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

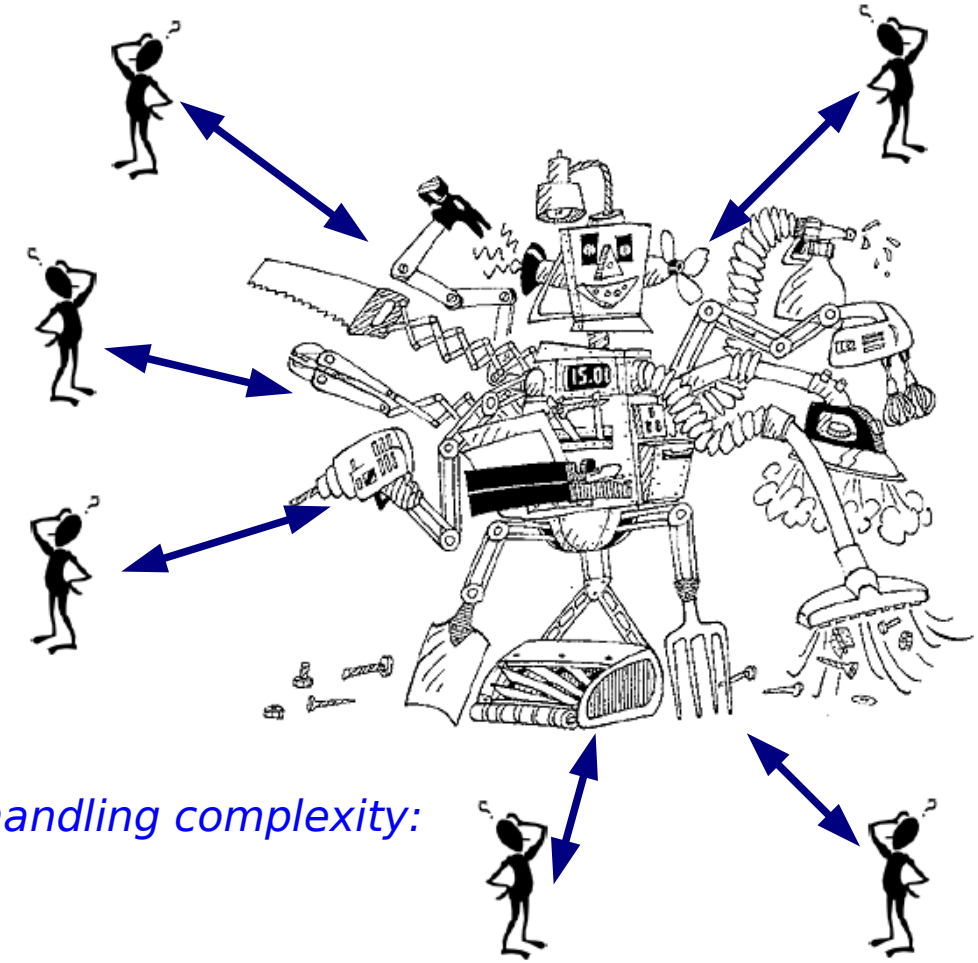
http://www.hs-ulm.de/lotz
http://smart-robotics.sourceforge.net/
http://www.zafh-servicerobotik.de/

Hochschule Ulm

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.



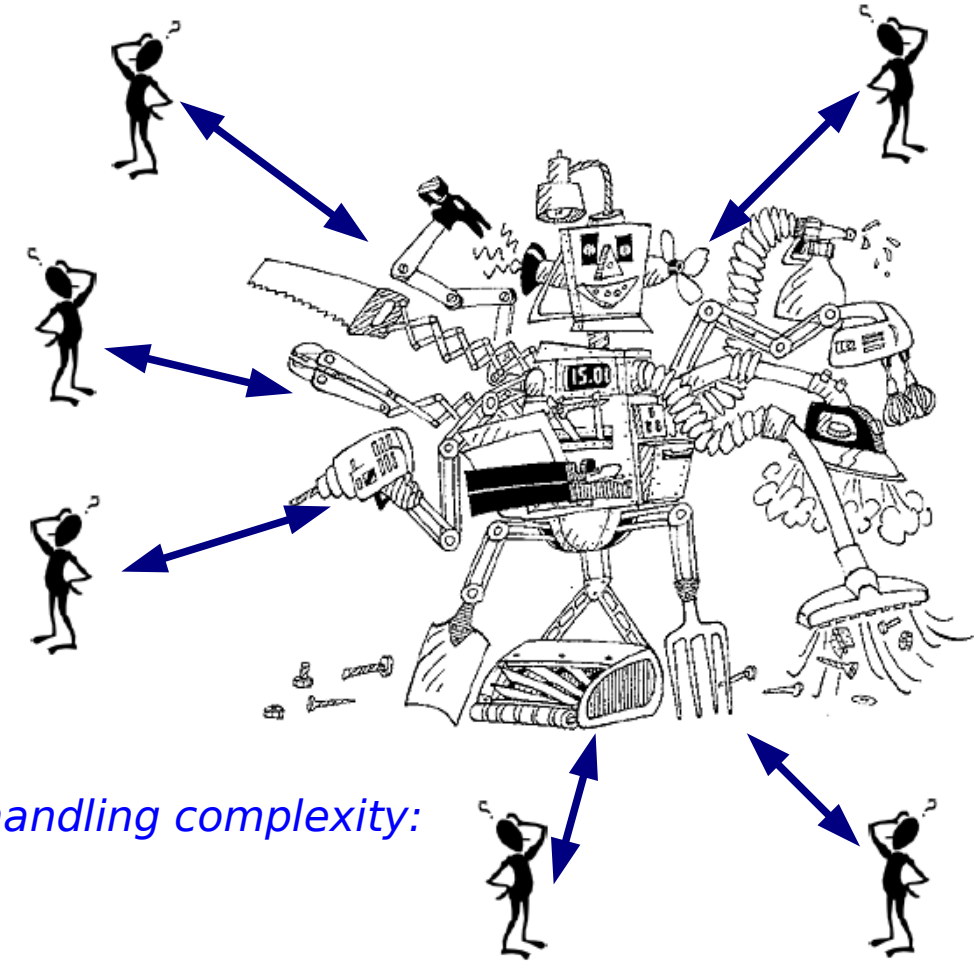
=> separation of roles and separation of concerns
=> this is a universal approach towards successfully handling complexity: applications, markets, sharing efforts / risks

service robotik

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.



