



servicerobotics

Autonomous Mobile Service Robots

SmartSoft MDSD Toolchain

7 May 2010 / SDIR V - Anchorage

Andreas Steck and Christian Schlegel

*Computer Science Department
University of Applied Sciences Ulm*

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

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





SmartSoft MDSD Toolchain

2010-05-07 SDIR 5 - Anchorage

- **Example 1: Servo (RTAI-Linux)**
- **Example 2: Navigation Task**
- **Example 3: “Follow Me” - RoboCup@Home**
- **Example 4: Cleanup Table Scenario**

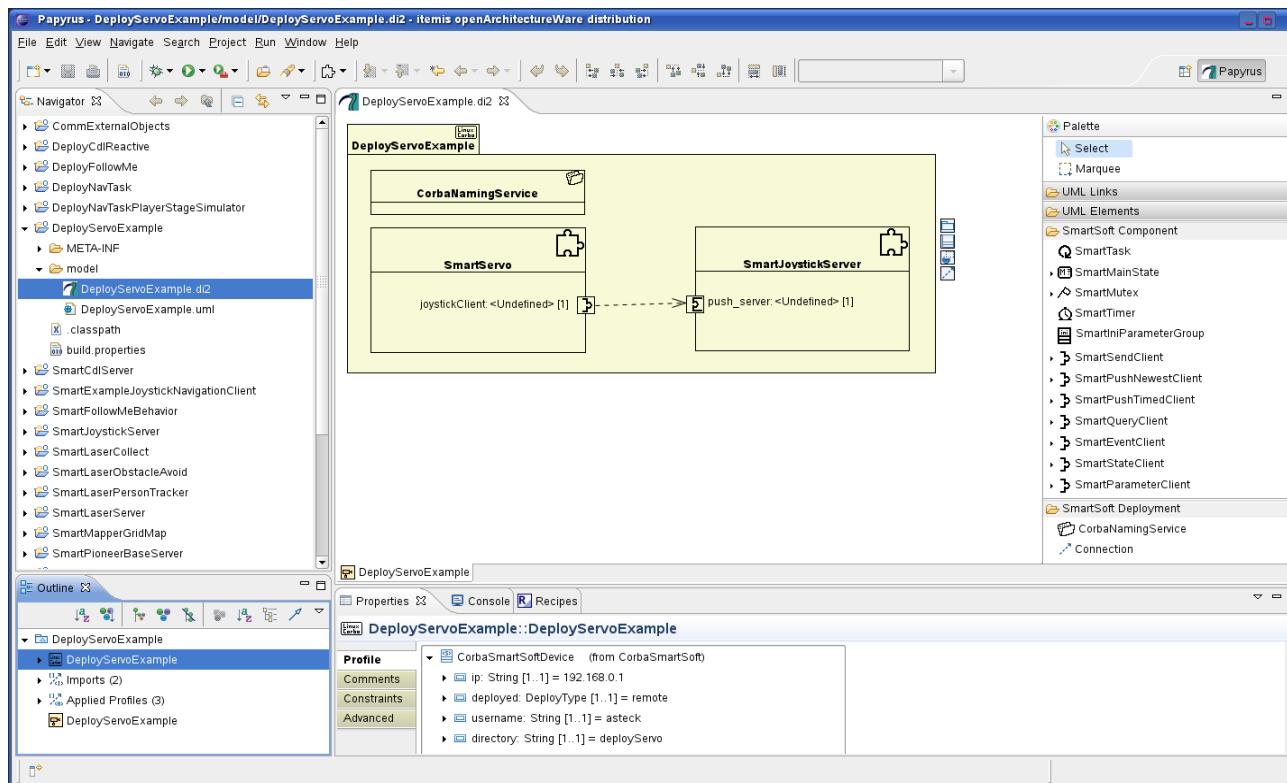




servicerobotics

Autonomous Mobile Service Robots

SmartSoft MDSD Toolchain



- Eclipse



- openArchitectureWare



- PapyrusUML



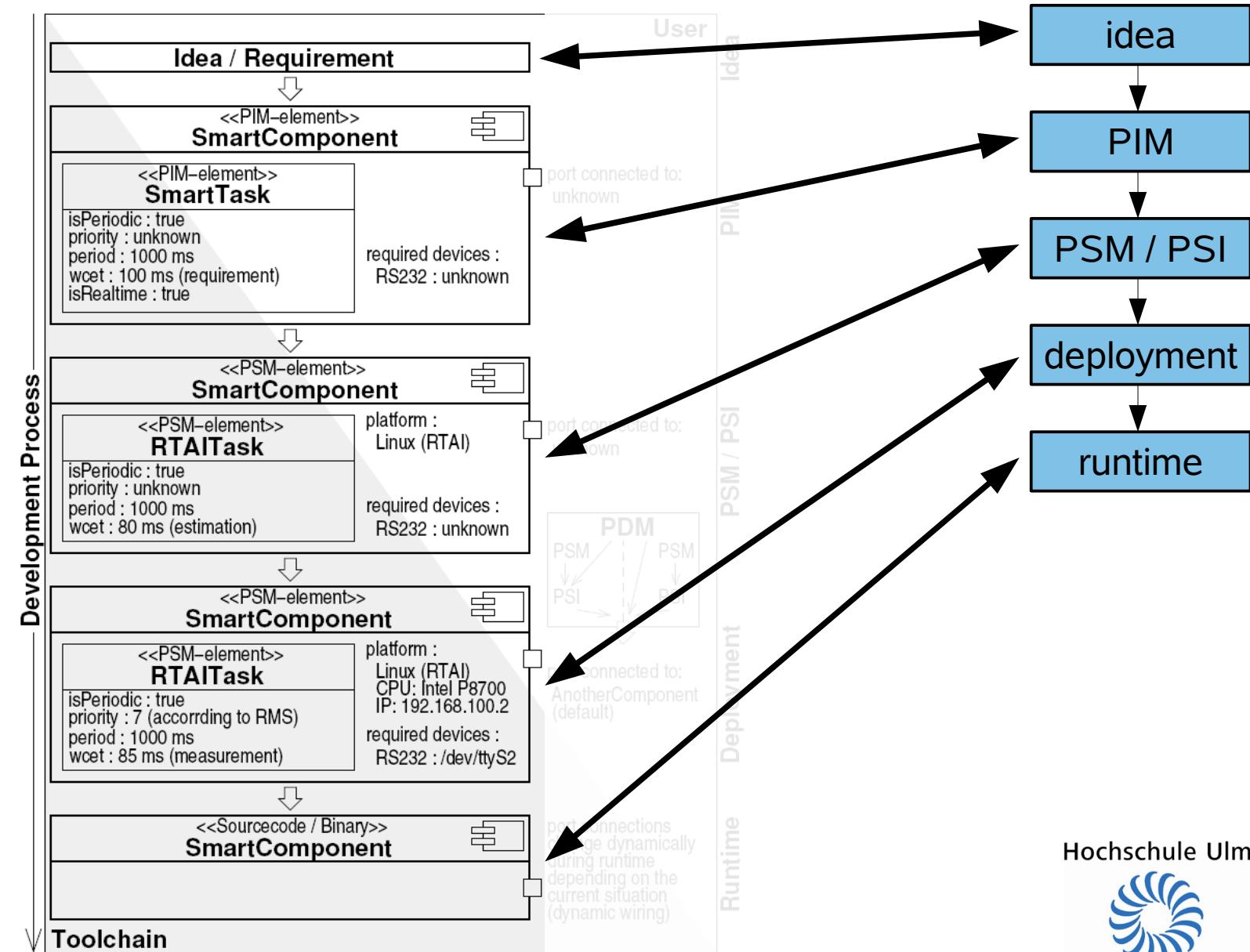
Hochschule Ulm





servicerobotics

Autonomous Mobile Service Robots





SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

overview
servo example



Hochschule Ulm

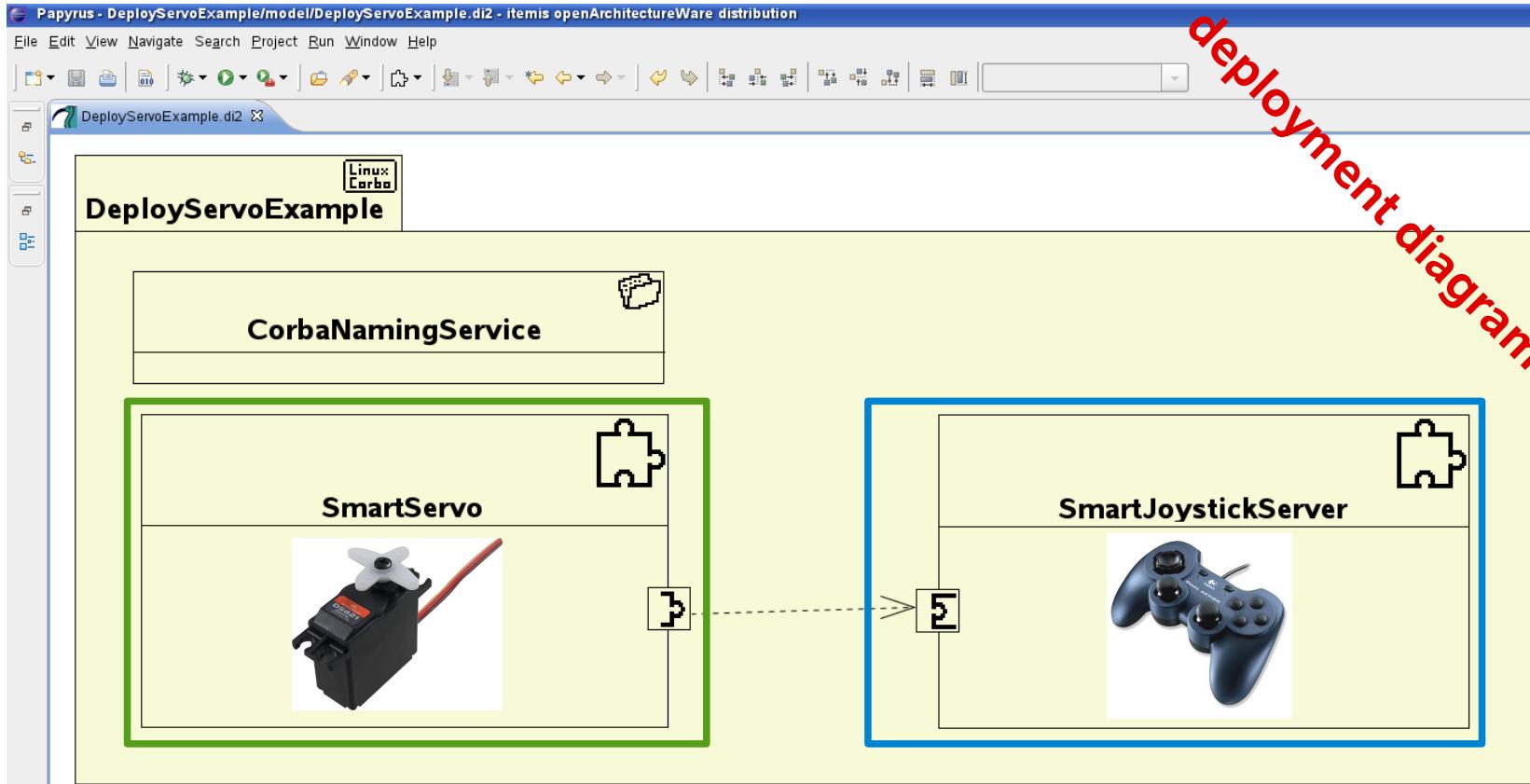




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

overview
servo example



SmartServo component
- will be created in the following demo

SmartJoystickServer component
- preexisting component (COTS)

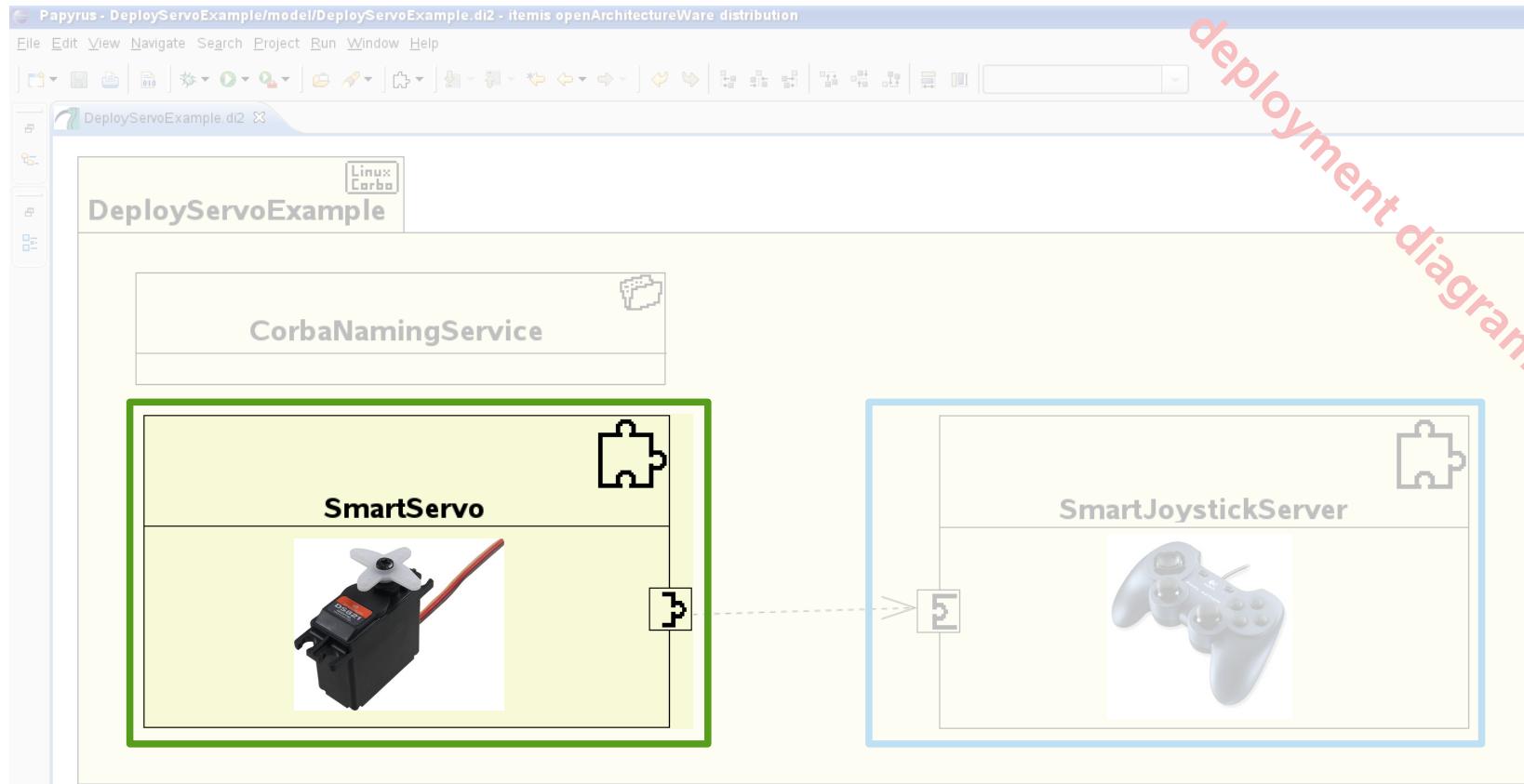




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

overview
servo example



SmartServo component
- will be created in the following demo

SmartJoystickServer component
- preexisting component (COTS)

