Workflow-driven Product Derivation

Arnaud Hubaux, Ebrahim Abbasi, Andreas Classen, Patrick Heymans

PReCISE Research Centre, Faculty of Computer Science, University of Namur
Namur, Belgium
{ahu, eab, acs, phe}@info.fundp.ac.be

Variability models, feature diagrams (FDs) ahead, are commonly used to document product line (PL) requirements. They are also often used during product derivation where they are fed to configuration tools which support semi-automated feature selection. Configuration tools facilitate this task by automatically propagating the decisions made and by ensuring their overall consistency. However, most feature-based configuration tools assume that there exists a single monolithic FD and do not account for configuration processes that are distributed among various stakeholders who have specific concerns and who intervene at different moments. Our collaborations with industry have confirmed the need for techniques and tools that support such complex configuration processes.

In order to provide a modelling and reasoning framework for this process, we built upon our earlier work on formal semantics for FDs [1] and proposed *feature configuration workflows* (FCWs) [2], a formalism that combines the workflow language YAWL [3] with FDs.

An FCW, such as the one shown in Figure 1, is a workflow where tasks (such as Web Administrator or PloneMeeting Manager) are associated with FDs. In our work, we used YAWL as workflow modelling language. The configuration process follows the workflow. A FD is only configured when the task to which it is linked (through a start link) is executed. Generally, a task is assigned to a stakeholder, and the FD that is configured during the task captures her concerns and responsibilities. The second kind of node in a workflow is the condition (e.g. ▶ , pm or ■) which designates a point in time. Linking a FD to a condition (through a stop link) means that the FD has to be fully configured when the condition is reached. In the example, it means that once pm is reached, there cannot be any decision left open in the FD of the web administrator and PloneMeeting manager.

The workflow is not the only link between FDs. Constraints can also be added across FDs, such as the $\ll excludes \gg$ link between features *Standard workflow* and *Archived* in Figure 1. Since the task and the stop of a FD are not necessarily directly linked to each other, the configuration decision can be postponed until the condition is reached. For instance, the stop of the Web Administrator , *viz.* pm , is placed only after PloneMeeting Manager . The decision of including or excluding the feature *Standard workflow*, for instance, does not have to be taken during task Web Administrator , because the person executing PloneMeeting Manager is also able to make this decision, the feature *Standard workflow* being the same.

The original implementation strategies for FCWs we studied are available in [2]. These strategies have been later completed to address FCW normalisation and decision

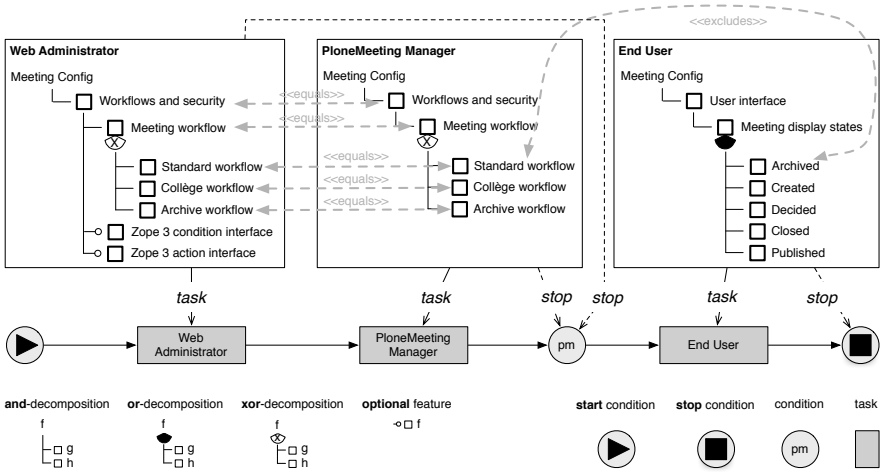

Fig. 1. Example FCW taken from the PloneMeeting configuration menu.

postponement problems [4]. Yet, an efficient implementation of these solutions is still to be provided.

In order to automate FCW editing and configuration, we are developing a tool based on YAWL and SPLOT [5]. YAWL provides support for workflow editing, task management and proposes a repertoire of advanced checks on workflows like soundness or weak soundness [6]. SPLOT provides support for FD configuration and constraint propagation [5]. Since YAWL and SPLOT are web-based applications, we are building a tool that relies on web-services to maintain the connection between the task management and FD configuration environments, and to enforce FCW semantics.

The next step on the agenda is to implement algorithms that evaluate whether an FCW is in *normal form* [4] and whether deadlocks can occur. In addition, the merging of inconsistent configurations obtained in concurrent environments (e.g. distributed offline configuration) will have to be dealt with.

Acknowledgements

This work is sponsored by the Interuniversity Attraction Poles Programme of the Belgian State, Belgian Science Policy, under the MoVES project and the FNRS.

References

1. Schobbens, P.Y., Heymans, P., Trigaux, J.C., Bontemps, Y.: Feature Diagrams: A Survey and A Formal Semantics. In: RE'06. (September 2006) 139–148
2. Hubaux, A., Classen, A., Heymans, P.: Formal modelling of feature configuration workflow. In: SPLC'09, San Francisco, CA, USA (2009)
3. van der Aalst, W., ter Hofstede, A.: Yawl: yet another workflow language. Information Systems **30**(4) (2005) 245–275

4. Classen, A., Hubaux, A., Heymans, P.: Analysis of feature configuration workflows (poster). In: Proceedings of the 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, USA (2009)
5. Mendonça, M.: Splot. http://www.splot-research.org/ (May 2010)
6. van der Aalst, W.M.P., van Hee, K., ter Hofstede, A., Sidorova, N., Verbeek, H., Voorhoeve, M., Wynn, M.T.: Soundness of workflow nets:classification, decidability, and analysis. Technical report, Technische Universiteit Eindhoven (2008)