



servicerobotics

Autonomous Mobile Service Robots

Andreas Steck

University of Applied Sciences Ulm, Germany



Research Interests:

- *Model-Driven Software Engineering for Robotics*
- *Three Layer Architectures*
- *Middleware for Robotics*
- *Task Coordination for Robotics*
- *Service Robotics*

My research work contributes to the SmartMARS meta-model and the SmartSoft MDSD Toolchain. I am convinced that model-driven software engineering is mandatory in service robotics to bridge the gap between lab prototypes and service robots suitable for everyday use. Furthermore, I am working on Three Layer Architectures to bridge the gap between symbolic and subsymbolic mechanisms of information processing.

<http://www.hs-ulm.de/steck>

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

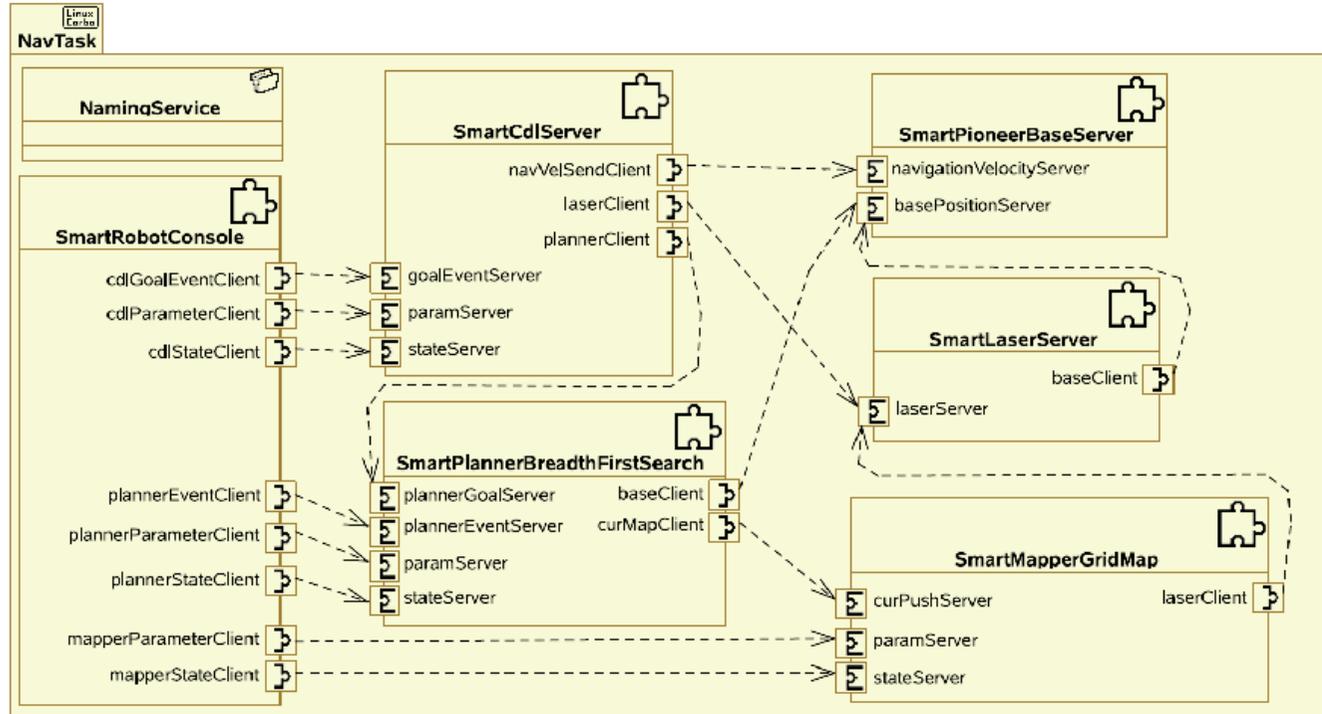
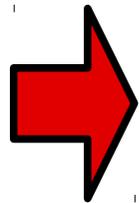
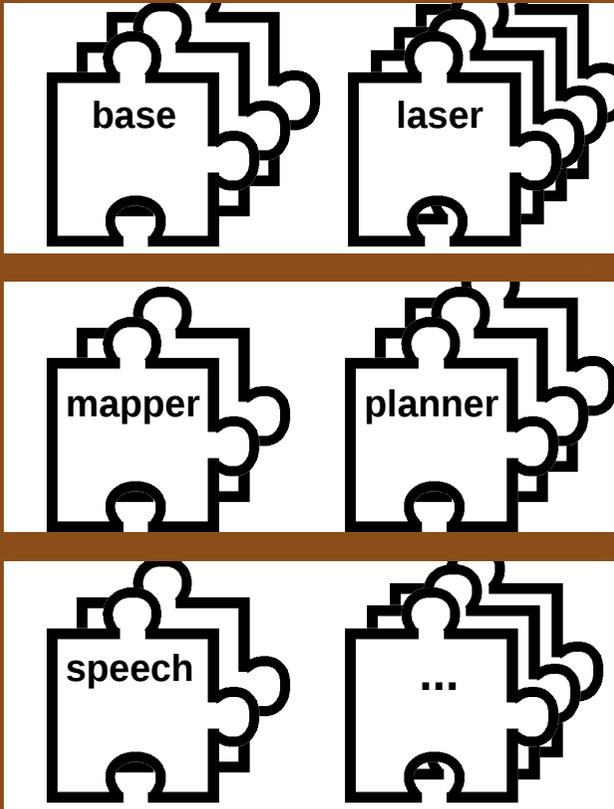
<http://www.zafh-servicerobotik.de/ULM/en/index.php>





