A Tool for the Visualization of Variant Structures in Virtual Reality

Lucas Kuentzer, Georg Rock

Trier University of Applied Sciences
Schneidershof 1
54293 Trier
Tel.: +49 (0)651 / 81 03 - 596
Fax: +49 (0)651 / 8103 - 454
E-Mail: Vorname.Name@hochschule-trier.de

Abstract: Over time models and tools had to be developed to comprehend the increasing complexity of technological systems and their variant components. In this paper, we present a vision and a design for a VR tool that visualizes these variant structures and provides the tools for collaborative review and development. The tool is especially suited for feature models and product structures. We analyze different visualization methods and their fit for virtual space as well as techniques to reduce visual clutter and improve clarity. Our technical approach and schedule for development and evaluation is outlined and the application design is explained in detail. The proposed visualization is based on a cone tree layout in adjustable orientation and features a multi-instance system for convenient display of cross-tree constraints between remote subtrees. A gesture-based input system complements established interaction concepts and the tool supports CAD-model import for engaging collaborative work in immersive virtual experience.

Keywords: virtual reality, variant management, feature model, graph visualization, interaction concept

1 Introduction

Nowadays, a product's technological advantages and associated unique features are often not enough to win over a new customer or keep the loyalty of an existing one. As a result, product developers and manufacturers are always looking for new and engaging ways to interact with the customer on an emotional level and to fulfill the customers need for individual product customization within the boundaries of the brand [Klu18]. The automotive industry is a pioneer in that regard. Instead of vehicles they sell driving experiences and offer highly customizable cars, which can result in one single part coming in thousands of different variants, as with the BMW X3 doors (3000) or headliner (90.000) [Klu18]. Not all of these variants make sense in a combined configuration and the total number of eventual product variants is lower than feared, but this shows the complexity of a modern product life cycle. In order to conquer this complexity at a reasonable cost, manufacturers have introduced product

Veröffentlicht durch die Gesellschaft für Informatik e.V. 2020 in B. Weyers, C. Lürig, D. Zielasko (Hrsg.):
GI VR / AR Workshop 2020, 24. - 25, September 2020, Trier.

http://dx.doi.org/10.18420/vrar2020_35

lines, reusing components among them like building blocks. These product lines still need to be modeled throughout the whole product development process. Feature models (FM) are an well known among researchers and a promising method for managing product structures and component constraints [KL13]. Unfortunately and to the best of our knowledge, FM trees are not yet well established in the industry and with customers. Instead, users prefer the use of tables and matrices, that have become familiar to them during their education and training in business or engineering.

Visualized as a tree or not, the larger the model the harder it is to comprehend and to visualize, especially on two-dimensional (2D) media like paper or a screen. Virtual Reality (VR) has recently become more relevant than ever as a visualization tool in education, training and the product development process [KHR19]. In late 2019 we conducted a survey among second-year engineering students, who were about to work on their first structural design project. The VR tool we used was poor in features, but natively integrated into the CAD-application familiar to them. The majority of students deemed the tool helpful for understanding the context of their work among other positive effects and potentials, but one of the most requested features was the representation of the product structure in familiar tree hierarchy.

In this paper we present our prototype of a VR tool for the visualization of variant structures like abstract feature models or specific product structures. We want to find out to what extend users benefit in their understanding of complex product structures from a spatial representation in VR over 2D visualizations or projections. Furthermore, our research goals include finding and validating means to deal with model scale, the user's sense of orientation within the model as well as the virtual space, and different preferences of users regarding visualization and interaction metaphors as opposed to simplistic and functional design themes.

2 Related Work

Trinidad et. al [TCBS08] propose to visualize the variability of software product lines as feature cone trees (FCT). Feature cone trees are hierarchical structures in three-dimensional space that has a given node as the tip of a cone and positions its children around the base [RMC91]. Adding the third dimension of depth, Cone trees make better use of limited screen space in 2D screen projections on the pretense of an adjustable view perspective and are a valuable alternative to 2D layouts for representing large hierarchical structures [TCBS08, RMC91]. Feature cone trees lead to an increased understanding of structure by the user and are capable of highlighting session-important or context-relevant information through depth [RMC91, CM00]. Orientation in 3D space as well as perception of depth and distance are among the biggest strength of the virtual reality technologies, resulting in it being a promising medium for these kind of visualizations [KHR19]. Depending on the test setup and application design users might be slower when performing certain data analysis tasks using cone trees as opposed to working with 2D layouts [CM00]. However, the users are more engaged and enthusiastic when working with the 3D cone trees, especially when presented in visually-pleasing aesthetics [CM00, RMC91]. Engaging visuals and pleasing aesthetics are associated with impressiveness of a virtual reality environment and can be linked to the user's task performance and the application's effectiveness [AVM18]. Under the right circumstances and with a good application design, users have been found to perform considerably faster working on cone trees than using a traditional directory structure [CK95].

