



ERICSSON

REFLECTIONS, LESSONS LEARNED AND POSSIBLE FUTURE DIRECTIONS OF MODEL DRIVEN DEVELOPMENT WITHIN ERICSSON

Diarmuid Corcoran: Expert Software Architecture / DURA
System & Technology

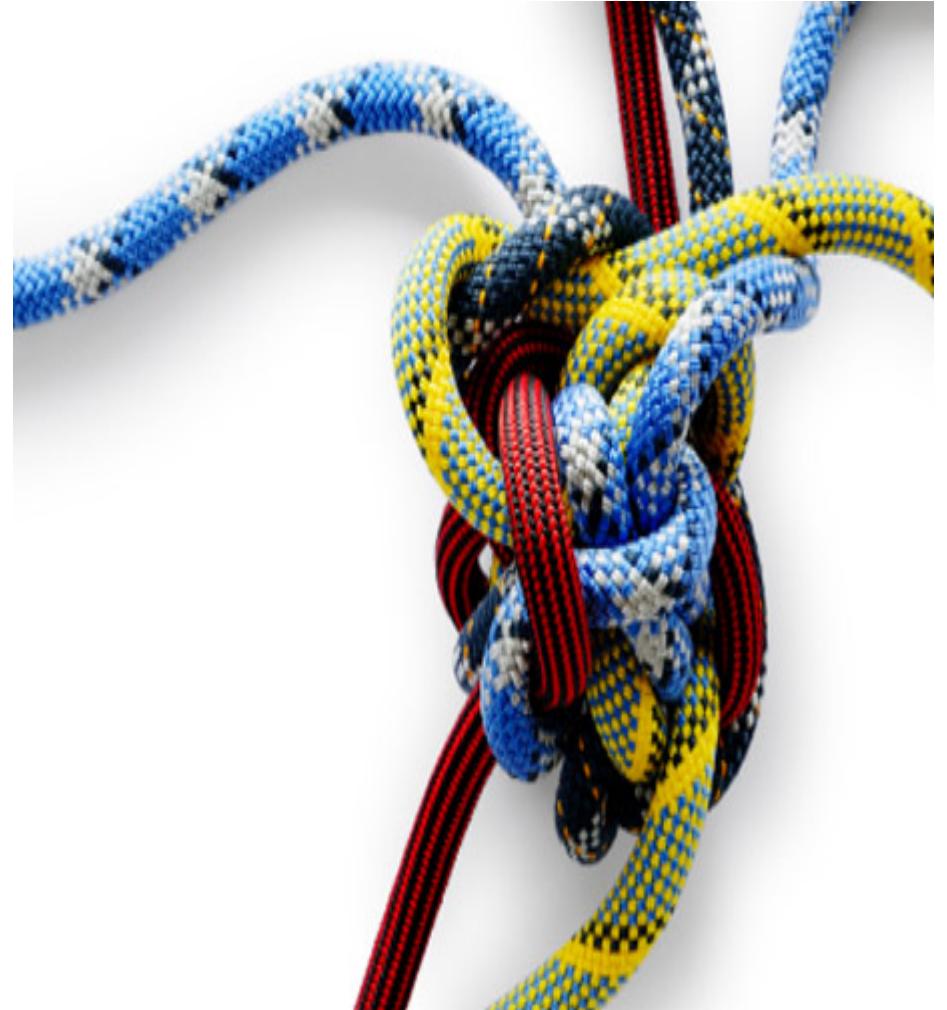
WHO AM I?

Worked at Ericsson since 1992. Primarily in software and system roles.

Have been very active in the Ericsson modelling community through the years, though recently less so.

Hold a position as “Expert in Software Architecture”

Primarily working on software research activities to support next generation radio networks.



DISCLAIMER :



- These are my personal reflections and options.
- The presentation captures my personal journey in MDD.
- Lots of stuff I won't mention : For example - Model Driven Test.

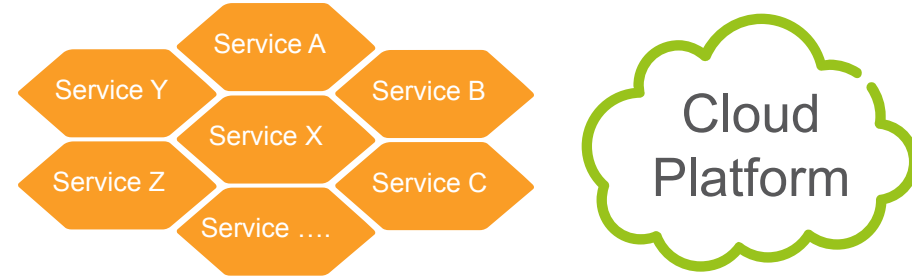


ERICSSON SYSTEM DOMAIN



Software for system management and services

Service & Management



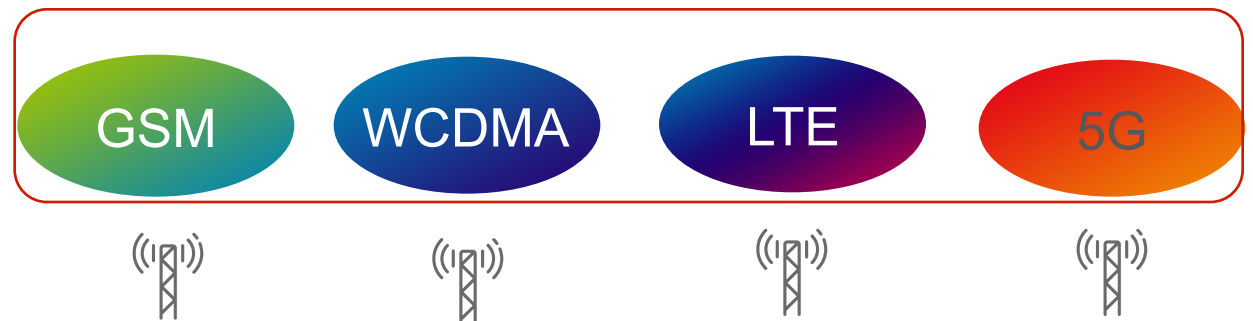
HW platforms and software for core network

Core Network



Diverse radio access platforms: HW and software

Radio Access



IPR and standards for system access

Devices



THE STORY LINE



1996-

“The Early Days” : Experimentation with MDD

2001-

Large Scale System Development

2009-

DSL’s Start to Dominate

2012-

Agile and Open Source

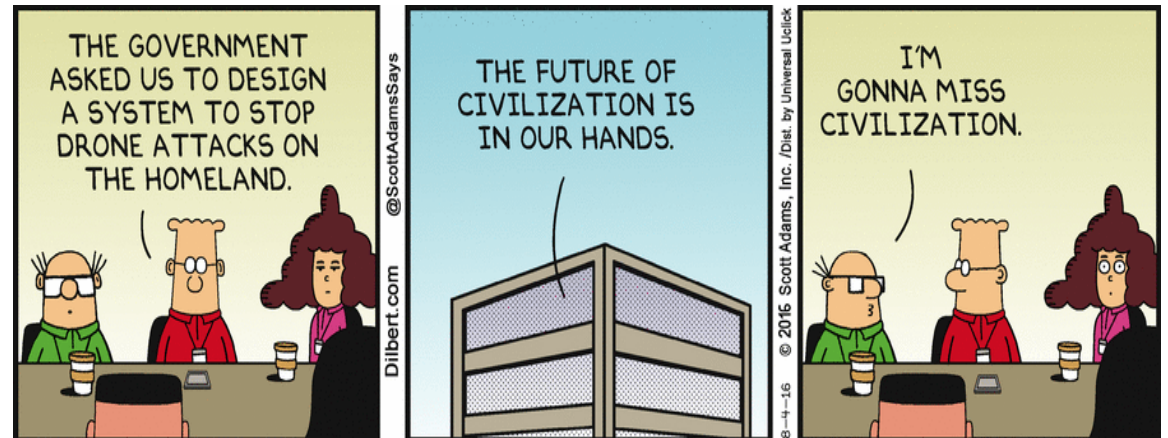
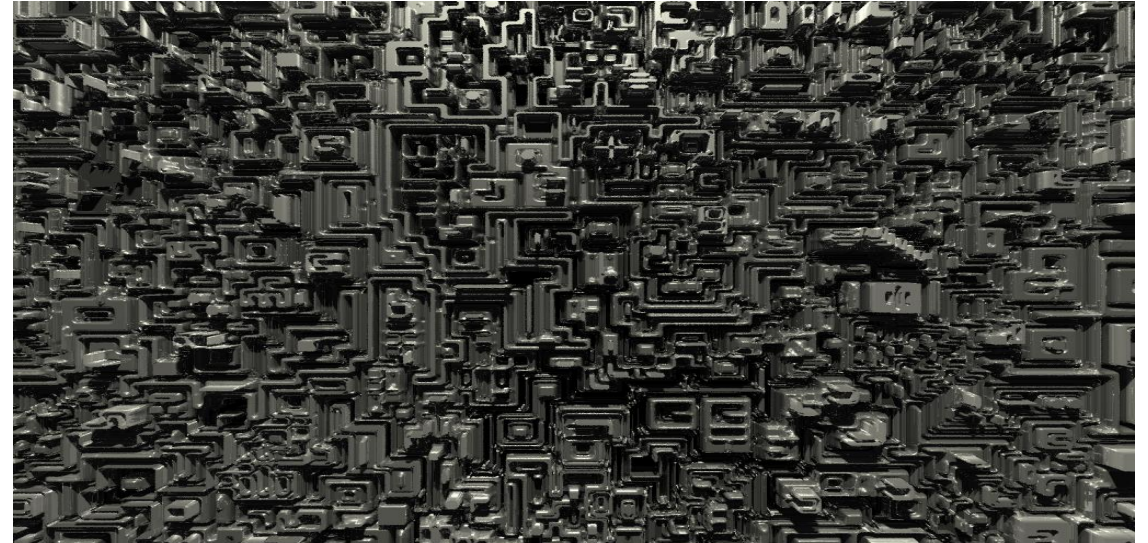
2017-

