, pages 126–129. ACM, 2014.

[24] L. Zhu and Y. Liu. Model driven development with non-functional aspects. In *Proceedings of the 2009 ICSE Workshop on Aspect-Oriented Requirements Engineering and Architecture Design*, pages 49–54. IEEE Computer Society, 2009.