Unfortunately, visual clarity and informational effectiveness diminishes severely in cone tree visualization exceeding 10 layers or 1000 nodes or more [CK95]. There are many suggested methods for dealing with the challenges of large hierarchical structures, such as feature models [TCBS08, RMC91, CM00, CK95, OR15]. These methods include the coloring, reshaping, resizing, grouping and labeling of nodes. The user should be able to search and filter feature nodes, constraints or whole substructures, hiding them from display completely. Sub-trees should be collapsible or extractable to be disregarded or focused on. Additional information, such as labels of leaf nodes, cross-tree constraints and feature-attached attributes should only be displayed when the user is focusing on the node or area or otherwise makes a selection, also highlighting the path from root to selected node. Additionally, a manually or automatically triggered fish-eye view enables the user to focus on details which are far away from them [Fur86]. These are just some of many techniques to remove visual clutter and to increase clarity and scalability.

In terms of graph interaction, Huang et. al [HFL+17] propose camera-based hand tracking for a gesture input system in virtual space. Their user study revealed that the performance of the gesture-based interaction is a matter of familiarity and the design of the input system. Users reported fatigue and feeling uncomfortable performing some of the gestures, but overall tasks were deemed easier to accomplish in comparison to 2D input devices.

3 Approach

The VR tool we are designing is meant to be used as an extension of the Glencoe web app (Fig. 1), which employs modern web technologies to combine a stylish and user-friendly interface with a suite of concurrently working SAT-solvers in the back-end [SBR18]. While the web app excels in the detailed modeling and analysis of complex feature models and their complicated constraints, the VR tool shall emphasize the process of understanding the visualized (sub-)structures, virtual collaboration between stakeholders and simple, but engaging graph manipulation within an immersive experience. The tool shall largely mirror Glencoe in functionality, wrapping it into an immersive and engaging VR user experience. Until systems are in place for both tools to work concurrently on a live model, Glencoe provides an interface for automatic analysis requests. We use consumer-grade hardware for our prototype and try to optimize the application to lower the bar of entry for midsize companies or institutions when it comes to VR. The controllers, that are included with the VR hardware system we use, combine traditional button, stick and touchpad controls with proximity sensors, estimating the hand pose. This enables us to implement established interaction

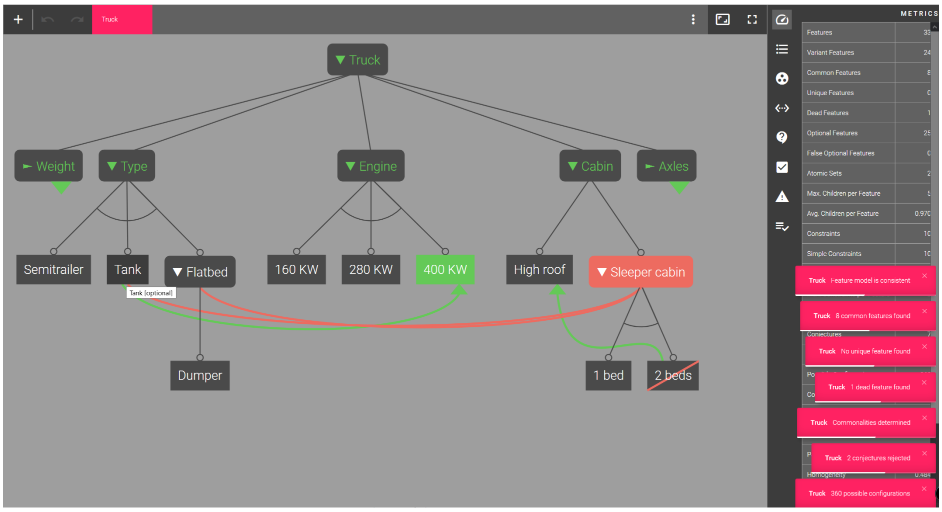

Figure 1: The Glencoe Web App displaying a feature model of a mock-up truck. The user has run automatic model analysis including inconsistencies and conjecture solvers, and is hovering over the Tank node which implies 400 kw and excludes Sleeper cabin.

