



servicerobotics

Autonomous Mobile Service Robots

SmartSoft MDSD Toolchain

15th November 2010 / SIMPAR Workshop, Darmstadt

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>





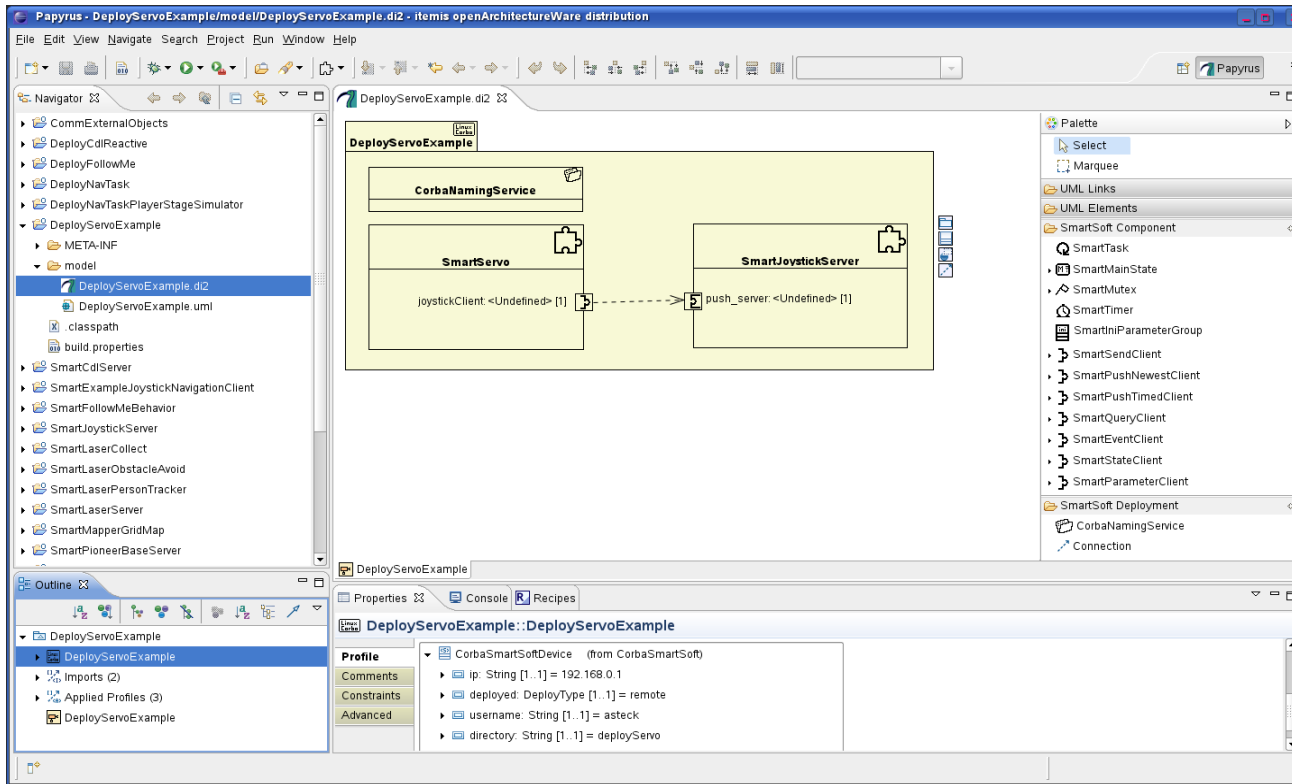
SmartSoft MDSD Toolchain

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





SmartSoft MDSD Toolchain



- Eclipse



- openArchitectureWare

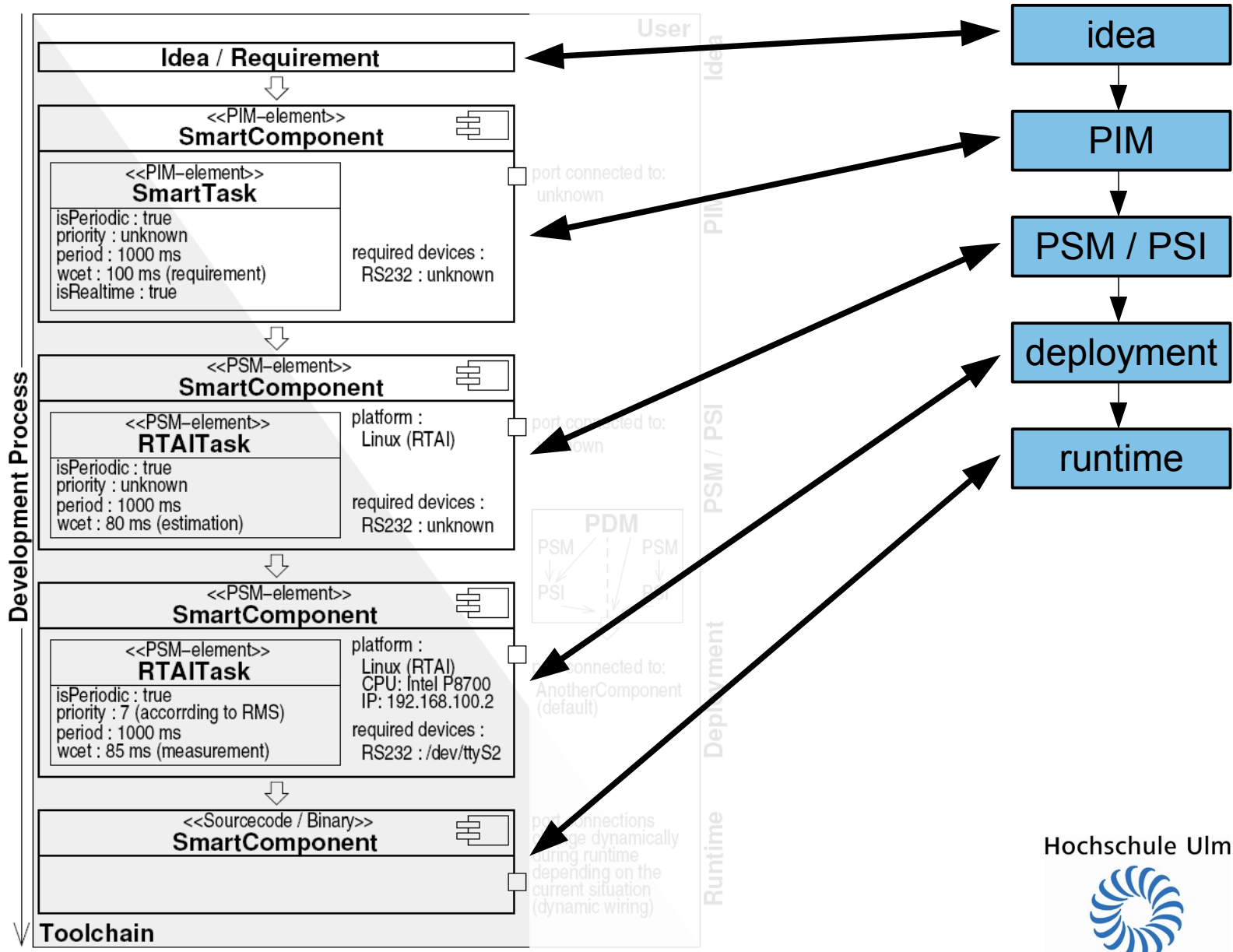


- PapyrusUML



servicerobotics

Autonomous Mobile Service Robots



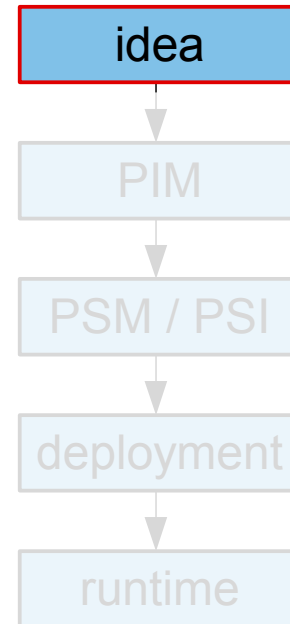
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

