Achieving Separation of Roles and Separation of Concerns in Robotics Software by Model-Driven Software Development

euRobotics Forum 2012, Odense, Denmark

Alex Lotz, Andreas Steck and Christian Schlegel

Computer Science Department
University of Applied Sciences Ulm, Germany

http://www.hs-ulm.de/lotz
http://smart-robotics.sourceforge.net/
http://www.zafh-servicerobotik.de/
What is the Challenge in Robotics?

- The current situation in software for robotics can be compared with the early times of the *World Wide Web* where one had to be a computer engineer to setup web pages.

- The *World Wide Web* turned into a universal medium only since the availability of tools
  - which have made it accessible to everyone
  - which allow domain experts (like journalists) to provide content without bothering with technical details
  - which ensure sustainability / availability of contents independently of preferred operating systems, browsers etc.

=> *separation of roles and separation of concerns*

=> *this is a universal approach towards successfully handling complexity: applications, markets, sharing efforts / risks*
What is the Challenge in Robotics?

- The current situation in software for robotics can be compared with the early times of the *World Wide Web* where one had to be a computer engineer to setup web pages.

- The *World Wide Web* turned into a universal medium only since the availability of tools
  - which have made it accessible to everyone
  - which allow domain experts (like journalists) to provide content without bothering with technical details
  - which ensure sustainability / availability of contents independently of preferred operating systems, browsers etc.

=> *separation of roles and separation of concerns*

=> *this is a universal approach towards successfully handling complexity: applications, markets, sharing efforts / risks*

**separation of concerns**

=> *e.g. model-based approaches like MDSD to explicate structures / properties*

=> *e.g. DSLs to allow non-roboticists to use robotics technology*
Separation of Roles
What is the problem?

Component Builder
Systems Integrator

? 

Source Code
Non-Functional Properties (NFP)
Parameter
QoS
Resource Information
Ports (Interfaces)

Relevant information is hidden in the source files.

→ source code has to be analysed

The robot has no access to that (hidden) information to reason on it at run-time.

Robot
Separation of Roles
Separation of Concerns

"freedom from choice“ in order to ensure system-level conformity
Separation of Roles
Separation of Concerns

Component Builder

Object recognition

Component Builder

Navigation

Base

Navigation

Speech

...
Separation of Roles
Separation of Concerns

Component Builder

Object recognition

Component Builder

Navigation

System Integrator

Base navigation

Speech

System Integrator

Black-box view

Make system-level bindings and adjustments
Separation of Roles
Separation of Concerns

Component Builder
- Object recognition
- Navigation

Component Builder
- Speech

Base Navigation

System Integrator

End User

System Integrator

Hochschule Ulm
Separation of Roles

How it works? → Use Component Models!

Component Shelf → Component Builder → System Integrator

Parameters, Properties, Ports (Interfaces), Resource Information, ...

ARE EXPLICATED IN THE MODELS
AND ARE NOT HIDDEN IN THE SOURCE FILES

Name: planner
Ports:
  mapClient : pushNewestClient<gridmap>
  goalServer : pushNewestServer<goal>
  plannerEvent : eventServer<goalStatus>
  state : stateServer
  param : parameterServer
...
States:
  active, neutral, ...

March 7th, 2012
euRobotics Forum 2012 / Lotz, Steck, Schlegel
Separation of Roles

Bridge between Design-time- and Runtime-Models

Models allow to ...

• explicate information
• provide a black-box view
• provide a hull that supports to switch the level of abstraction (e.g. from Component Builder View to System Integrator View)
• use MDSD tools to run automated model checks

As models are computational, also the robot can access the explicated information.

=> The robot can take this information into account to make decisions at run-time.
Illustration of the Development Process

- Implemented as UML 2.0-Profile for Robotics Software Components
- supports Component Development, System Integration, Deployment
- based on standards: UML 2.0, Papyrus, Eclipse Modeling Project, etc.
- different Runtime-Platforms, Middleware-Systems etc.

2-step transformation workflow (framework builder view)
The SmartSoft Component Model
Mapping to different Middlewares

- meta-information for user-code
  - e.g. platform constraints
  - runs only on QNX
  - runs only on RTAI-Linux

stable interface towards user-code
stable interface towards other components
internal implementation of the component hull is generated depending on the parameters
The SmartSoft Communication Patterns
- send: one-way communication
- query: two-way request/response
- push newest: 1-to-n distribution
- push timed: 1-to-n distribution
- event: asynchronous conditioned notification

The SmartSoft Services
- param: component configuration
- state: activate/deactivate component services
- wiring: dynamic component wiring
- diagnose: introspection of components

(internally based on communication patterns)
Model-Driven Approach
Component Builder View

PIM Graphical Representation

PIM Files
PSI Files
PIM outline
Attributes / Tagged Values
Model-Driven Approach

Screencast “Simple Navigation”

http://youtu.be/04SqzrS6Udw
Model-Driven Approach
System Integrator View

Deployment Model
Imported Components
Deployment Properties
Graphical Representation of Deployment Model
Palette
Button
Model-Driven Approach

System Integrator View – Buttler Scenario

Task Coordination

Human Robot Interaction

Localization

Learning from Demonstration

Mobile Manipulation

Navigation

Task coordination components connect to state and parameter servers of all components (not displayed) and to all ports that appear disconnected in this diagram.
Model-Driven Approach

System Integrator View – Buttler Scenario
Further References

Recent book chapters (open access PDFs)


http://smart-robotics.sourceforge.net/

http://www.youtube.com/user/RoboticsAtHsUlm
Selected publications for Model usage at Run-Time

  Get as PDF

- Andreas Steck, Alex Lotz and Christian Schlegel, "Model-Driven Engineering and Run-Time Model-Usage in Service Robotics", in Proc of the 10th ACM international conference on Generative programming and component engineering (GPCE '11), Portland, Oregon, USA, October 2011
  Get as PDF

- Andreas Steck and Christian Schlegel, "Towards Quality of Service and Resource Aware Robotic Systems through Model-Driven Software Development", in Proc. 1st International Workshop on Domain-Specific Languages and models for ROBotic systems (DSLRob/IROS), Taipei, Taiwan, October 2010
  Get as PDF