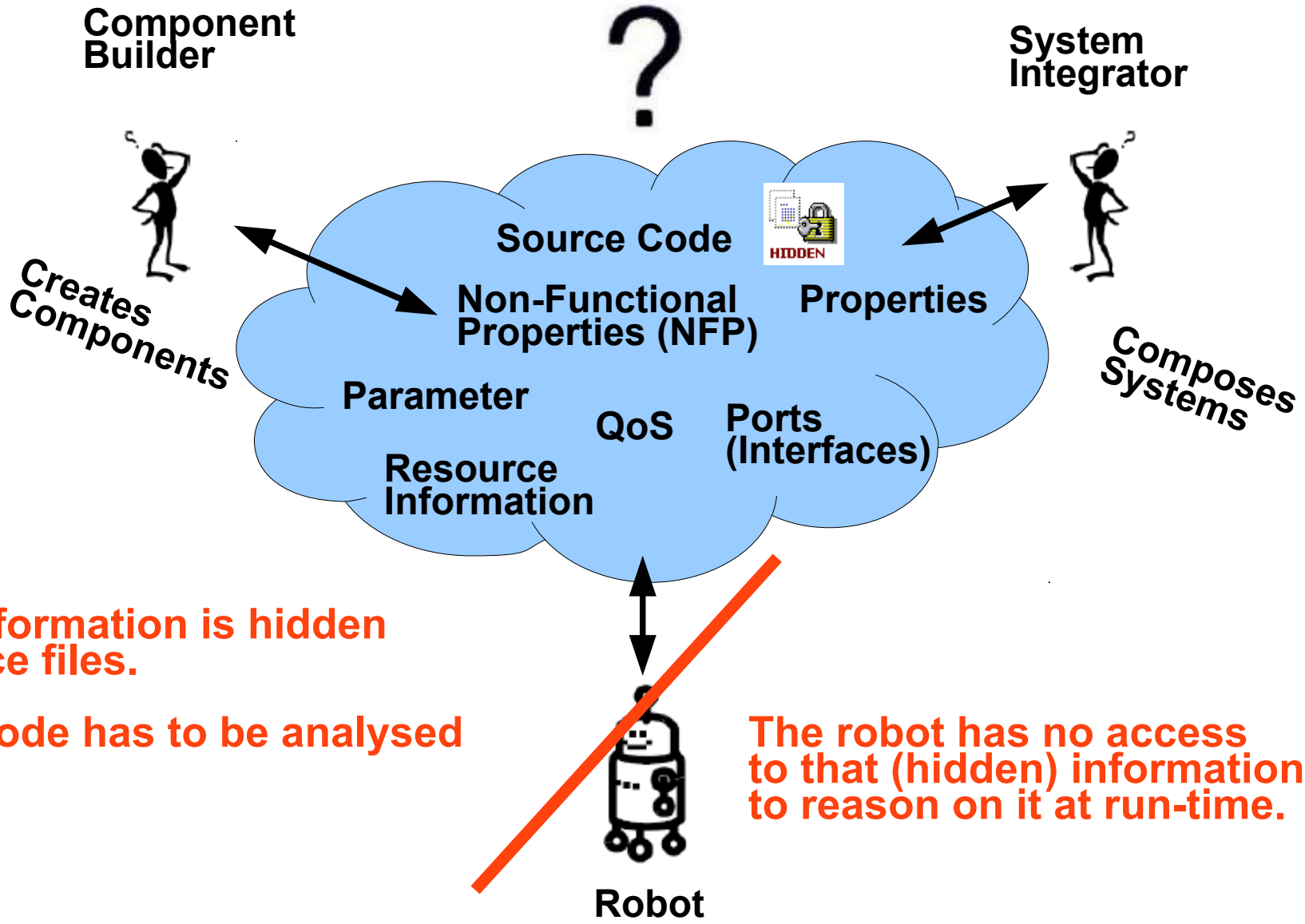
=> separation of roles and separation of concerns
=> this is a universal approach towards successfully handling complexity:
applications, markets, sharing efforts / risks

separation of concerns
=> e.g. model-based approaches like MDSD to explicate structures / properties
separation of roles
=> e.g. DSLs to allow non-roboticists to use robotics technology

Hochschule Ulm

Separation of Roles

What is the problem?

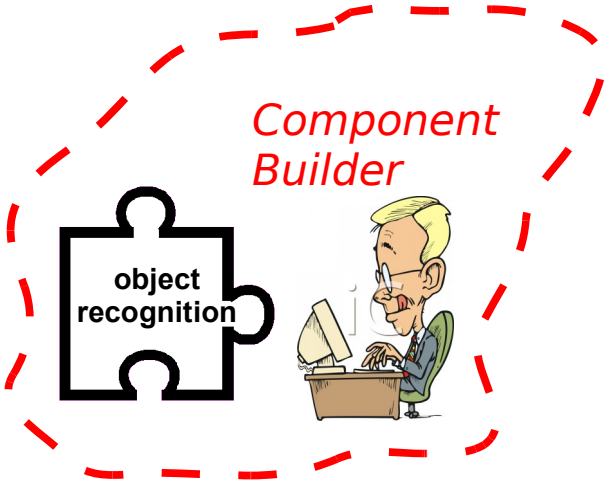


Relevant information is hidden in the source files.

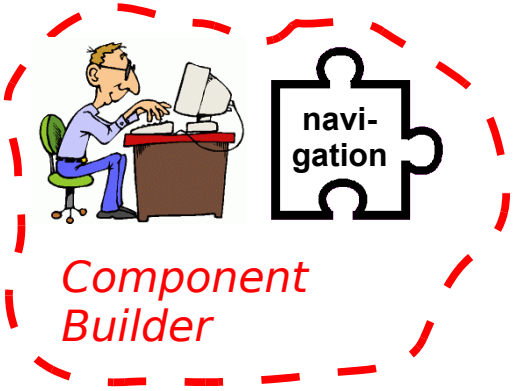
→ source code has to be analysed

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

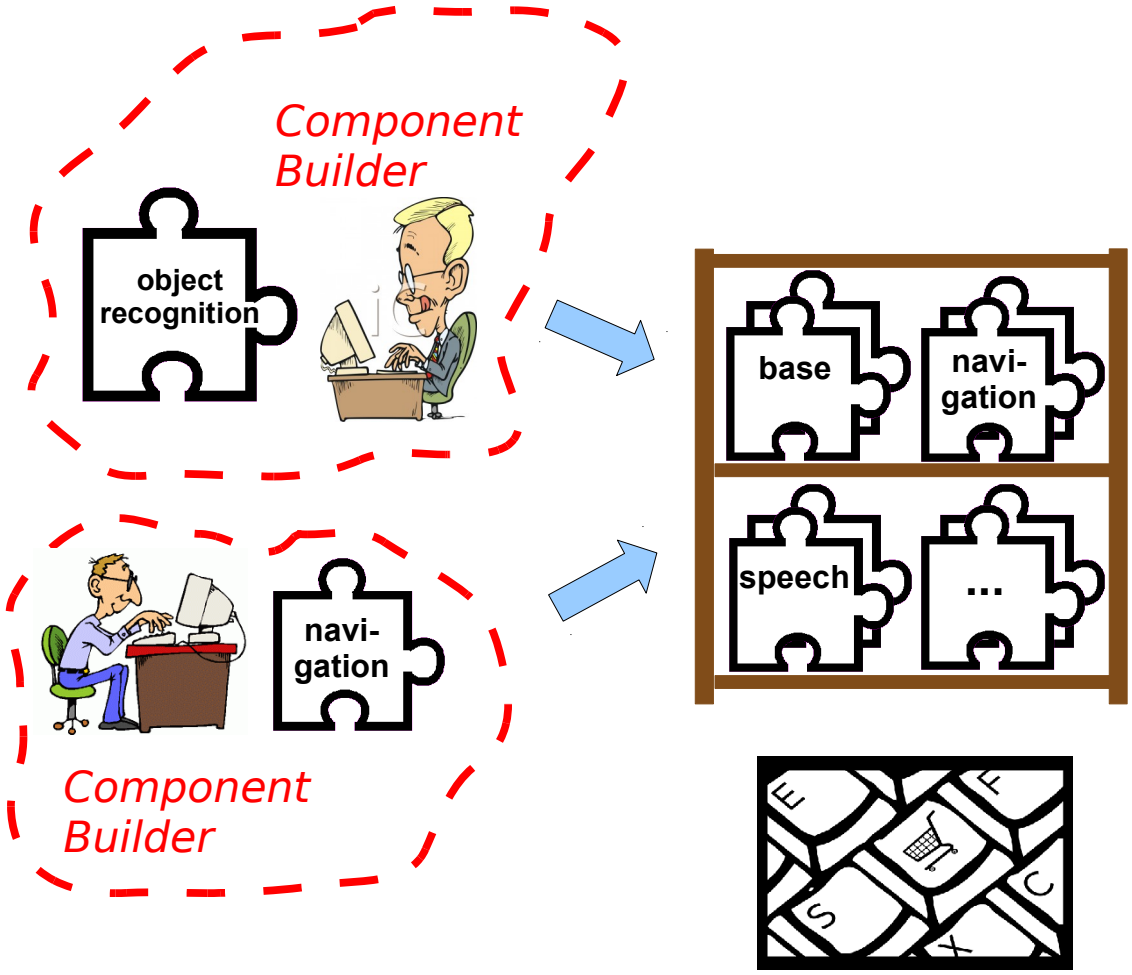
Separation of Roles
Separation of Concerns



