Project #4:
Knowledge gain from Simulations and Experiments

Ulf Eliasson and Rogardt Heldal
Challenge: Mechatronic Systems
Lack of knowledge
Experimenting to gain Knowledge

Two case studies

Plant models

Learning

Make better decisions
Preliminary Results:
... Preliminary results

- Quick feedback
- Aware of holes in knowledge
- Allowed for exploring solutions

- However, faulty assumptions on plant models caused issues in the software.
... Preliminary results:

Constraints

• Modeling language has to be simple to use
• Support within the organization
• Executable
• Good code generation
• Good test environment
Success so far ...

Conclusions:

• Agile: early learning, less assumptions
• But, wrong plant models can be costly

Next Sprint:
– How to be able to integrate ECUs continuously.

Grants:
– ASSUME (5.2 Msek,Vinnova)
– xEVCO2 (To be submitted)
– xxxxWe, Horizon 2020

Papers (8):

Håkan Burden, Rogardt Heldal and Jon Whittle. Comparing and Contrasting Model-Driven Engineering at Three Large Companies. Submitted to ICSE 2014 Industrial Track, May 2014


Assumption Management Framework for Automotive Embedded Software Development. Vinnova/FFI project 2013-02646


Adry Ferwerda, Kenneth Lind, Niklas Mellegård, Rogardt Heldal, and Michel Chaudron, On the Introduction of MD*/DSL in Software Maintenance: An Industrial Case Study. The paper is more or less ready and will be submitted to IEEE Trans. Softw. Eng.
We aim for automating further parts of the testing and reach continuous integration and deployment. Both in a model-based virtual test environment but also for hardware-in-the-loop testing and finally to deploy the new software to prototype vehicles.
Software Development Tools to Support Continuous Deployment

Rogardt Heldal
Tools are needed

Complex software
Many people

Requirements
Testing
Architecture...

Diagram showing complex software architecture with various components and connections.
Good tools

• Easy to use
• Good support for the task at hand
• Syntax support
  – Less to remember
• Validation support
  – Building the correct software
• Automation
  – Less work
Hypothesis

• Our hypothesis is: **software development tools made by the companies themselves are simpler to use, more flexible, and easier to change**

• ... or built on open source

• ... and one avoids vendor lock-in
Strategies

• Investigate three different strategies:
  – make your own tools (in-house),
  – tools based on open source,
  – and use of commercial off the shelf-tools.

• What conditions decide what strategy to use?
• What factors are important for success?
Moreover (Future Sprint)

• How important is automation?

• Comments:
  – some researchers claim that Lean processes make companies more productive, while some other argue that the level of automation plays a key role
Continuous deployment

- Continuous deployment:
  - Require continuous modification of secondary software such as tools
    - Avoid delay due to tools
    - Avoid technical depth in tools
    - Avoid work around in code due to tools
      - Technical depth in code