New Opportunities in MDD

WHY MODEL DRIVEN DEVELOPMENT?



- › We build complex systems with lots of dependencies.
- › Complexity has a technological aspects and a human aspect.
- › MDD is a tool to help fight this complexity
- › Complex system and software development will continue.
- › Next generation systems will have much more autonomous responsibility.





“The Early Days” : Experimentation with MDD

RATIONAL ROSE AND JDI

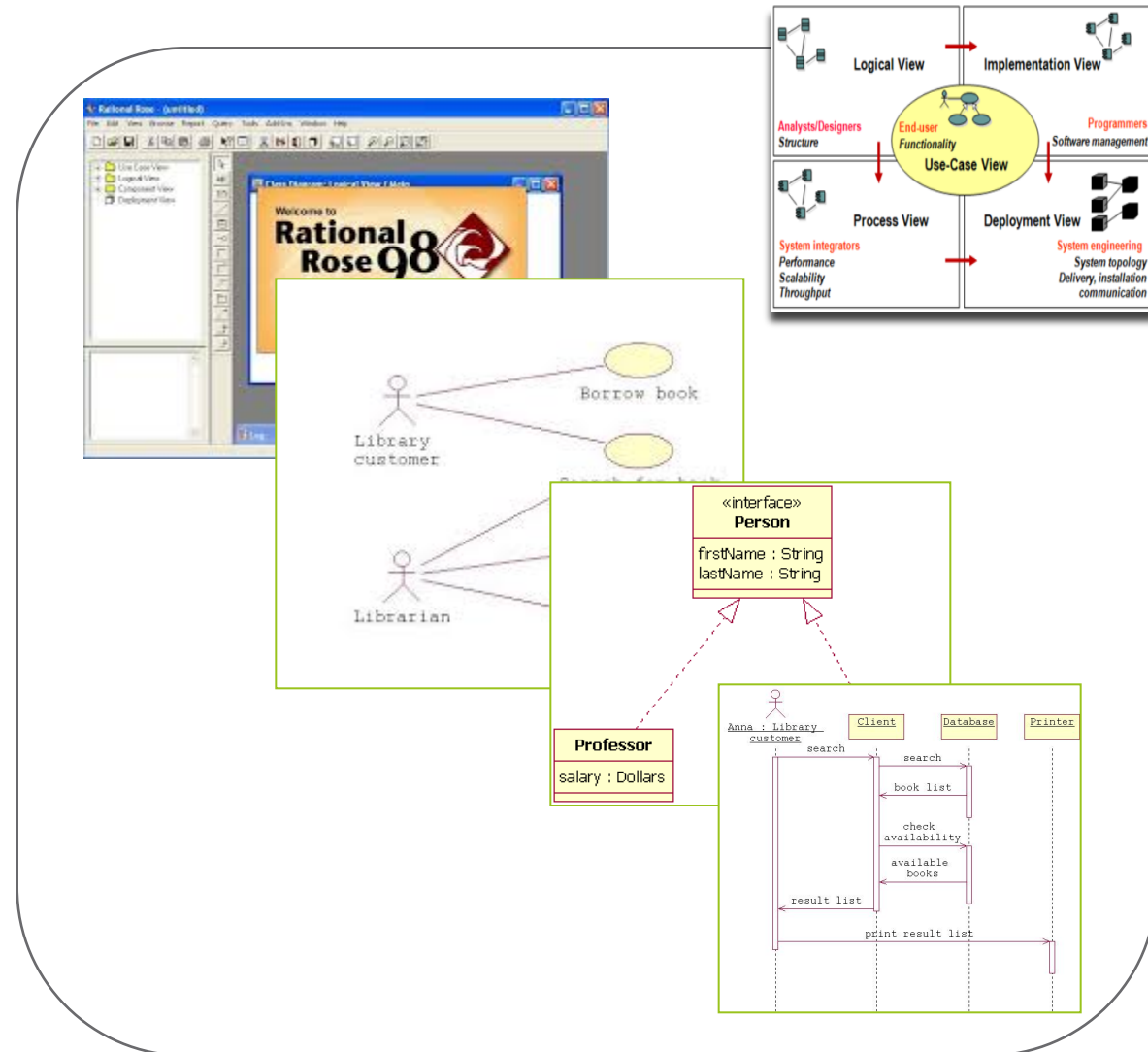


JDI (Joint development Initiative) was first large scale cooperation between Rational and Ericsson.

Much effort on integrating RUP and 4+1 view on our legacy Ericsson processes. Integrating with requirements view

We went a bit crazy drawing sequence diagrams.

And were generally left with an unsatisfactory feeling that something was missing.....



OBJECTIME : MAKING THE MODEL DO STUFF

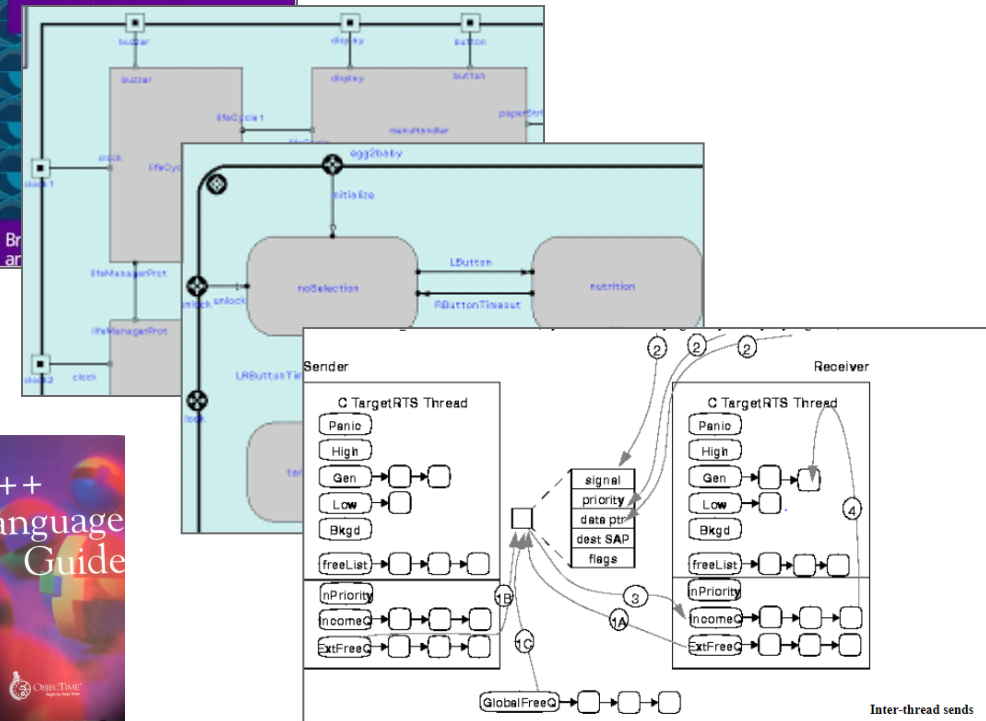
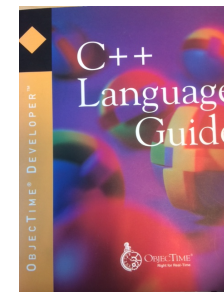
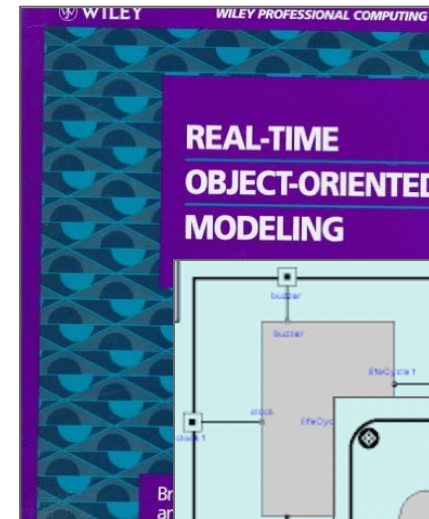


Objectime was a revelation when we first discovered it! **We could finally make our models do stuff!**

ROOM was a simple DSL that was easy to learn and matched our embedded telecom domain very well.