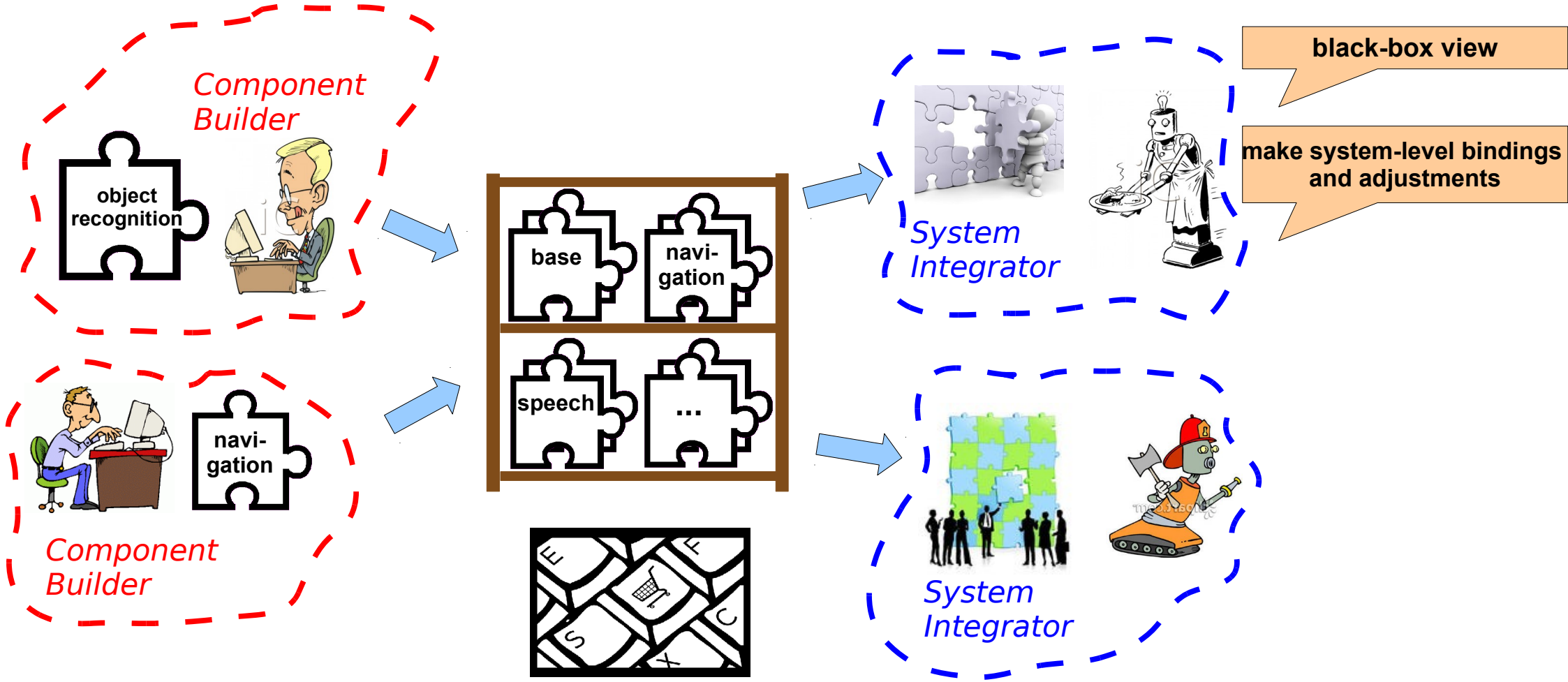
„freedom from choice“ in order to ensure system-level conformity



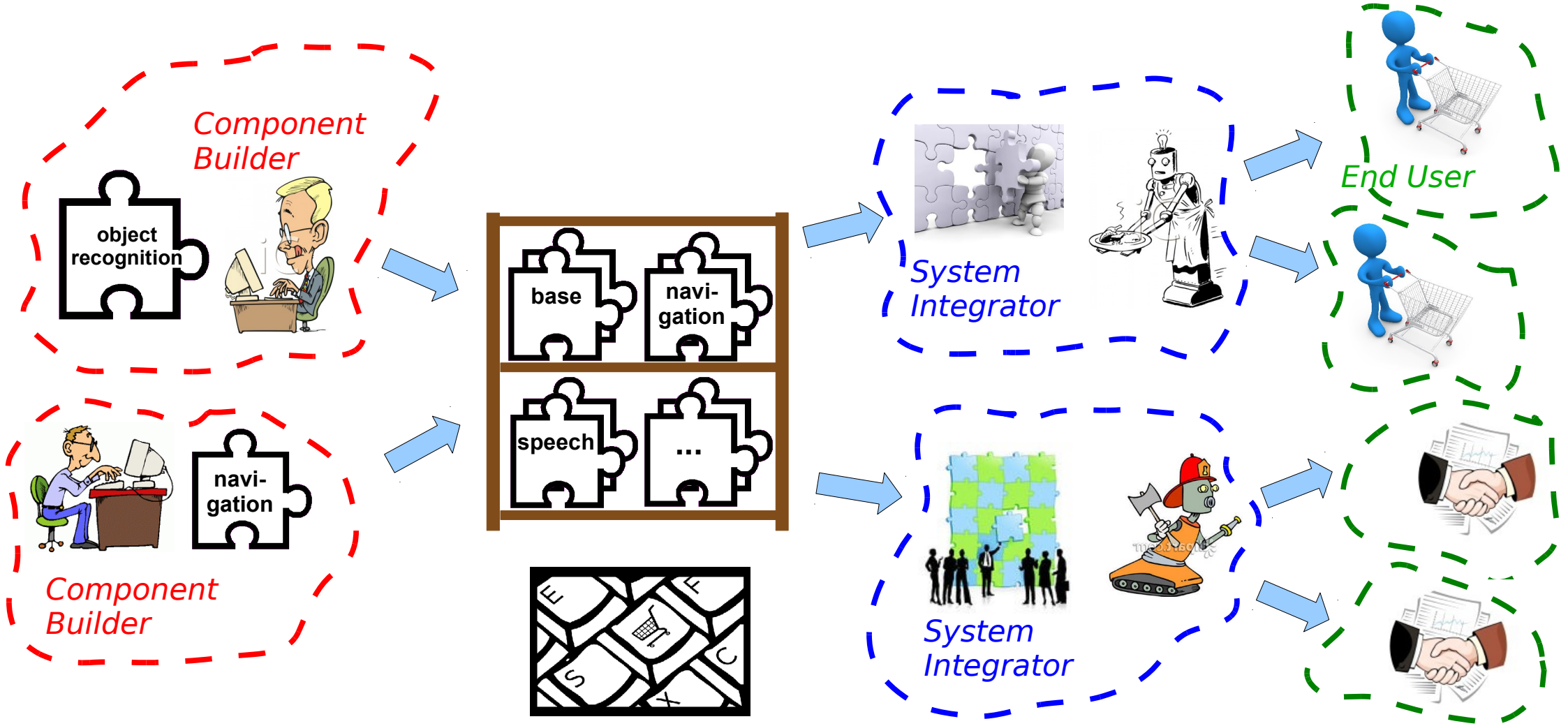
Separation of Roles
Separation of Concerns



Separation of Roles
Separation of Concerns

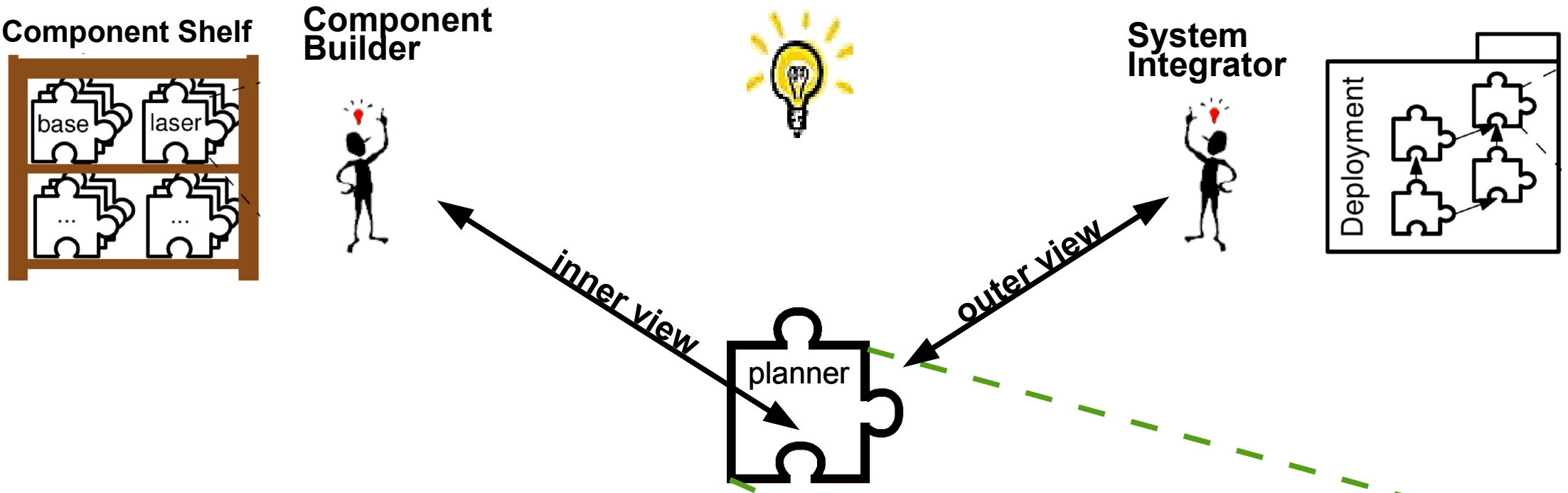


Separation of Roles
Separation of Concerns



Separation of Roles

How it works? → Use Component Models!