Building robots by composing reusable components

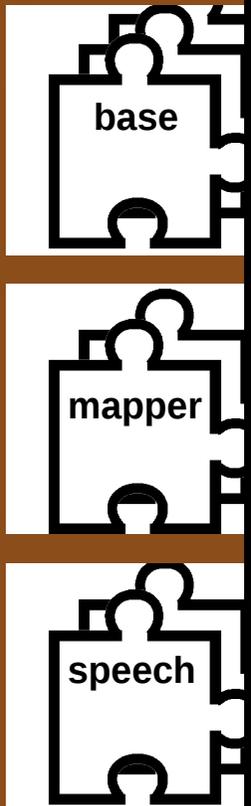
Component Shelf



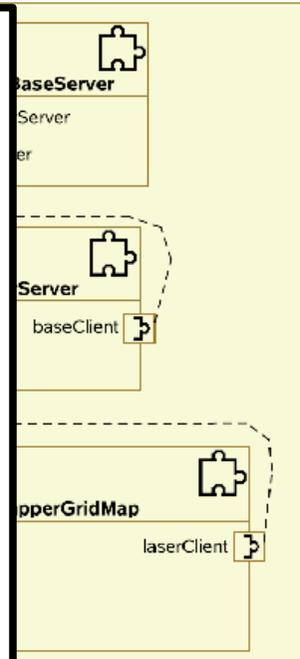
Building robots by composing reusable components


 NavTask

Comp



- precisely defined component services
→ component model
- loosely coupled components with local responsibility
- middleware independence
→ is achieved with MDSD
- Resource Awareness and Quality of Service (QoS)
- orchestration of the components





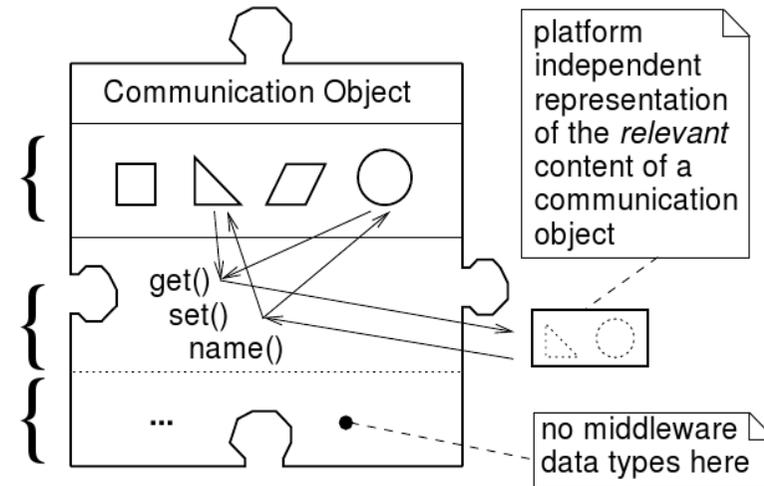
Precisely Defined Services

services are defined by an interaction pattern and a communication object

data structures

framework interface

arbitrary user member functions



The SmartSoft Interaction Patterns

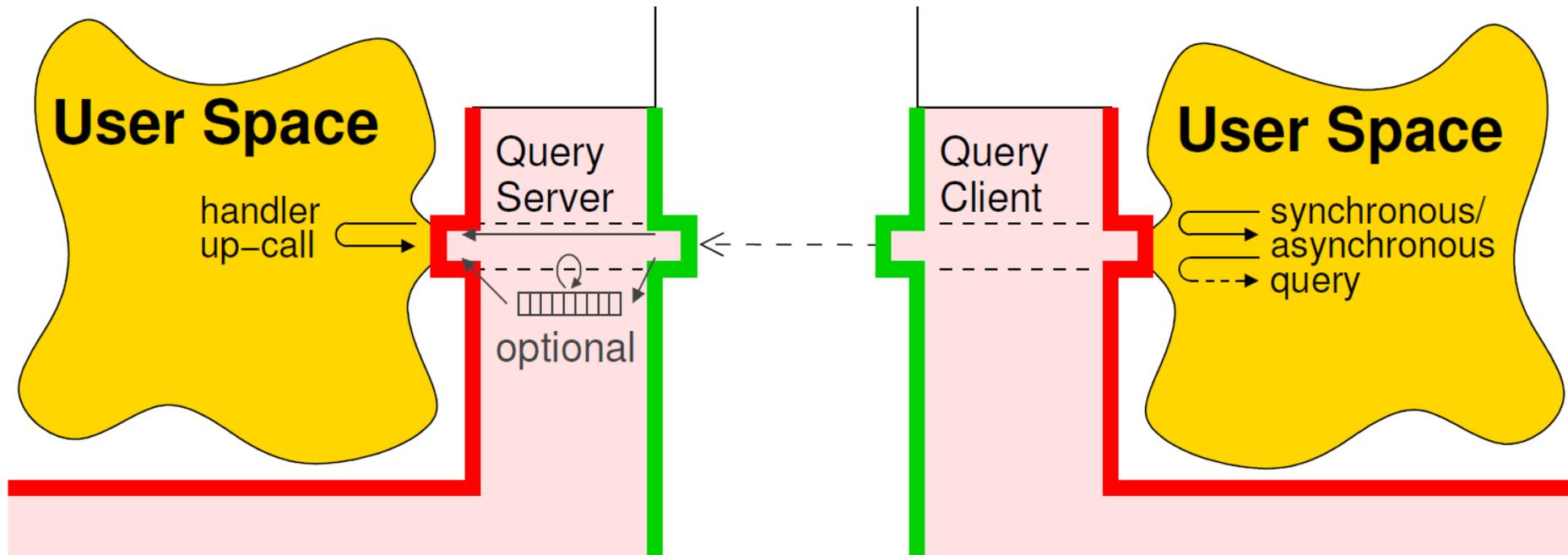
send	one-way communication
query	two-way request/response
push newest	1-to-n distribution
push timed	1-to-n distribution
event	asynchronous conditioned notification
state	activate/deactivate component
wiring	dynamic component wiring

