
Gurvan Le Guernic

DGA Maîtrise de l'Information

DSM – Domain Specific Modeling Workshop
October 20th, 2019
Goal: give some feedback on an experiment to develop a model-based process from a document-based one.

1. Context of the work
2. Overview of the process and supporting tool developed
3. Feedback on the experiment
4. Discussion on some observations
Context
An Atypical Context

- 2 main stakeholders equally involved with the models
  - covers 2 independently owned processes

- Models used to:
  - communicate on design decisions ... 
  - ...and their justifications
  - not to generate implementation artifacts

- Context applicable to:
  - involved procurement agencies + contractors
  - certification agencies + vendors
Overview of the Engineering Process

- **CcS**: ~ Security Target
- **DEC**: ~ Architecture Design Documentation
Realization
Precise Definition of the Modeling Process

Focus on the precise definition of the process

Every task is decomposed in 5 to 10 linear subtasks

Based on the existing document templates/requirements

Define precisely: what to do, when, and how
Security Analysis: Step 1

Model Initialization

![Model Explorer](image)

- View: Black box
  - Actors
  - Assets
  - TOE Life cycle
  - Security services
  - Security functions
  - Non security services
  - TOE environment
  - External communications
Security Analysis: Step 2
Initialization of the Attack Tree: Feared Events
Security Analysis: Step 2
Threat Analysis relying on the Attack Tree

G. Le Guernic
Specialized Multi-view Multi-stakeholder Model-Based Engineering Framework
Security Analysis: Step 3
Security Requirements Elicitation

- Generation of the “Security Target” document (CdS)
- Export of the model to the design teams
Architecture Design: Step 1
Model Initialization

[Diagram showing a model structure with views for Black Box, Implementation Independent White Box, and Implementation Focused White Box, including subcategories like Actors, Assets, TOE Life cycle, Security services, Security functions, Non security services, TOE environment, External communications, Implementation independent architecture, Power supply state machine, Authentication state machine, Functional state machine, Implementation focused architecture, Implementation focused allocation]
Architecture Design: Step 2
Architecture Design (a.k.a. SysML Modeling)
Overview of the SysML Profile
Usability-Related Observations: Successes

- Clear and explicit definition of the associated methodology
  (*who does what when and with which information pieces*)
  - facilitate the design of the model-based process
    - shapes the meta-model and its “fragments”
  - help users apply the model-based process
  - information reshaped as a “recipe” separating: what to do; and how to do it

- Tool customization
  - in association of the previous point, only show relevant entities

- Automatic generation of (intra and inter) coherent documentation
Usability-Related Observations: Failures

- Difficult to respect the model hierarchy/organization over time
  - in a diagram, Papyrus creates objects under the diagram’s root objects
  - more tool customization?

- Difficult to preserve model consistency
  - many invariants exist that have to be preserved
  - some (OCL/Java) rules help, but may be difficult to design, in part due to inter-relations between those invariants

- Implementation difficulties causing stability issues
Design-Related Observations

- For (dynamic) information-related models, nearly always distinguish between content and container.

- Lots of SysML objects for representing the same “domain” object:
  - consistency issues
  - does not follow a Model-View-ViewModel
Discussion
SysML: a Meta-model of “Views”

Multiplication of “modeling objects”

- A single service or security function is represented by:
  - a UseCase instance in order to capture accessibility by users and inclusion relations in UseCase diagrams;
  - an Activity instance in order to describe its behavior in an Activity diagram and for linking with other model elements;
  - some CallBehavior instances to reflect execution of the corresponding function in another service or function behavioral description;
  - an Operation instance in order to register which architecture block implements this function;
  - some Behavior Execution Specification instances to represent its execution in Sequence diagrams;
SysML: a Meta-model of “Views”

Multi-models vs Model-View-ViewModel

G. Le Guernic
Specialized Multi-view Multi-stakeholder Model-Based Engineering Framework
SysML: a Meta-model of “Views”

Analogy with technical engineering drawing
Syntax vs Semantic

- Syntax entities are *interpreted* to give rise to semantic entities.
- Interactions are done on the syntax with a meaning on the semantics.
- Standard modeling tools explicit the definition of the (abstract) syntax, but not its semantics.
- Optimizations can be achieved by decorrelating the (abstract) syntax and semantics.
## Conclusion

G. Le Guernic

Specialized Multi-view Multi-stakeholder Model-Based Engineering Framework
Conclusion

- Importance of an explicit and detailed definition of the process supported and the information manipulated
- Importance of a dedicated and customized supporting tool
- Difficulty to develop dedicated tooling supporting a specific MBE process
- Difficulty to embed complex metamodel into SysML
  - use SysML as a communication and import/export interface, but maybe not as the core metamodel