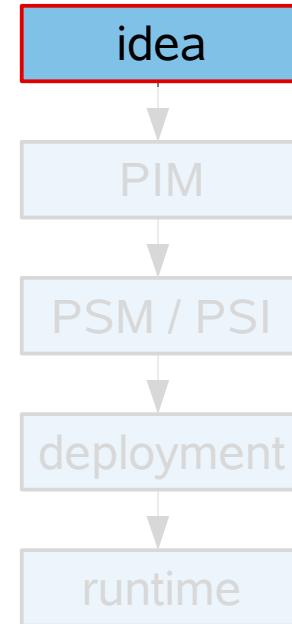
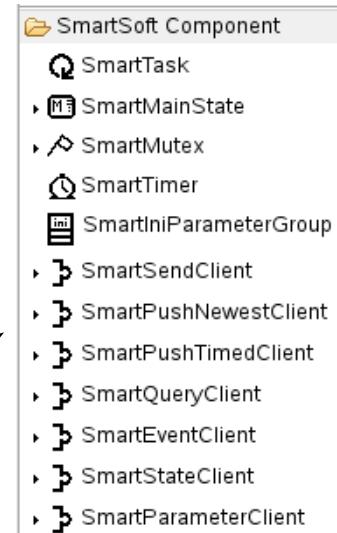
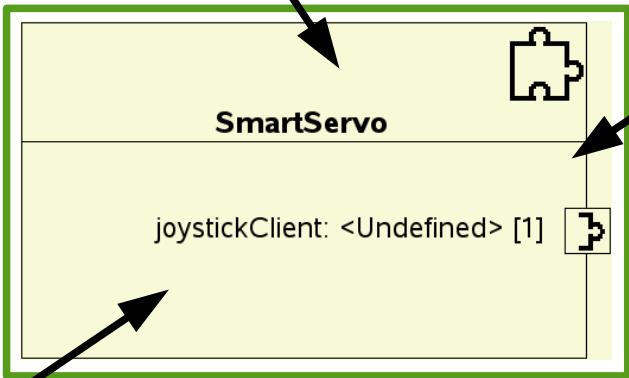
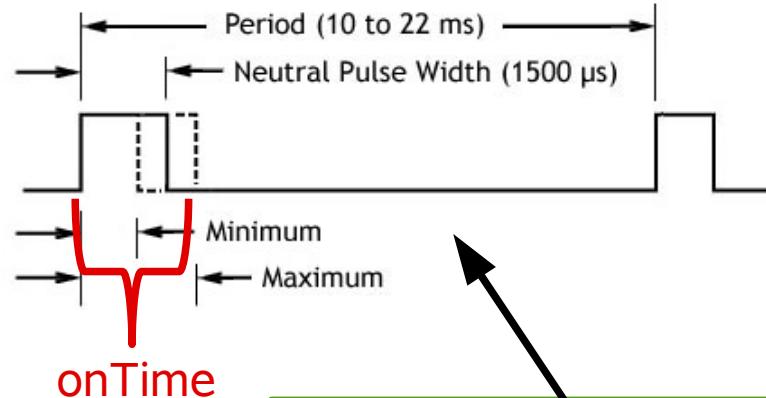




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component idea

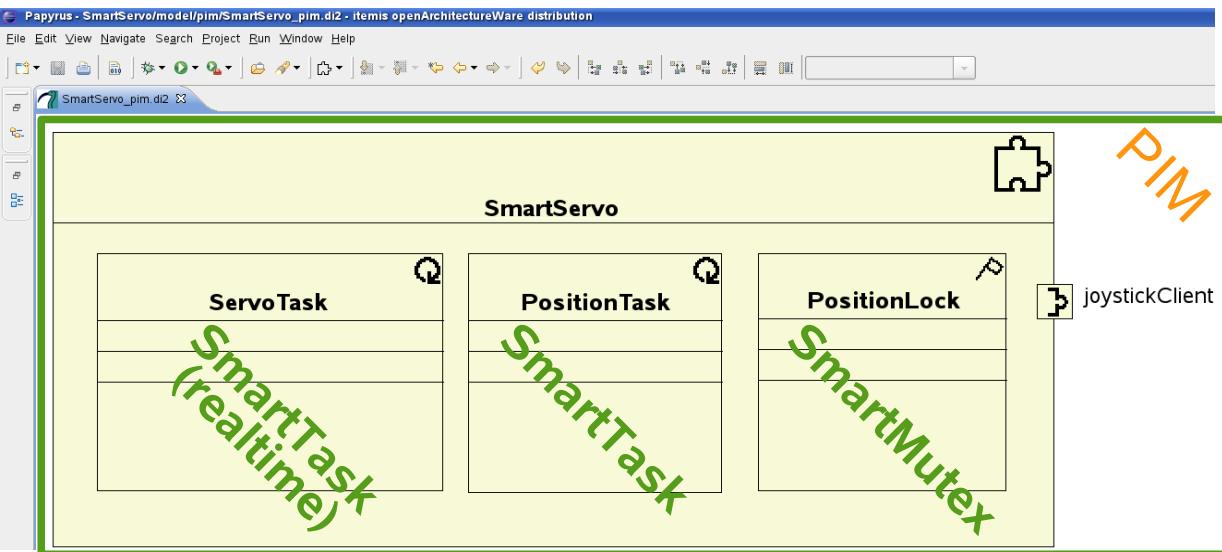




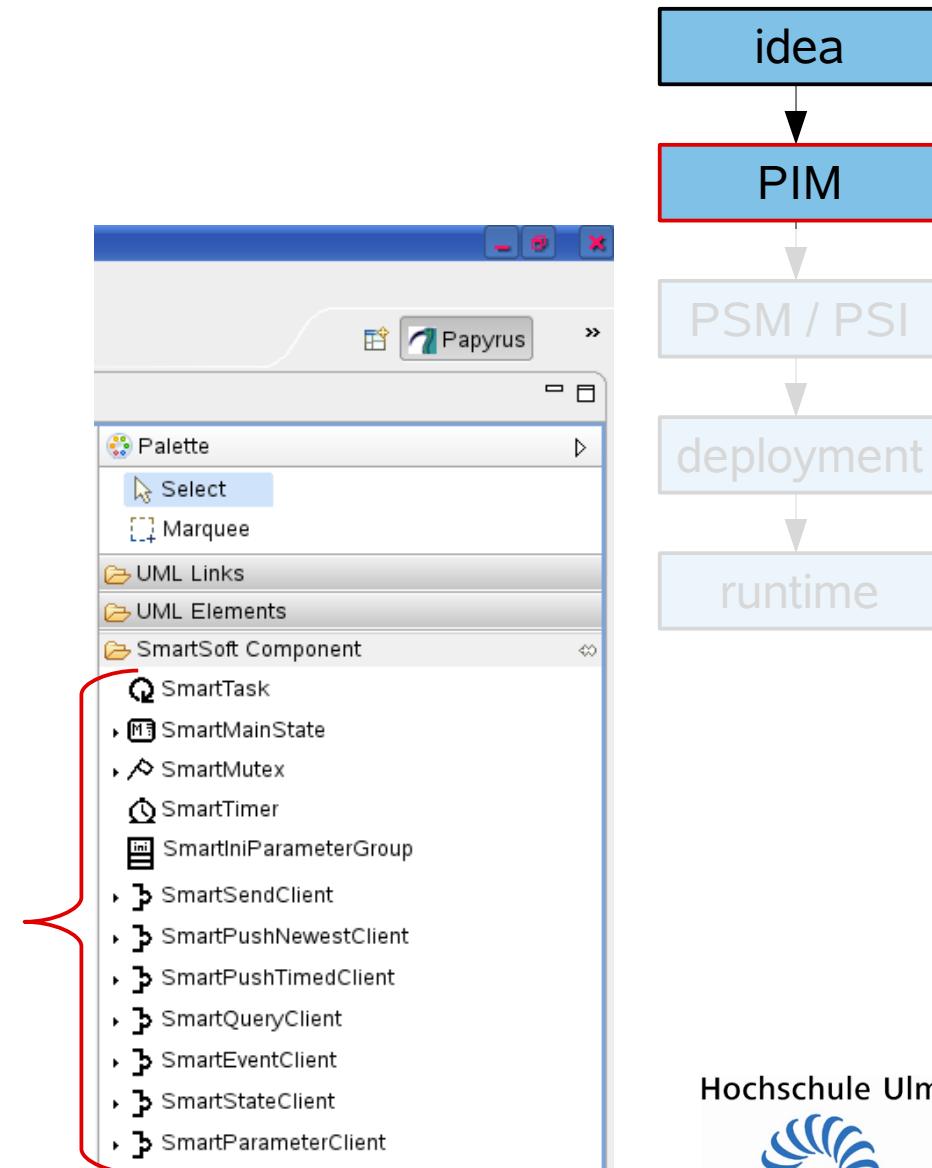
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component PIM



create PIM using
the elements
offered by the
SmartSoft MDSD
Toolchain palette

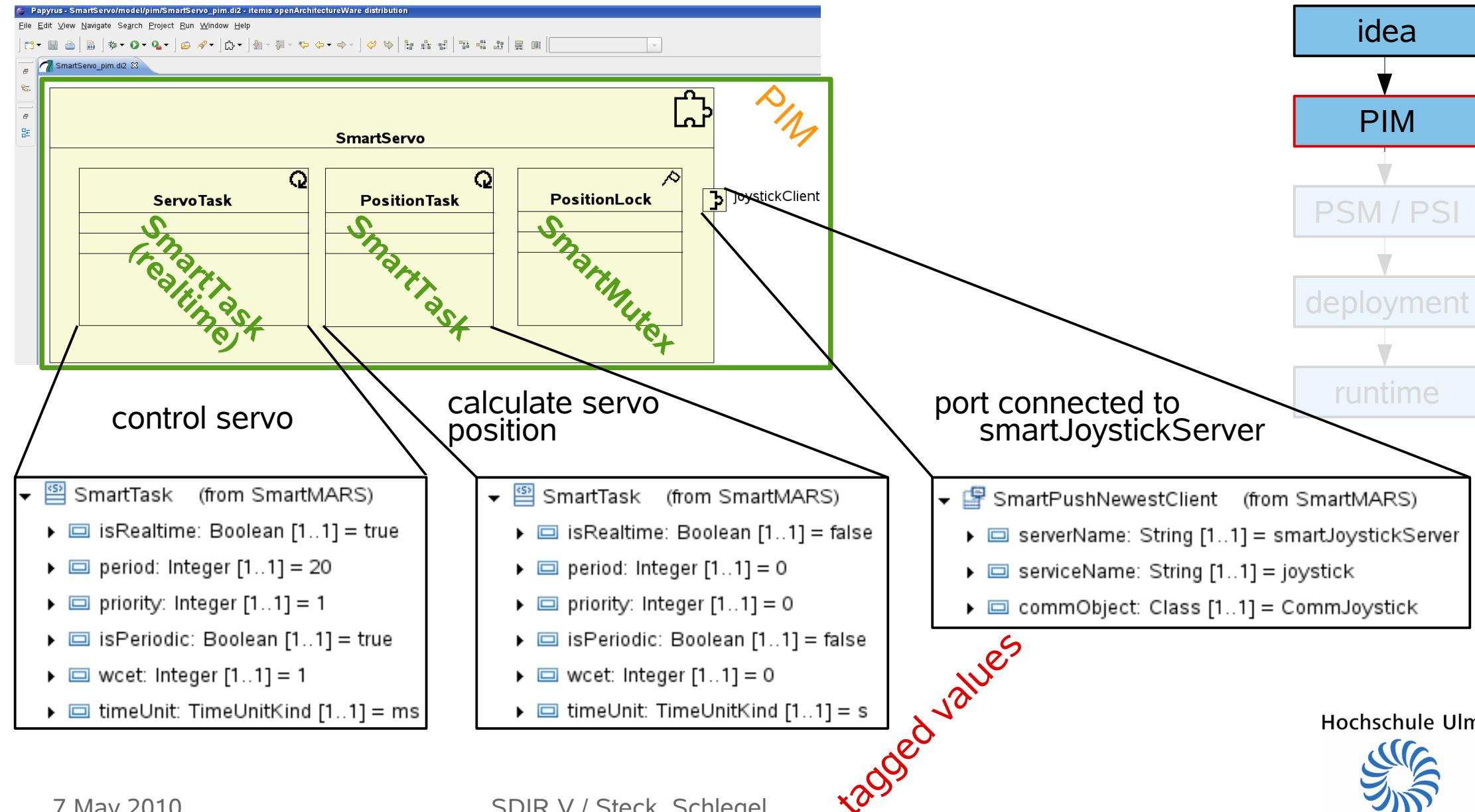




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component PIM

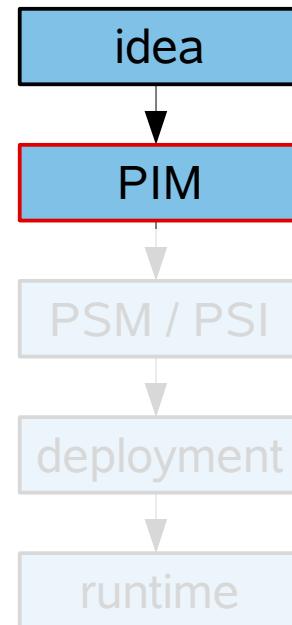
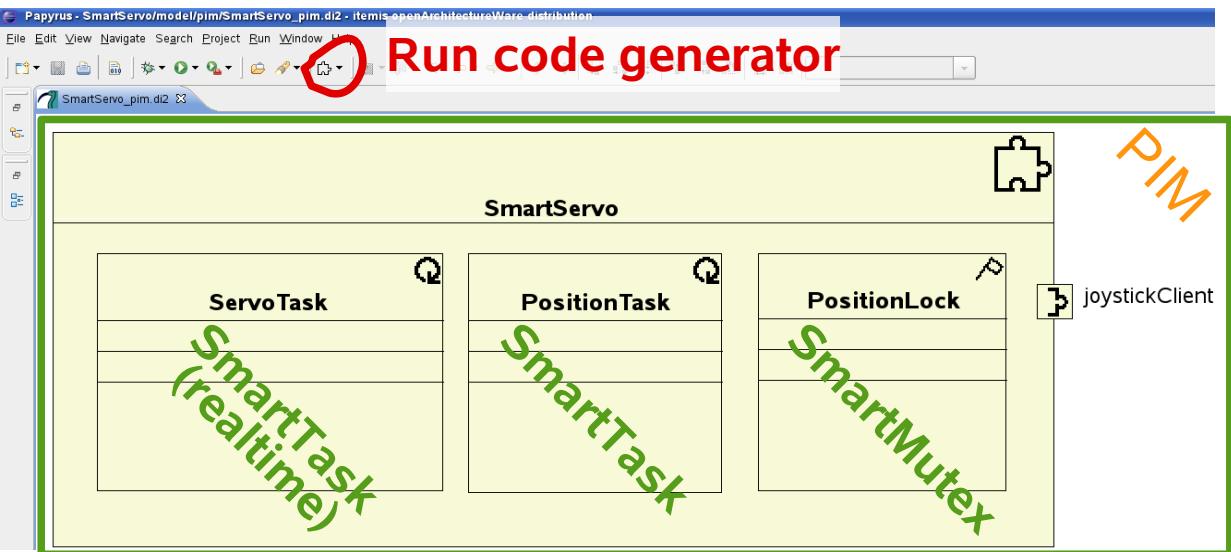




