HyVar NEWSLETTER N°2, Nov 2016

In this issue:

- ✓ Hybrid Feature Models
- ✓ Hybrid Reconfiguration Engine HyVarRec

In Next Issue:

- ✓ HyVar DSVL Toolchain
- ✓ HyVar Cloud Infrastructure

Hybrid Feature Models

Scalable Hybrid Variability for Distributed Evolving Software Systems

The goal of HyVar is to be able to model Software Product Lines which adapt to their current surroundings.

In software product line (SPL) engineering, feature models (FMs) are a common used technique to model conceptual commonalities and variabilities. To this end, features are structured in a hierarchical way and can be grouped to **or** and **alternative** groups. Moreover, cross-tree constraints (CTCs) can be used to define dependencies between features which cannot be captured by the hierarchy of the FM.

HyVar goes beyond the state-of-the-art by proposing hybrid variability

However, in standard SPL engineering, it is not possible to model the external influences on the SPL. To be able to capture the influence of the surroundings, we introduce **hybrid feature models** (HyFMs). With HyFMs, it is possible to model contextual information representing the surrounding and its influence on the SPL. We provide three types of contextual information: Boolean, String, Integer and pre-defined Enumerations. Additionally, each contextual information may have a domain, indicating valid value ranges for that context.

To model the impact of the context on the SPL, we introduce the concept of **validity formulas** (VFs). VFs can be annotated to each feature and indicate in which context a feature is selectable. Figure 1 depicts an example of a HyFM for a car. Four contextual information are modeled (bottom right side), capturing the ***pollution*** and ***weather*** around the car, the number of accompanying ***passengers*** and the current ***time***.

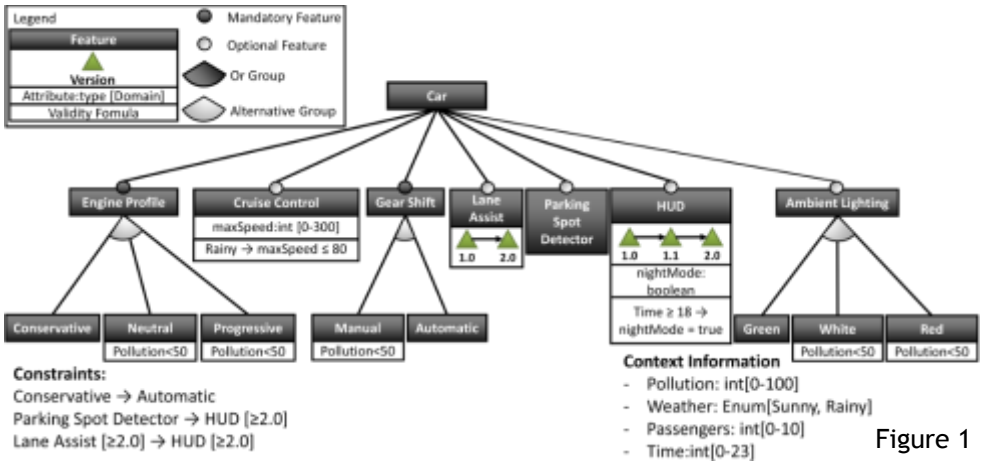

Figure 1

Moreover, multiple features are annotated with VFs.
For instance, the ***Manual*** gear shift can only be selected if the current ***pollution*** is less than ***50***.
Additionally, we extended standard feature models by advanced concepts, such as feature versions, representing evolved implementations of features, and feature attributes, allowing more fine-grained variability.
In the example, the Lane Assist is available in two different versions, 1.0 and 2.0. We implemented HyFMs in the DarwinSPL which is open-source and available on GitHub (www.darwinspl.de)
Together with HyVarRec (next topic in this newsletter), it is possible to realize SPLs automatically adapting to their surroundings, based on the modeled contextual information and validity formulas.

