

NWADSL 2016: FROM THE COALFACE

Facts, observations and personal opinions
Ulf Olsson and Lennart Holm, Ericsson AB

STRUCTURE

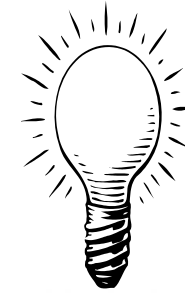
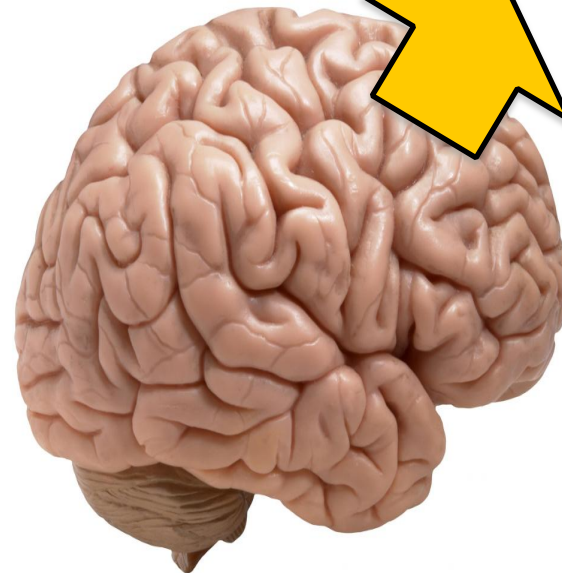
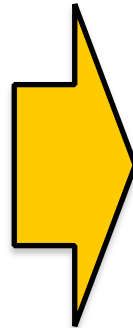
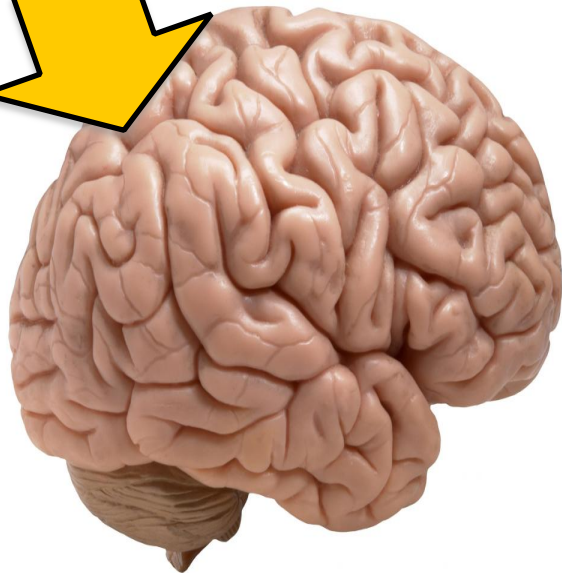
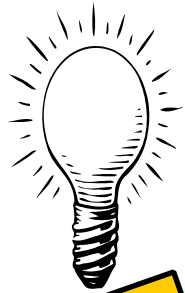


- › The mission
- › Timeline
- › What have we done
- › What could be improved
- › Next steps

THE MISSION



“We need to ensure
architecture and product
consistency across the
company”



THE METHOD



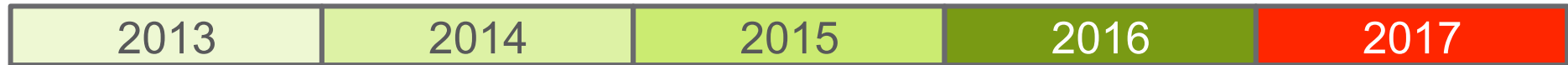
- › Functional modeling with interfaces works well to discuss and agree end-to-end behavior
 - Hierarchy of logical functions
 - Expandable logical interfaces
 - Information elements
 - Use-cases, signal flows
 - Complemented with topological models
- › Adapting the graphics worked; Papyrus mostly does what it says on the tin
 - But takes some effort... (more on that later)
- › Major takeaway from the first year:
Collaboration tools are just as important as graphics (if not more so)!



TIMELINE



Prototype
Pathfinder projects
All domains modeled
Models cleaned up,
tool upgraded



Platform	Homegrown	Luna	Still Luna...	Mars	Neon
Focus	Explore requirements	Basic modeling	Collaboration	Drawing conclusions (tool and models)	UX improvements Widening the user community

IT SEEMED A GOOD IDEA AT THE TIME... (AKA WE ASKED THE USERS)



1. What do we get when we ask our users?
 - This was reinforced by the Consortium GUI study



Fix the GUI!

2. Make it possible to use the knowledge captured in the model in more convenient ways!

- Architects interact with:
 - > Other architects
 - > Designers
 - > Product management
 - > Planners



DETAILS



- › The benefit of Eclipse is that it can be made to do anything.
- › The drawback of Eclipse is that it can be made to do anything.
- › A typical user is frequently shown commands and presentation options that are irrelevant for the task at hand
 - Creates apparent complexity and increases the opportunity for user error
 - Good user guides help, but is really not the way of solving the problem
- › Export to other formats is a prime requirement
 - HTML via MMA works, but has its idiosyncracies
 - Word via Gendoc, but lacks recursive descent (essentially a Word problem, but anyway)
 - Editable Powerpoint: unsolved!
- › Collaboration via Git is still a mystery to most people
 - Most frequent source of fatal model corruption
 - Configuration management now on file level; IMHO it should have been on model element level (Model Element Management rather than Source Code Management?)