overview
servo example



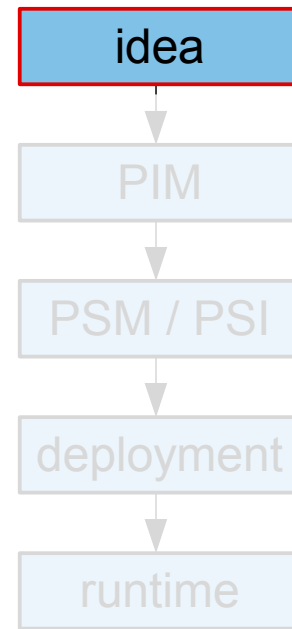
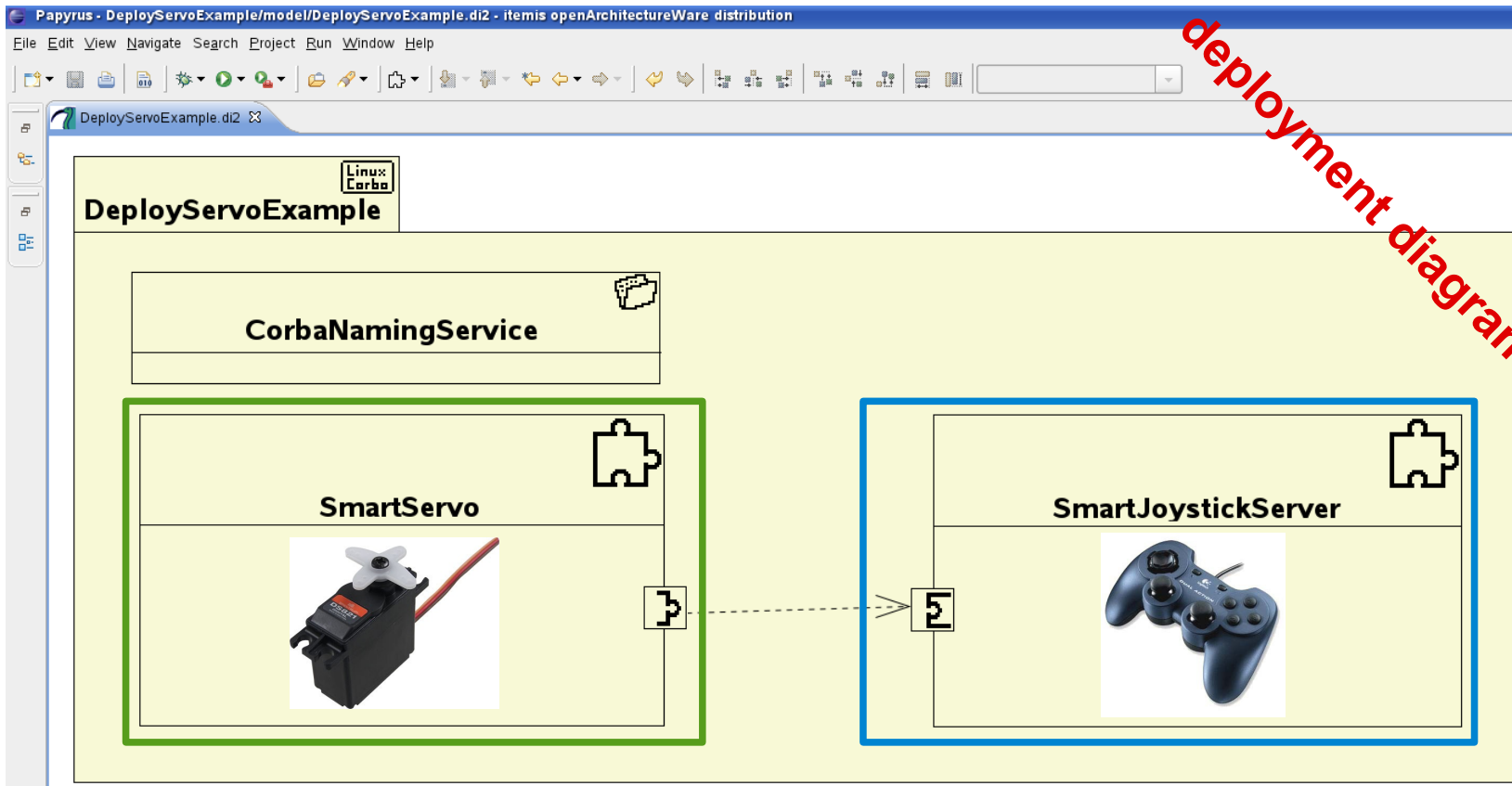
- introduction into overall MDSD process via simple example
- hard realtime in servo component





SmartSoft MDSD Toolchain Example 1: Servo (RTAI-Linux)