The build environment and run-time system hid much of the initial platform complexity.

But there was something missing.....The tool was poor at handling the system view.



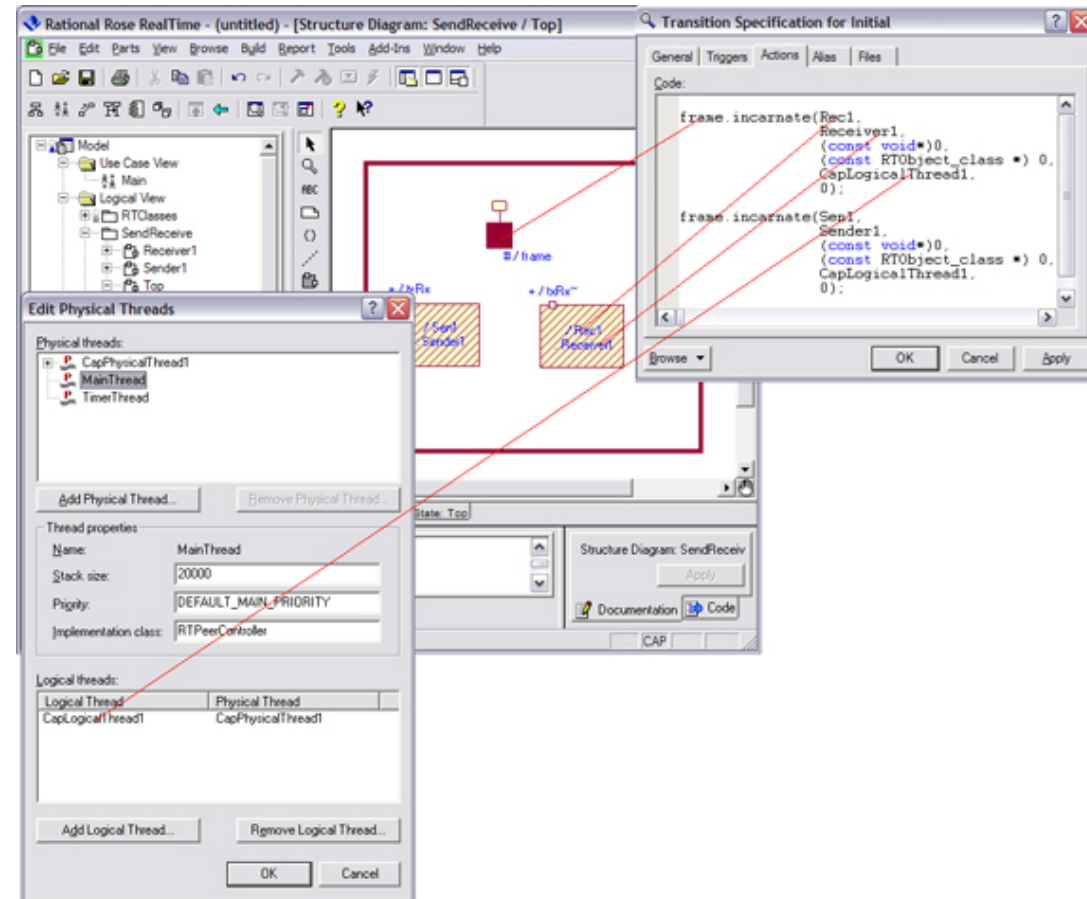
RATIONAL ROSE REAL-TIME



Objecttime and Rational Rose integrated. Important as it combined the best of both tools.

Provided the base for large scale MDD for many years. Still very much in use today

Eventually migrated into an Eclipse plugin as RSA-RT. **A process which took far too....long!**



MEANWHILE: SHLAER-MELLOR AND EXECUTABLE UML

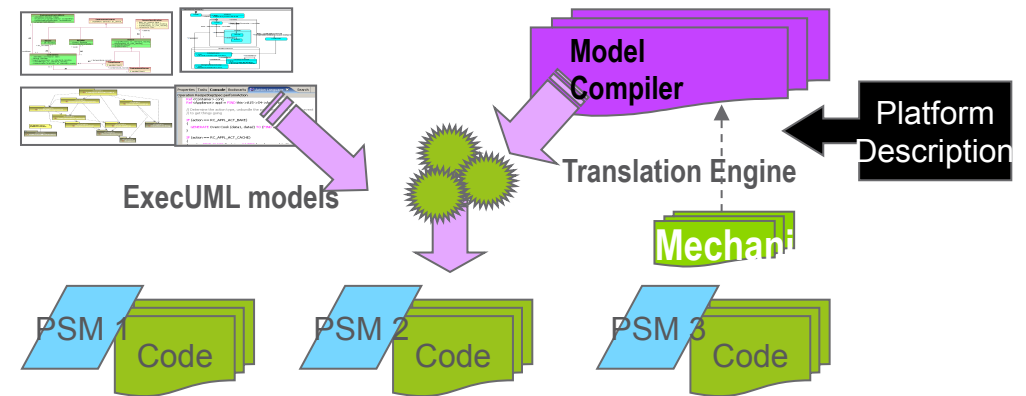
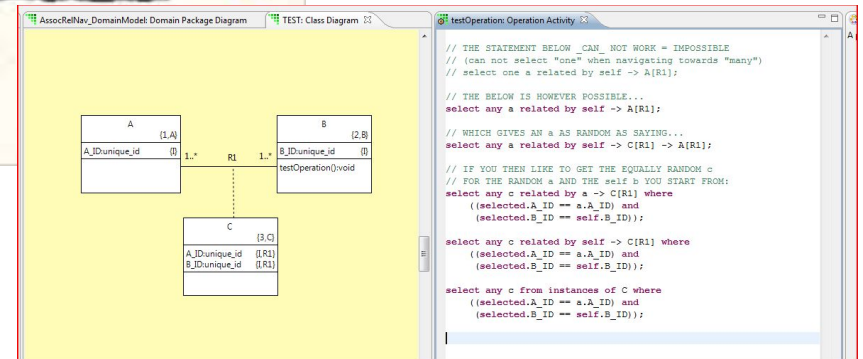
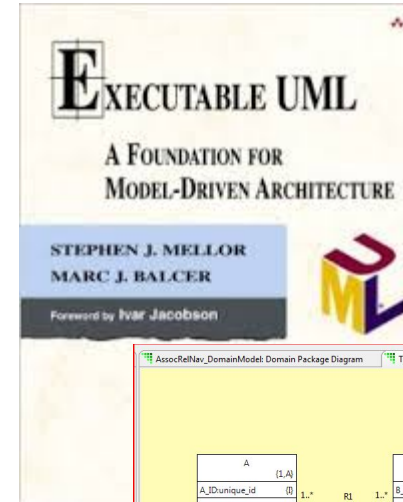


Some of the first “large” MDD experiments actually used the S-M method. Both Kennedy-Carter and Bridgepoint toolsets.

1996-1997 there was a very ambitious effort to use S-M for parts of AXE based BSC (Base Station Controller).

Many other efforts continued through the years but ultimately the strangeness off S-M notation and the domination of UML lead to sidelining this approach.

We did very much like the idea of an implementation independent action language.



WHAT DID WE LEARN?



- The importance of covering all views and flows in one tool.
- MDD tools are complex and difficult to learn.
- UML is large and not always easy to apply.
- But that MDD did help us to combat complexity.