These patterns are sufficient since they offer request/response interaction as well as asynchronous notifications and push services.



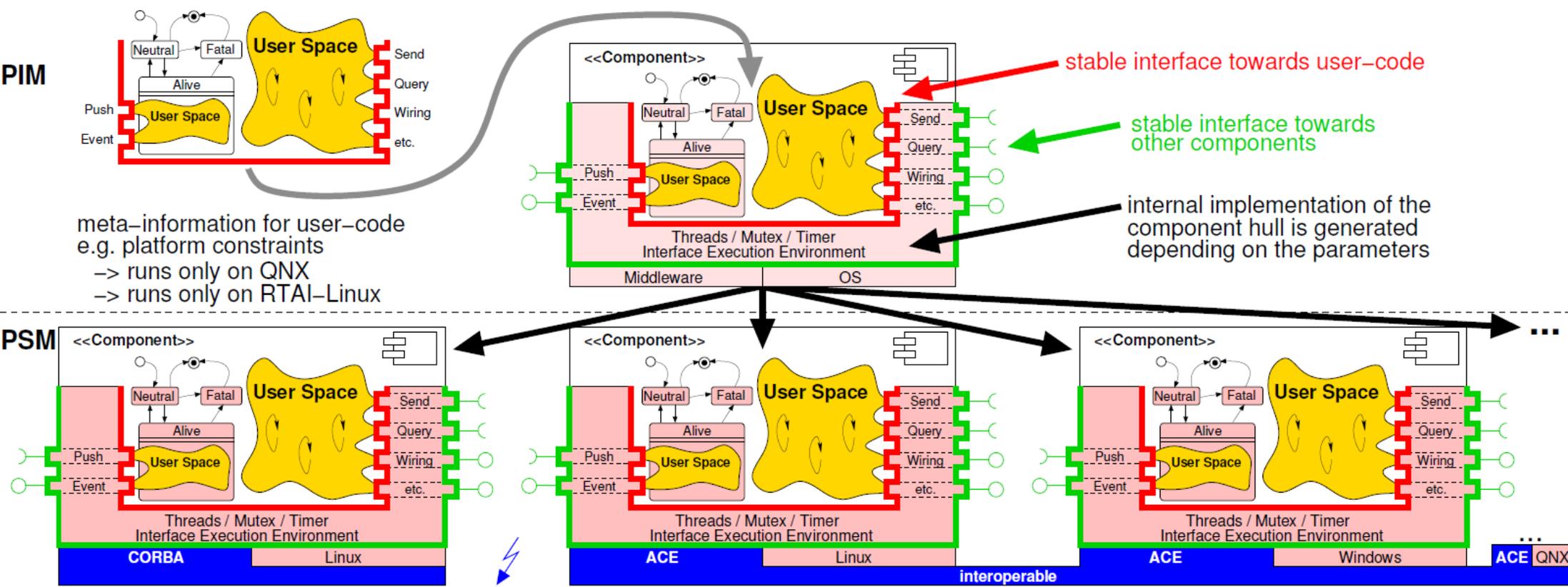
Loosely Coupled Components

Example: Query-Pattern



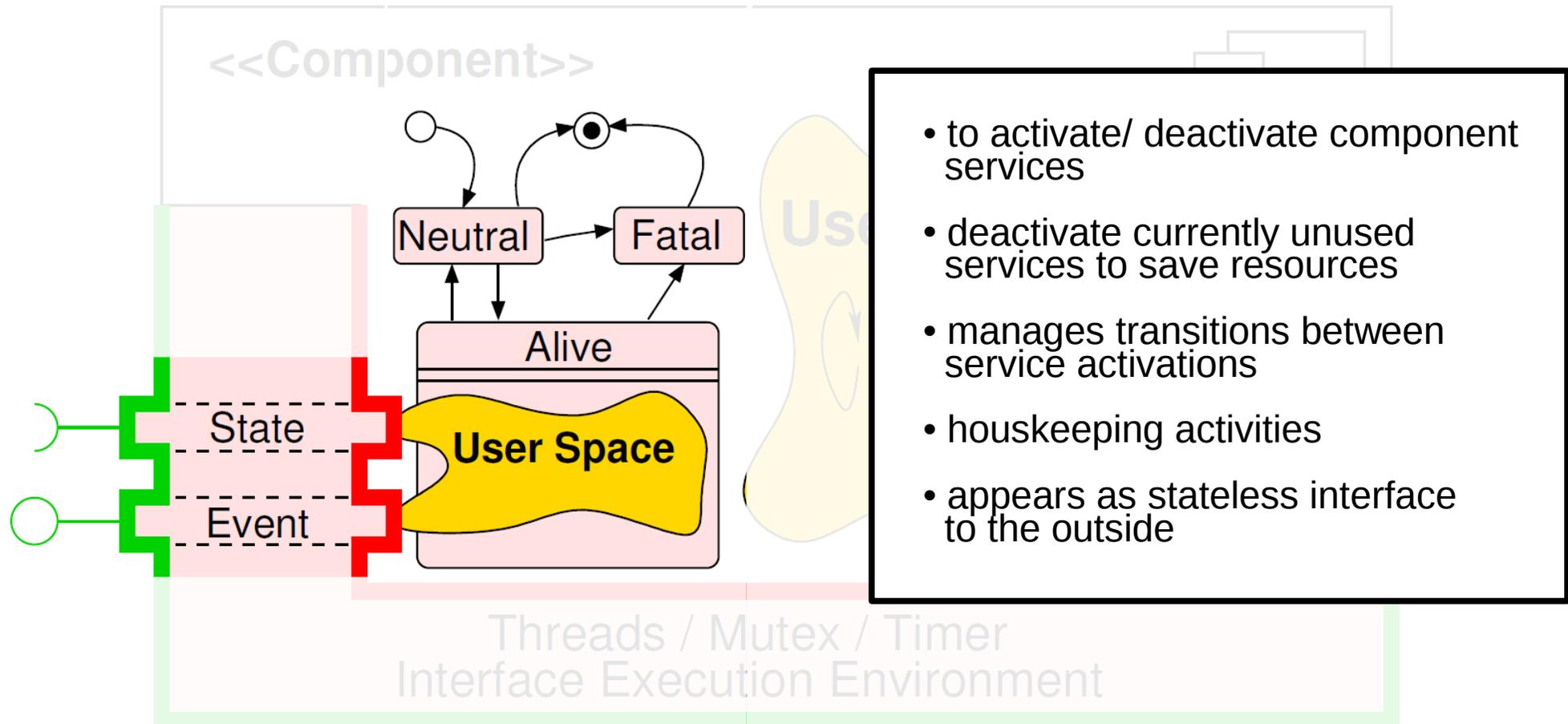
- query requests can optionally be queued (active / passive handler)
- calling the handler in active mode is done by a dedicated task.
- whether the handler should be active or passive can simply be changed in the model (parameter `isActive = true | false`)
- depending on the parameter `isActive` a different component hull is generated by the toolchain. No modification in the User-Code necessary