overview
servo example



SmartServo component
- will be created in the following demo

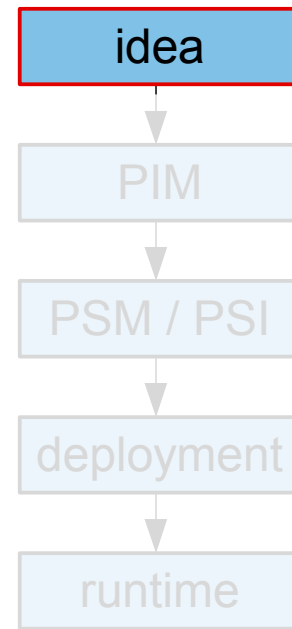
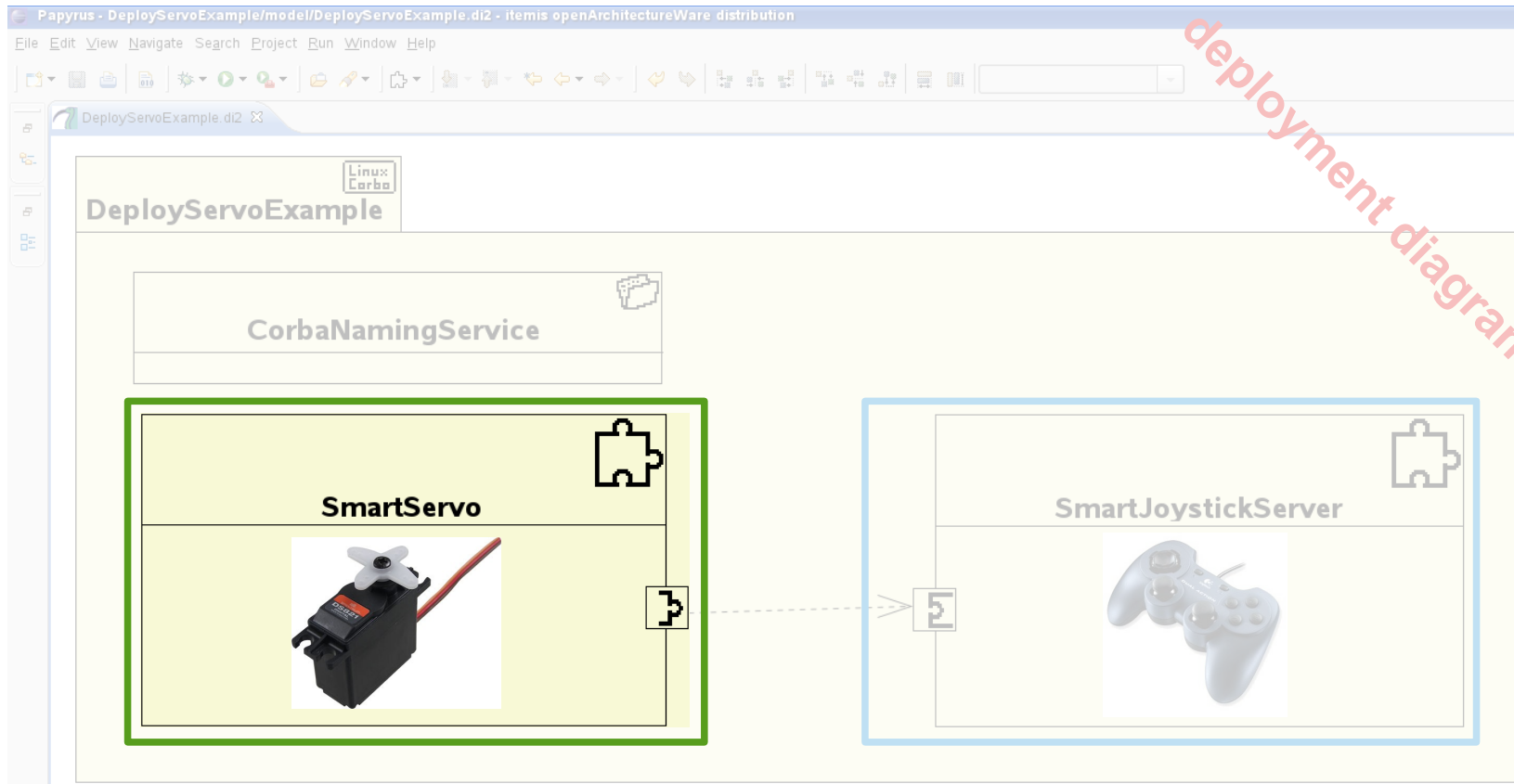
SmartJoystickServer component
- preexisting component (COTS)