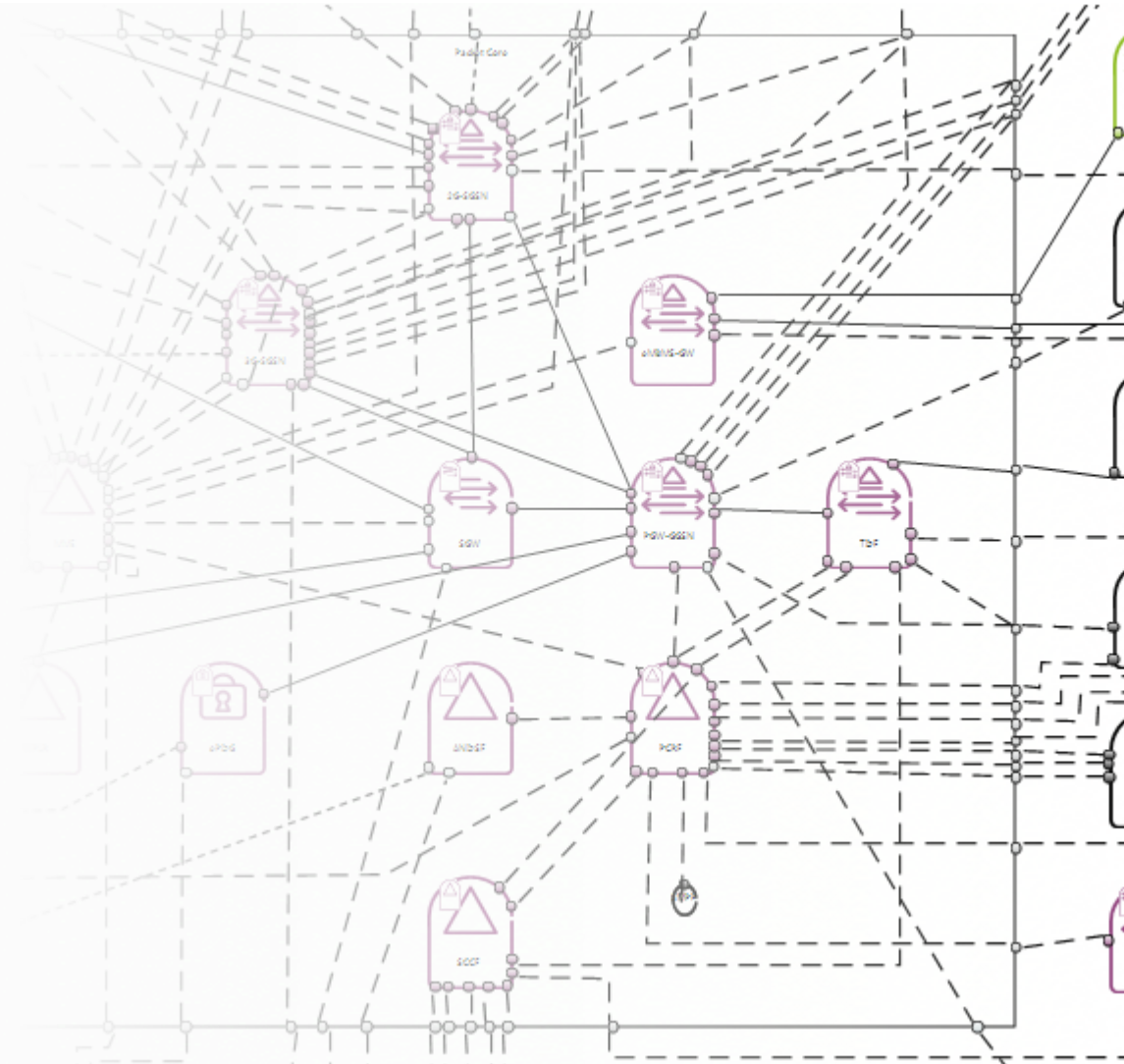
WHERE ARE WE?



- › 12 domains modeled
 - 27 models in total
- › 3693 components
- › 553 diagrams

- › Large?
Not really, but lots of information in there

(OTOH, we do have major performance issues...)



MAJOR CONCLUSIONS



- › Creating a DSL on top of Papyrus works, but... (non-exhaustive list)
 - Much more dependent on code than expected
 - Writing code against Eclipse/Papyrus requires detailed understanding that is sometimes hard to acquire
 - UML (?) sometimes places unexpected restrictions on what we can do
 - › Example: no component instances in component diagrams, but ok in class diagrams?
 - Sometimes hard to predict side effects
 - › Example: linking signals to components requires ports, but then dependencies are between ports and not components...
 - The projects are busy working on release n+1, whereas our plugin developers need support on release n (or n-1...)
- › Large models uncover performance issues
 - Hard to debug without access to actual data -> access issues
 - Some faults triggered by garbage left from previous versions



WHERE DO WE WANT TO GO?



- › Model based reviewing
 - EMFCompare now stable enough to be usable, but needs a GUI that is closer to what the modeler works with (“DSLCompare”)
- › Access control on multiple levels
 - Model, branch, package, model element, ...
- › Less UML, more DSL
- › Linking to design models
 - Two-way traceability
- › The model(s) as the natural and authoritative source of information about what our systems can do and how they do it

“Don’t search the intranet, use the model!”



ERICSSON