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
generate PSI

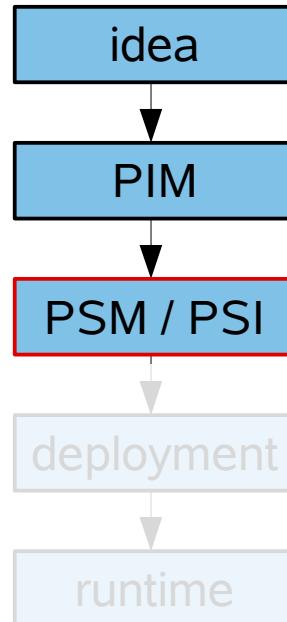
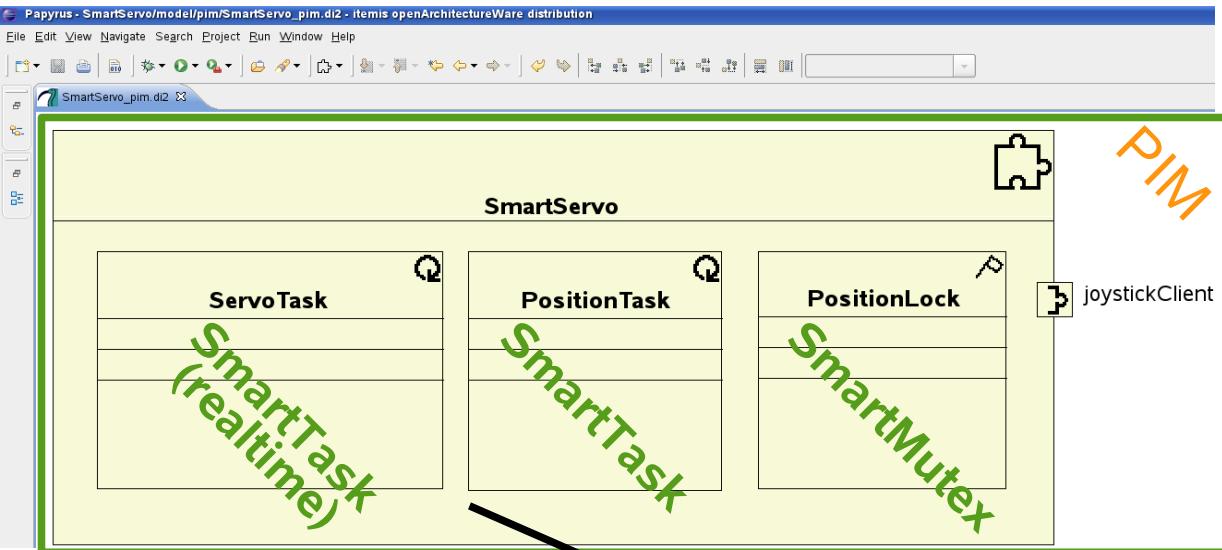




SmartSoft MDSD Toolchain

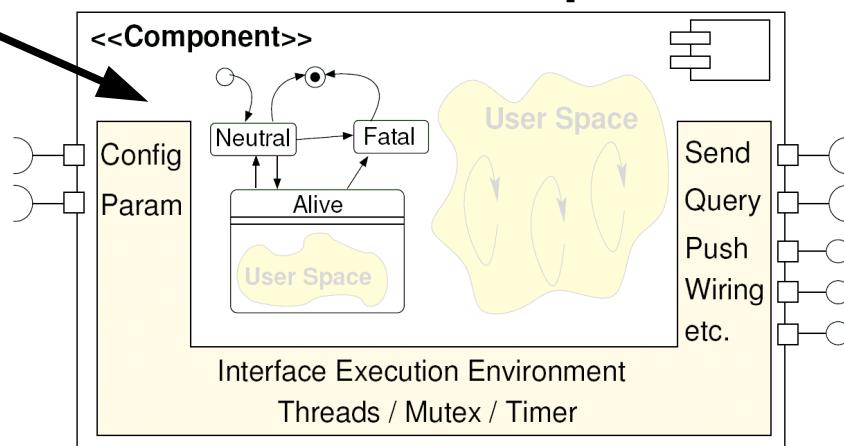
Example 1: Servo (RTAI-Linux)

creating a component
generate PSI



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

executable component

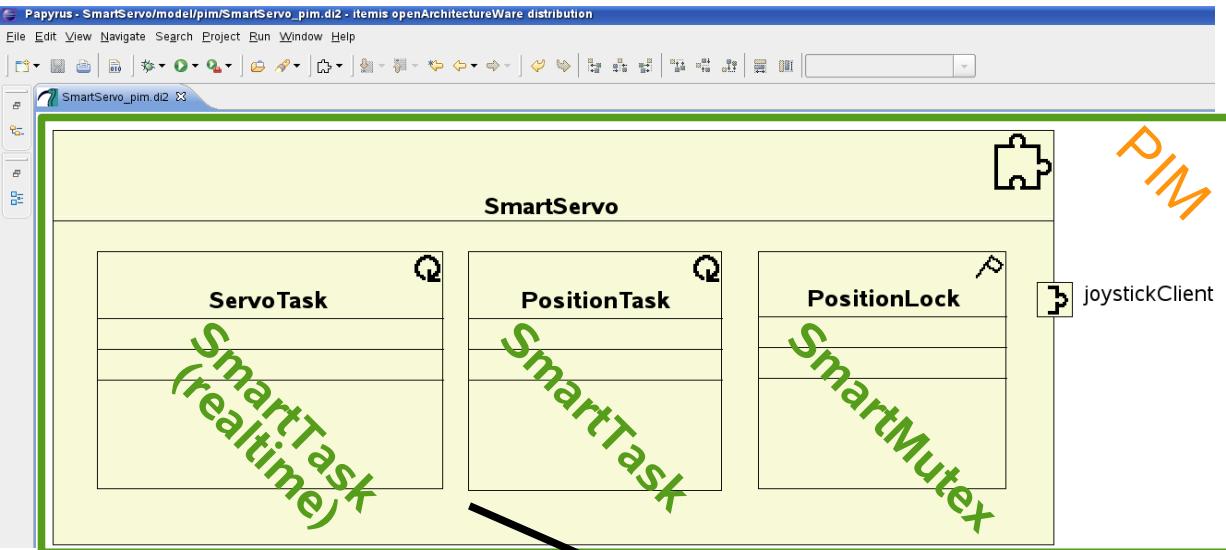




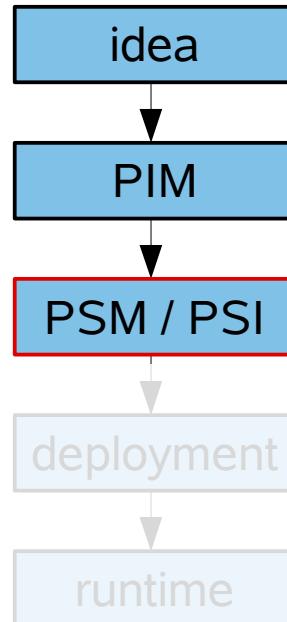
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
integrate user-code

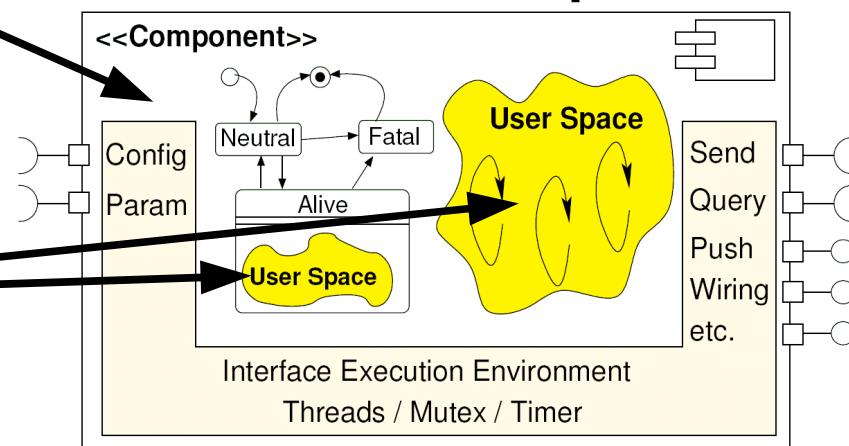
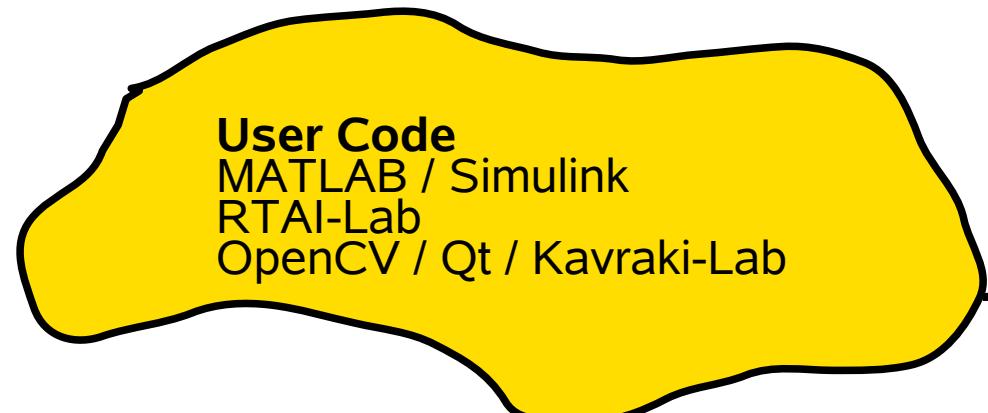


PIM



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

executable component





SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
generate PSM

Papyrus - SmartServo/model/pim/SmartServo_pim.di2 - items openArchitectureWare distribution

File Edit View Navigate Search Project Run Window Help

SmartServo_pim.di2

PIM

SmartServo

ServoTask PositionTask PositionLock

isRealtime = false
isPeriodic = true

isRealtime = false
isPeriodic = false

PositionLock

joystickClient

mapping PIM elements to PSM elements (is done by toolchain)

toolchain generates elements to emulate periodic tasks

PSM

SmartServo_pim.di2 model_psm.uml

platform:/resource/SmartServo/model/psm/model_psm.uml

- <Model> model_psm
 - <><smartCorbaComponent><Component> SmartServo
 - <><smartCorbaPushNewestClient><Port> joystickClient
 - <><smartCorbaTask><Class> ServoTask
 - <><smartCorbaTimer><Class> ServoTaskTimer
 - <><smartCorbaConditionMutex><Class> ServoTaskCondMutex
 - <><smartCorbaMutex><Class> ServoTaskMutex
 - <><smartCorbaTask><Class> PositionTask
 - <><smartCorbaMutex><Class> PositionLock

CorbaSmartSoft

pathmap://CORBASMARTSOFT_PROFILE/CorbaSmartSoft.profile.uml

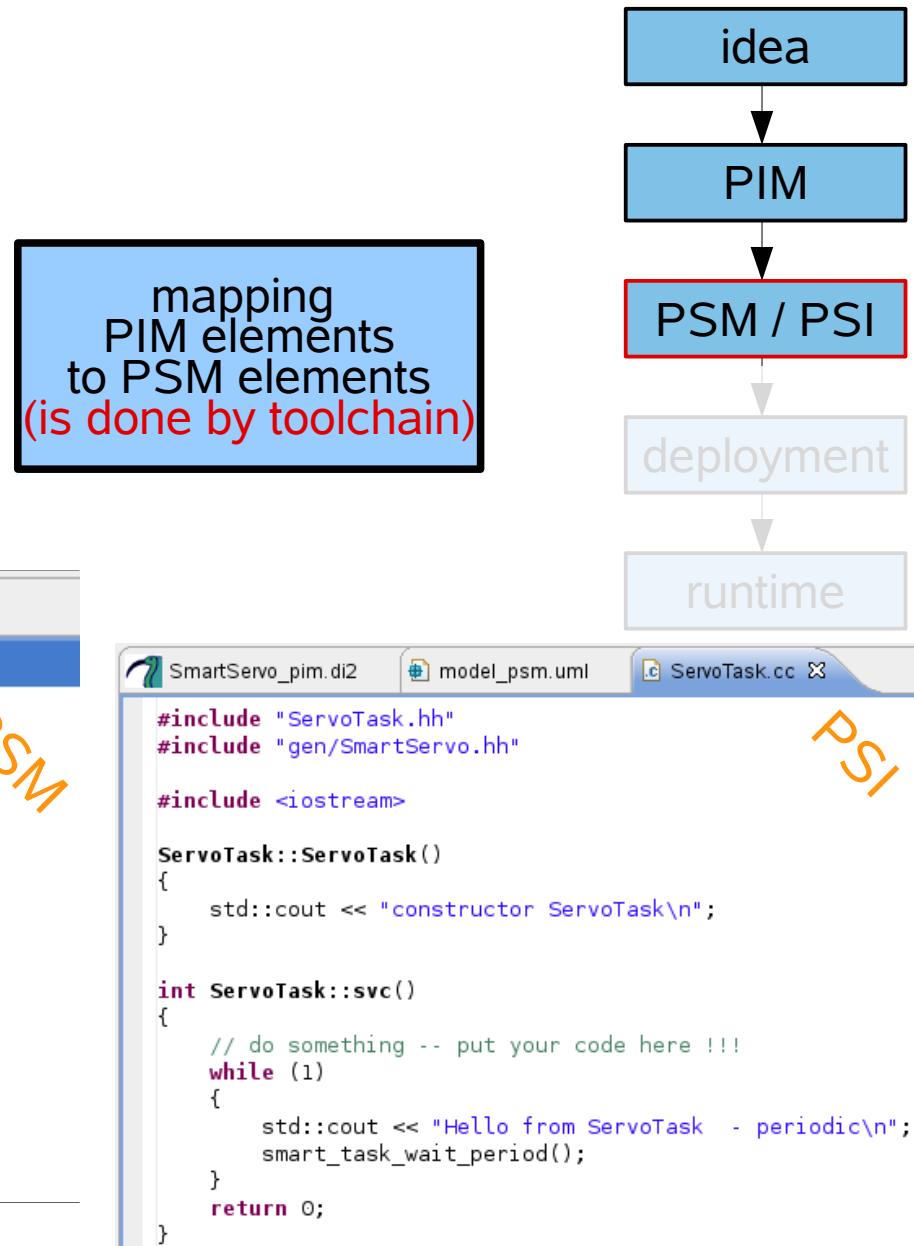
PSI

```
#include "ServoTask.hh"
#include "gen/SmartServo.hh"

#include <iostream>

ServoTask::ServoTask()
{
    std::cout << "constructor ServoTask\n";
}

int ServoTask::svc()
{
    // do something -- put your code here !!!
    while (1)
    {
        std::cout << "Hello from ServoTask - periodic\n";
        smart_task_wait_period();
    }
    return 0;
}
```