SmartSoft MDSD Toolchain Example 1: Servo (RTAI-Linux)

overview
servo example



SmartServo component
- will be created in the following demo

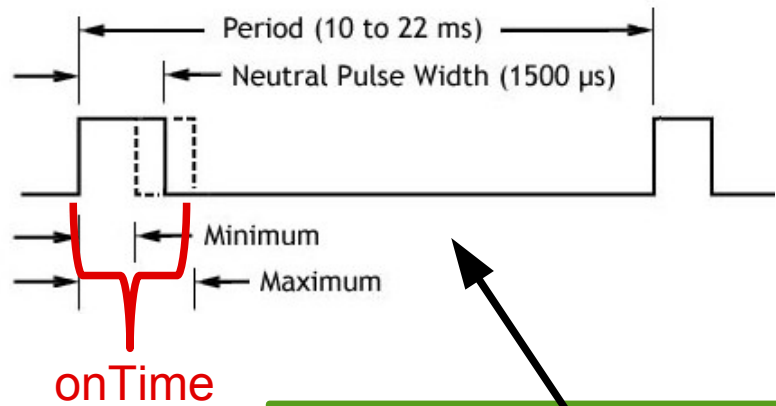
SmartJoystickServer component
- preexisting component (COTS)



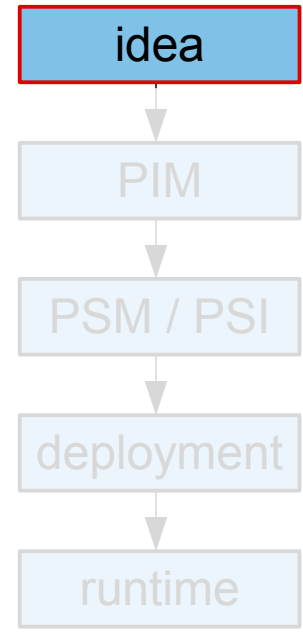
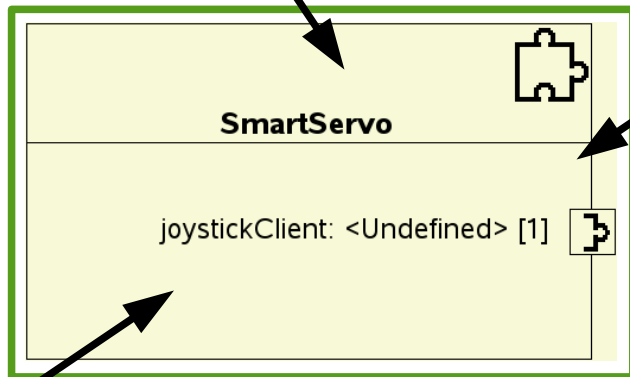