Middleware Independence Reuse of User-Code



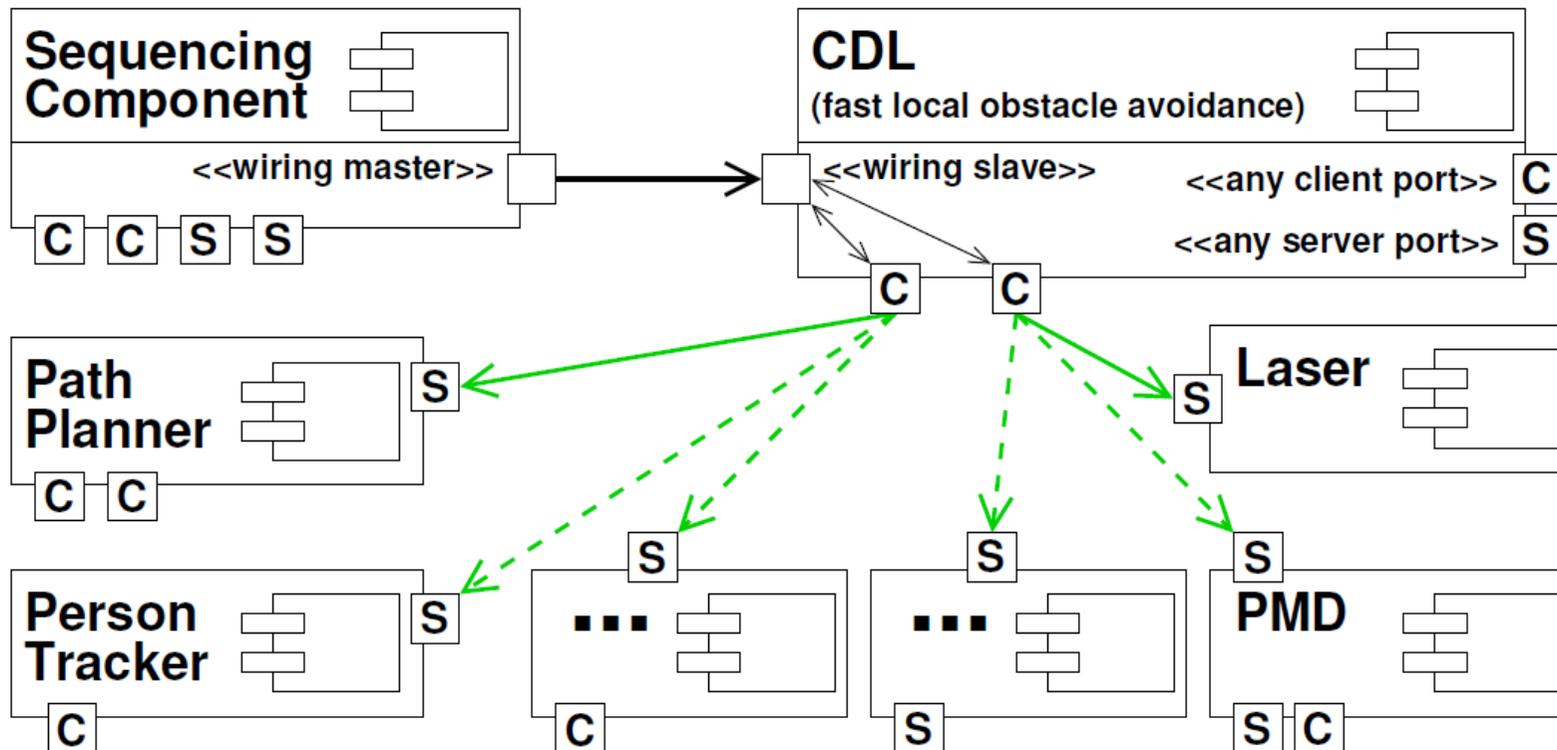
Resource Awareness

Example: State Pattern – State Automaton



Resource Awareness

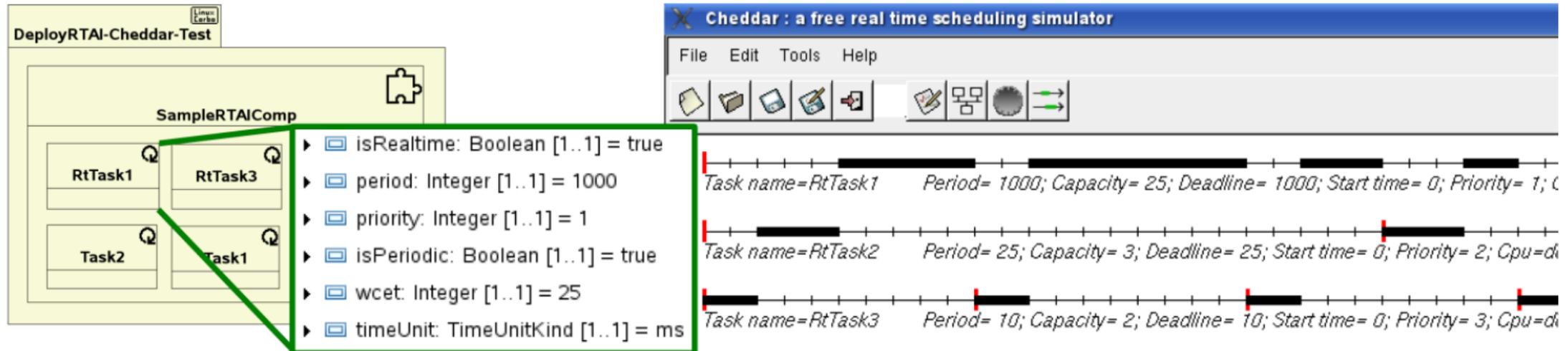
Example: Dynamic Wiring of Services



- a wiring master port can be used to rearrange the client connections of components at run-time
- e.g. CDL component can receive its intermediate goals either from a path planner or a person tracker and its distance information from either the laser or the PMD component

Resource Awareness and Quality of Service

Example: Schedulability Analysis (CHEDDAR)