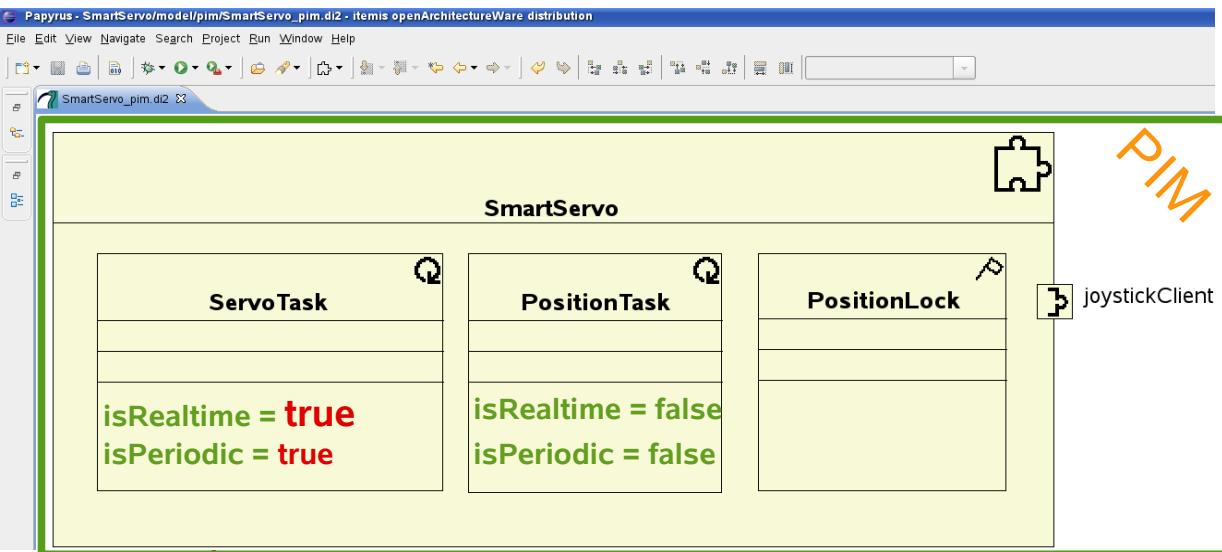




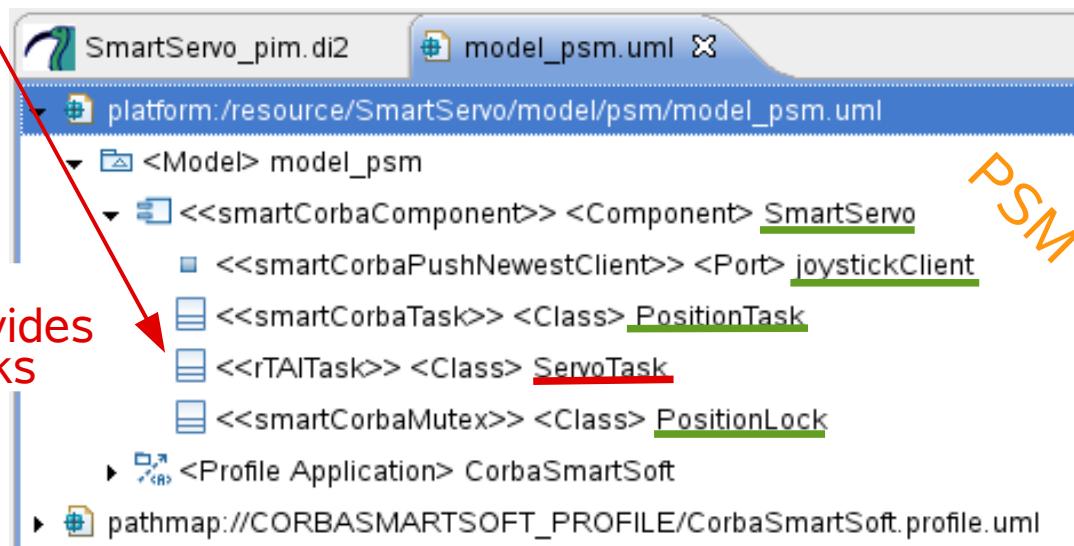
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
generate PSM



mapping PIM elements to PSM elements (is done by toolchain)



```
#include "ServoTask.hh"
#include "gen/SmartServo.hh"

#include <iostream>

ServoTask::ServoTask()
{
    std::cout << "constructor ServoTask\n";
}

int ServoTask::svc()
{
    // do something -- put your code here !!!
    while (1)
    {
        std::cout << "Hello from ServoTask - periodic\n";
        smart_task_wait_period();
    }
    return 0;
}
```

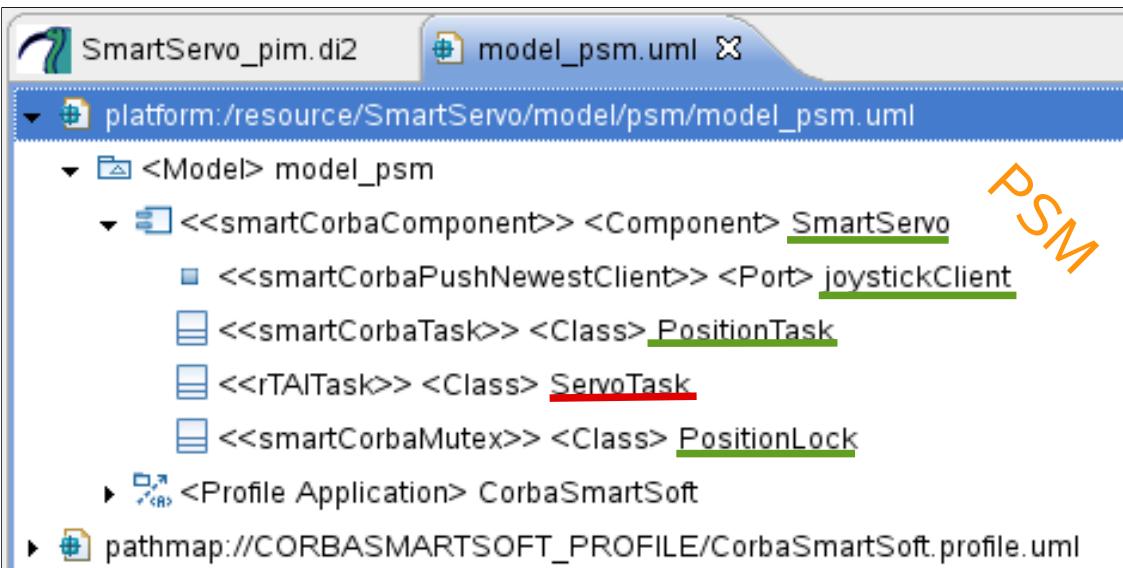
PSI



SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
generate PSI



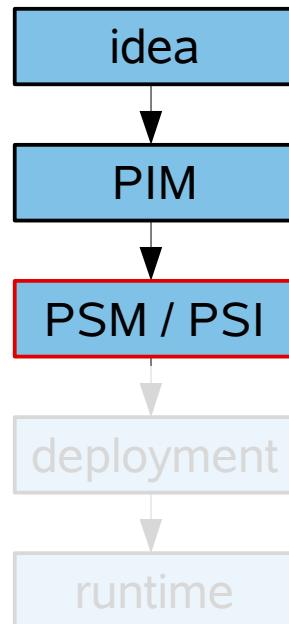
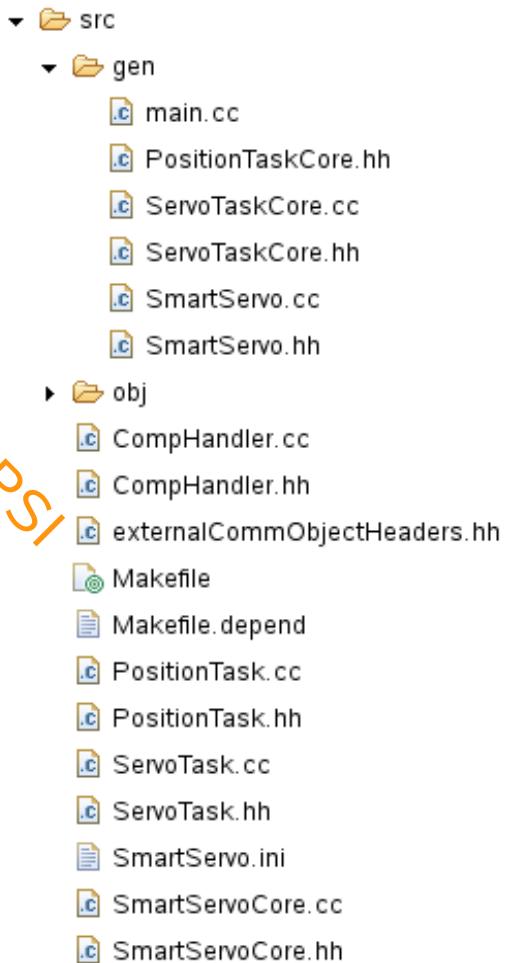
generate PSI
out of
PSM elements
(is done by toolchain)

```
// include communication objects
#include "externalCommObjectHeaders.hh"

void CompHandler::onStartup()
{
    std::cout
        << "startup - put your startupCode

    CHS::StatusCode status;

    // connect to all services
    std::cout << "connecting to: " << COMP->ini
        << ";" << COMP->ini.joystickClient;
    status = COMP->joystickClient->connect(COM
        COMP->ini.joystickClient.serviceName);
    while (status != CHS::SMART_OK)
    {
        usleep(500000);
        status = COMP->joystickClient->connect
            COMP->ini.joystickClient.serv
            COMP->ini.joystickClient.servicename);
    }
    std::cout << "connected.\n";
    COMP->joystickClient->subscribe();
}
```

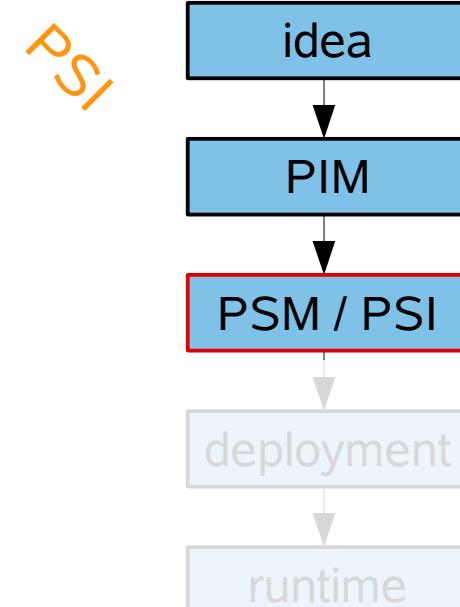




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a
component
integrate
user-code



```

SmartServo_pim.di2
model_psm.uml
PositionTask.cc
ServoTask.cc
SmartServoCore.cc

#include "SmartServoCore.hh"

// constructor
SmartServoCore::SmartServoCore()
{
    std::cout << "constructor SmartServoCore\n";
    servoOnTime = 1500000; code added by developer
}

.h SmartServoCore.hh

#ifndef _SMARTSERVOCORE_HH
#define _SMARTSERVOCORE_HH

#include <iostream>

class SmartServoCore
{
private:

public:
    SmartServoCore();
    int servoOnTime; code added by developer
};

#endif

```





SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
integrate user-code

PSI

```

SmartServo_pim.di2      model_psm.uml      PositionTask.cc X      ServoTask.cc
#include "PositionTask.hh"
#include "gen/SmartServo.hh"

#include <iostream>

PositionTask::PositionTask()
{
    std::cout << "constructor PositionTask\n";
}

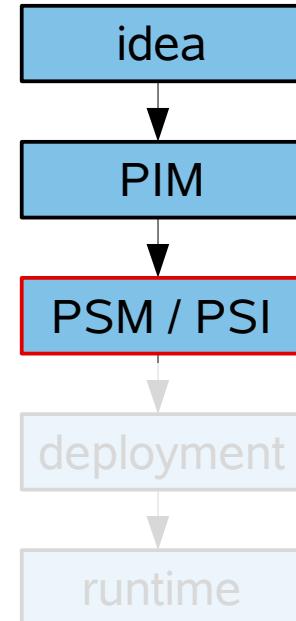
int PositionTask::svc()
{
    // do something -- put your code here !!!
    while (1)
    {
        CHS::StatusCode status;
        Smart::CommJoystick cmd;
        status = COMP->joystickClient->getUpdateWait(cmd);
        if(status == CHS::SMART_OK)
        {
            double x = cmd.get_x();
            COMP->PositionLock.acquire();
            COMP->servoOnTime = 1500000 + (x * 500000);
            std::cout << "servoOnTime: " << COMP->servoOnTime << std::endl;
            COMP->PositionLock.release();
        }
    }
    return 0;
}

```

wait for position update (joystickServer)

code added by developer

calculate servo position (servoOnTime)

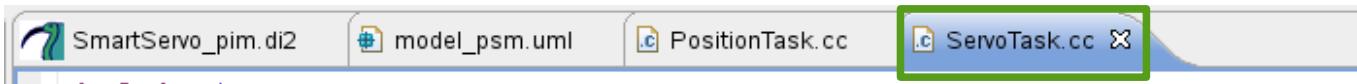




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a
component
integrate
user-code



```

SmartServo_pim.dii    model_psm.uml    PositionTask.cc    ServoTask.cc x

```

```

#include <iostream>
#include <sys/io.h>

#define LPT_PORT 0x378

ServoTask::ServoTask()
{
    std::cout << "constructor ServoTask\n";
}

int ServoTask::svc()
{
    RTIME on_time;
    // servo position
    on_time = nano2count(1500000);

    while (1)
    {
        // ontime
        COMP->PositionLock.acquire();
        on_time = nano2count(COMP->servoOnTime);
        COMP->PositionLock.release();

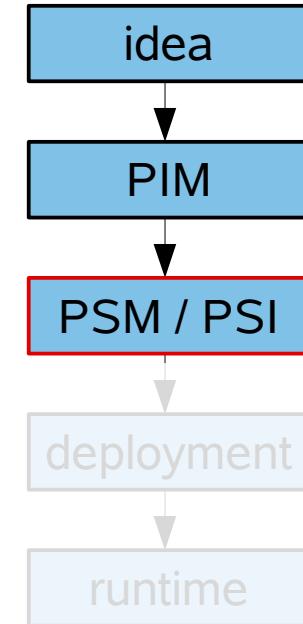
        // switch on
        outb(0xff, LPT_PORT);
        rt_sleep(on_time);

        // switch off
        outb(0x00, LPT_PORT);
        smart_task_wait_period();
    }
    return 0;
}

```

A red curly brace on the right side of the code highlights the section from the constructor to the final closing brace of the while loop. The text "code added by developer" is written in red next to the brace.