SmartSoft MDSD Toolchain Example 1: Servo (RTAI-Linux)

creating a component idea



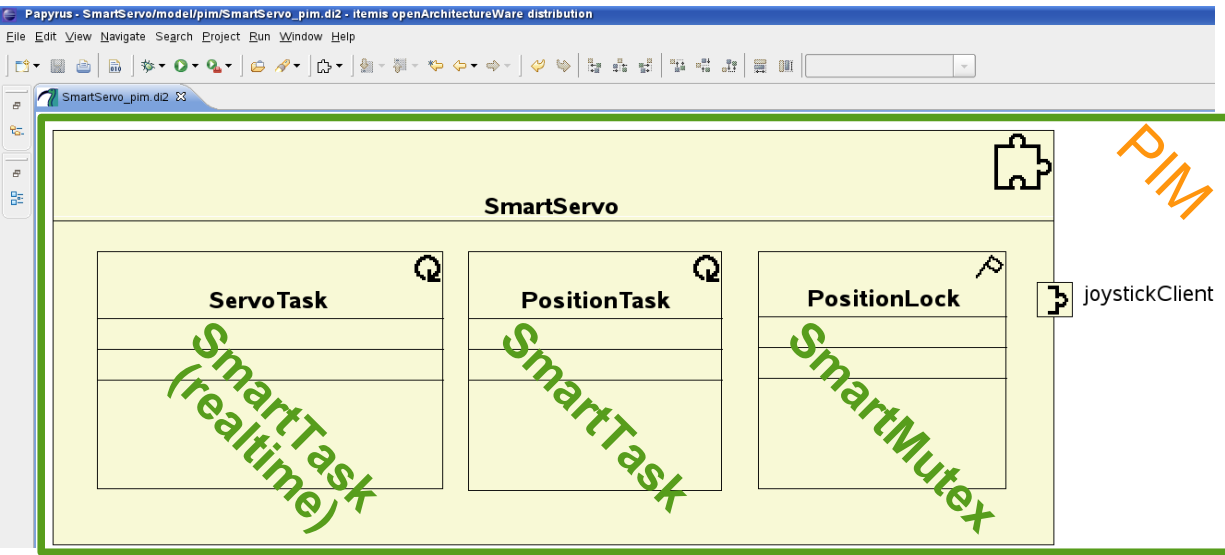
- SmartSoft Component
 - SmartTask
 - SmartMainState
 - SmartMutex
 - SmartTimer
 - SmartIniParameterGroup
 - SmartSendClient
 - SmartPushNewestClient
 - SmartPushTimedClient
 - SmartQueryClient
 - SmartEventClient
 - SmartStateClient
 - SmartParameterClient



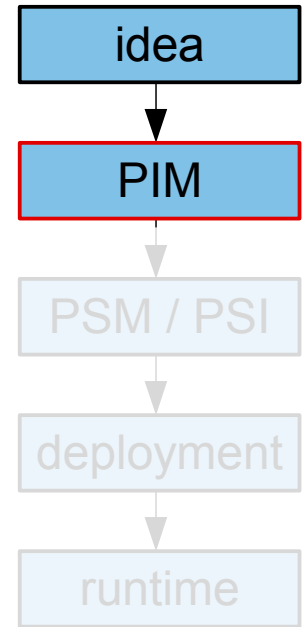
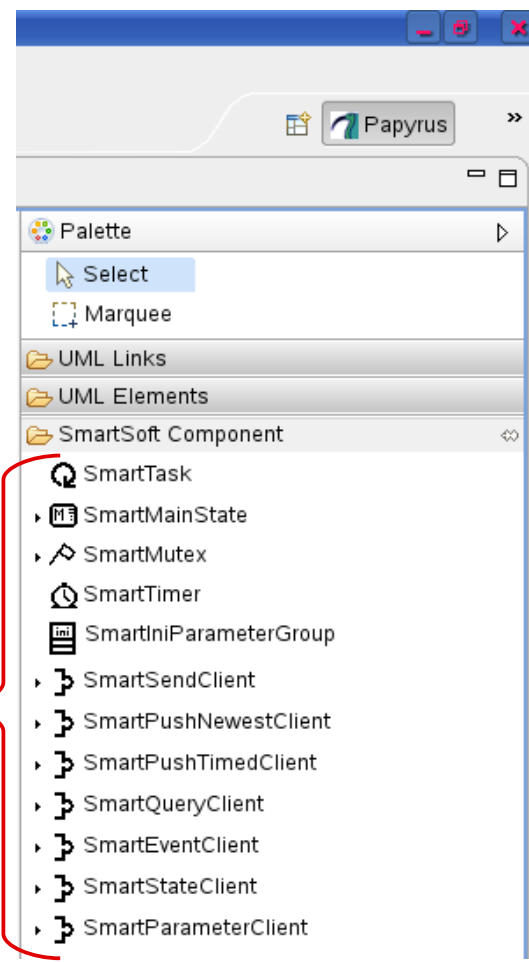
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component PIM



