

Panel

Models and Analysis

Panelists:

Sébastien Gérard, CEA-List

G. Gogniat, UBS-CNRS, Lorient

Dorina Petriu (chair), Carleton University

Bernard Rumpe, TU Braunschweig

Bran Selic, IBM

Jean-Luc Voirin, Thales

Current context

- **MDD changed the focus of software development from code to models**
- **Analysis and verification of non-functional characteristics NFCs (such as performance, schedulability, security, reliability, etc.) for UML models will enhance the benefits of MDD**
 - **UML may model the application software and the underlying platforms (OS, middleware, hardware)**
- **Many formalisms, methods and tools for NFC analysis have been developed over the years**
 - **e.g., Queueing networks, Petri nets, stochastic process algebra, fault tree, formal logic, schedulability analysis,...**
 - **these NFC analysis models are usually more abstract than UML software models used for software development**

Challenges

- **Scalability of analysis methods and tools**
 - **ability to apply these to industrial scale systems and obtain trustworthy accurate results**
 - **overcoming the cultural bias against or ignorance of formal analyses in software.**

- **Distinguish between “use of analysis to understand the modeled system” vs. “analysis of the model itself”.**
 - **model quality is an attribute on its own (and independent of system quality – is it?)**
 - **is it worth to have modeling guidelines and analyse those?**

Bridging the gap

- **Bridge the gap between modeling languages and tools (primarily UML) and NFC analysis models – one NFC at a time**
 - **Probably the biggest challenge facing us right now, since it has no general solution but needs to be solved anew for each case. In fact, it remains to be proven that automating such domain mappings are feasible/practical.**
 - **Advance model transformation methods and tools for automatic or semi-automatic (computer assisted) derivation of analysis models from UML models.**
 - ◆ **the transformation should bridge a large semantic gap; the source and target model are usually at different levels of abstraction.**
 - ◆ **e.g., QVT**

Bridging the gap (cont)

- **Develop new UML profiles (standard or not yet) for annotating UML models with extra-information necessary for the analysis of different NFC**
 - ◆ e.g., SPT, MARTE, UMLsec, etc.
- **Develop better analysis result reporting**
 - ◆ in UML form as much as possible.
- **Tool interoperability:**
 - ◆ UML editor + transformation engine + analysis tool + result reporting
- **Implications on software methodologies and development process - integrate NFC analysis throughout the software development process (one NFC at a time)**
 - ◆ e.g., existing processes: Software Performance Engineering, Risk-Driven Development.

Model Management

- **Version control for models – especially needed for large teams**
 - **already supported by some tools**
- **Managing the evolution of models (e.g., check consistency between refinements or advanced diff of models)**
- **Managing model results – usually, many experiments are performed for different settings, configurations, workloads, etc.**
- **Quality of model artifacts**

Combining the analysis of multiple NFCs

- **Methodology and process issues**
 - in what order are verified the NFCs (is there a dominant NFC?)
- **How to make trade-offs and judge the relative merits of different design alternatives using different NFC criteria**
 - e.g., security and performance
- **Managing models and results**
 - more versions and more results
- **Transformation from UML to multiple analysis models for different NFCs**
 - the starting point for each NFC is probably a different UML model view annotated with a different profile.
 - how to keep the consistency of these views with an evolving UML model?
 - how to combine the results of the different analysis techniques or even to combine the analysis techniques themselves.
 - ◆ e.g., how to combine schedulability and performance analysis?

Further discussion on "Analysis"

- **If Analysis consists of considering/checking the result of the design against some expected properties of the product:**
 - **Can the product be really designed without incorporating the NFC in the design process?**
 - **If not, how to weave different properties & NFC (at model level)?**
 - **How to express the way each NFC impacts/constrains the design?**
 - **How to measure impact of design choices in one NFC, on other NFC?**
 - **How to help in reconciling these constraints?**
 - **When detecting a discrepancy or inadequacy wrt expected properties, how to find the way to improve?**

... more on “Analysis”

- **If Analysis consists (also) of defining “what to do” (as opposed to “Design”):**
 - **How to define (model, check) the “operational need” (tasks to be completed by the final user of the product, and expected contribution of the product)? And more, how to limit its complexity?**
 - **How to “extract” from this model, the expectations (requirements) towards the product? Especially NFC?
Is model transformation enough? What has to be added (“invented”), and how to help?**
 - **How to check that the requirements obtained for the product comply with the real need?**
 - **How to check that these requirements are feasible? Development Cost, accessible technologies, performance, safety, ease of use, user ownership cost...**

General considerations

- **Do not focus on models, but rather on how to make the solution emerge, and how MDD can help.**
- **Consider how to formalize and support intellectual process to build the solution:**
 - **models are [only] “bus stops” in the travel; how to run from one to another?**
- **Consider building a true knowledge management system, able to formalize and exploit the engineering know-how (design rules, compromise criteria, checking rules...) and the way model definition/transformation can support this.**