PSI

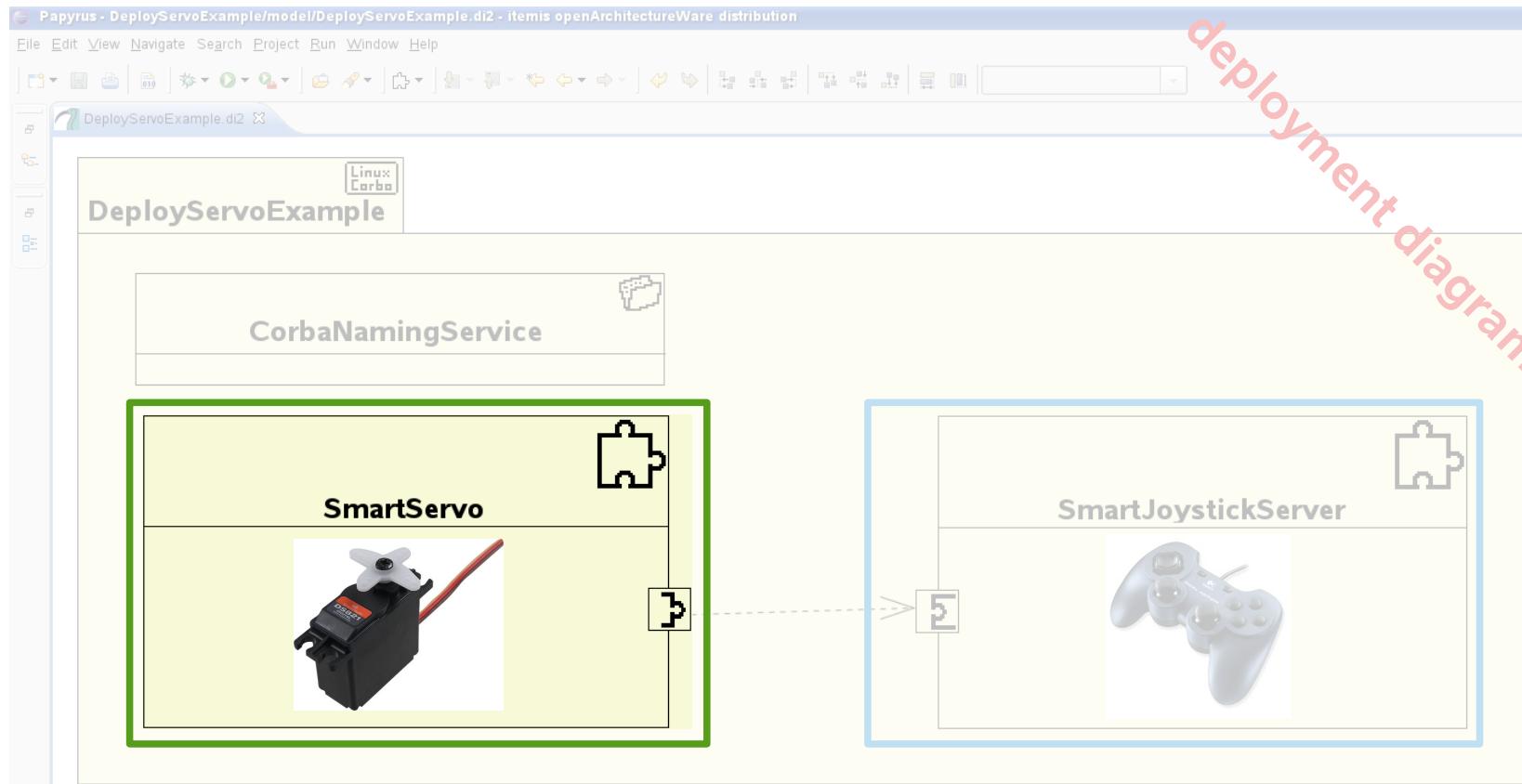




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

servo example
overview



SmartServo component
- created

SmartJoystickServer component
- preexisting component (COTS)

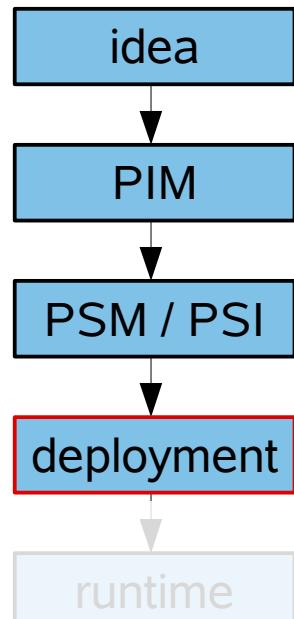
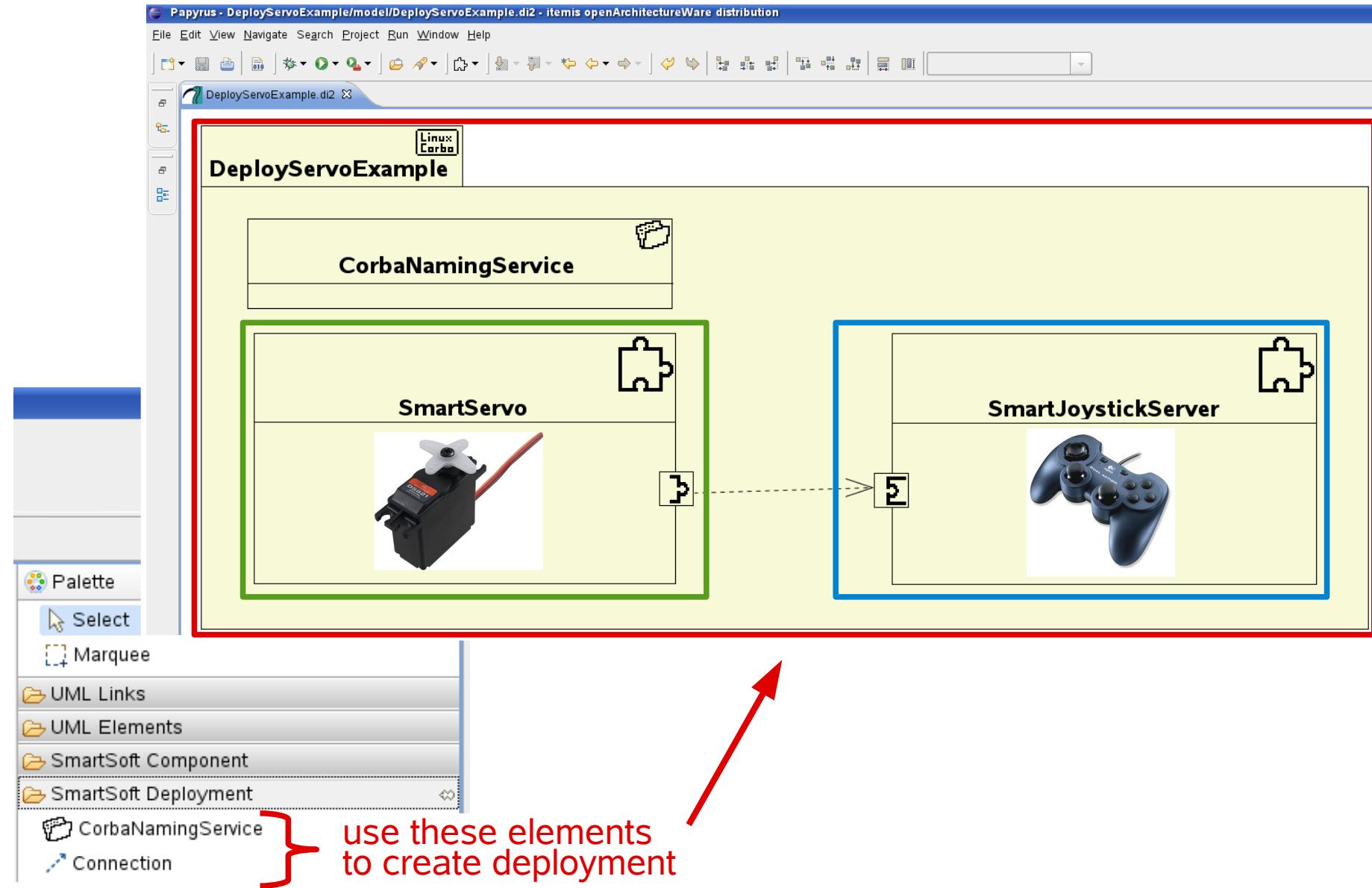




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

create
deployment

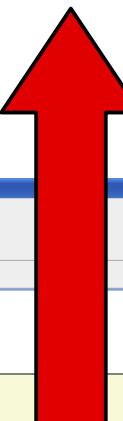
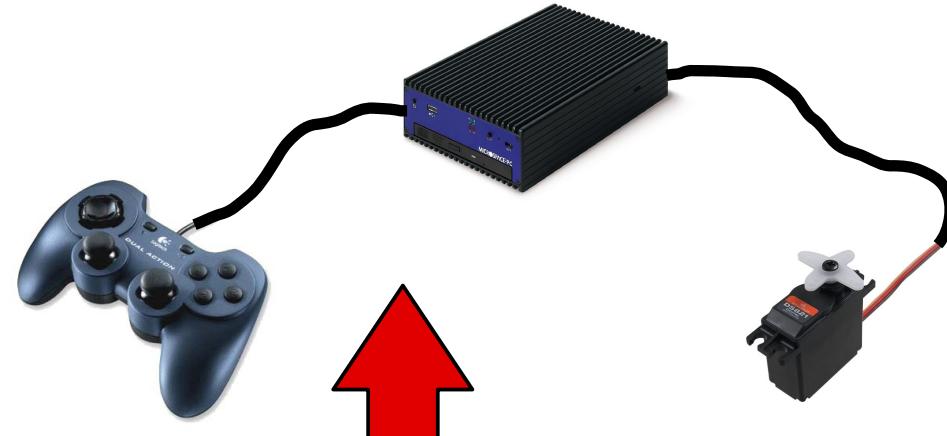
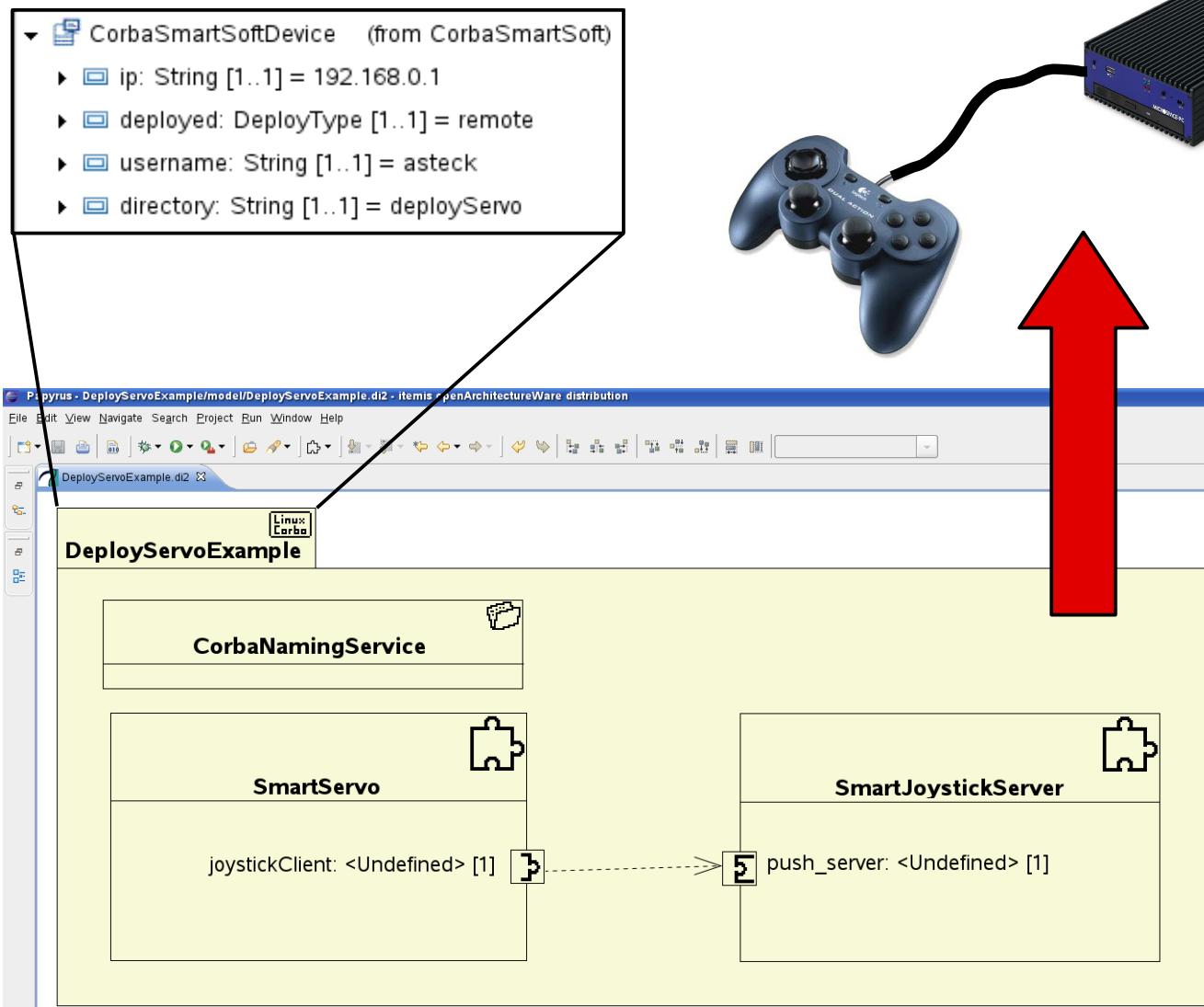




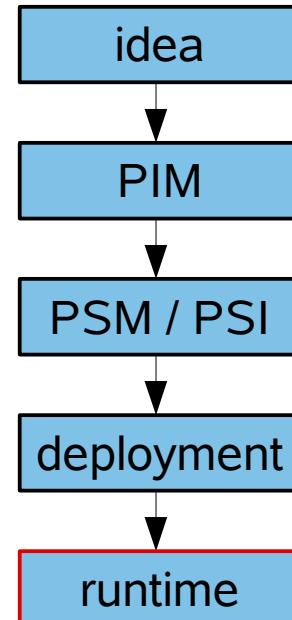
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

create
deployment



deploy to target system
and run startscript

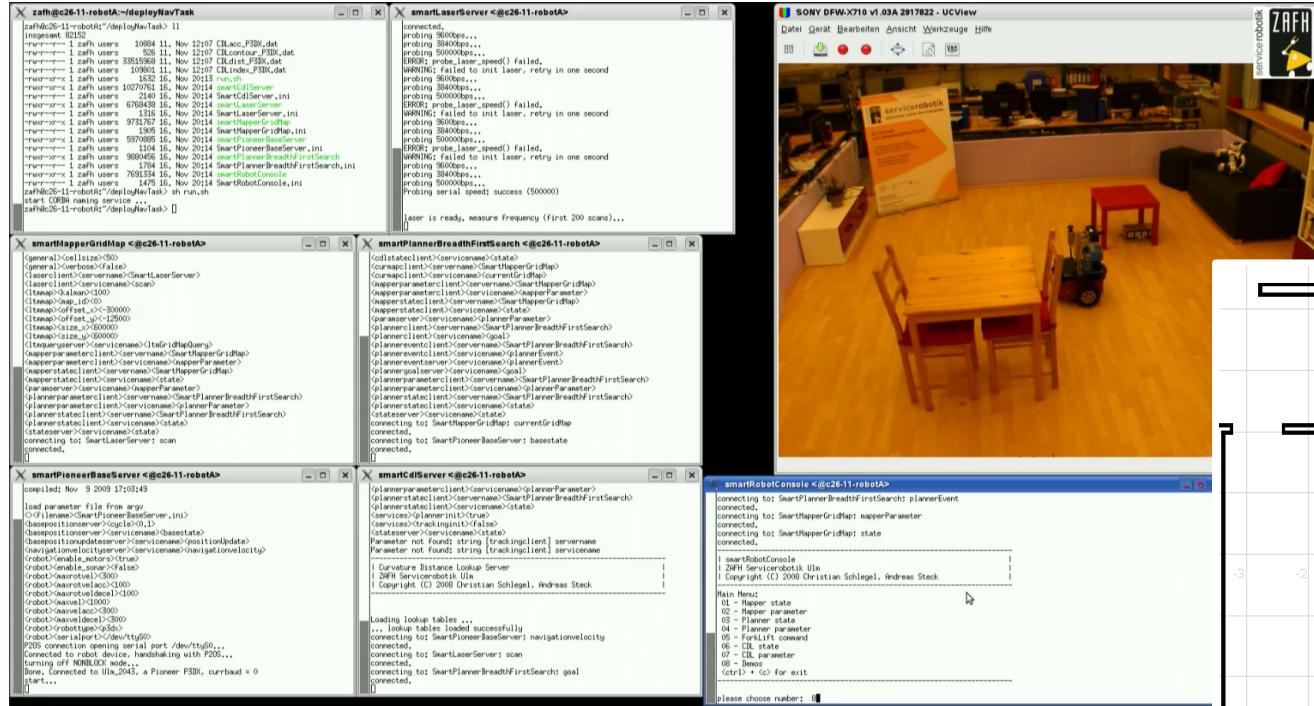




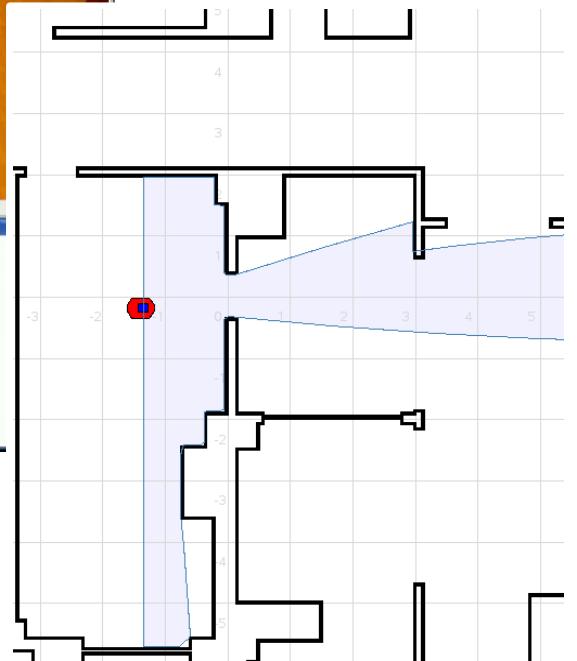
SmartSoft MDSD Toolchain

Example 2: Navigation Task

navigation task
overview



- deployment of COTS components
- simple replacement of components (P3DX/SICK ↔ Player/Stage)
- robot drives to positions given by operator
- collision avoidance, map building, path planning, etc.