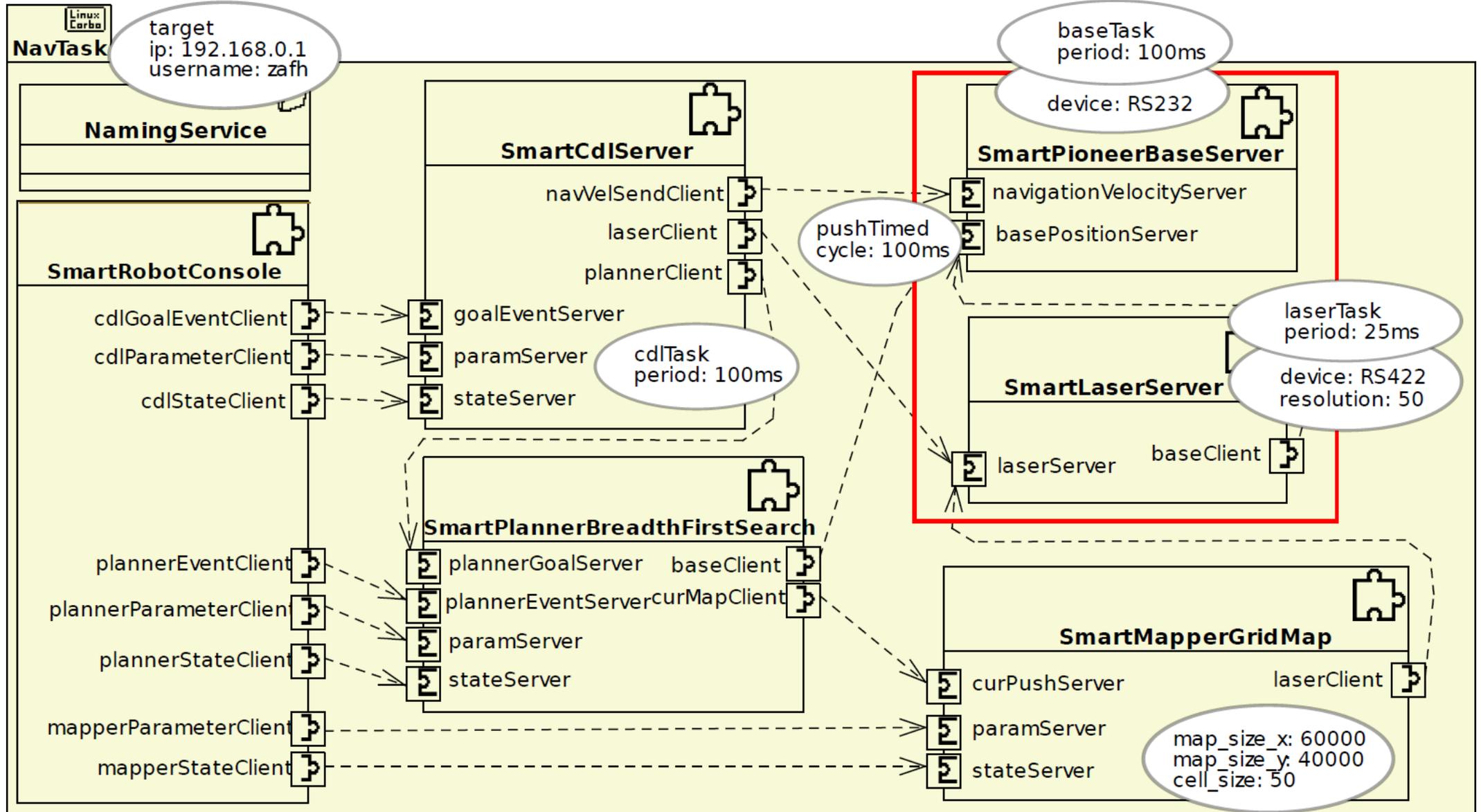


- PSM is transformed into a CHEDDAR specific analysis model
- only the realtime tasks are taken into account (period, wcet)
- analysis of parameters explicated in the modeling level
- realtime schedulability analysis



Resource Awareness and Quality of Service

Example: Navigation Task





SmartSoft MDSD Toolchain Technical View

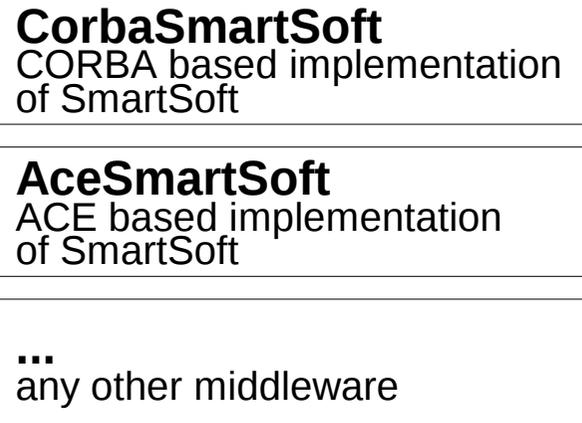
PIM

PSM

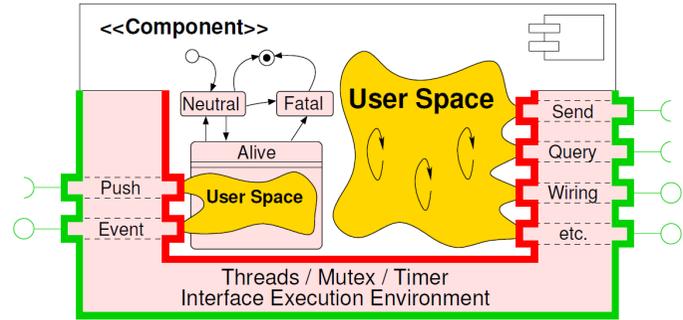
PSI

SmartMARS – Metamodel
(Modeling and Analysis of Robotics Systems)

M2M
oAW
xTend
check



M2T
oAW
xPand
check



- UML2-Profile
- platform independent stereotypes
 - SmartComponent
 - SmartTask
 - SmartMutex
 - SmartQueryServer
 - SmartEventClient
 - ...