HyVar project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644298

Hybrid Reconfiguration Engine HyVarRec

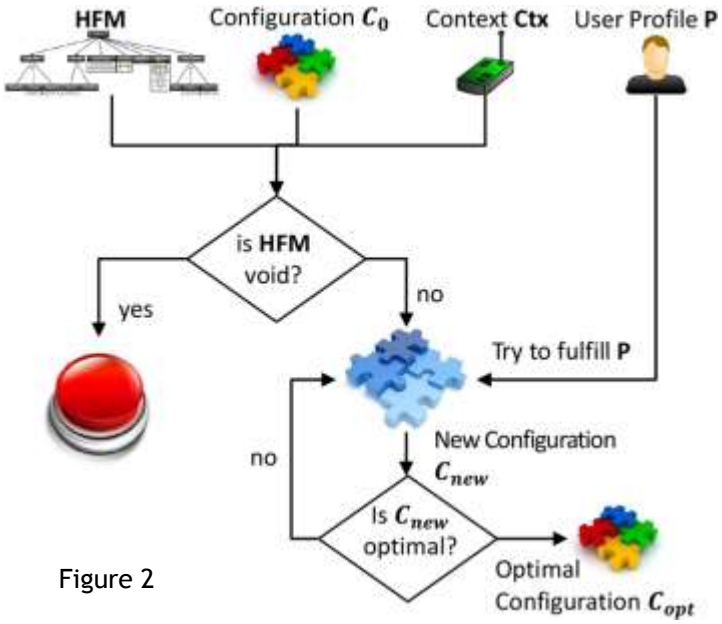

Figure 2

HyVarRec

HyVarRec is a tool that allows to reconfigure an existing configuration for a given Software Product Line when it is subject to contextual changes. It is one of the cores tools developed within the HyVar project and responsible to compute what are the features of the update that will be sent to the car.
HyVarRec requires different sources of input: the Hybrid Feature Model describing the software product, the current configuration C of the software product, the values of the contextual information Ctx, and the user profile P (see Figure 2).
The primary function of the contextual reconfigurator is to provide a valid configurations C' for the context Ctx that maximize the preferences of the user.

In case of two configurations of equal quality for the user, the one that minimizes the difference between the initial configuration C is provided.
This means that HyVarRec first tries to minimize the number of feature removals needed to transform C into C' and, later, to maximize the number of attributes which values could be kept the same.
As a simple example consider the configuration depicted in Figure 3 describing the configuration of a car located in Europe that uses

i) the European Emergency call (i.e., feature ECall),
ii) the package for European languages (i.e., EU_Languages),
iii) the GPS.

The ECall feature can be selected only if the location of the car is in Europe.

Indeed, as can be seen from the picture, the feature ECall has associated a validity formula that exclude its selection if the contextual property "Position" is equal to "Europe".
When the car leaves Europe, entering for instance Russia, its configuration is not valid anymore and needs to be changed.
Thanks to the use of HyVarRec we can compute the new valid configuration depicted in Figure 4. The feature ECall has been deselected and replaced by the feature EraGlonass, i.e., the emergency call version for Russia. Moreover, since EraGlonass requires the language Russian and the use of the Glonass (see the cross tree constraints associated to the feature model of the car) these two features are selected.
At the same time the support for the European languages is deselected since the feature EU_Languages is mutually exclusive with respect to the Russian language support. All these changes are computed automatically by HyVarRec that present in output directly the new valid configuration.

HyVarRec could be easily installed using Docker container technology available for the majority of the operating systems. It can therefore be used simply sending an HTTP Post request to the server deployed by using Docker.

HyVarRec is written in python, open source, and freely available from https://github.com/HyVar/hyvar-rec

Constraints:
EraGlonass ↔ Russian
EraGlonass → Diagnostic
eCall → EU_Languages
eCall → GPS
EraGlonass → Glonass

Context Information
- Position: Enum[Russia, Europe]

Figure 3

Constraints:
EraGlonass ↔ Russian
EraGlonass → Diagnostic
eCall → EU_Languages
eCall → GPS
EraGlonass → Glonass

Context Information
- Position: Enum[Russia, Europe]

Figure 4

HyVar Events

- ✓ Integration meeting Braunschweig, July 4-5 2016
- ✓ Review meeting Brussels, September 2016
- ✓ ISOLA 2016 Corfu, October 2016 http://www.isola-conference.org/isola2016/ Track on Variability modelling for scalable software evolution
- ✓ General Meeting Torino, October 19-20 2016
- ✓ FOSD - 7th **International Workshop on Feature-Oriented Software Development** Amsterdam, November 2016 http://www.fosd.net/workshop2016

Next Events

- ✓ Integration meeting Torino, February 2016
- ✓ General Meeting in Braunschweig, June 2016
- ✓ iFM - 13th **International Conference on integrated Formal Methods** Torino, September 2017 http://ifm2017.di.unito.it/
- ✓ ESOCC - 6th **European Conference on Service-Oriented and Cloud Computing** Oslo, September 2017 http://esocc2017.ifi.uio.no/