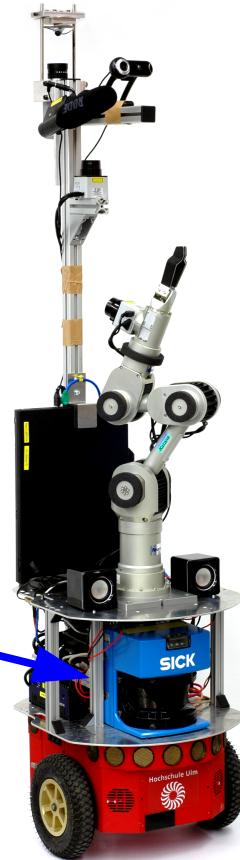
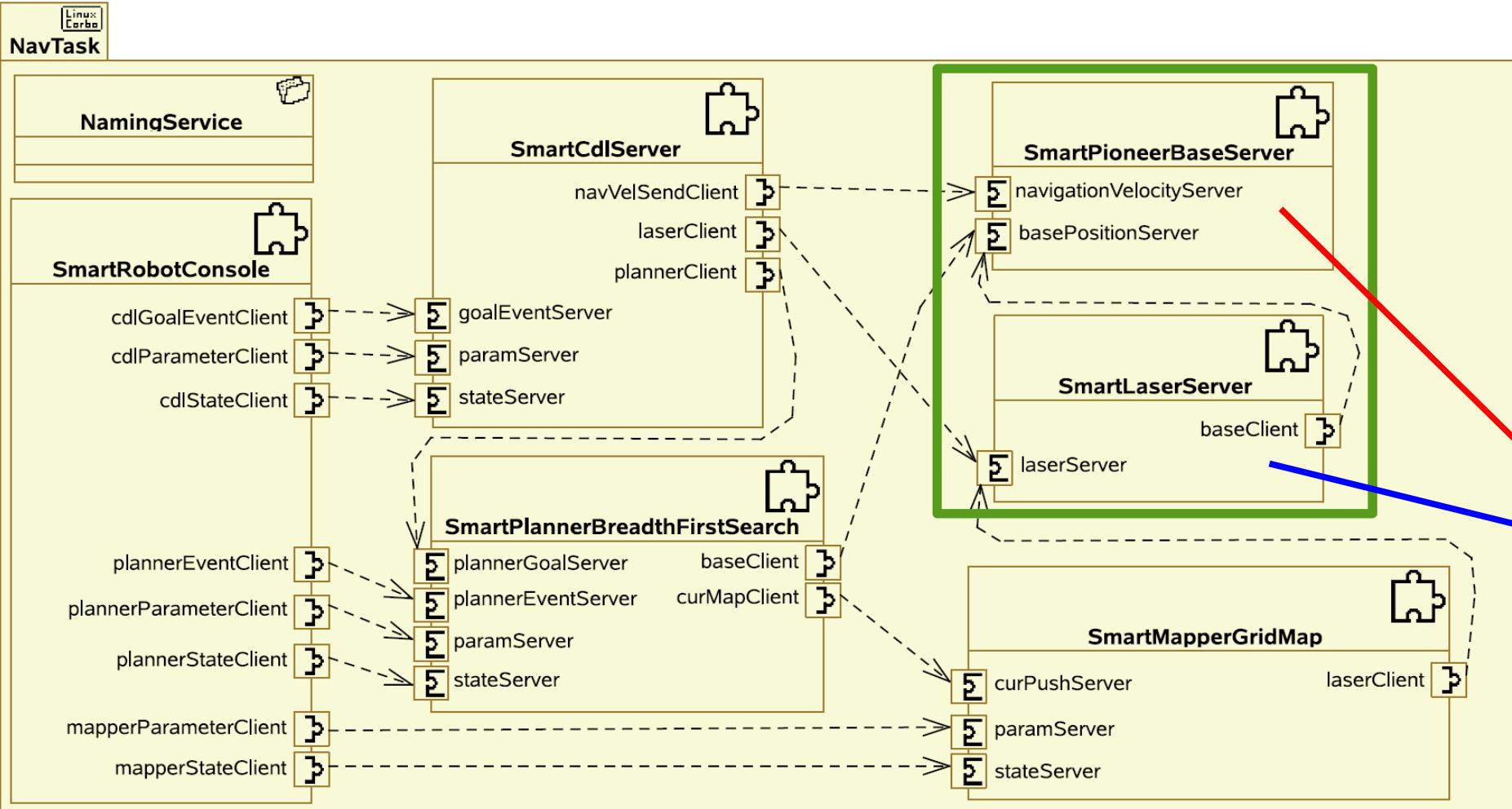




SmartSoft MDSD Toolchain

Example 2: Navigation Task

deployment
navigation task
p3DX/SICK

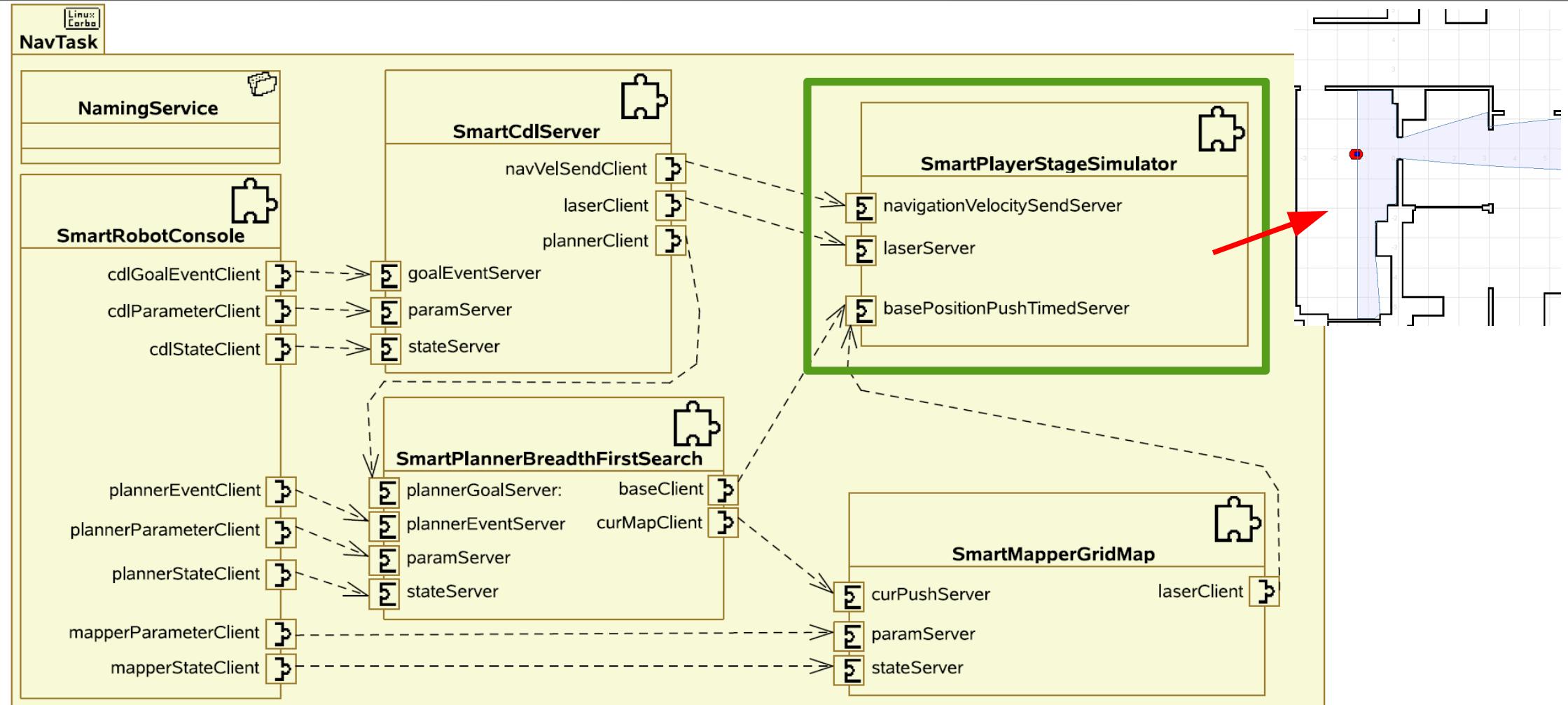




SmartSoft MDSD Toolchain

Example 2: Navigation Task

deployment
navigation task
Player/Stage





SmartSoft MDSD Toolchain

Example 2: Navigation Task

navigation task
overview

Live Demo

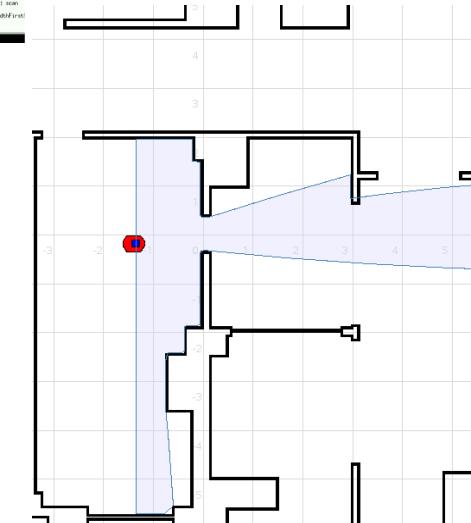
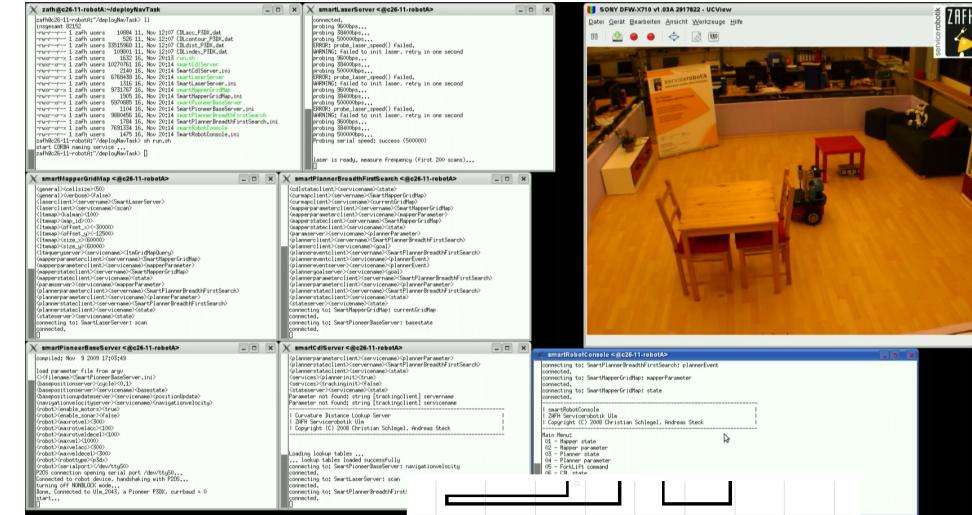
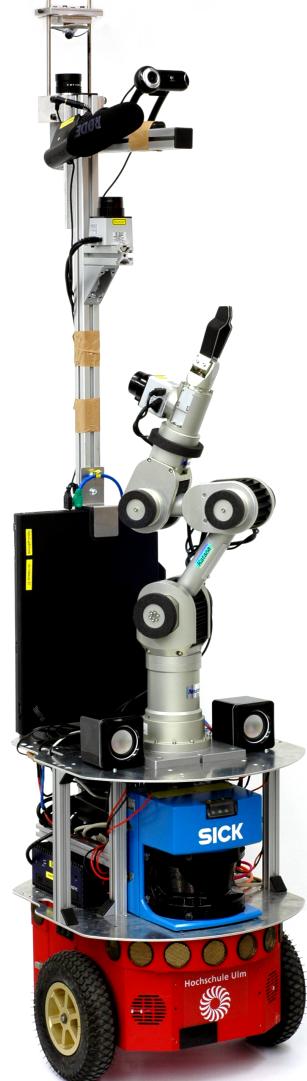
- how to create a deployment in SmartSoft MDSD Toolchain
- define the initial wiring between the components
- add platform specific details
- add target platform settings
- deploy scenario to target
- run scenario in Player/Stage simulator

watch on youtube:

<http://www.youtube.com/watch?v=4dZnDPgw3fA>

attached file:

2009-12-02-deploy-toolchain.mp4



Hochschule Ulm



SmartSoft MDSD Toolchain

Example 3: “Follow Me” - RoboCup@Home

overview



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

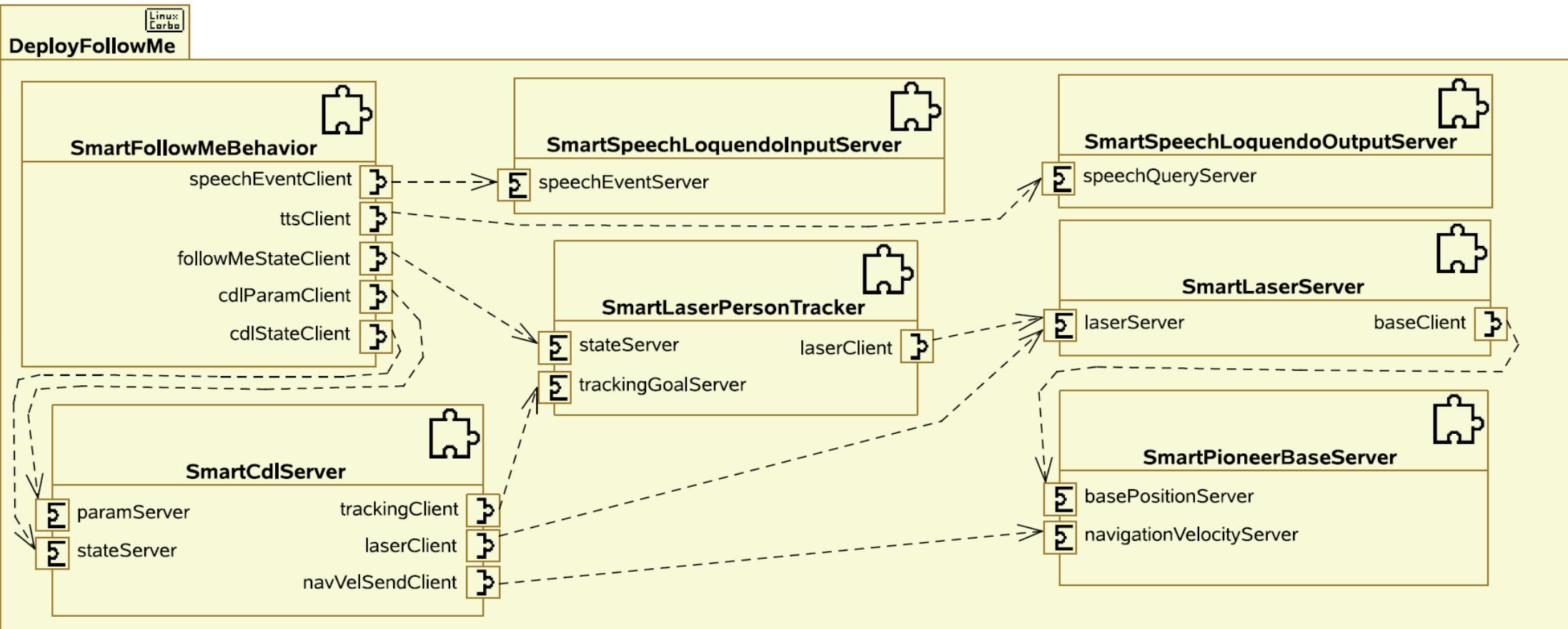




SmartSoft MDSD Toolchain

Example 3: “Follow Me” - RoboCup@Home

deployment





SmartSoft MDSD Toolchain

Example 3: “Follow Me” - RoboCup@Home

overview



watch on youtube:

http://www.youtube.com/watch?v=-xvO_bbrb-I
http://www.youtube.com/watch?v=grf_-32oY0

attached files:

