variED: A Collaborative, Real-Time Feature Model Editor

Elias Kuiter
Otto-von-Guericke-University
Magdeburg, Germany
kuiter@ovgu.de

Sebastian Krieter
Harz University of Applied Sciences
Otto-von-Guericke-University
Wernigerode & Magdeburg, Germany
skrieter@hs-harz.de

Jacob Krüger
Otto-von-Guericke-University
Magdeburg, Germany
jkrueger@ovgu.de

Thomas Leich
Harz University of Applied Sciences
Wernigerode, Germany
tleich@hs-harz.de

Gunter Saake
Otto-von-Guericke-University
Magdeburg, Germany
saake@ovgu.de

In the paper attached below, we discuss the formal foundations of collaborative, real-time feature modeling. To this end, we describe an extension of the *Multi-Version Multi-Display* (MVMD) technique, which we employ to detect problematic situations (i.e., conflicts) that may arise due to concurrent actions of multiple collaborators. Furthermore, we describe a simple resolution process for such conflicts. The described technique allows for concurrent editing of feature models without compromising editing efficiency due to the optimistic nature of the MVMD technique.

To demonstrate the feasibility of our approach, we have implemented the core concepts presented in the paper in the web-based prototype *variED*. This prototype is freely available in our GitHub repository.[1] In our GitHub repository, we describe how to deploy variED to several cloud infrastructure providers, such as Amazon AWS, as well as a manual build procedure. For convenience, we have also set up an instance of variED that is available for public use.[2] This instance is restarted regularly, so changes will not be persisted longer than 60 minutes. We proceed to describe the course of a typical editing session in variED.

Initiating a New Session. Upon visiting variED in a web browser, a new collaborative feature modeling session can be initiated in three ways:

- Choose *Tools > Command Palette... > Join collaborative session* and one of the provided example feature models (e.g., *FeatureIDE > FeatureIDE*). Items in the command palette can be filtered by typing on the keyboard.
- Choose *File > New...*, fill in (arbitrary) values for *Project* and *Artifact* and click *Create*. A new, empty feature model is created that only consists of a *Root* feature.
- Choose *File > New...*, fill in (arbitrary) values for *Project* and *Artifact* and upload a FeatureIDE-compliant feature model.[3] Thus, existing feature models can also be used with variED.

Inviting Collaborators. To invite others to the current feature modeling session, choose *File > Share... > Copy* and share the URL with your collaborators. To simulate multiple collaborators, variED can be also opened in a second web browser or a private browsing window. (Due to technical constraints, it is not possible to open variED twice in the same browser instance.)

[1] https://github.com/ekuiter/variED
[2] http://varied.herokuapp.com/
[3] e.g., https://raw.githubusercontent.com/FeatureIDE/FeatureIDE/master/plugins/de.ovgu.featureide.examples/featureide_examples/FeatureModels/APL/model.xml

Feature Modeling. All core functionality available for feature modeling in FeatureIDE and pure::variants has been implemented. Most feature modeling operations are available by left, right or double-clicking on a feature and are straightforward to use. However, some operations, such as constraint-related ones, are currently only available from the command palette (*Tools > Command Palette...*). Note that when creating or updating a constraint from the command palette, variED currently expects the constraint in a Lisp-like prefix notation. That is, the cross-tree constraint *FeatureModeling* $\Rightarrow$ *FeatureIDE* is represented as `(-> FeatureModeling FeatureIDE)`. Similarly, $A \vee B \Leftrightarrow C \wedge D$ is denoted as `(<=> (| A B) (& C D))` or, equivalently, `(equals (or A B) (and C D))`.

Provoking a Conflict. Our implementation of the MVMD technique allows every editing operation to be applied immediately, without waiting for the server or other collaborators. After applying an operation, it is synchronized with other collaborators and possibly, a conflict is detected and displayed. This procedure can easily be triggered, for example, when two collaborators simultaneously rename the same feature `A` to different names `B` and `C`. In case of low network latency, probability for conflict is low, so provoking one intentionally may be difficult. In that case, choose *Tools > Command Palette... > Developer: Simulate message delay* and enter a number of milliseconds by which all communication with other collaborators should be delayed. When a conflict has been detected, the local editor is frozen until the conflict is resolved. We are currently working on integrating the conflict resolution process described in the paper—as of now, variED only summarizes the conflict situation and does not allow resolution yet. However, it is still possible to switch to or create another feature model as described above.

Export. After an editing session, variED allows to export a feature model in a number of graphics formats. These export options are available from the *File > Export as* menu. View-specific options, such as feature model layout and collapsed features, are preserved. variED also allows to customize some aspects of the feature model diagram, such as font, colors and margins (*Tools > Settings...*).

More Information. We provide more detailed build instructions and API documentation for variED in our GitHub repository. To get a deeper insight into variED, open the web browser's JavaScript console, where every interaction with the server is logged. In debug mode (*Tools > Command Palette... > Developer: Toggle debug mode*), detailed information about conflict detection is logged as well.