Large Scale System Development

MDD BECOMES A KEY PHILOSOPHY IN WCDMA, LTE AND 5G SYSTEMS

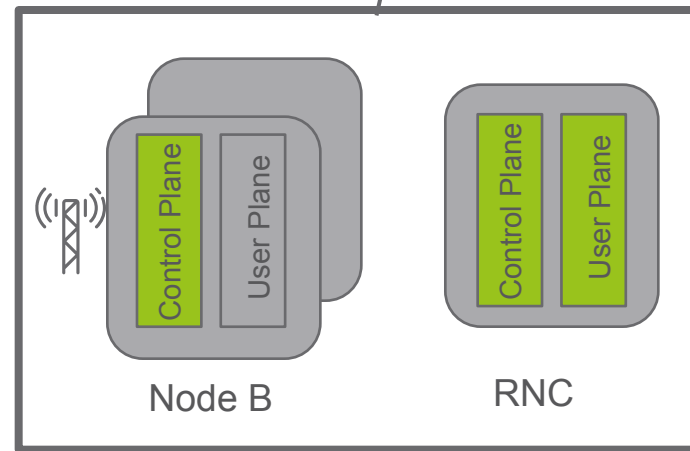
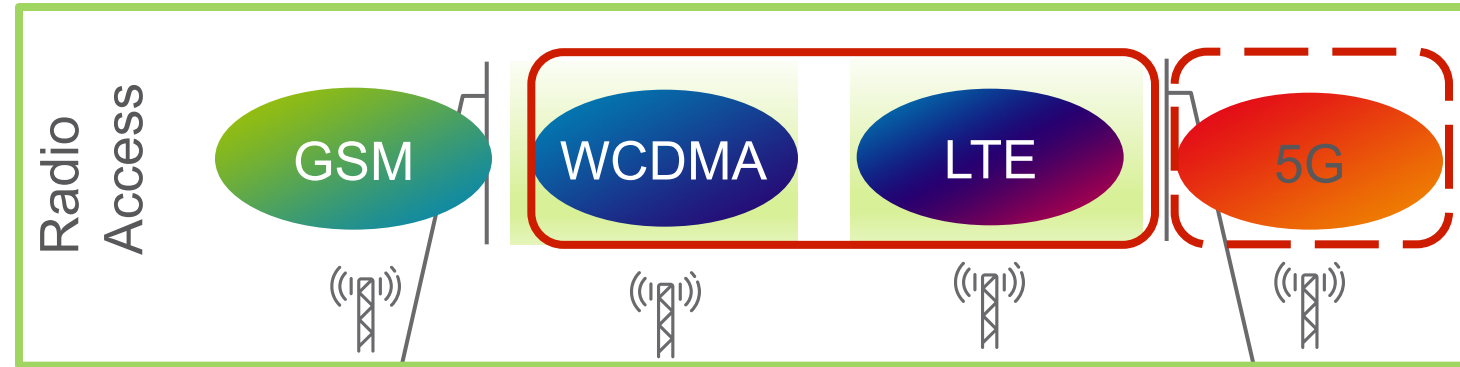


Model Based System Engineering at WCDMA RAN

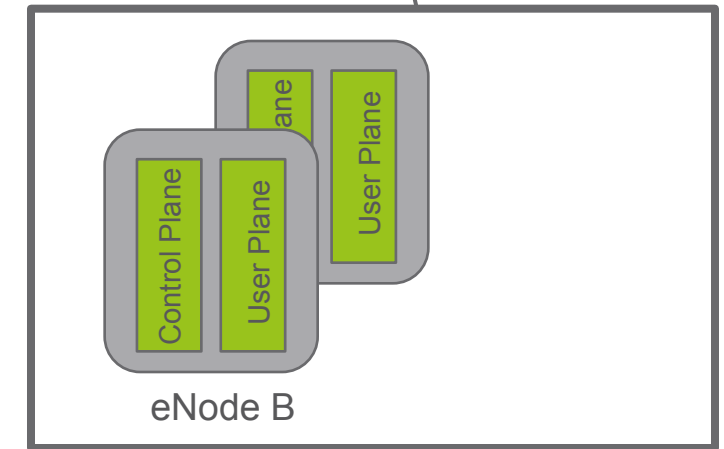
Software Design and Implementation of WCDMA RBS and RNC Control and User Plane.

Software Design and Implementation of LTE RBS Control Plane.

Software Design and Implementation of LTE RBS User Plane using DSL's.



WCDMA RAN



LTE RAN

MODEL BASED SYSTEM ENGINEERING – WUM (WCDMA UNIFIED MODELLING)

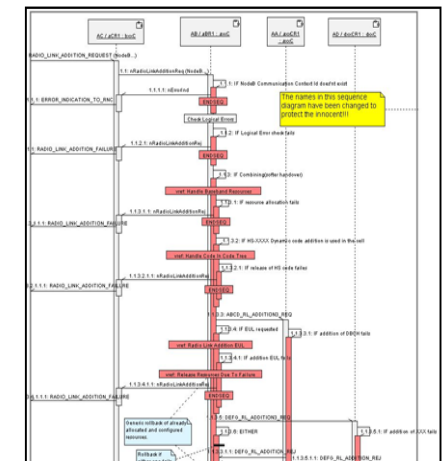
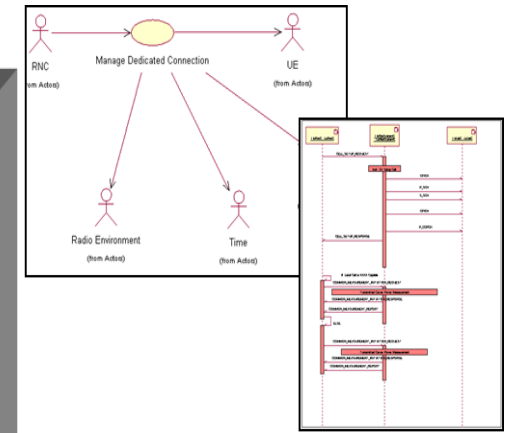
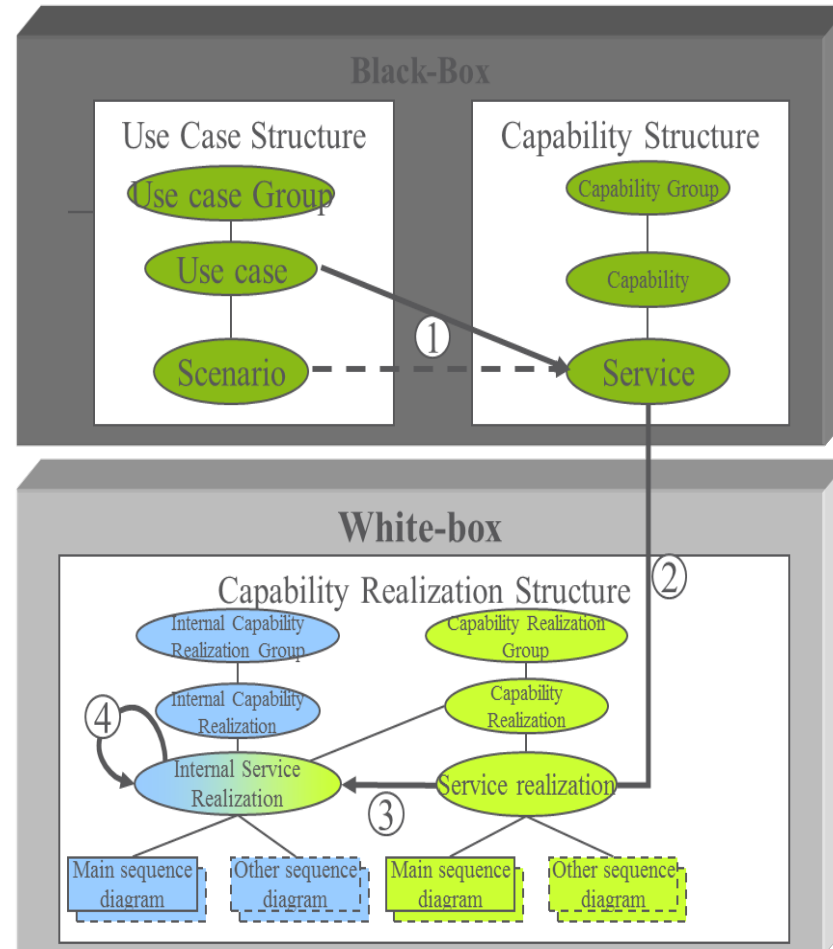


MDD systems approach using a process called “**MBSE**” (Model Based System Engineering).

Move from highly document driven process to repository “single source of information” approach.

Objective: Significant reduction in specifications faults.

UML-RT based with UML2 + SysML like extensions.

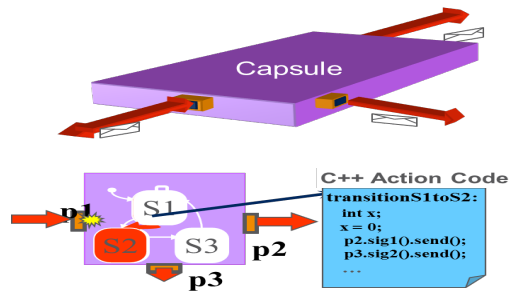


SOFTWARE DESIGN AND IMPLEMENTATION USING MDD WITH UML-RT

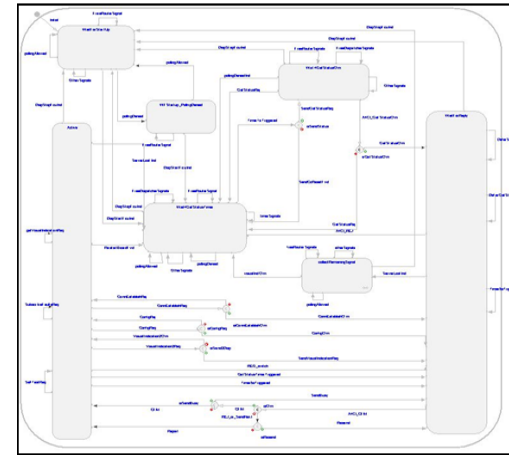


Objective: Battle design implementation complexity using a subset of UML called UML-RT.