2009-09-10-followMe.mp4
2010-02-25-followMe.mp4

Hochschule Ulm

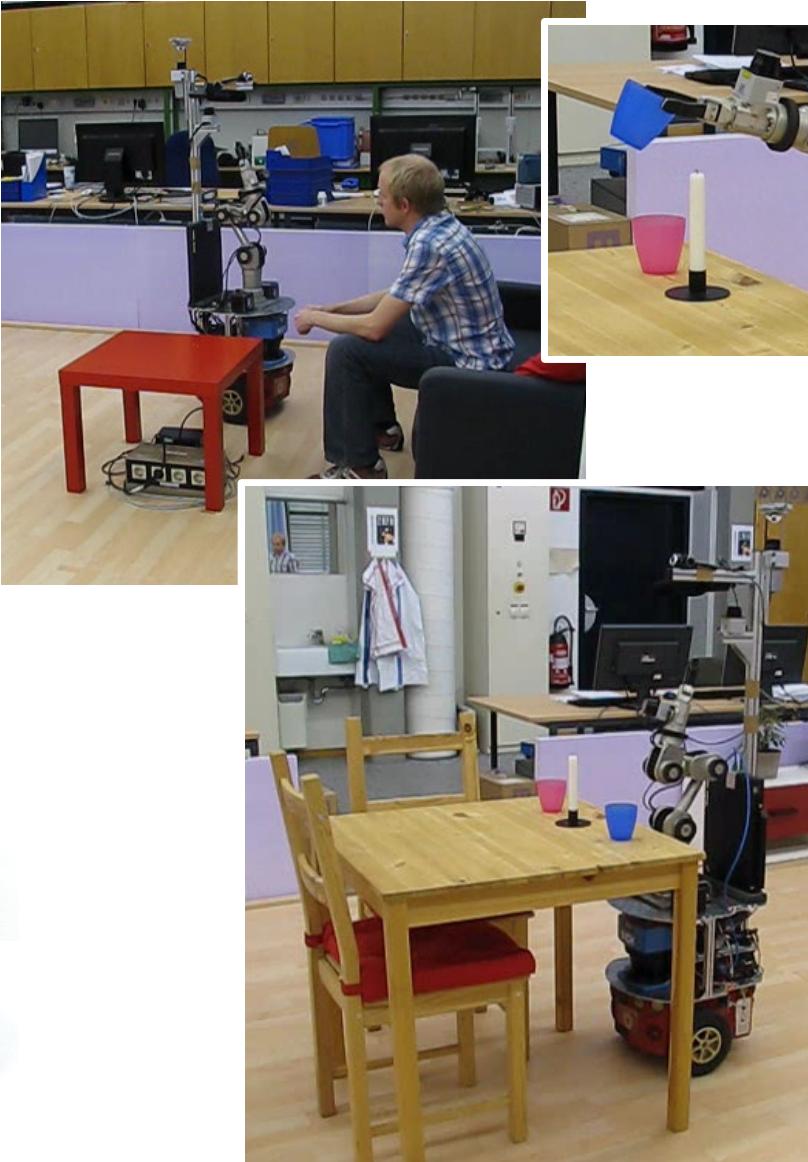




SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario

overview



- deployment of COTS components
- complex real world scenario in everyday environment
- robot cleans up the table

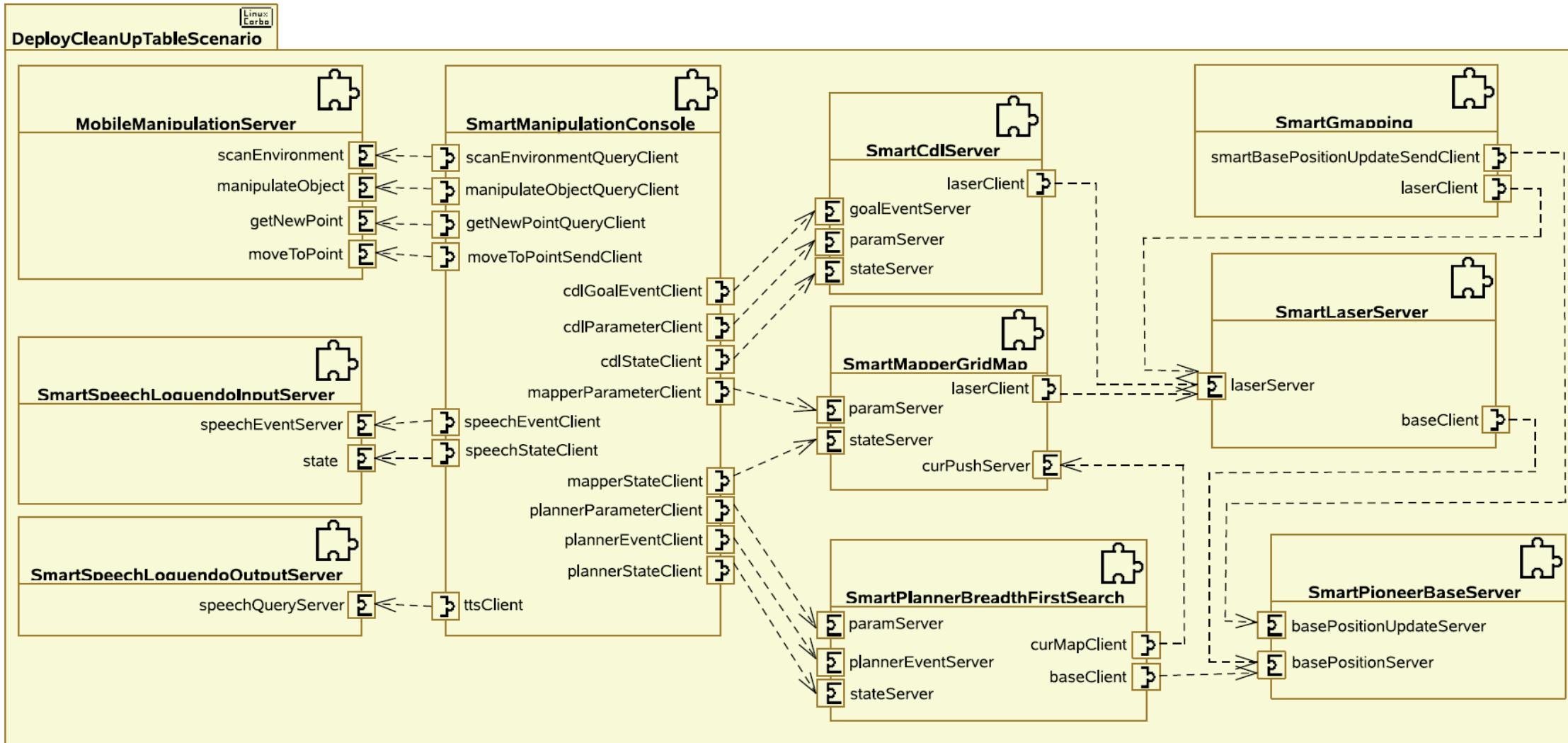




SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario

deployment

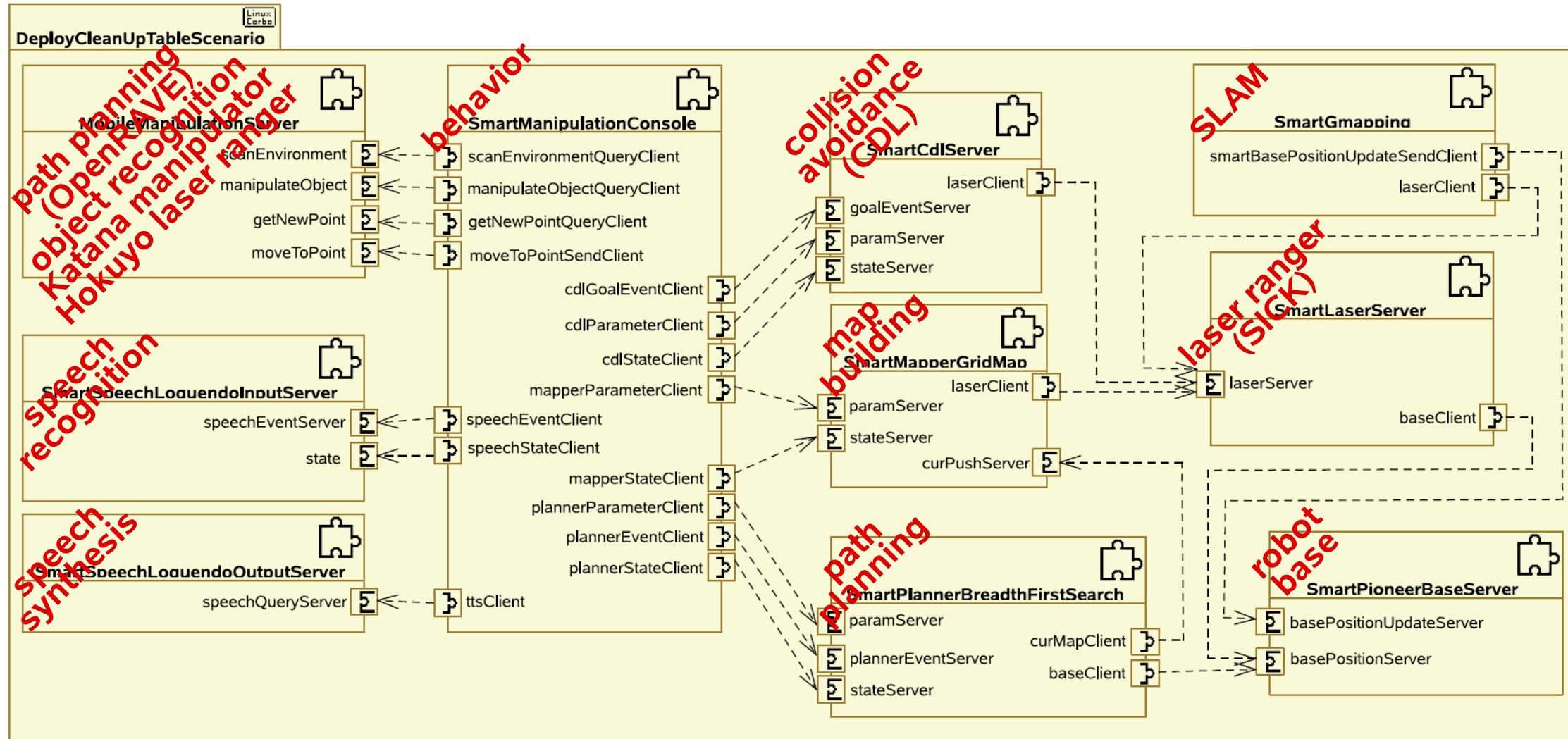




SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario

deployment

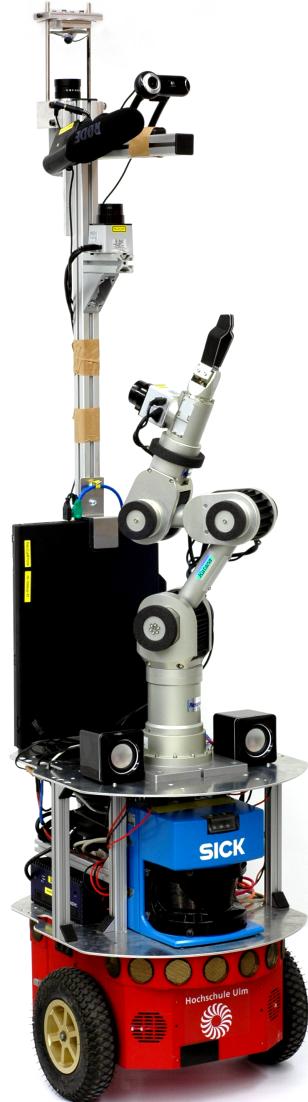




SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario

overview

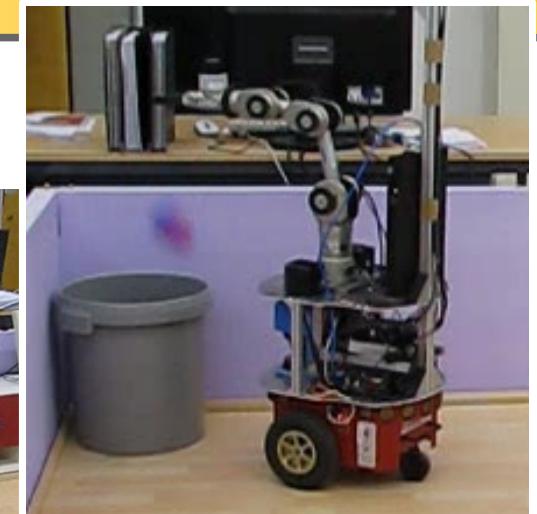


[watch on youtube:](#)

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

[attached file:](#)

2010-02-23-cleanUpTable.mp4





SmartSoft MDSD Toolchain Links

SmartSoft - Mozilla Firefox <2>

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

SmartSoft Components and Toolchain for Robotics

What is SmartSoft?

standardized components whose inter requirements.