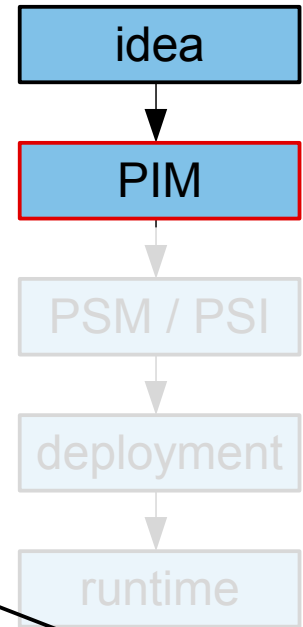
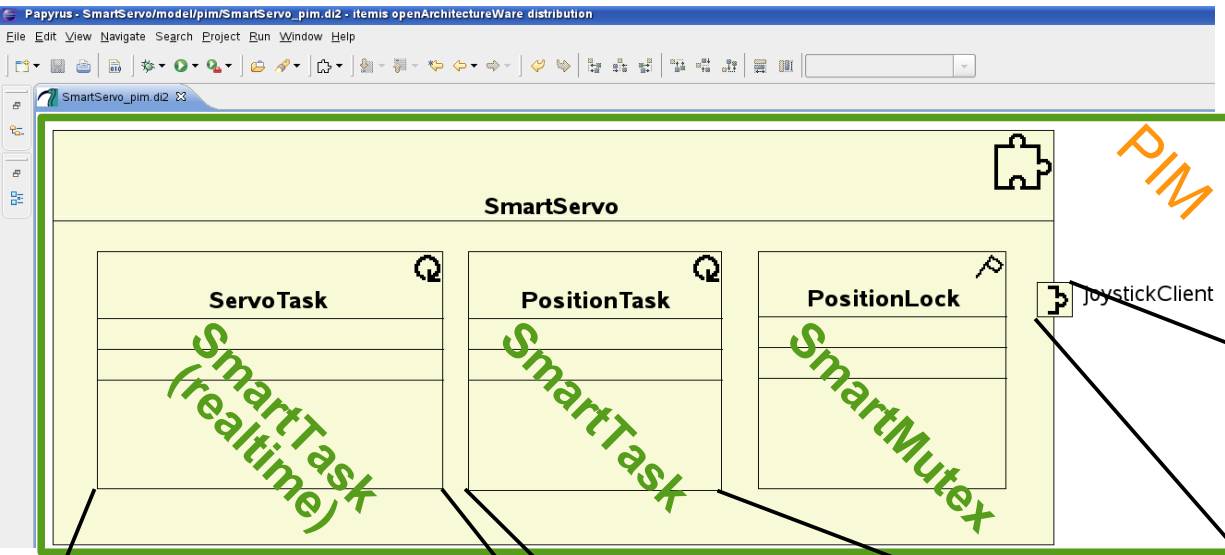
create PIM using the elements offered by the SmartSoft MDSD Toolchain palette



SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component PIM



control servo

- SmartTask (from SmartMARS)
 - isRealtime: Boolean [1..1] = true
 - period: Integer [1..1] = 20
 - priority: Integer [1..1] = 1
 - isPeriodic: Boolean [1..1] = true
 - wcet: Integer [1..1] = 1
 - timeUnit: TimeUnitKind [1..1] = ms

calculate servo position

- SmartTask (from SmartMARS)
 - isRealtime: Boolean [1..1] = false
 - period: Integer [1..1] = 0
 - priority: Integer [1..1] = 0
 - isPeriodic: Boolean [1..1] = false
 - wcet: Integer [1..1] = 0
 - timeUnit: TimeUnitKind [1..1] = s

port connected to smartJoystickServer

- SmartPushNewestClient (from SmartMARS)
 - serverName: String [1..1] = smartJoystickServer
 - serviceName: String [1..1] = joystick
 - commObject: Class [1..1] = CommJoystick

tagged values



SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
generate PSI

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

File Edit View Navigate Search Project Run Window

Run code generator

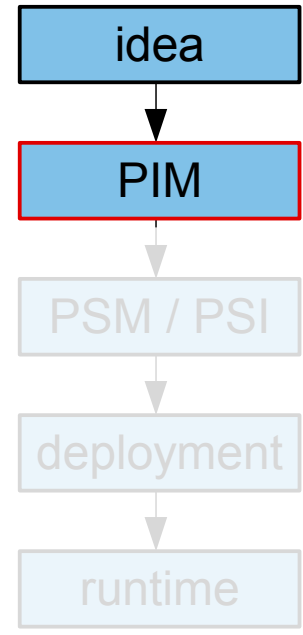
SmartServo_pim.di2

SmartServo

ServoTask <i>SmartTask (realtime)</i>	PositionTask <i>SmartTask</i>	PositionLock <i>SmartMutex</i>
---	---	--

