Requirements Engineering for Control Systems

Dominik Schmitz*, Hans W. Nissen, Matthias Jarke, Thomas Rose

RWTH Aachen University,
Cologne University of Applied Sciences,
Fraunhofer FIT
ZAMOMO Project

• „Integration of model-based software and model-based control systems engineering“

• Innovations in cars nowadays mainly driven by software, but control system engineering and software engineering currently not interacting ⇒ methodological supplement is hindered

• Application domain: combustion engine controller
ZAMOMO Project

- Integration of model-based software and model-based control systems engineering
- Innovations in cars nowadays mainly driven by software, but control system engineering and software engineering currently not interacting ⇒ methodological supplement is hindered

Application domain: combustion engine controller

Characteristics
- Interdisciplinary
- Importance of hardware, in particular sensors and actuators
- Flexible, innovative, customer-oriented small- & medium enterprises
- RE as part of offer development (timing, costs, reuse decisions, …)
- Project-driven work, frequent innovations
**i* based RE**

- control systems requirements
- software requirements
- common problem understanding with customer
- model-based requirements capture
- transition to (mathematical) design
- traceability, configuration management
- common comprehensible representation

- tool and analysis support
- model-based requirements capture
- transition to (mathematical) design
- traceability, configuration management

**Transition to (Mathematical) Design**

- Pot 
- Variables
- Requirements
- Interfaces
- Implementation
- Design

**Traceability**

- Agent
- Goal

**Common Comprehensible Representation**

- Agent
- Goal

---

June 7-8th 2010

iStar 2010

3/12
**i* based RE**

**i* based capture of control requirements**

- Model-based
- Functional and non-functional aspects
- Combined, inter-disciplinary investigation
- Capture any (kind of) stakeholder
- Sensors and actuators as resource dependencies

(control systems requirements)

(software requirements)
Domain Model-based RE
Domain Model-based RE

Features

- Common starting point
- Accelerates modeling
  - Eliminate non-applicable elements
  - Add project-specific elements
- Tailoring, update possible
Challenges during Offer Development

Offer development

- Model requirements from SMEs point of view
- Senior engineer manually selects similar projects, inspects, and includes reusable artifacts
- Prepare cost calculation

Two dangers

- Reusable artifacts not found (too many projects, too less time) ⇒ offer too expensive
- Artifacts actually not reusable ⇒ project loss
Challenges during Offer Development

Offer development

- Model requirements from SMEs point of view
  - Support engineer in identifying similar projects
    - Similar = requirements match
  - Reduce and focus number of relevant projects to be inspected in-depth

- Reusable artifacts not found (too many projects, too less time) ⇒ offer too expensive
- Artifacts actually not reusable ⇒ project loss
### Similarity Search

**Pre-defined queries (currently 11)** referring to the domain model

**Ad-hoc, user-defined queries**

---

**Query weights, sum up to 1**

**Earlier projects stored in the database**

**Objects from current project that occur also in the earlier project**

**Overall ranking**

---

#### Table: Comparing Current Project with Projects in Database

<table>
<thead>
<tr>
<th>Project</th>
<th>Customer req.</th>
<th>Combustion engine block</th>
<th>Cylinder positioning</th>
<th>No. of cylinders</th>
<th>Fuel</th>
<th>Overall similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>50%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>15%</td>
</tr>
<tr>
<td>Project 2</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
<td>100%</td>
<td>95%</td>
</tr>
<tr>
<td>Project 4</td>
<td>75%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>93%</td>
</tr>
</tbody>
</table>

Weights:

<table>
<thead>
<tr>
<th>Weights</th>
<th>0.3</th>
<th>0.1</th>
<th>0.2</th>
<th>0.1</th>
<th>...</th>
<th>0.3</th>
</tr>
</thead>
</table>

---

**Diagram:**

- Controlled system: combustion engine block
- Carburettor
- Boxer
- Rod
- Cylinder positioning
- Common rail
Domain Model Evolution

- Adjust domain model to reasonable size and to reflect new innovations

- Problems for model comparison ⇒ push domain model changes to finalized projects

June 7-8th 2010
Domain Model Evolution (2)

- Suggestions for reductions, extensions from usage in previous projects
- Computation of extension in 5 steps
  - Identify project-specific extensions
  - Compute similarity between them (shared anchor objects in domain model)
  - Group similar extensions
  - Inspection and implementation by engineer
  - Store results
Domain Model Evolution (2)

- Suggestions for reductions, extensions from usage in previous projects

Computation of extension in 5 steps:
- Identify project-specific extensions
- Compute similarity between them (shared anchor objects in domain model)
- Group similar extensions
- Inspection and implementation by engineer
- Store results

Extension 1+2

Extension 1
- controller: electronic control unit
- sensor: cylinder pressure
- controlled system: combustion engine block

Extension 2
- measure cylinder pressure
- control cycle

Extension 3
- actuator: waste gate
- turbine
- biturbo

Refinement path
- cylinder positioning
- stroke
- no. of cylinders
- cooling
- oil
- fuel
- gasoline
- pump injection
- direct injection
- gasoline
Transformation to Simulink

1. Take design decisions
   - Manual Support for checking readiness
   - Refined model (PIM)

2. Derive Matlab/Simulink skeleton
   - Matlab/Simulink model (PSM – Matlab)
   - Matlab/Simulink model (PIM – RCP)

3. Incorporate hardware details
   - Interactive Add RCP platform specific libraries
   - Specific libraries considered (PSM – RCP)

Implementation Details

- ConceptBase
  - i* module
  - i* framework
  - i* models
  - Model-to-model
    - query i* model, reformat results via answer format
  - Simulink module
    - Simulink framework
    - Simulink models
  - Model-to-text
    - XML export SimEx-Tool by IT Power Consult

June 7-8th 2010
iStar 2010
Summary

interdisciplinary methodology

consider non-functional requirements

equal treatment of

continuously model-based

domain- & project-oriented reuse

traceability, configuration management

June 7-8th 2010

iStar 2010
Future Work

- Enhance domain model evolution support, e.g. by additional heuristics
- Simulation at RE level
  - against abstract model of controlled system?
  - against detailed Matlab/Simulink model?
- Investigate sociality of $i^*$ actors in control system setting
- Apply domain model based approach to other fields
• **i* Wiki** – [http://istar.rwth-aachen.de](http://istar.rwth-aachen.de)
  – Quick guide
  – Tool comparison
  – Growing community
  – *i* workshops
  – *i* news, e.g. on iStarML, upcoming events, project ideas, …
ZAMOMO – Publications