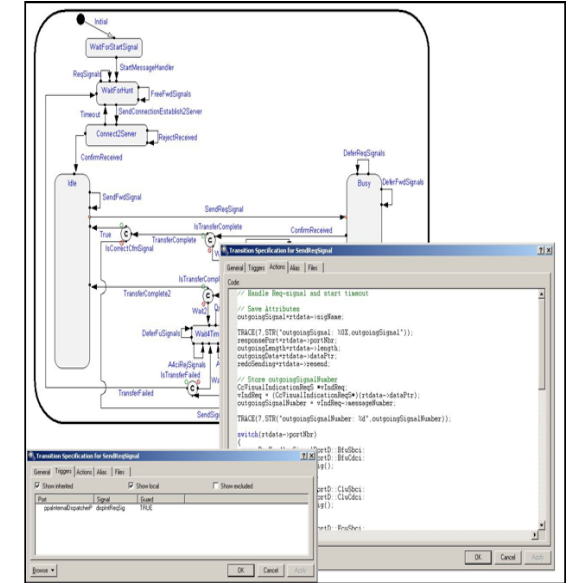
Capsule and state-machine abstractions provide very powerful abstractions for dealing with reactive embedded systems.



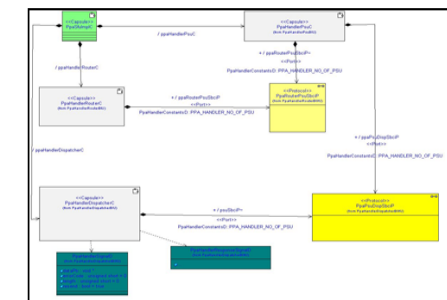
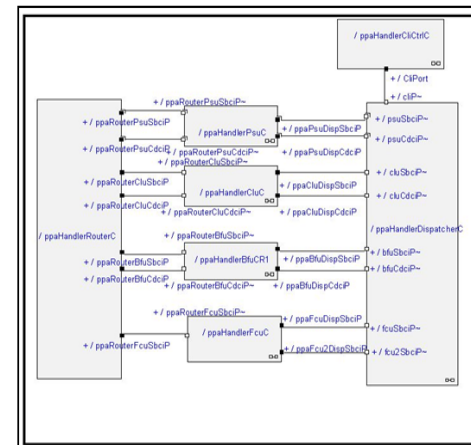
C++ action language. Run-time system abstraction to hide the underlying platform.



Behavior – State Machines



Behavior – Action Code



Structure – Class Diagrams

MBSE - WUM - WHAT DID WE LEARN?



Semi-formal specification.

Great tool for system analysis.

Fantastic way to store system knowledge.

Clear and useful connection to requirements.



Scalable configuration management.

Lack of executability in the model.

Impedance in changing mindset.
Superimposing on legacy ways of working.

Toolset lock-in.

UML-RT IMPLEMENTATION - WHAT DID WE LEARN?



Common unified design patterns.

Simple and concise abstractions
Capsule / States / Ports

Model is the code philosophy.

Very useful run-time layer.



Overuse (abuse) of UML-RT constructs.

C++ actions language. Mixing abstractions in model.

Complete lack of control of code generation process.

Toolset lock-in.

Configuration management.



DSL's Start to Dominate

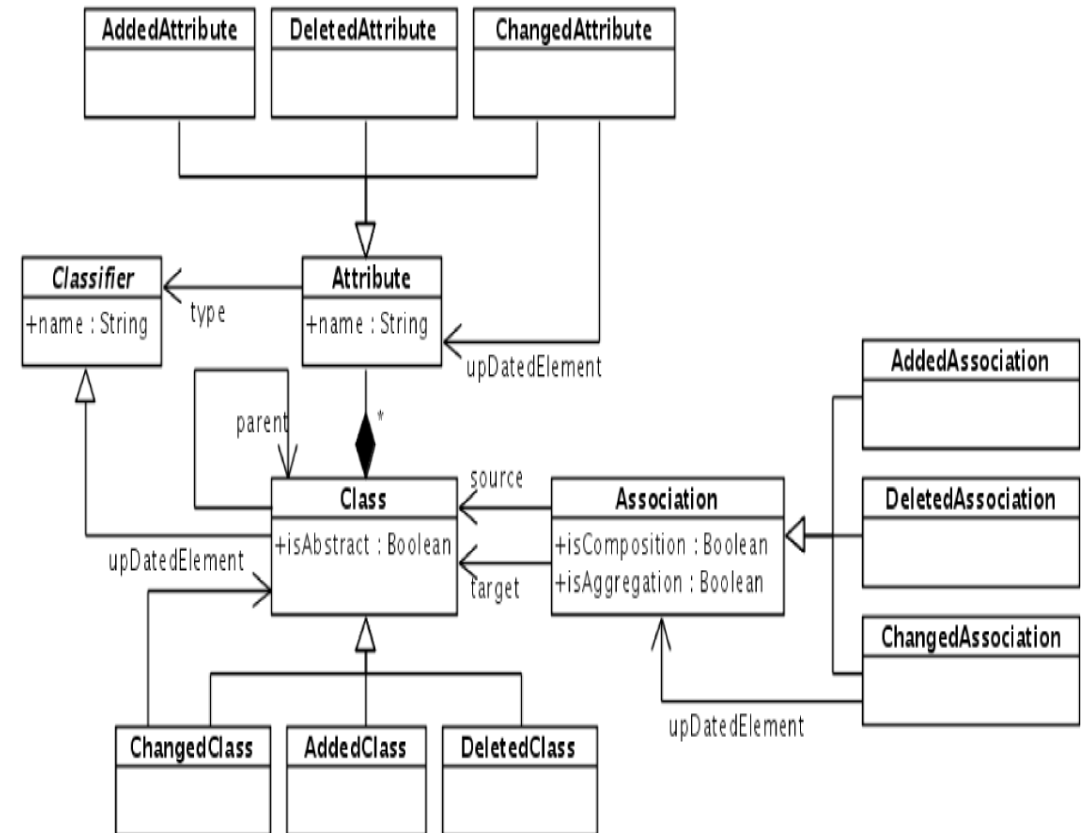
THE JOURNEY TO DSL'S



Prerequisite: Made the transition to Eclipse based UML tools. But still by and large proprietary.

WOW!: UML has a meta-model and we can navigate and manipulate it!

But: Now we could tune surface syntax of UML and code generate as we needed.



TECHNOLOGY ENABLER: THE ECLIPSE ECOSYSTEM



- Eclipse provided a lot of the fundamental building blocks for enabling DSL development.

- Tools for M2M transforms.
- Tools for M2T transforms.
- Tools for editor and syntax development.
- Frameworks for generated target code for c, java, etc.
- All integrated into one environment.