concepts in VR, like personal navigation, alongside a context-based gesture input system when working with the variant structure representation or CAD models. The development can be separated into four major phases, according to an evolutionary development model. In Phase 1 the development focuses on the import and cone tree visualization of a given feature model as well as back-end feature model manipulation and the interface to the original Glencoe system. The second phase is about the implementation of user interaction and the immersive VR experience, including setting up the context-based gesture input system and modules that estimate user focus and intent. For the third phase we plan to implement the import and (semi-)automatic linking of 3D models from computer-aided design. Phase 4 aims to improve on the visuals and polish the user experience as well as expanding on the tree visualization by setting up tools for model import, session management, save and load persistence and online multi-user sessions. Optimization and testing efforts regarding performance and usability run concurrently to these phases. So far a phase 1 technical prototype is nearing completion of its first major iteration. The concept of the prototype has been validated together with stakeholders from the Glencoe project. For our first user study before the end of summer we will prepare the demo of a vehicle with variant components and its visualized feature model as well as its product structure. We will ask engineering students and partners from the product development industry to try out the tool and provide feedback in a questionnaire centered on tool acceptance, styling and interaction design.

4 Application Design

The feature model is exported from Glencoe and parsed into the VR tool, populating an internal model which calculates additional node information such as their depth in the tree and the node count within subtrees. This information is used to lay out either a horizontallyoriented cone tree (also: CAM-Tree [RMC91]) for deep trees like product structures or a vertically-oriented cone tree around the user for broad, but shallow trees. The user can change the tree orientation according to their preference. The visual appearance of the application can also be changed between different themes, some providing a blank design space and others more elaborate environment metaphors. The nodes are spread around the base of their parents' cone according to the space requirements estimated from the size of their subtree and the cones are shrunk or stretched according the size of their nodes' subtree and the displayed maximum height of a subtree can be set. Our initial vision had the nodes then arrange themselves in a force-based layout in conjunction with a Universe metaphor. Early prototypes made us to abandon this concept as default visualization as it threatens to lower tool acceptance by taking control away from the user, only making orientation and interaction more challenging. The user can choose to scale the FCT to a small tree originating in their palm or as to life-sized structure to walk among the nodes and constraints (Fig. 2). After phase 4 the user will have tools to already filter the displayed feature model on a virtual workbench upon import for the sub-trees relevant to them, before the tree even gets visualized. In general, the focus is shifted from the whole structure to substructures and single nodes, their constraints and place in the surrounding system. The nodes are labeled