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

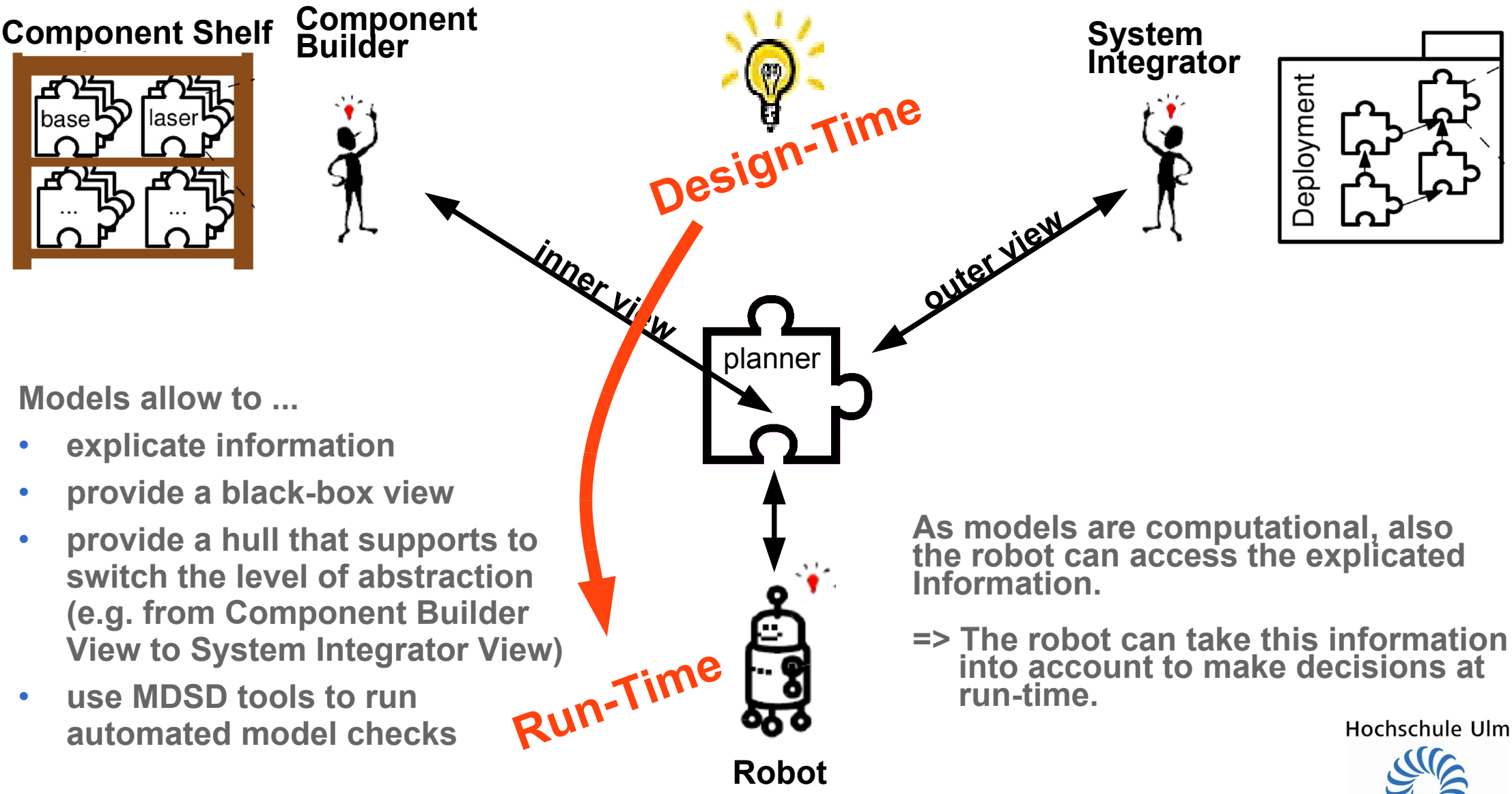
ARE EXPLICATED IN THE MODELS

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, ...
```

Separation of Roles

Bridge between Designtime- 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

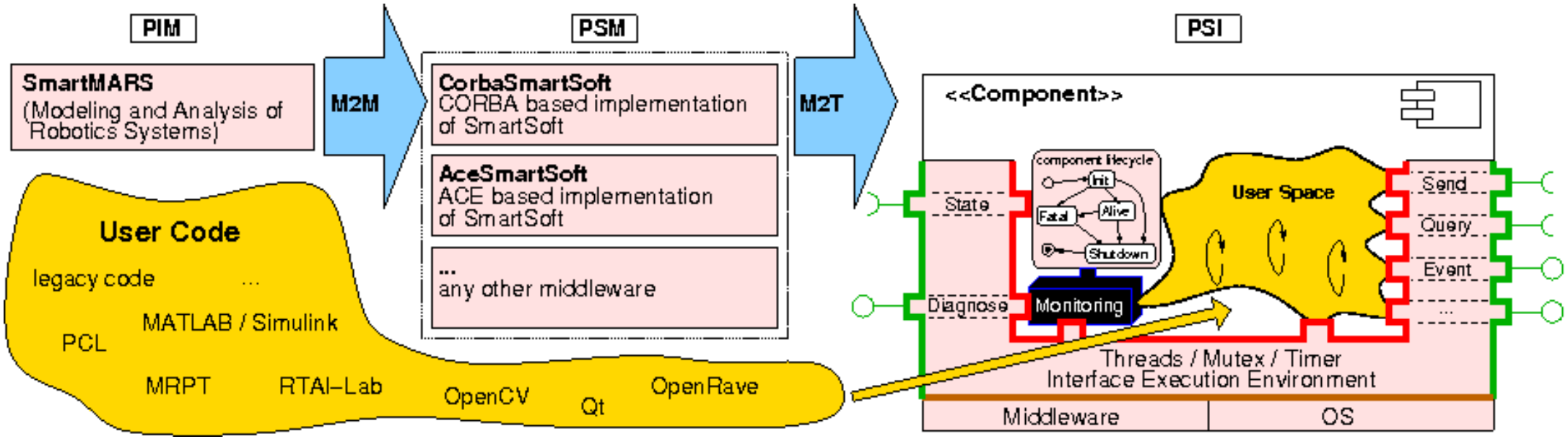
As models are computational, also the robot can access the explicated Information.

=> The robot can take this information into account to make decisions at run-time.

Model-Driven Approach
SmartMDSD

Illustration of the Development Process

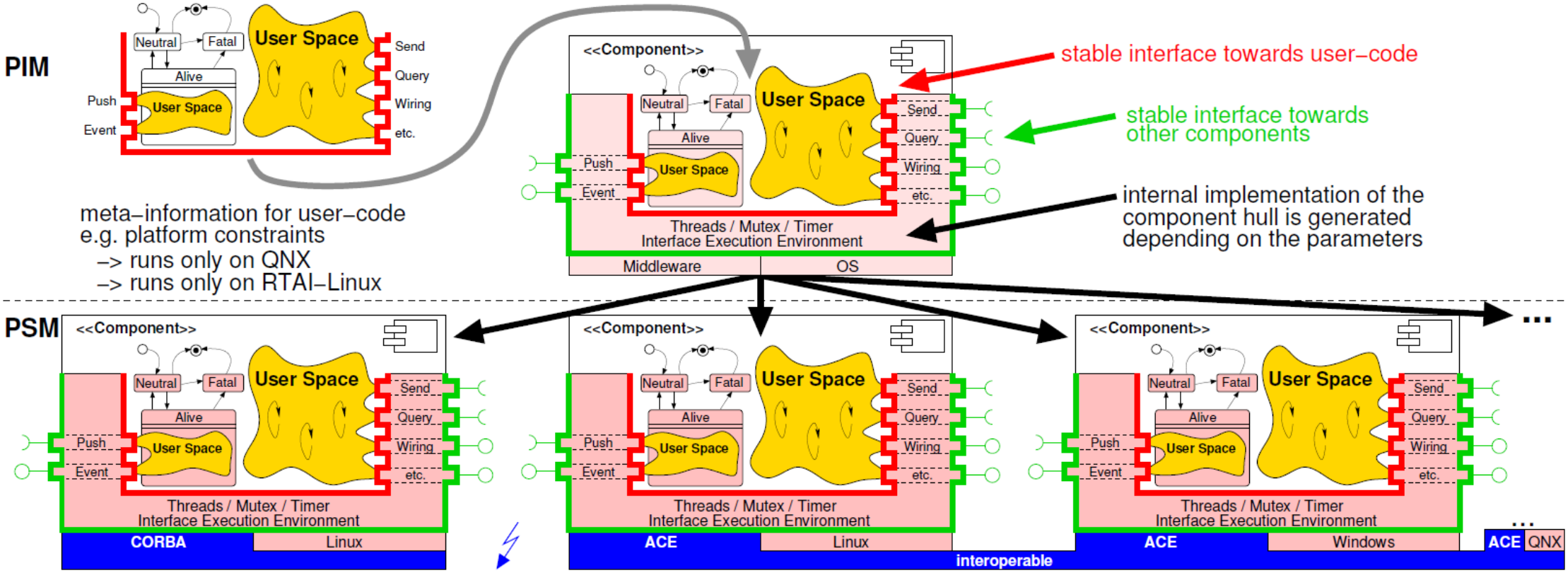
- Implemented as UML 2.0-Profile for Robotics Software Components
- supports Component Development, System Integration, Deployment