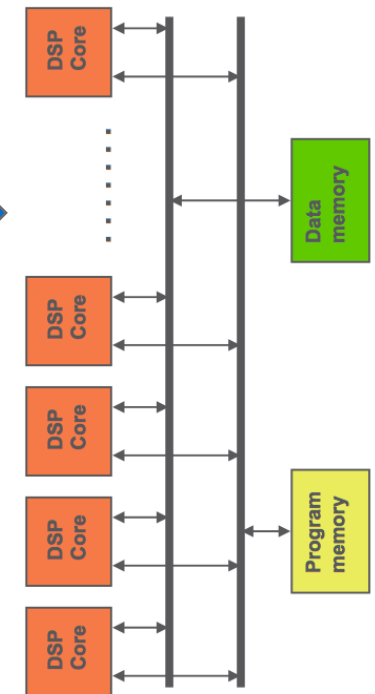
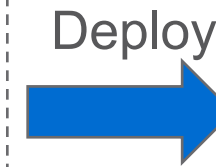
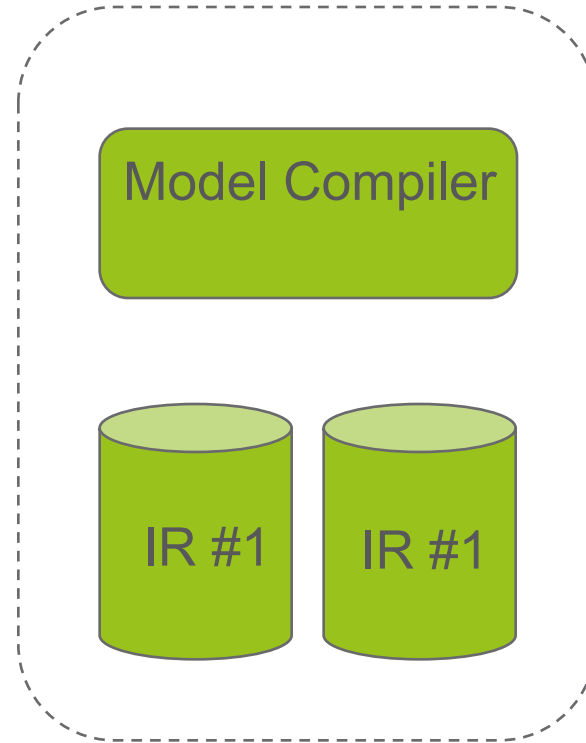
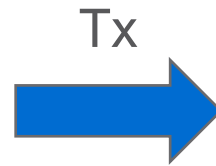
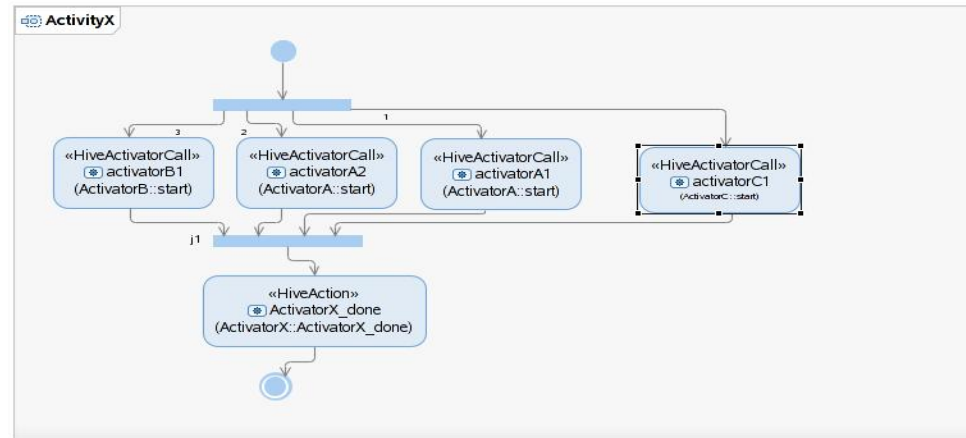
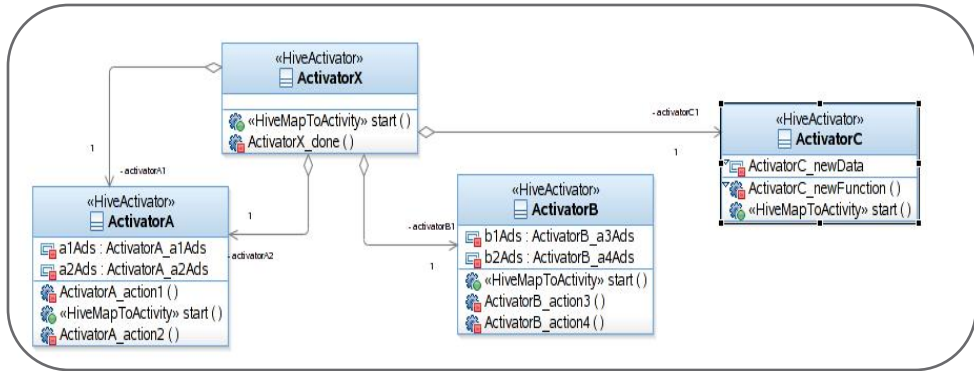
LANGUAGE ENGINEERING FOR EVERYONE!

Xtext is a framework for development of programming languages and domain-specific languages. With Xtext you define your language using a powerful grammar language. As a result you get a full infrastructure, including **parser**, **linker**, **typechecker**, **compiler** as well as editing support for **Eclipse**, **IntelliJ IDEA** and your favorite **web browser**.

[Learn more...](#)

The screenshot shows the Eclipse IDE interface. On the left, a package explorer shows a project named 'my-home' with sub-packages 'HomeRules' and 'src-gen'. The main editor displays a DSL rule file with text like 'Device Window can be OPEN, SHUT', 'Rule: 'Close Window, when heating turned on!', and 'Rule: 'Switch off heating, when window gets opened!'. Below the text is a state machine diagram with states 'State OFF', 'ON - Heating OFF', 'ON - Heating ON', 'OFF - Heating OFF', 'OFF - Heating ON', 'OPEN - Window OPEN', and 'SHUT - Window SHUT'. The IDE also shows a console at the bottom with the message 'No console to display at this time.'

DSL : GENERATING AND DEPLOYING CODE FOR MASSIVELY MULTICORE SYSTEMS



- Extension of UML-RT for Structure
- Extension of Activities to describe parallelism
- Action code in C

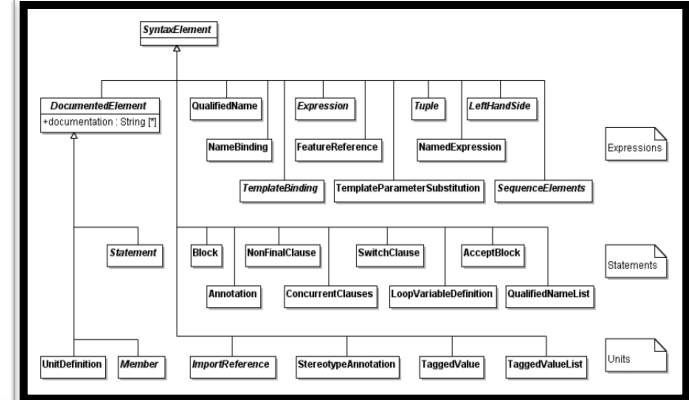
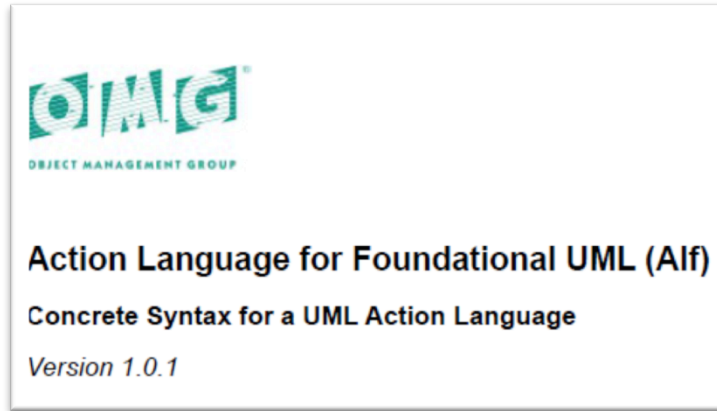
- Model compiler framework

- Deployment on many-core platform

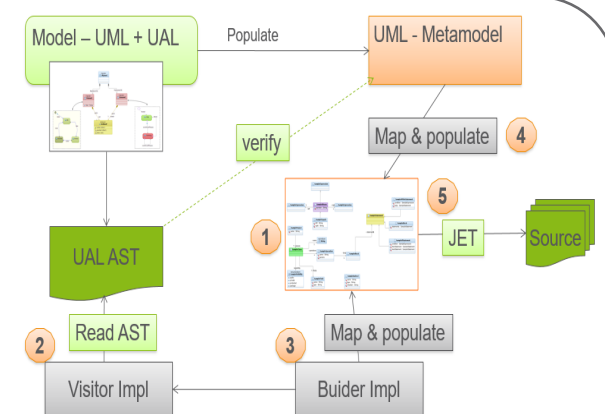
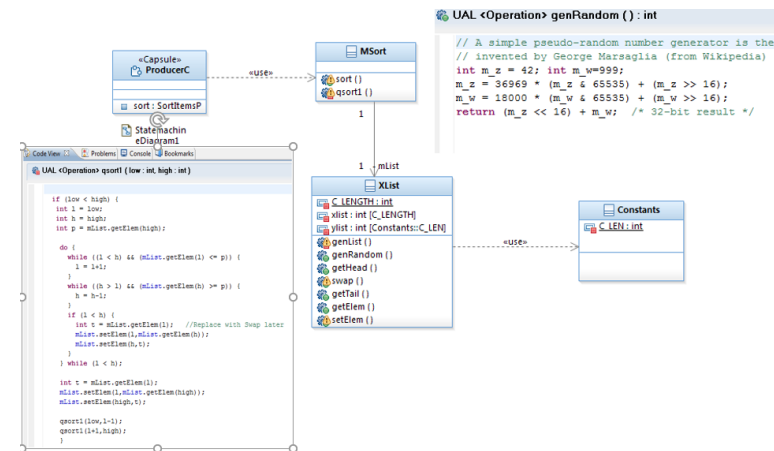
AND WE ATTEMPTED TO INTEGRATE AN ACTION LANGUAGE INTO UML



- We did like the idea of an action language as part of UML.
- Between 2007-2011 we worked on standardising ALF.



