UML OR DSML?

YOU CAN NOW HAVE BOTH WITH PAPYRUS 1.0!

Ulf Olsson  
Toni Siljamäki  
Francis Bordeleau  
Ericsson AB

NWADSL@EclipseConEurope2104
OUTLINE

› What’s the problem?
› How did we go about solving it?
› Experiences
› Conclusions
WHAT IS THE PROBLEM?

› Large development organization
  - 25000 R&D personnel out of 114000 world wide

› Wide range of products
  › Radio, IP, transport, mobility, communications, OSS/BSS, cloud infrastructure, …

› Continuity and compatibility
  - Modeling support
    › Language, validation, transformation
  - Collaboration support
  - Stakeholder support
    › Portfolio, design, implementation, marketing

› Efficient tool development
  - Leverage community development momentum
THE GOAL:

Get into the head of the next guy

Find the tools to bridge this gap.
And the gap to the machine!
Behind this:

Old method: 3GPP standardization by telco industry consensus

You find this:

Problem: this only a part of our scope, and the world moves faster now…
So, what do we do instead of paper and drawing tools?
MODELING APPROACH

› Functional modeling with interfaces works well to discuss and agree end-to-end behavior
  – Hierarchy of logical functions
  – Expandable logical interfaces
  – Information elements
  – Use-cases, signal flows
  – Complemented with topological models
COLLABORATION APPROACH

Central coordination, distributed way of working

All workbenches hold the complete model(s)
Cycle: evolve/propose/commit/distribute
Supports studies based on stable architecture

Git repository managed by Gerrit Central
“We need to ensure architecture and product consistency across the company”
Network Architecture Modeling DSL
NWADSL – Short Background

- **NWADSL** – Short Background
  - Very Domain Specific Language – Modeling of Network Architectures
  - Guiding document: Network Architecture Evolution Strategy
  - Defining the NWA language – Concepts, Modeling Elements, Colors
  - Modeling Rules, WoW, Guidelines, Wiki
  - Example Models (PowerPoint slide-kit)
  - Technical Icons Library

- **NWADSL** – Start Modeling
  - PowerPoint Modeling
  - Modeling in Visio
  - Whiteboard
Technical Icons Library
The first NWADSL tooling (1)

Top-level diagram:

Domain-level diagram:

Function diagram:

Interface diagram:
The first NWADSL tooling (2)

Sequence diagram:

Sub-sequence diagram:
Question: Why Not UML ???

- Why not Open Source UML tools for NWADSL instead of in-house tools?
  - Evaluation of Ericsson DSL’s and tooling on Papyrus was ongoing.
  - **Why not create an NWADSL in Papyrus UML?**
  - (an EMF-based NWADSL was also discussed, but we like to reuse capabilities of UML and open source tooling, not reinventing them)

- Initial discussions with the Papyrus vendor CEA
  - Example graphics in Papyrus by CEA
  - Testing stereotypes, SVG shapes, stylesheets
  - …showing that it could be done in Papyrus 😊

- First pilot: Test Papyrus Graphics Capabilities
  - **Problem:** Replace UML elements with SVG’s
  - The initial support in Papyrus was primitive...
  - …but the first pilot project was successful.
  - …showing GREAT potential for improvements.
Customized NWADSL in Papyrus

- NWADSL - Main Parts
  - NWA Profile
  - Customized Palette
  - SVG Graphics Library
  - CSS stylesheet rules for graphics rendering

(rules based on stereotypes and property values)
NWADSL Modeling in Papyrus (1)

Top-level diagram:

Function diagram:

Domain-level diagram:

Interface diagram:
NWADSL Modeling in Papyrus (2)

Sequence diagram:

Sub-sequence diagram:
Multiple DSL’s in Same Model

- **NWA Vision:**
  - Integrate NWADSL with other DSLs and plain UML.

- **Some NWA Use Cases:**
  - De-compose into finer-grained NWA Functions. (done)
  - Combine with NWA Info Modeling. (done)
  - Combine with plain UML modeling. (done)
  - Combine NWA modelling with BPM.
  - De-compose NWA Functions into ”design” models, like Executable UML.

- **Proof-of-concept showcase models needed:**
  - Combine NWADSL with another Ericsson DSL in the same model.
  - Combine NWADSL with a ”design language”, like Executable UML.
**Executable UML DSL Prototype**

- **Quick DSL Prototyping:**
  - Develop an Executable UML Profile (basic OCL restrictions)
  - Develop Customized UI Palettes
  - Then migrate an existing model...

- **Model Compiler Prototyping:**
  - M2M + M2T capabilities required
  - Template capability is also required
  - QVTo and blackboxing using StringTemplates can solve this 😊

- **NWA Modeling showcase thinking:**
  - When linking these technologies to NWADSL we prove we can go from Network Modeling down to code.
Integrate NWA + ExecUML DSL’s

NWA model

Exec UML model

decompose
Integrate NWA + Baseband DSL’s

NWA model
Baseband model

decompose

LIFE AT THE COALFRONT

The good
- 103 people on the mailing list
- 16 active architects
- …and this is just the beginning!

The bad
- Our goal is to work on a model-based level of abstraction, therefore all tools must work on the same level
  - Luna 1.0 had text-based compare only, which caused corrupt models. Very off-putting…
  - The EMFCompare project (Obeo, EclipseSource) has made major progress in the past months
    - Possible to create corrupt models if upstream commits are not synchronized
    - Pushback from the users: too complex, “looks like programming”
WISH LIST (EXCERPTS)

› Simple UI
  - As few irrelevant items as possible

› Scripting
  - For local adaptation

› Creation of navigation-enabled HTML
  - For information sharing with non-modelers

› Improved focus on graphical interaction
  - Example: UI for EMFCompare is on the right track, but not intuitive enough
  - Things done in the diagrams sometimes has unintended effects on the model

› …
CONCLUSIONS

We need:

› Flexibility
  – To achieve the right abstraction

› Integration
  – For efficiency in the whole development and integration chain

› Ease of use
  – To lower the threshold for architecture-level users

› Efficient collaboration
  – To support the entire organization

…and a community approach
  – To leverage innovation and creativity inside and outside the company
UML OR DSML?

› Not a question of “OR”, we have both!

› Presents a DSML to the user:
  - Adapted to the problem space (actually, spaces!)
  - Customized: no UML expertise required

› Based on UML:
  - Leverages years of work of top modeling and tool implementation experts
  - Builds on the semantic richness of UML

› Benefits:
  - Facilitates integration with other languages
  - Accommodates other UML-based technologies (present and future)
  - Provides a full language support infrastructure (editors, renderers, compilers, debuggers etc.)

› Drawbacks:
  - More complex to define that “pure” (built from scratch) DSML – UML expertise required