Information and Software Technology 55 (2013) 595–606

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Experience from model and software reuse in aircraft simulator product line engineering

Henric Andersson [a,b,*], Erik Herzog [b], Johan Ölvander [a]

[a] Department of Management and Engineering, Linköping University, SE-581 83 Linköping, Sweden
[b] SAAB Aeronautics, SE-581 88 Linköping, Sweden

ARTICLE INFO

Article history:
Available online 21 July 2012

Keywords:
Software Product Line
SPL
Knowledge Based Engineering
Configurator
Model Based Development
PDM

ABSTRACT

Context: "Reuse" and "Model Based Development" are two prominent trends for improving industrial development efficiency. Product lines are used to reduce the time to create product variants by reusing components. The model based approach provides the opportunity to enhance knowledge capture for a system in the early stages in order to be reused throughout its lifecycle. This paper describes how these two trends are combined to support development and support of a simulator product line for the SAAB 39 Gripen fighter aircraft.
Objective: The work aims at improving the support (in terms of efficiency and quality) when creating simulation model configurations. Software based simulators are flexible so variants and versions of included models may easily be exchanged. The objective is to increase the reuse when combining models for usage in a range of development and training simulators.
Method: The research has been conducted with an interactive approach using prototyping and demonstrations, and the evaluation is based on an iterative and a retrospective method.
Results: A product line of simulator models for the SAAB 39 Gripen aircraft has been analyzed and defined in a Product Variant Master. A configurator system has been implemented for creation, integration, and customization of stringent simulator model configurations. The system is currently under incorporation in the standard development process at SAAB Aeronautics.
Conclusion: The explicit and visual description of products and their variability through a configurator system enables better insights and a common understanding so that collaboration on possible product configurations improves and the potential of software reuse increases. The combination of application fields imposes constraints on how traditional tools and methods may be utilized. Solutions for Design Automation and Knowledge Based Engineering are available, but their application has limitations for Software Product Line engineering and the reuse of simulation models.

1. Introduction

Reuse and *Model Based Development* are two prominent trends to improve efficiency in industrial product and software development. Product lines enable shorter lead-time to create product variants by reusing and sharing components (be it software or hardware) [1,2]. The model-based development approach provides the opportunity to increase the knowledge captured of a system in the early phases and to maintain the information in a model throughout its lifetime [3]. This work reports on experience from feature modeling that connects:

(1) complex product development (e.g. aircraft systems) using Model Based Development, and
(2) a simulator product line that contain constraints from 1.

Today, there are many advanced domain modeling and simulation environments that allow detailed simulation prior to components' and products' realization. Those environments evolve continuously regarding both languages and modeling techniques, and the use of tools/languages such as Simulink® [4], Modelica® [5], UML® [6], and VHDL-AMS [7] is increasing in industry. For a heterogeneous system such as an aircraft, there is a need to combine and integrate simulation models developed in different environments into a virtual product in order to simulate the complete system. Simulations are used to predict the behavior and performance of system configurations not yet realized, but also to simulate real product configurations for verification activities and for training operational users. Measured behavior of the real system

* Corresponding author at: Department of Management and Engineering, Linköping University, SE-581 83 Linköping, Sweden. Tel.: +46 13 182506; fax: +46 13 281101.
E-mail addresses: henric.andersson@liu.se, henric.andersson@saabgroup.com (H. Andersson).

0950-5849/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2012.06.014

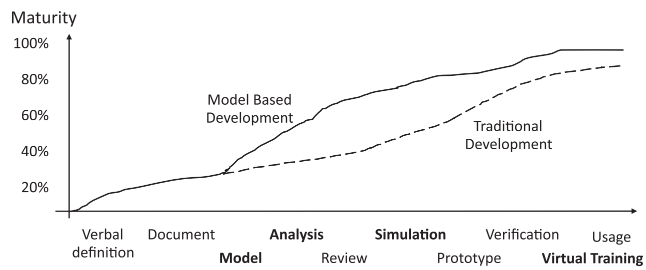

Fig. 1. Product maturity as a function of development progress. With model-based systems engineering the system's maturity increases faster by enabling analysis and simulations of the system from an early stage.

is fed back to the models to improve the accuracy and quality of simulations. The benefit of using a model-based development approach, in terms of increasing product maturity over time, is illustrated in Fig. 1 and based on [8].