- based on standards: UML 2.0, Papyrus, Eclipe Modeling Project, etc.
- different Runtime-Platforms, Middleware-Systems etc.



2-step transformation workflow (framework builder view)

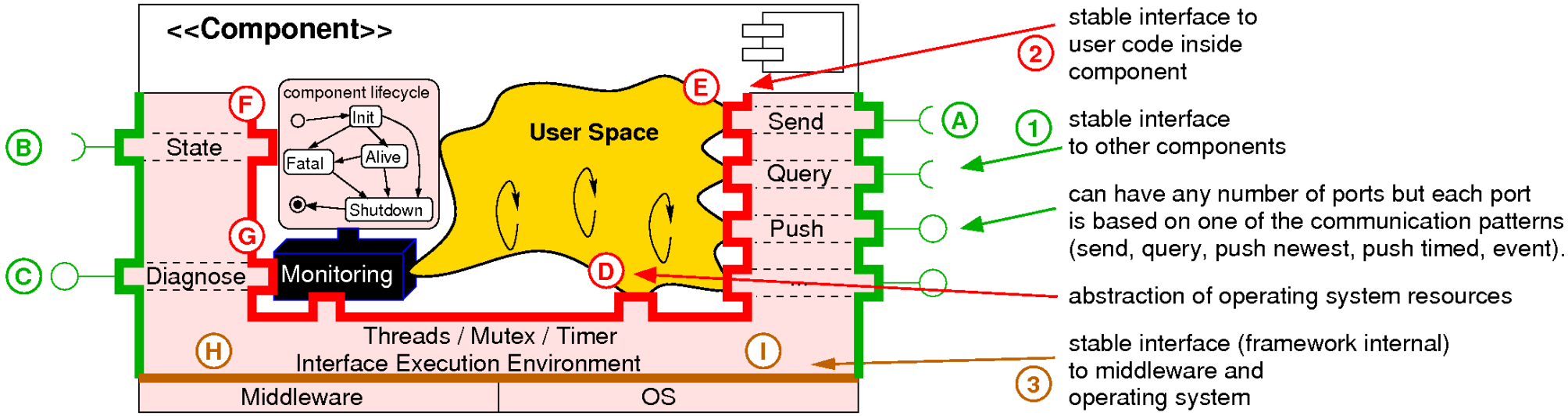
Hochschule Ulm

The SmartSoft Component Model
Mapping to different Middlewares



The SmartSoft Component Model

Decoupled Sphere of Influence



- *Services* are defined by a *Communication Pattern* and *Communication Objects*
- *Communication Objects* are communicated between components: platform-independent, by-value
- *Services* are offered / used by components via *Ports*

The SmartSoft Communication Patterns

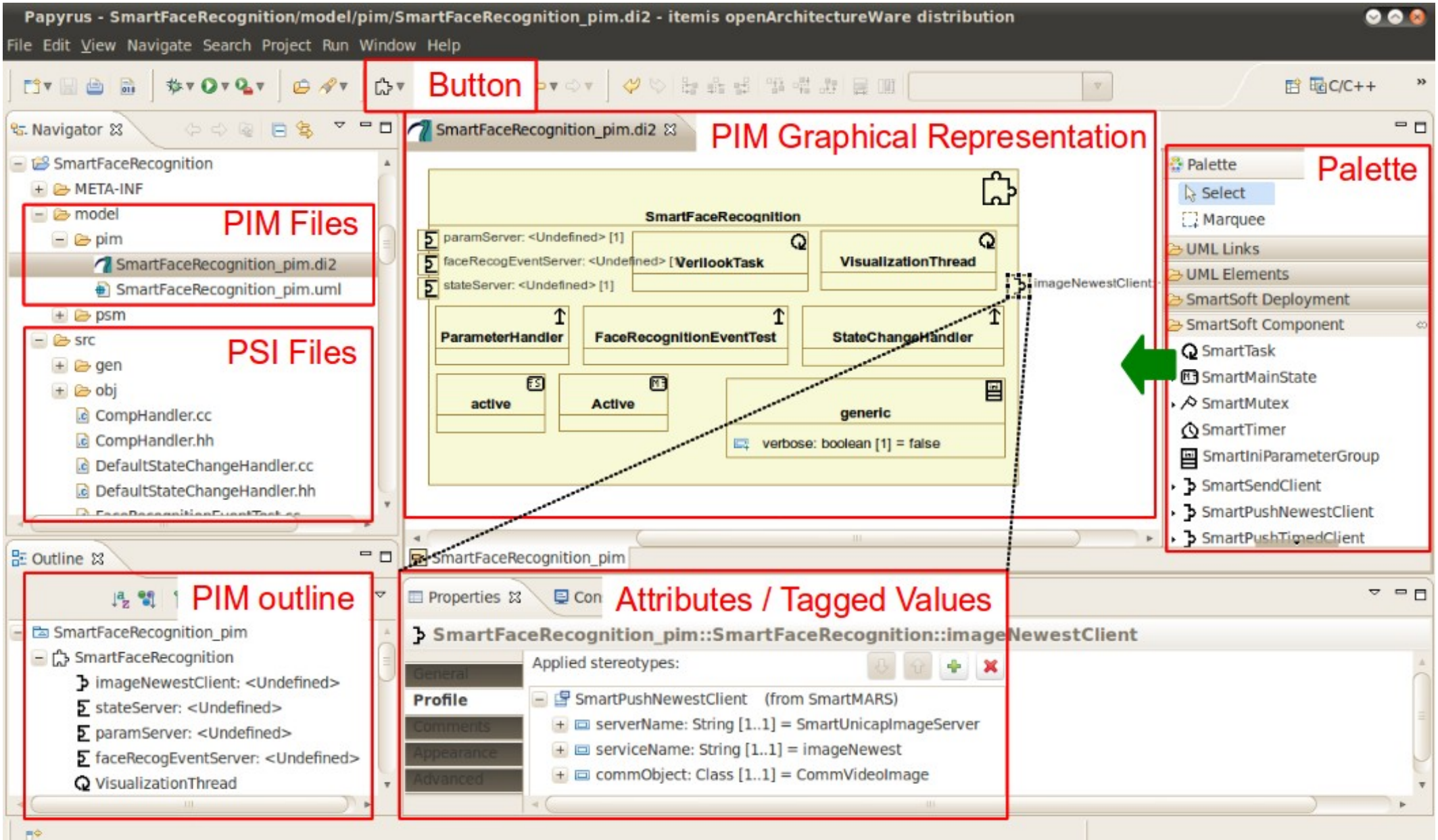
send	one-way communication
query	two-way request/response
push newest	1-to-n distribution