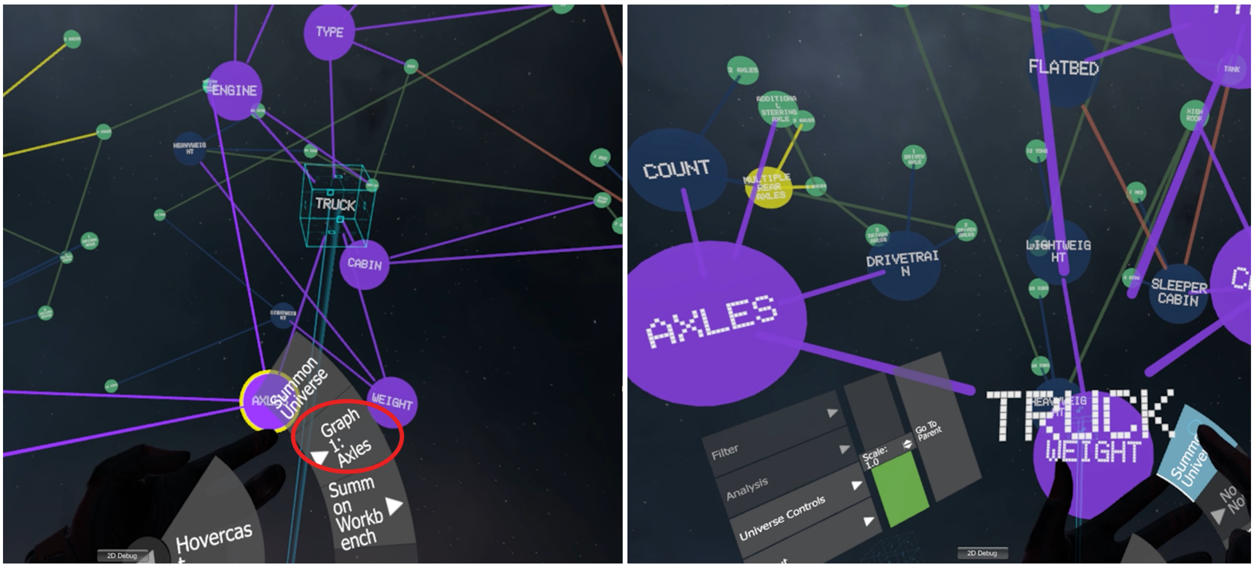

Figure 2: On the left side the user selects a node by poking it in a scaled-down feature model updating the palm-attached context menu. On the right side the user has just summoned the room-scale feature model to walk among or configure through the hovering control UI.

and children as well as nodes related through constraints are highlighted upon interaction. Nodes as well as constraints can be selected to access further information and context-based interaction options in a palm-attached UI. In later iterations of the application this UI menu can be placed on an individual basis for accessibility. The context menu enables the user to explore features and their constraints, to collapse or filter sub-trees and manipulate the tree structure by adding or removing nodes and constraints. A similar building-blocks technique as in Glencoe could be employed to let the user craft even complicated constraints or conjectures for requests to the Glencoe server. The user will be able to summon several sets of virtual workbenches dedicated to tasks like model import, session management or multi-instance management from their carry-on menu. An additional tree-instance-specific UI can be used to manage tree layout, scale, sub-tree navigation and physics interactions. For many of these initially UI-bound interactions the user will be able to use gestures or interaction metaphors instead, for example for adjusting the scale and rotation of subtree instances, the selection of nodes into a satisfiability configuration, the removal of nodes and constraints, and many more tree visualization, navigation and manipulation functions. The idea is to leave the degree of immersion up to the user in how they want to work and letting them really take advantage of their virtual presence.

Arguably the biggest challenge to the effectiveness of our VR FCT visualization and interaction remains the scale of the given feature model. A large number of nodes and constraints is sure to overwhelm the user and because of that we implement many of the techniques highlighted in other works as ways to reduce visual clutter and allow the user to direct their focus. Many of these techniques can be easily in 3D similarly to how they work in 2D (e.g.: coloring, sizing, grouping, collapsing, highlighting, etc.). Other techniques require a different approach due to UI-challenges in VR (e.g.: filtering, context menu, etc.). In our endeavor to reduce visual clutter we run the risk of cutting out information that is relevant to the user. We aim to limit the number of nodes being displayed to the user, but need a way for them to explore the relations between the nodes of remote sub-trees that might not be visualized in the same instance by default. Similarly to opening a second window on a different subtree in 2D, the user can summon a helper instance of the cone graph. While constraints from nodes in one instance to the other crossing window borders would be rather unusual and maybe even confusing, seeing these cross-tree constraints between remote sub-tress crossing the space between primary and helper instance already feels natural during development (Fig. 3).

When linked to an interactive visualization of a CAD-model, interactions with the FCT through navigation, manipulation or configuration can be mirrored by the 3D model through highlighting, part selection and variant exchange. This interaction between the construction design model and the feature model can be used for mutual validation, identification of dead features or impossible configurations as well as potential collisions and conflicts in the build space of a construction part.Multi-user capability enables local and remote collaboration, customer interaction as well as training and education.

5 Conclusion

In itself cone trees are not a very established method of 3D graph representation in desktop application, as navigation and orientation becomes bothersome. This changes in virtual space through the users sense of depth and spatial awareness, provided the scene lends itself to prominent orientation points. We suspect that the sense of immersion of both being able