- We worked with IBM to develop an implementation of ALF called UAL
- Part of RSA toolkit.



DSL'S - WHAT DID WE LEARN?



Streaming-lining UML makes sense

Actually easier to get buy-in for DSL variants of UML.

Open source and Eclipse ecosystem has fantastic support for DSL creation.

“Crappy” open source tools that we can “look inside” are more useful than locked proprietary tools.



ALF action language was not really a success. Not enough of interested parties.

Debugging DSL generated systems is not trivial.

The diversity of options in Eclipse was a headache.



Agile and Open Source

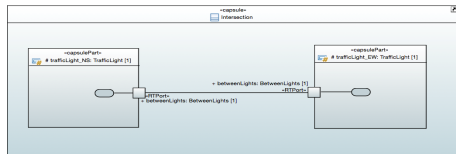
PAPYRUS AND POLARSYS



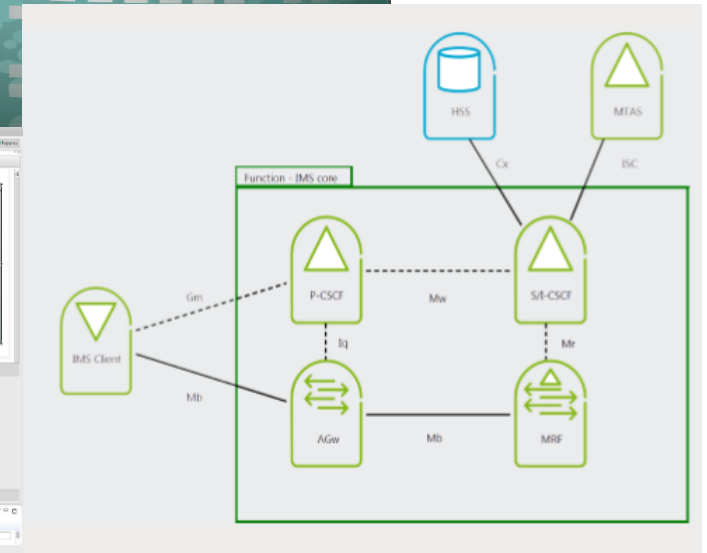
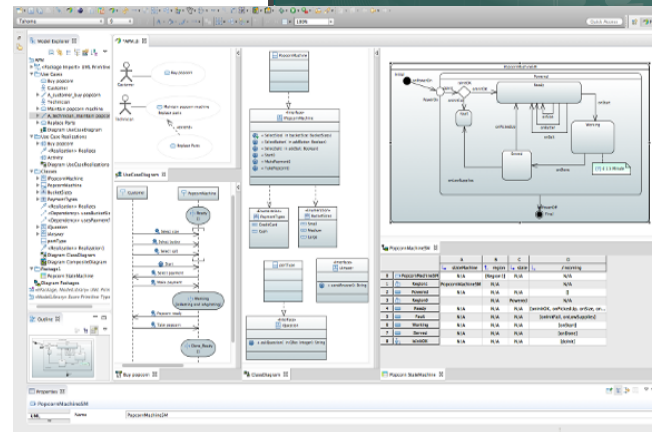
Ericsson has been investing in Papyrus ecosystem.

Internally we have started to use to use Papyrus for many projects.

Papyrus-RT:



Long Term: We have high hope for Papyrus eventually replacing proprietary tools.



IN PASSING : EFFECT OF AGILE WAYS OF WORKING

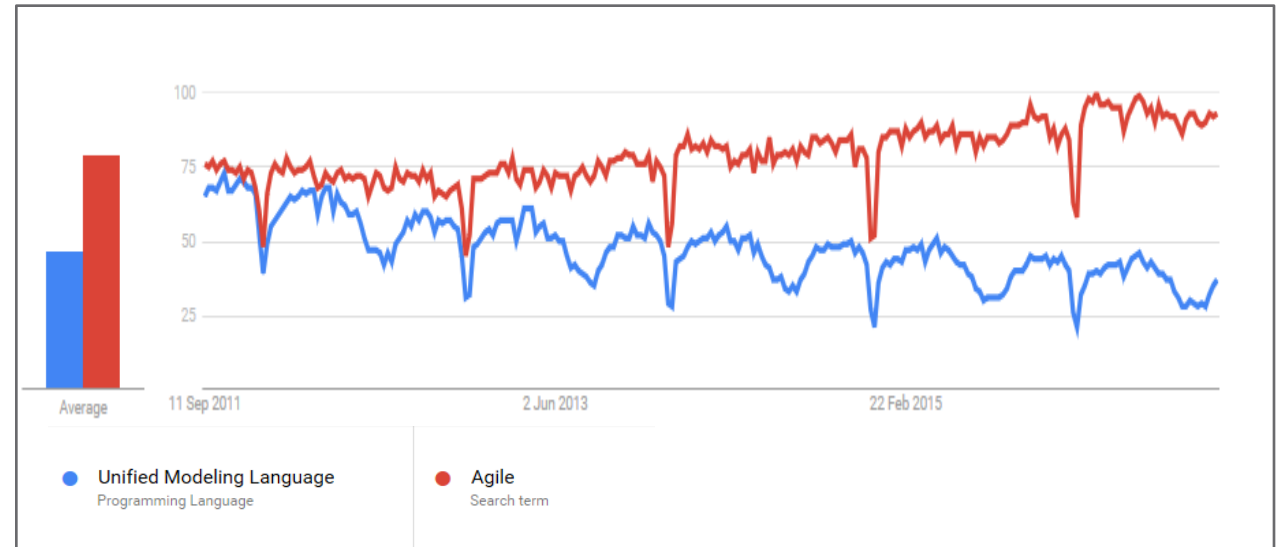


2012+ Ericsson started investing heavily in an agile transformation for R&D.

Driven by marked feature velocity.

Push towards very light-weight system and implementation tools.

In general UML centric MDD “in the large scale” start to be questioned after this.



OPEN SOURCE- WHAT HAVE WE LEARNT SO FAR?

- It's all about community involvement! IF you are not active in the community you can't steer technology.
- Convincing Ericsson policy makers about contribution to open source community is not trivial.
- Currently it is very clear that we will promote open source tools as preferred.
- Fact : 80% (+) of our software is currently open source.





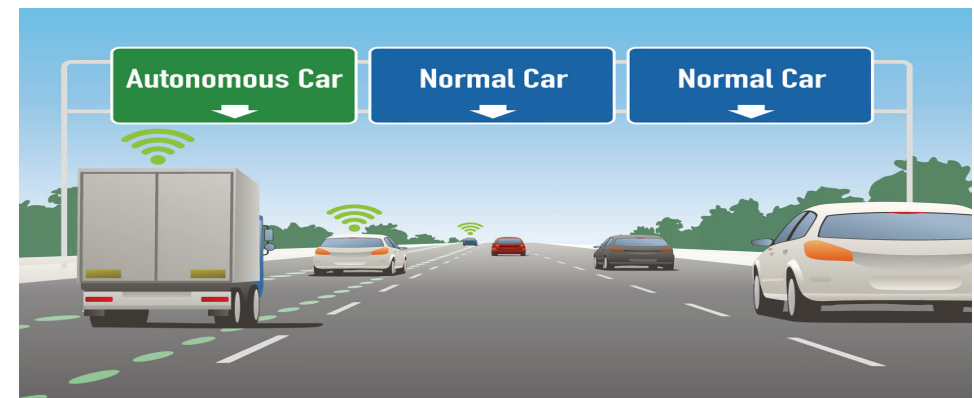
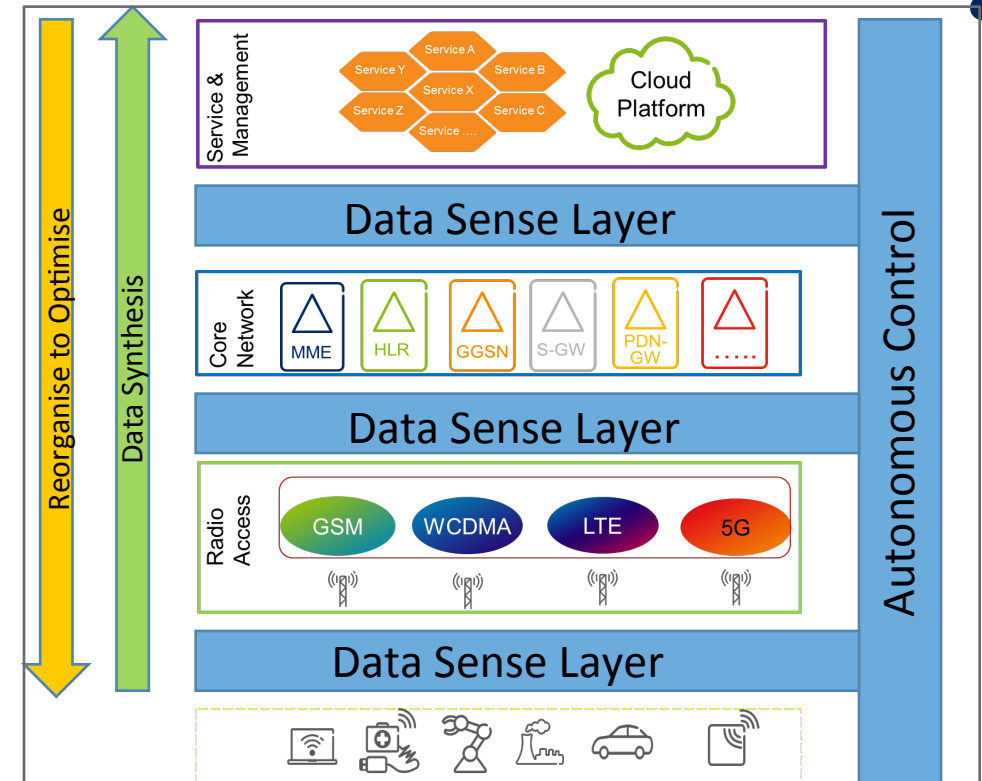
New Opportunities in MDD