push timed	1-to-n distribution
event	asynchronous conditioned notification

The SmartSoft Services

param	component configuration
state	activate/deactivate component services
wiring	dynamic component wiring
diagnose	introspection of components
(internally based on communication patterns)	

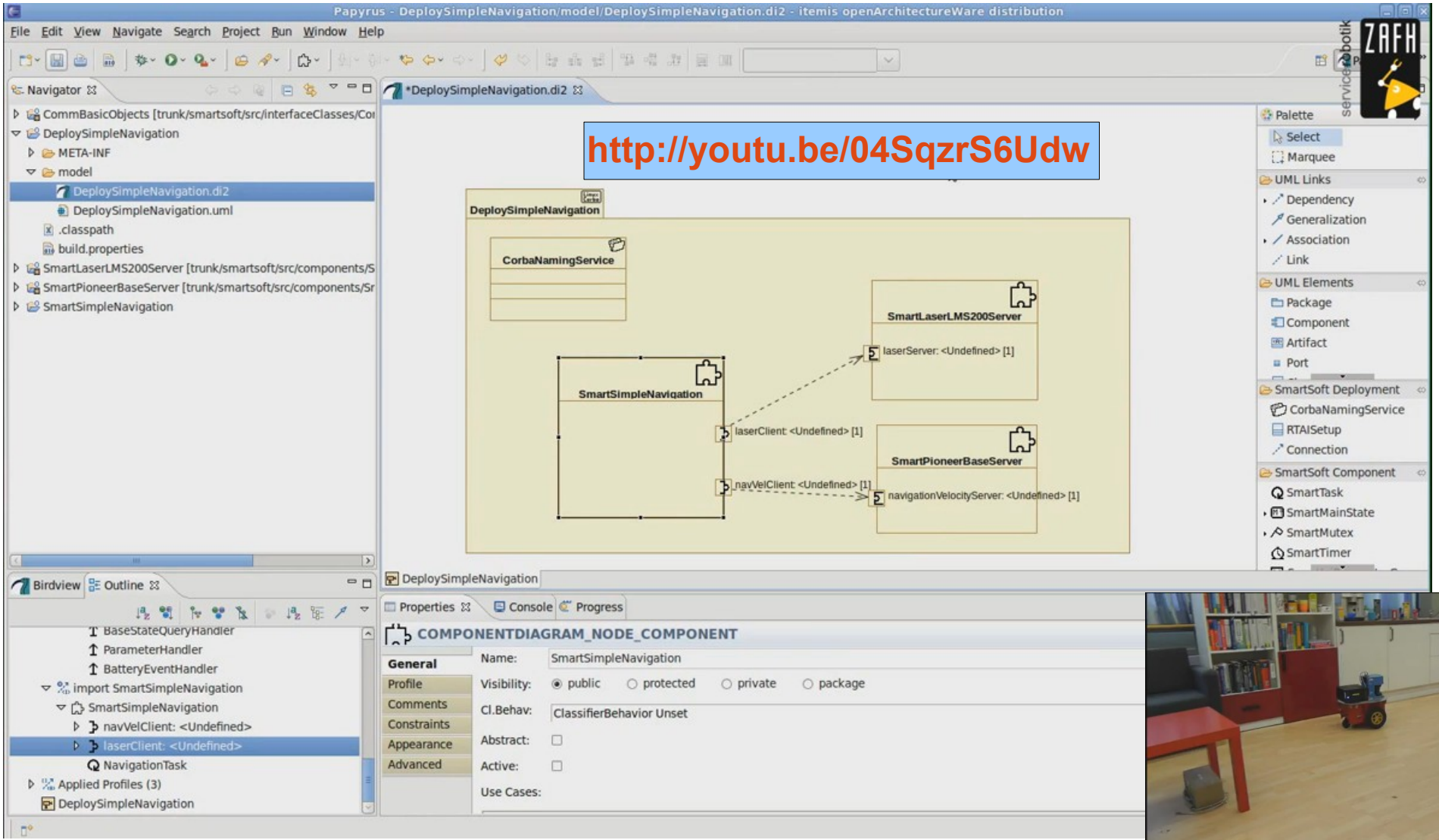
service robotik

Model-Driven Approach
Component Builder View



Model-Driven Approach

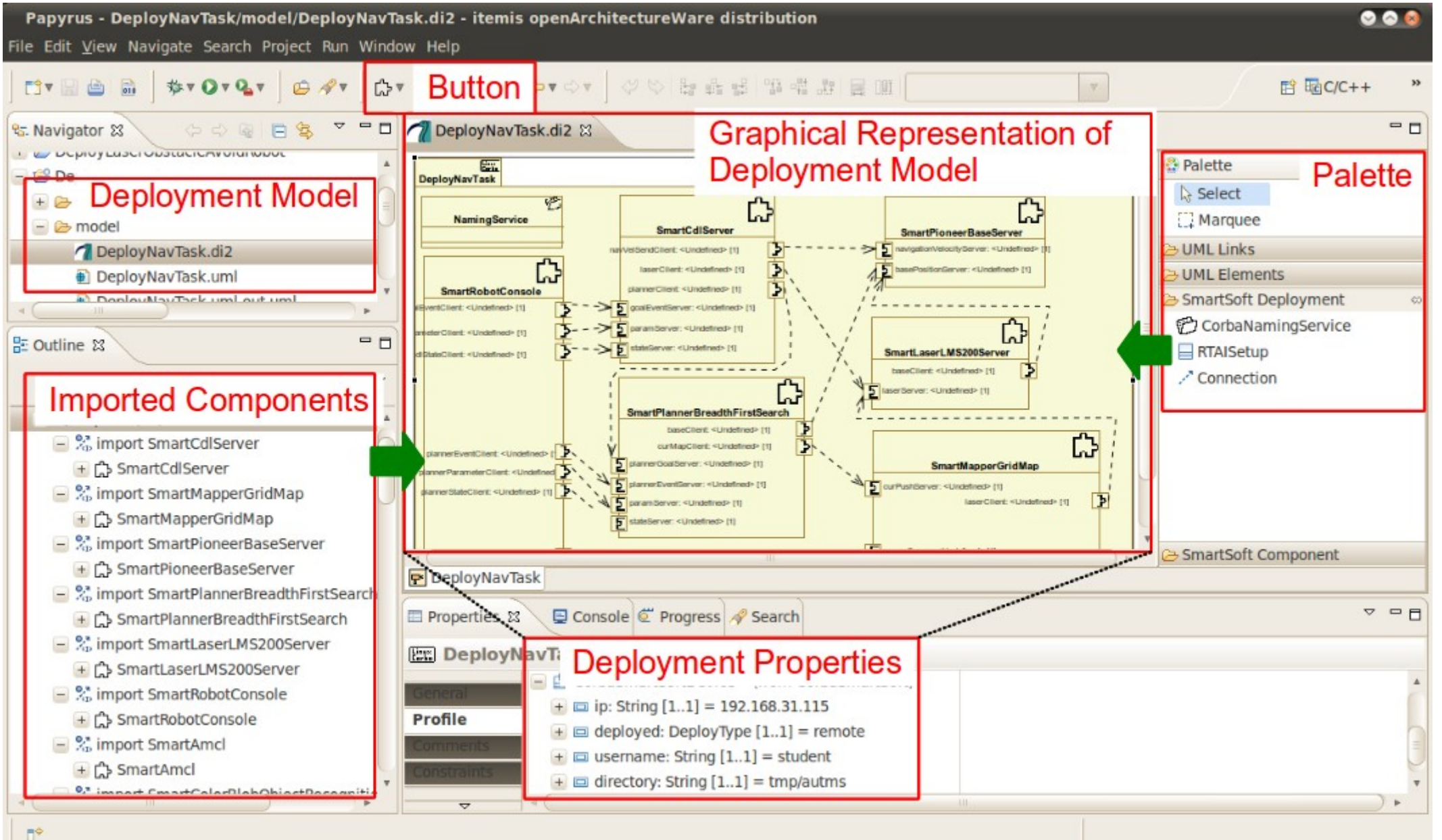
Screencast “Simple Navigation”



service robotik

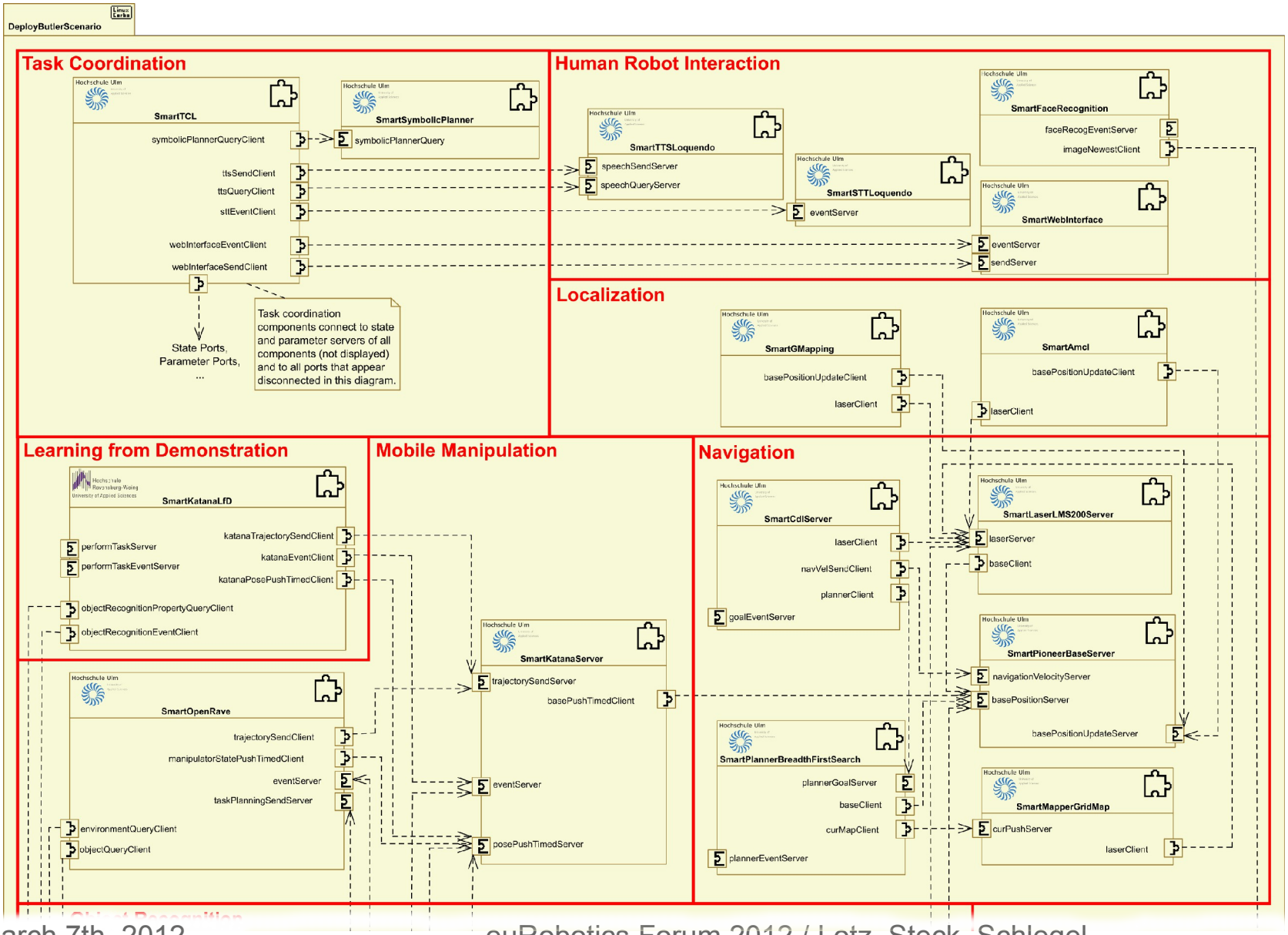
Model-Driven Approach

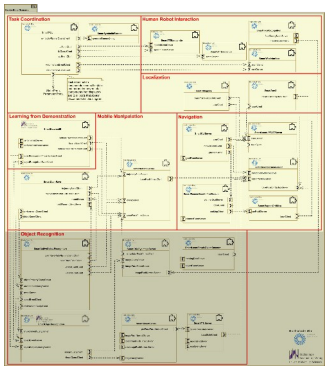
System Integrator View



Model-Driven Approach

System Integrator View – Buttler Scenario

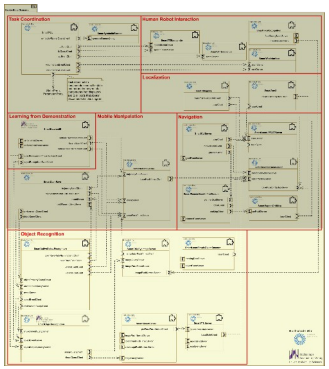


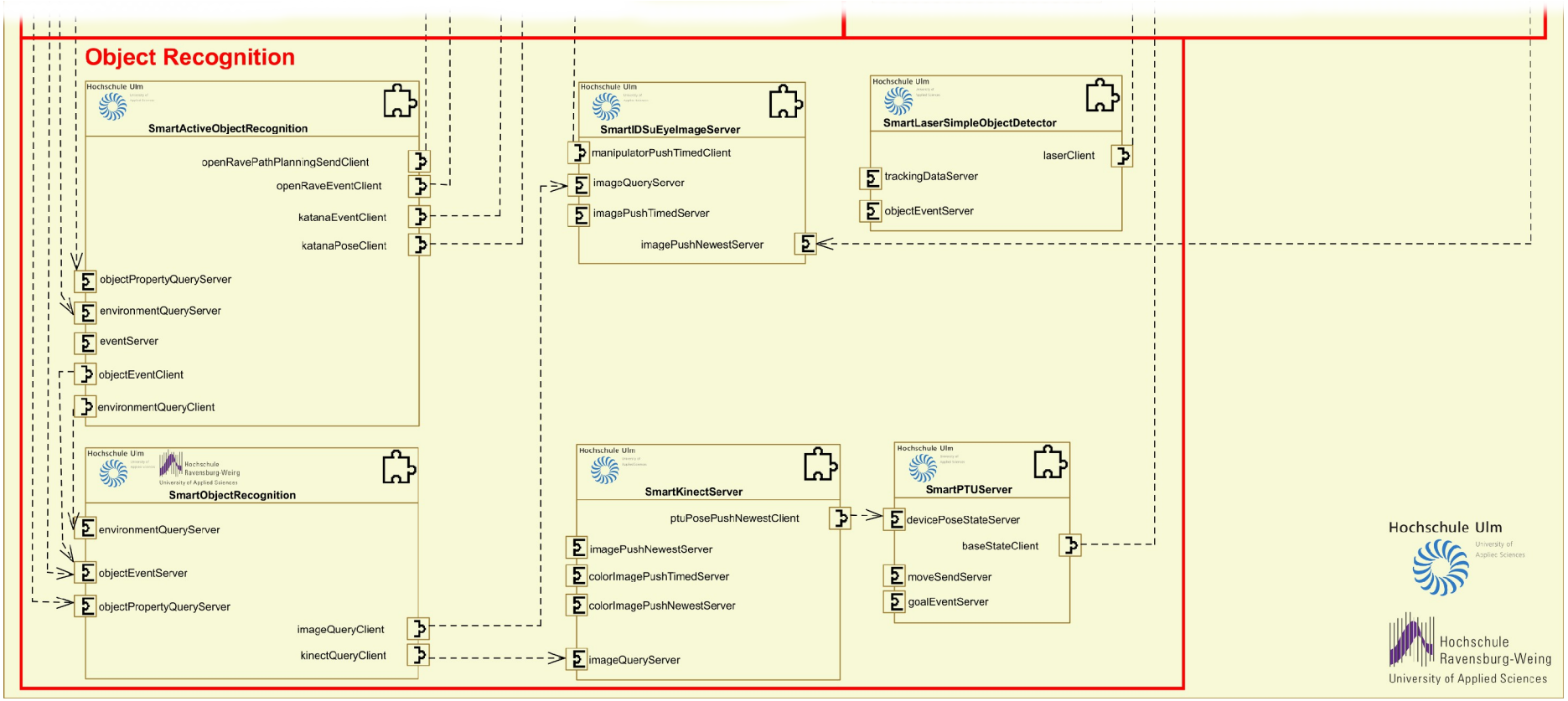