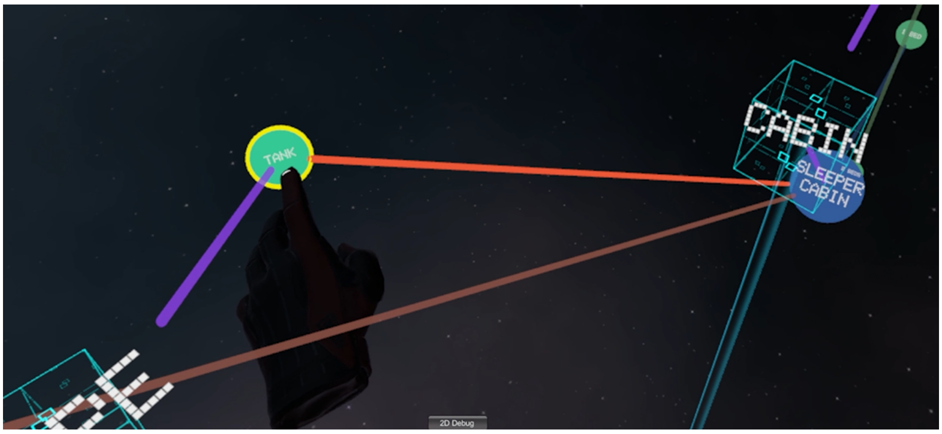

Figure 3: The user is hovering his index finger over the Tank feature in the primary feature model representation rooted on the Type subtree, cross-instance-highlighting the constraint to the Sleeper cabin feature which is displayed in a helper tree rooted on Cabin.

to observe the tree from afar as well as venture within it will lead to a deeper understanding and great acceptance and hope to prove that in our evaluation.

Our approach is meant for large-scale structural analysis, but aims to provide a engaging and creative communication tool for inter-departmental design, development and product line reviews focused on substructures of the given product or process. We see it as a valuable tool for improving the collaborative decision making process in product development or education. Outside of a development process our tool could be used as a multi-user configuration tool for a salesperson's interaction with private or personal customers of highly-customizable products like cars, prefabricated houses, construction or agriculture vehicles.

The tool we present is still in the early stages of development and only time will tell if we can overcome the challenges we have mentioned. However, we believe that our visualization approach and application design provide an engaging vision for future endeavors in VR-assisted variant management.

References

[AVM18] Devon Allcoat and Adrian Von Muhlenen. Learning in virtual reality: Effects on performance, emotion and engagement. *Research in Learning Technology*, 26, 11 2018.

[CK95] Jeromy Carriere and Rick Kazman. Research report. interacting with huge hierarchies: beyond cone trees. In *Proceedings of Visualization 1995 Conference*, pages 74–81, 1995.

[CM00] Andy Cockburn and Bruce Mckenzie. An evaluation of cone trees. In *People and Computers XIV: British Computer Society Conference on Human Computer Interaction*, pages 425–436. Springer-Verlag, 2000.

[Fur86] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '86, page 16–23, New York, NY, USA, 1986. Association for Computing Machinery.

[HFL+17] Yi-Jheng Huang, T. Fujiwara, Yun-Xuan Lin, Wen-Chieh Lin, and Kwan-Liu Ma. A gesture system for graph visualization in virtual reality environments. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 41–45, Los Alamitos, CA, USA, apr 2017. IEEE Computer Society.

[KHR19] Lucas Küntzer, Michael Hoffmann, and Georg Rock. Virtual reality as an e-learning tool in the student-centered vehicle development project protron. In Niels Pinkwart and Johannes Konert, editors, *DELFI 2019*, pages 157–162, Bonn, 2019. Gesellschaft für Informatik e.V.

[KL13] Kyo Chul Kang and Hyesun Lee. *Variability Modeling*, pages 25–42. Springer, 2013.

[Klu18] Florian Klug. *Logistikmanagement in der Automobilindustrie: Grundlagen der Logistik im Automobilbau.* Springer Berlin Heidelberg, 2018.

[OR15] Jürgen Ommen and Georg Rock. Visualization of variability in complex development structures. In *Proceedings of the Ninth International Workshop on Variability Modelling of Software-Intensive Systems*, VaMoS '15, page 109–116, New York, NY, USA, 2015. Association for Computing Machinery.

[RMC91] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, page 189–194, New York, NY, USA, 1991. Association for Computing Machinery.

[SBR18] Anna Schmitt, Christian Bettinger, and Georg Rock. Glencoe – a tool for specification, visualization and formal analysis of product lines. In M. Peruzzini et al. (Eds.), editor, *Transdisciplinary Engineering Methods for Social Innovation of Industry 4.0: Proceedings of the 25th ISPE Inc. International Conference on Transdisciplinary Engineering*, pages 665–673, Amsterdam, Netherlands, September 2018. IOS Press.

[TCBS08] Pablo Trinidad, Antonio Ruiz Cortés, David Benavides, and Sergio Segura. Three-dimensional feature diagrams visualization. In Steffen Thiel and Klaus Pohl, editors, *Software Product Lines, 12th International Conference, SPLC 2008, Limerick, Ireland, September 8-12, 2008, Proceedings. Second Volume (Workshops)*, pages 295–302. Lero Int. Science Centre, University of Limerick, Ireland, 2008.