- UML2-Profile
- platform specific stereotypes

has to be created by a middleware expert

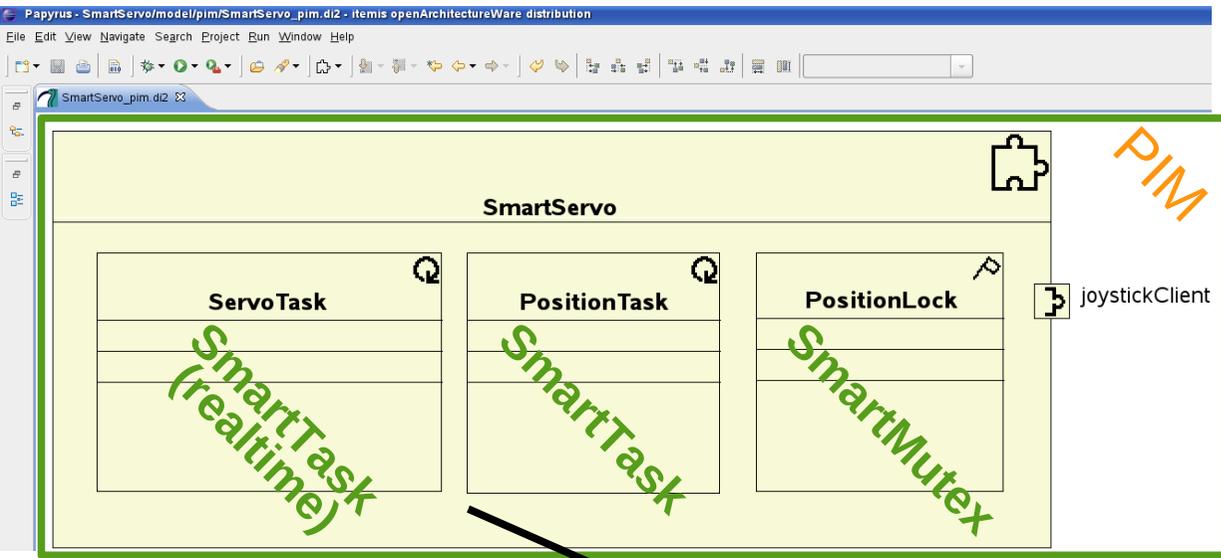
The User Space can contain arbitrary code and libraries

The User Space stays the same independent of the different platform specific models

Just the component hull will be created



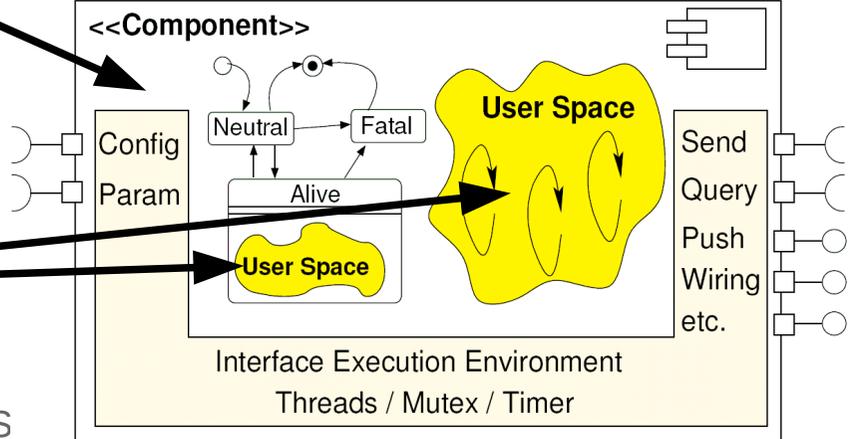
SmartSoft MDSD Toolchain User View



verification (e.g. QoS)+
 transformation
(is done by toolchain)