Traditionally, simulators for complex systems have been tailor-made and have included large portions of dedicated hardware to offer performance and behavior close to the real system. Such high-end simulators are expensive to realize and maintain and typically have a dedicated support staff, whose task it is to customize and operate the simulators. Customizing a simulation includes the realization of a suitable model configuration and parameter settings to satisfy the end-user needs, where an end-user could be a development team or a pilot instructor. In a typical simulation scenario, a development team needs to verify an upgrade of an aircraft subsystem, for example, the Environmental Control System (ECS) that provides cooling air to the *avionics* (that is *aircraft electronics*). Verification of the ECS behavior and performance is partly performed by simulation. To set up the simulation for the actual needs of the ECS team is defined as simulation customization.

The continuous increase in computing power allows the creation of simpler but still powerful simulators where most, if not all, functionality is implemented in software models. More computing power increases simulation capacity and software-based parts increase flexibility. More flexibility and computing power enlarge the number of simulation configurations that can be made available to engineers and other end-users. However, there are a number of challenges associated with the increase in the number of simulation configurations:

- It must be possible to tailor individual models' fidelity and execution time characteristics to the objective of each individual simulation and to the performance available in each simulator's computation engine, i.e. less computationally intensive model variants will be used in low-end simulators. Consequently, the variations and constraints of individual models must be clearly understood to avoid creating invalid or non-practical simulation configurations.
- The increased number of simulators implies that the number of product configurations that are created for simulation purposes will increase drastically. When there are multiple low-end simulators it is no longer possible to have a dedicated staff whose task it is to create valid simulation configurations. This task must be taken over by the development engineers themselves. Consequently, the model configuration-and-instantiation process to a particular simulator must be simplified such that engineers, who more seldom use a simulator, are able to define a correct model configuration.

Two kinds of product lines can be identified in the situation presented above. There is a *Primary product line* constituting the products that have been realized or are under realization and a *Secondary product line* of simulators and simulation models that can be combined to simulate (or represent) the primary product variants.

The generic parts (subsystems) of a simulator are shown in Fig. 2. Some of their functional responsibilities are described below:

1. **Simulation models** representing the simulated system/product, for instance an aircraft with its immediate surroundings. This part contains several sub-models, which are needed for the simulation of a complex product. It includes parameter libraries, mathematics library functions, and solvers needed for calculation and execution of the simulation. One of the sub-models is the Environmental Control System (ECS) model that is used as a running example throughout the paper. It is described in more detail in Section 5.1.
2. **Hardware in the loop** contains functions that enable the connection of vehicle Electronic Control Units (ECUs) so they can be part of the HILS – hardware in-the-loop simulation.
3. **Operational Environment** performs simulation of other vehicles/systems that interact with the simulated system/product. In military applications, it is called the 'tactical environment'.
4. **Audio/Visual Environment** creates visualization of the outside world, sensor images, presentation of the surrounding environment, and generation of sound/audio.
5. **Instructor/Operator Station** (IOS) & Other Tools includes the human–machine interface for control of the simulator and the operational/tactical simulation.
6. **Execution** contains hardware and software components for simulation execution. It includes time-management (real-time in some simulator types), simulation computers, operating system, data synchronization, and data distribution.

All parts have variable instantiation, but only the simulation models (part 1) have a strong relation to the primary product (the aircraft) and is the focus for this paper.

An additional challenge is that the structures used to manage product data in the primary product line, where information is managed in a Product Data Management (PDM) system, do not align with the structures used to manage the simulation models where Software Configuration Management (SCM) systems are

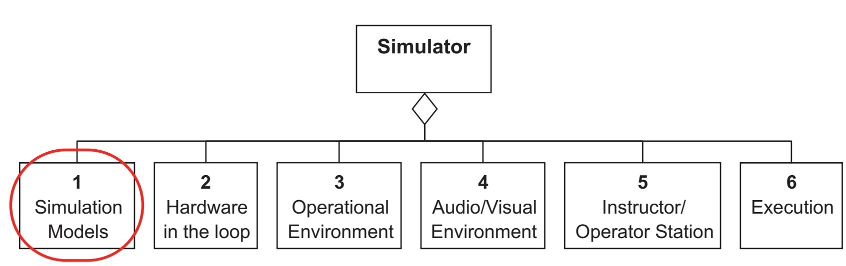

Fig. 2. The top level of a generic product structure for large-scale simulators. Part 1 – *Simulation models* is the focus for configuration support related to structures and data of the simulated product.