CHALLENGE #1 : AUTONOMOUS SYSTEMS



- Autonomous communication systems
- Deriving optimal operation based on massive data awareness.
- **Challenge: Modelling these system. Understanding emergent behaviour.**

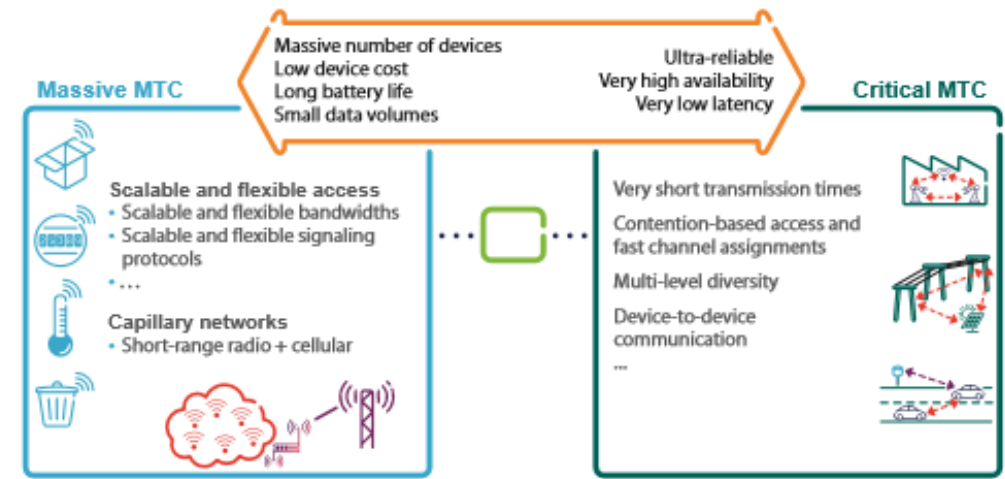
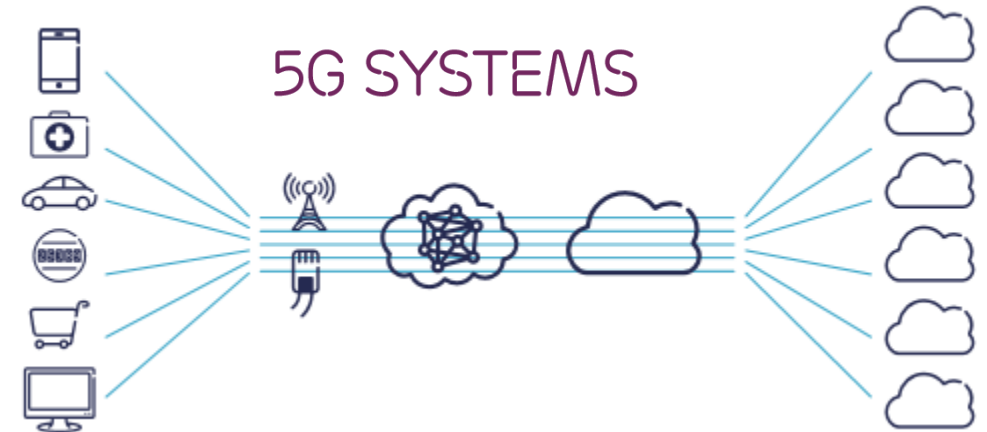
- Autonomous vehicles. Currently a very active research area.
- **Challenge: Modelling system stability and security.**



CHALLENGE #2 : MISSION CRITICAL RAN SYSTEMS



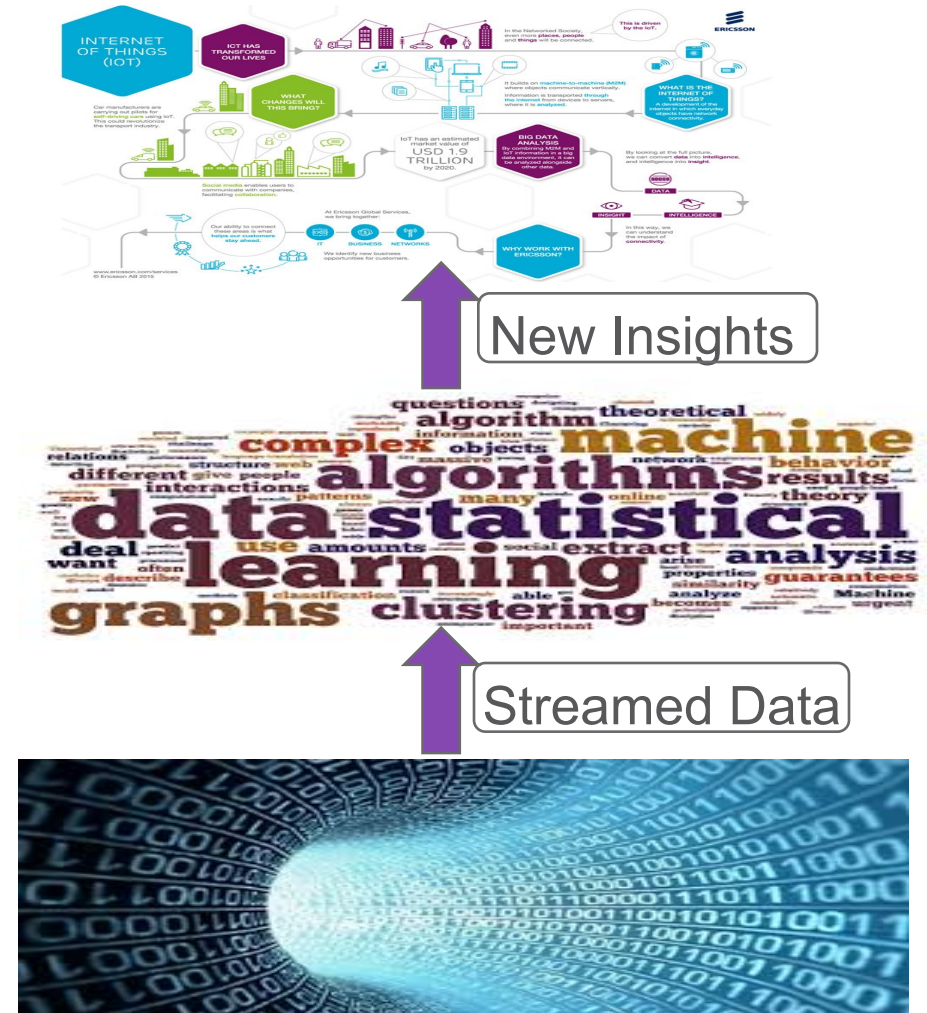
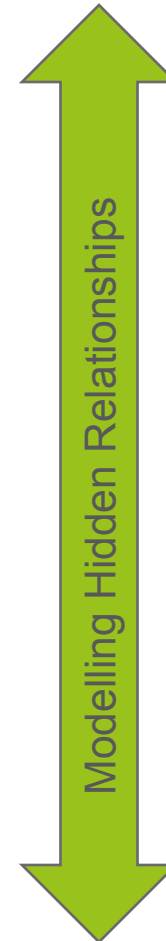
- 5G radio access will support mission critical systems.
- This will require new system modelling techniques
- **Challenge: How can we model mission critical system over an inherently unreliable medium**



CHALLENGE #3 : MACHINE LEARNING AND DATA DRIVEN ARCHITECTURES



- Much focus on new data driven architectures.
- New machine learning technologies make unravelling complex relationships in data feasible.
- This leads to new insights and possibilities.
- **Challenge: How can we use model driven approaches to empower data driven architectures.**



Q&A





ERICSSON