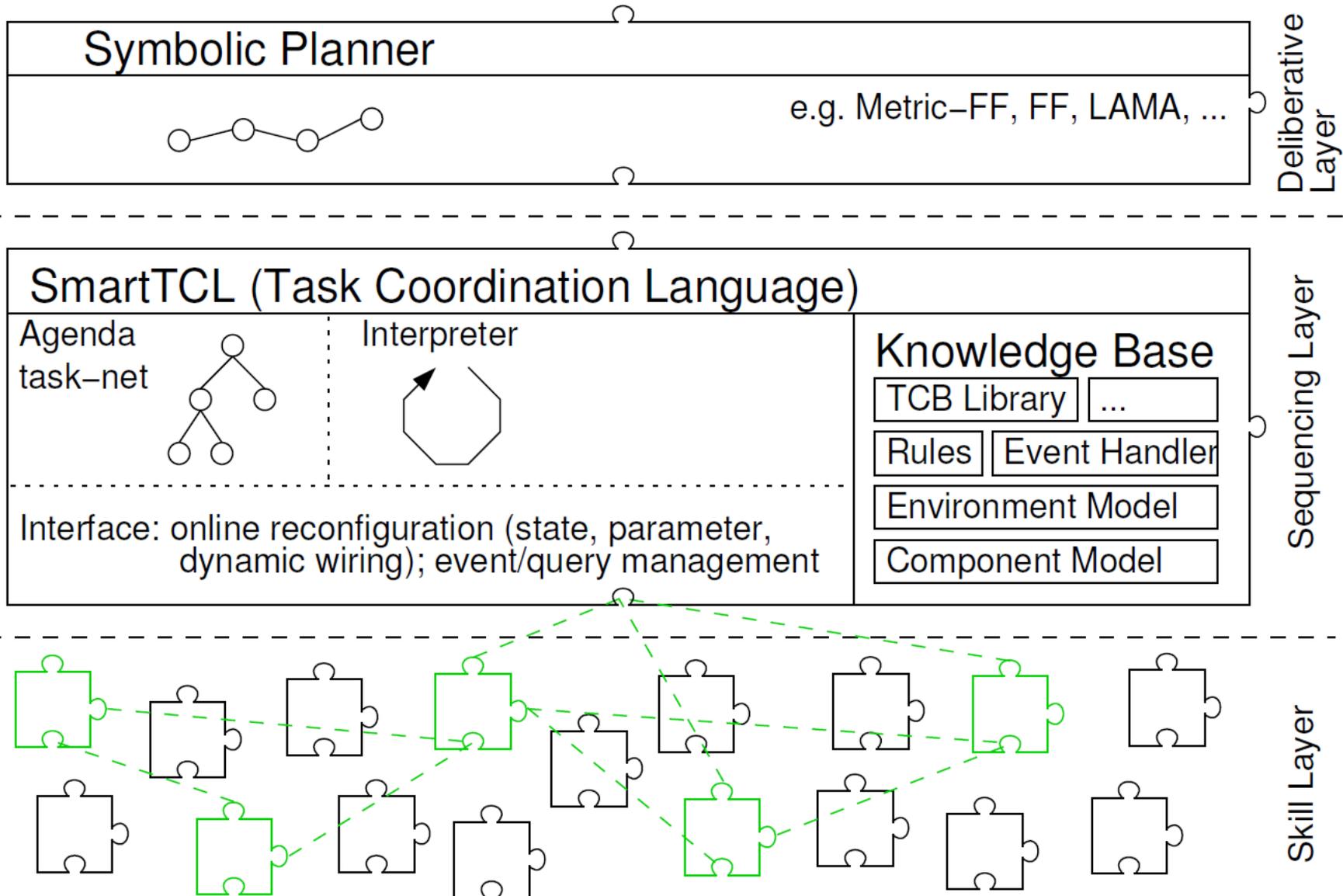
User Code
 MATLAB / Simulink
 RTAI-Lab
 OpenCV / Qt / Kavraki-Lab

executable component





The Three Layer Architecture based on the SMARTSOFT Concepts

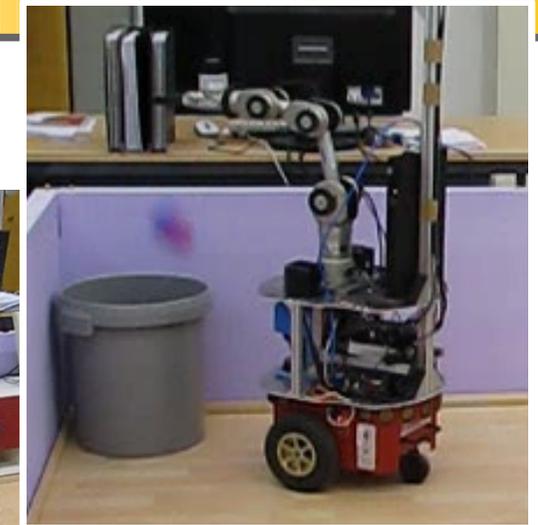
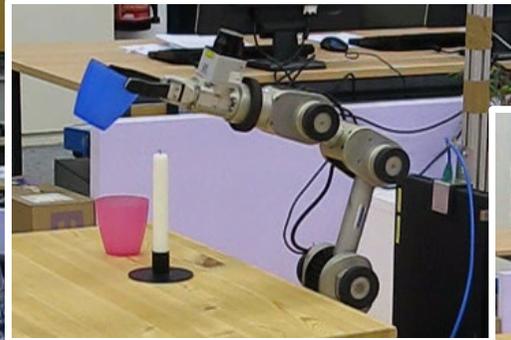


Example: “Follow Me” Reuse for RoboCup@Home



- deployment of COTS components
- reuse for RoboCup@Home
- robot follows a person through an unstructured environment

Example: “Cleanup Table Scenario”



[watch on youtube:](#)

<http://www.youtube.com/roboticsathsulm>

<http://www.youtube.com/watch?v=40d4Dlk5LCQ>



SmartSoft MDSD Toolchain Links



SmartSoft - Mozilla Firefox

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

SmartSoft Components and Toolchain for Robotics

Home
Overview
SmartSoft MDSD
CORBA / SmartSoft
ACE / SmartSoft
Components
Videos
Publications
Legal Notice

YouTube - Kanal von RoboticsAtHsUlm - Mozilla Firefox

<http://www.youtube.com/roboticsAtHsUlm>

Suchen

Konto erstellen oder anmelden

Abonnements Protokoll Video hochladen

Robotics@HS-Ulm Kanal von RoboticsAtHsUlm

Abonnieren Uploads

Suchen

Hinzugefügt am | Meist gesehen | Beste Bewertung

- Follow Me - SmartBots@Ulm - 15 views - vor 5 Tagen
- Mobile Manipulation using a Katana arm - 61 views - vor 1 Woche
- Who is Who? - SmartBots@Ulm - 112 views - vor 1 Monat
- Deployment of SmartSoft Components - Navigation - 156 views - vor 3 Monaten
- Visual SLAM - Lifelong Localization of a Mobile - 184 views - vor 3 Monaten