Hochschule Ulm

Model-Driven Approach

System Integrator View – Buttler Scenario

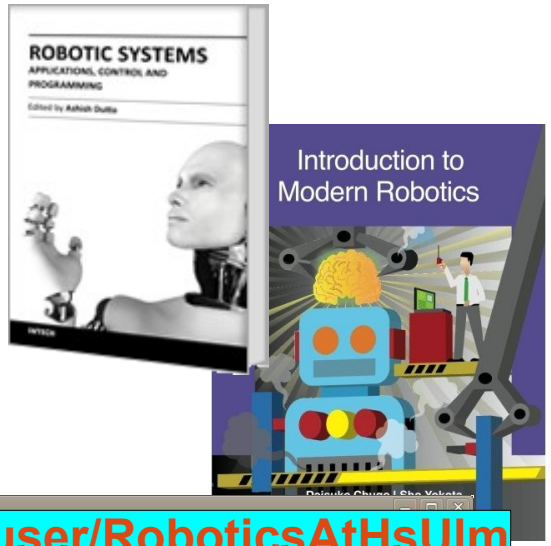




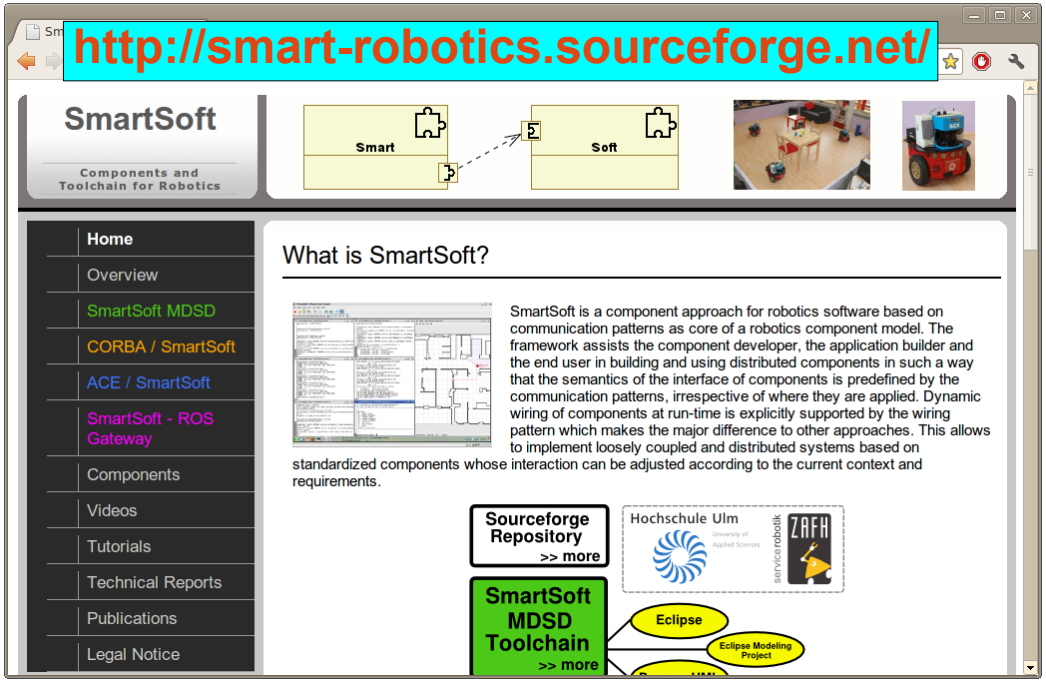
Further References

Recent book chapters (open access PDFs)

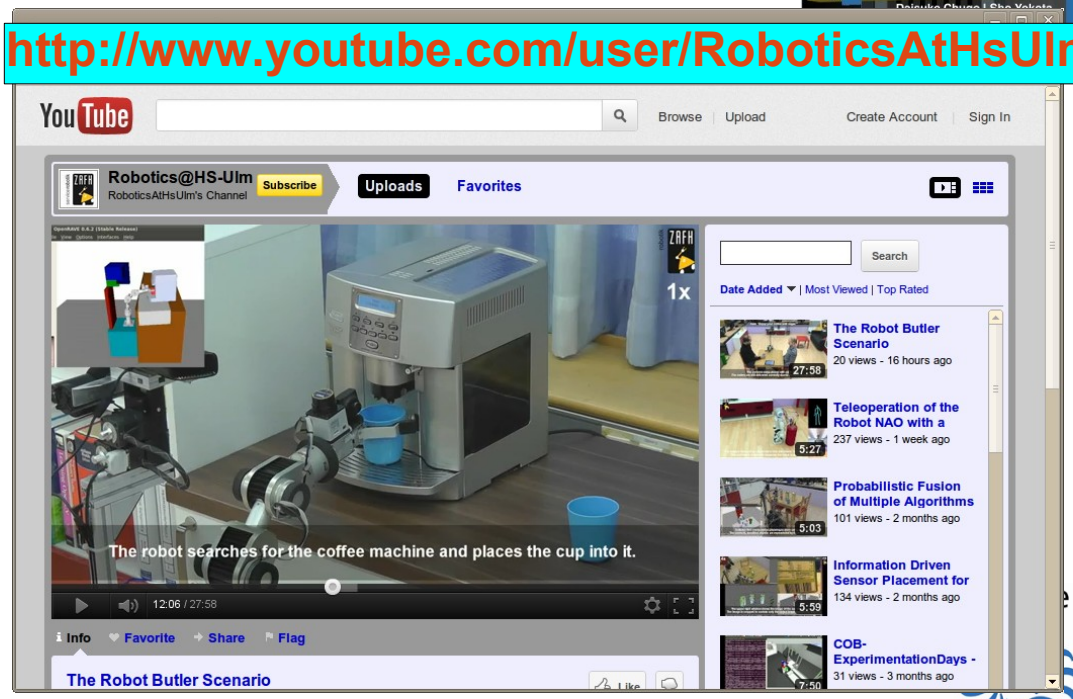
- Christian Schlegel, Andreas Steck and Alex Lotz, "**Robotic Software Systems: From Code-Driven to Model-Driven Software Development**", in *Robotic Systems - Applications, Control and Programming*, ISBN 978-953-307-941-7, InTech, 2012 **Download as PDF**
- Christian Schlegel, Andreas Steck, and Alex Lotz. "**Model-Driven Software Development in Robotics: Communication Patterns as Key for a Robotics Component Model**", in *Introduction to Modern Robotics*, ISBN 978-0980733068, iConcept Press, 2011 **Download as PDF**



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



http://www.youtube.com/user/RoboticsAtHsUlm



service robotik

Selected publications for
Model usage at Run-Time

- Christian Schlegel, Andreas Steck, Davide Brugali, Alois Knoll. **“Design Abstraction and Processes in Robotics: From Code-Driven to Model-Driven Engineering”**, in *2nd International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Pages 324-335, Darmstadt, Springer LNAI 6472, ISBN-10 3-642-17318-7, 2010
  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 Developmment"**, in *Proc. 1st International Workshop on Domain-Specific Languages and models for ROBotic systems (DSLRob/IROS)*, Taipei, Taiwan, October 2010
  Get as PDF

Hochschule Ulm