YouTube - Kanal von RoboticsAtHsUlm - Mozilla Firefox

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

Robotics@HS-Ulm Abonnieren

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



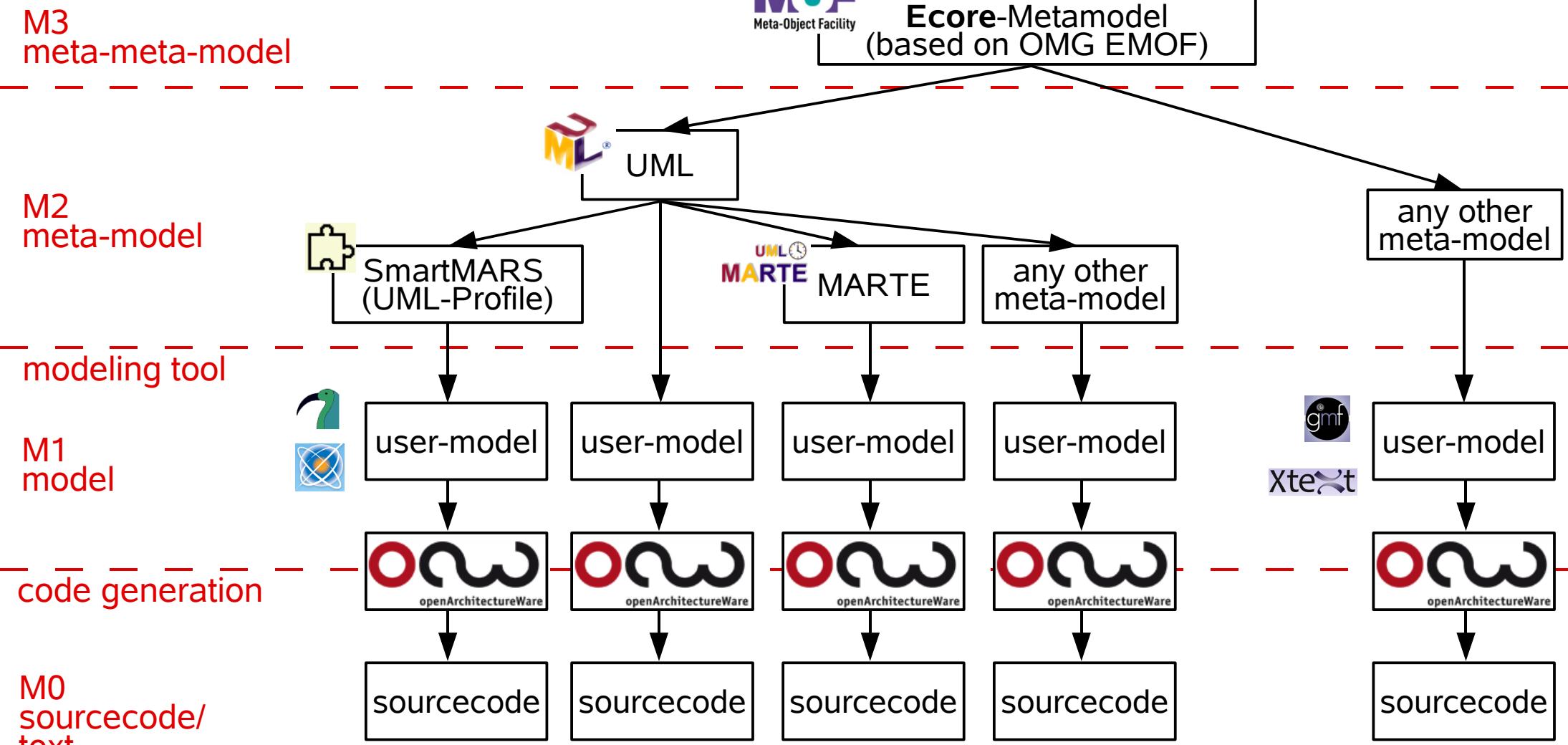


Addendum



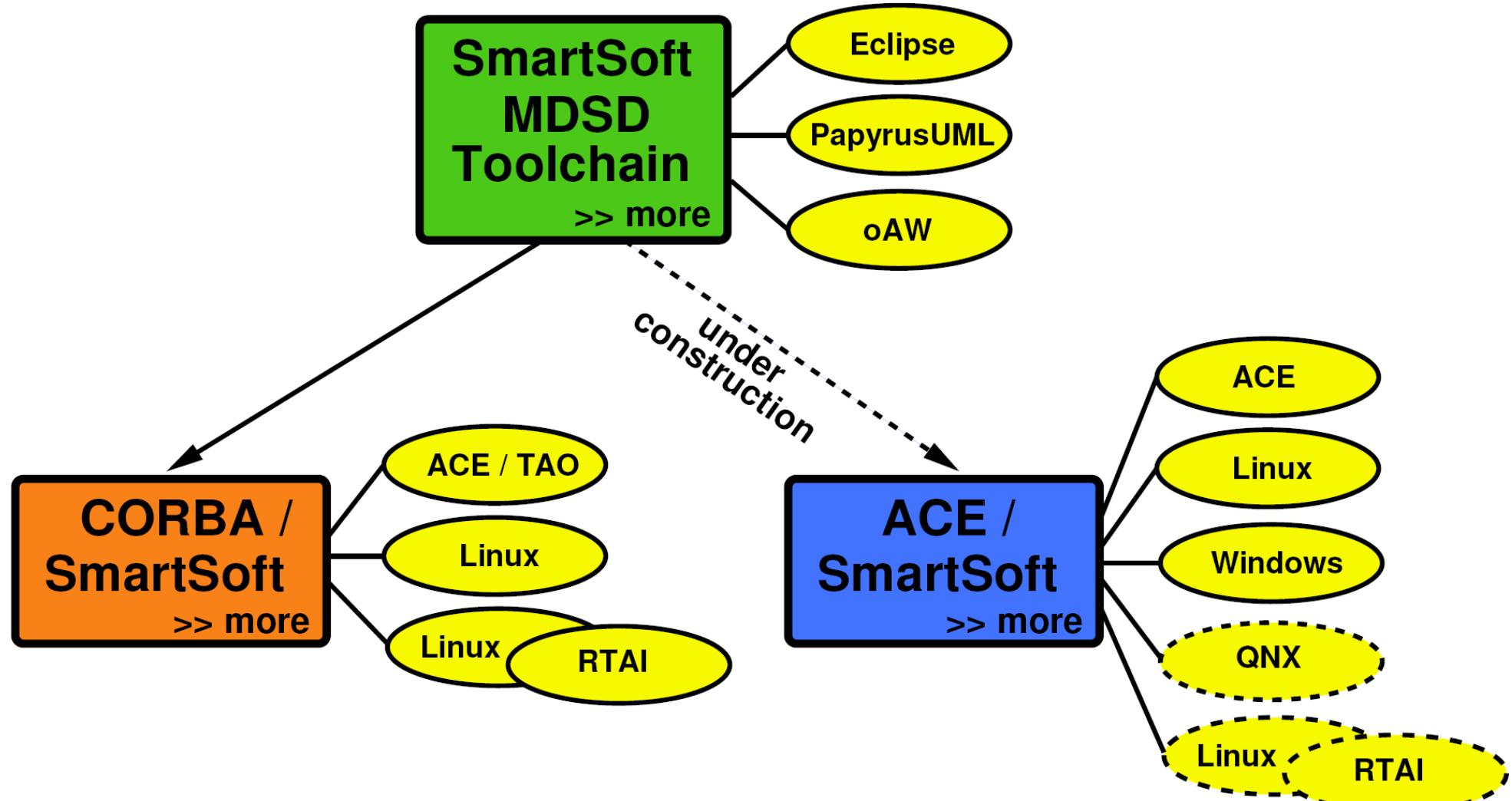


Eclipse Modeling Framework Meta-Modeling





SmartSoft MDSD Toolchain

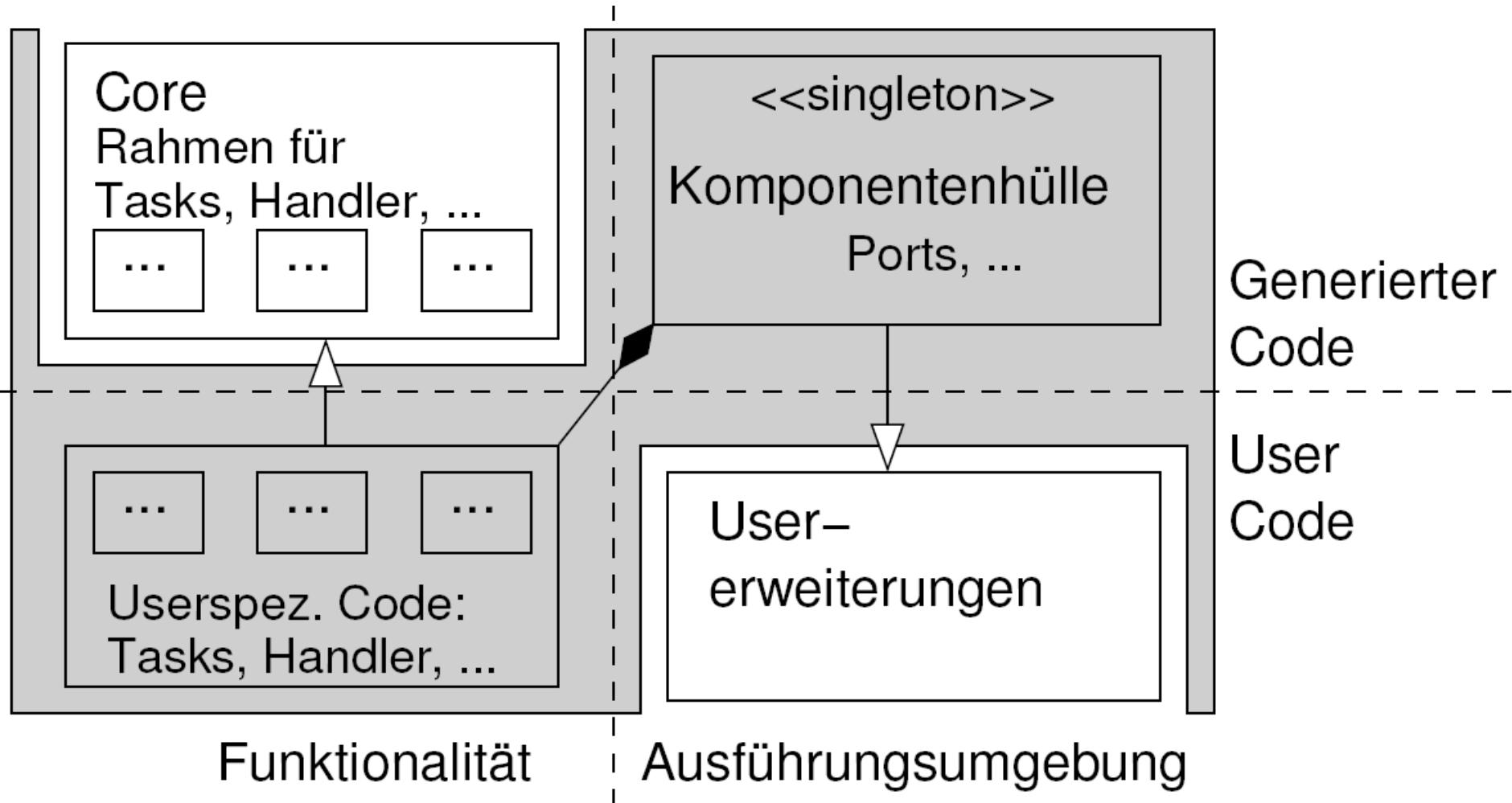


Hochschule Ulm



SmartSoft MDSD Toolchain

Generation Gap Pattern

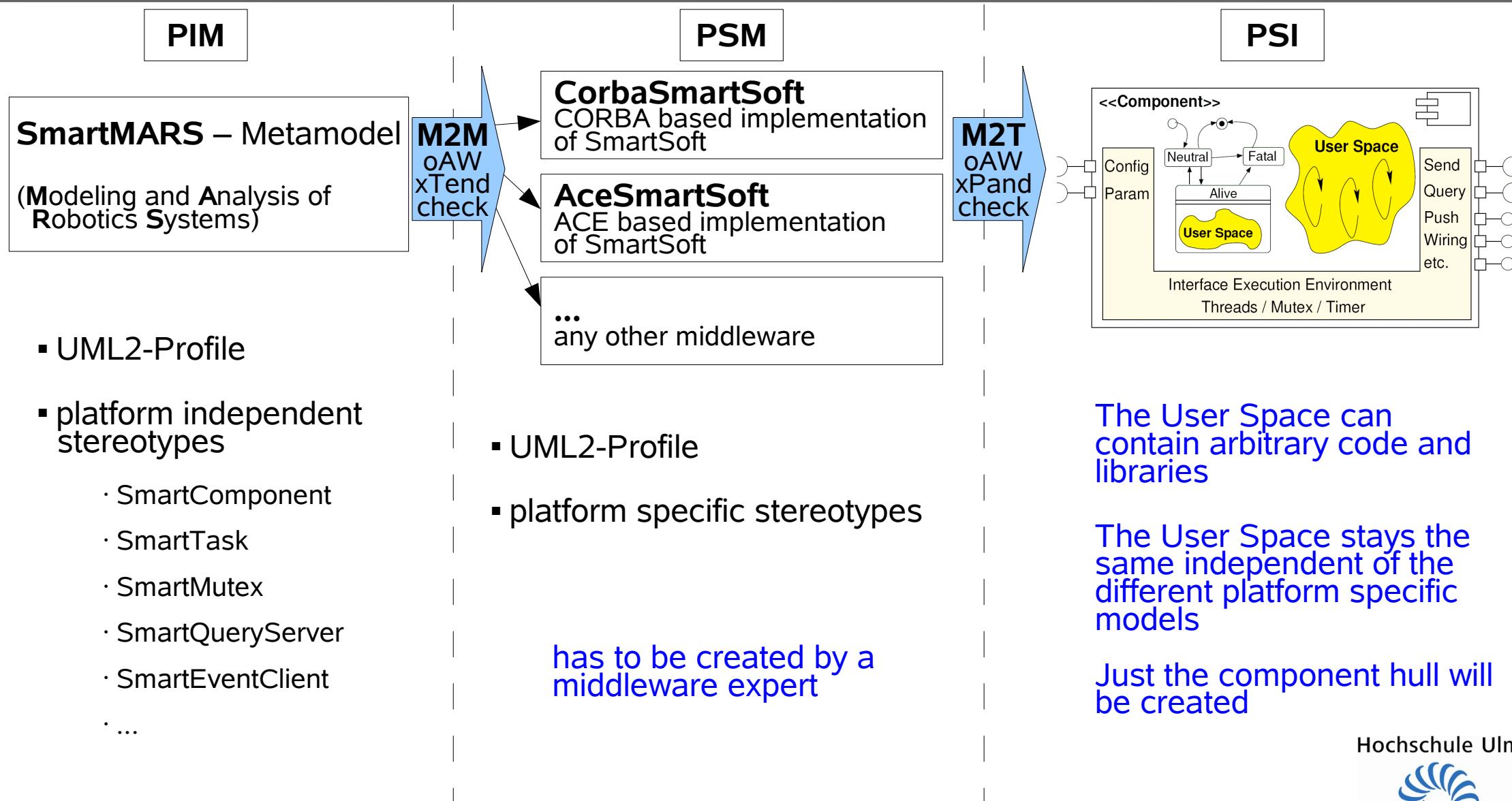




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

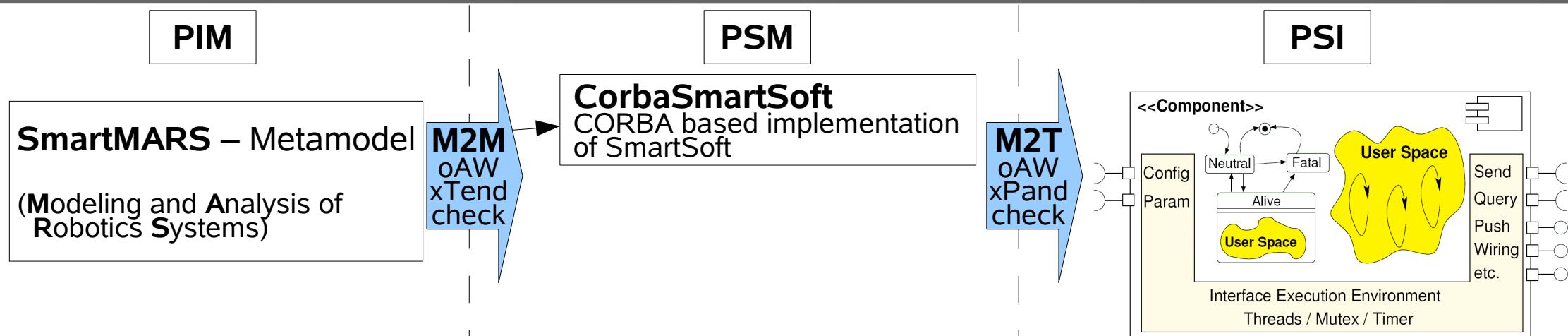
Creating a component workflow





SmartSoft MDSD Toolchain

Creating a component workflow



- 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

Creating a component workflow