joystickClient

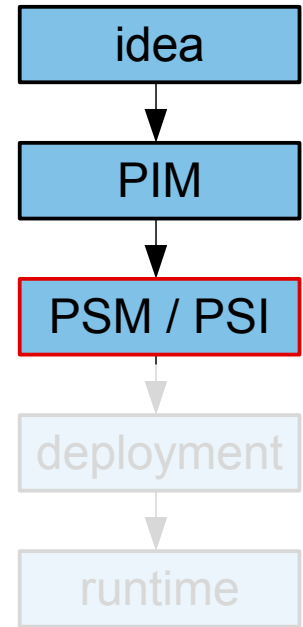
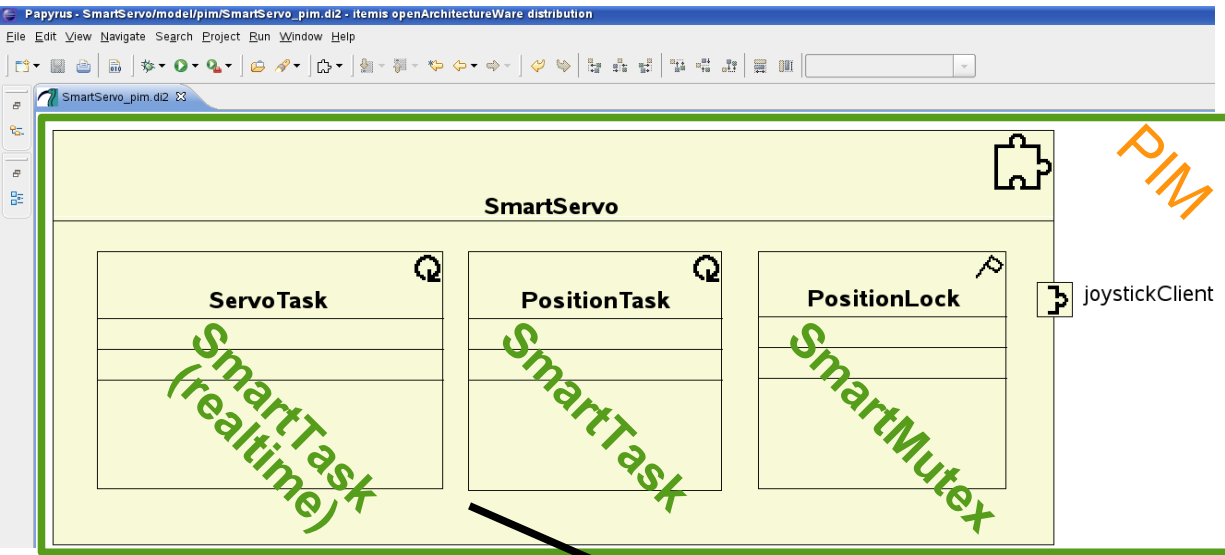
PIM



SmartSoft MDSD Toolchain

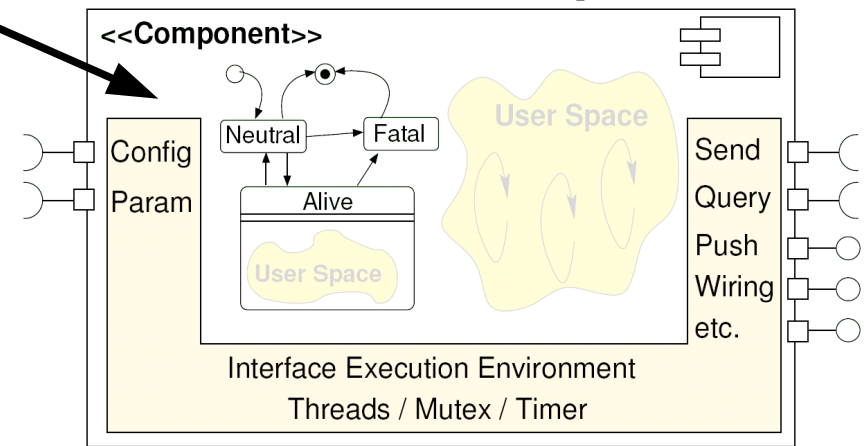
Example 1: Servo (RTAI-Linux)

creating a component
generate PSI



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

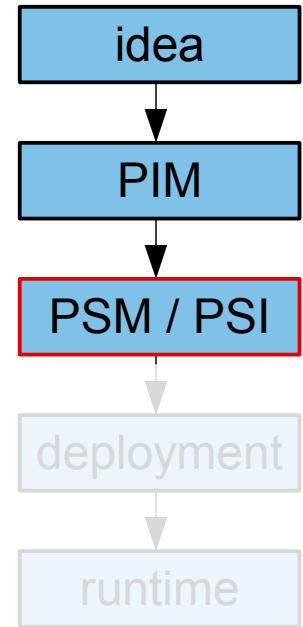
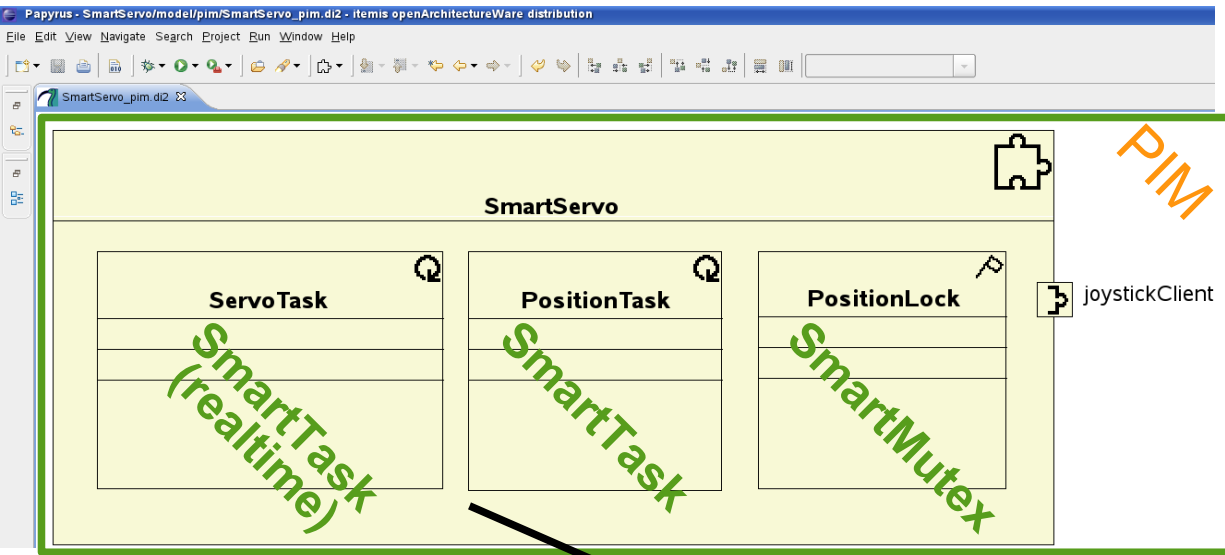
executable component



SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

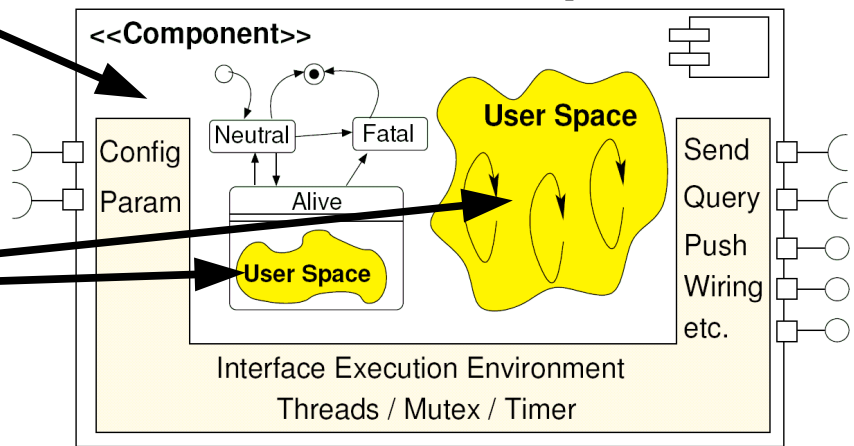
creating a component
integrate
user-code



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

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

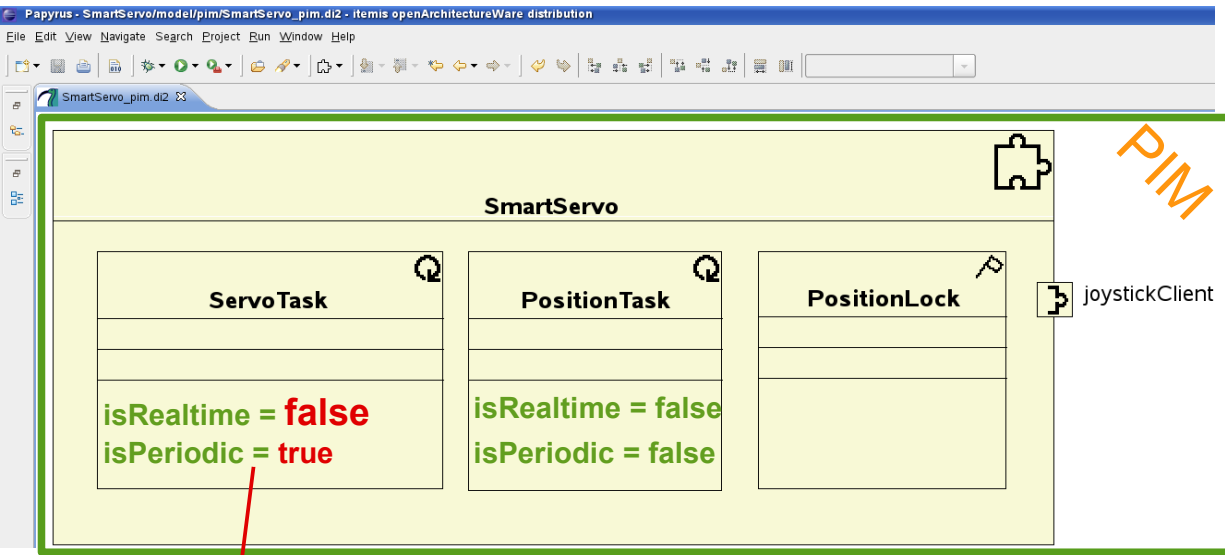
executable component



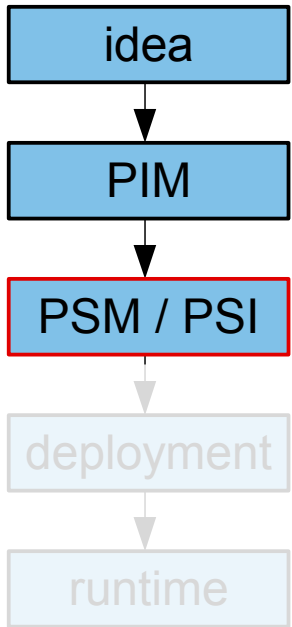
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

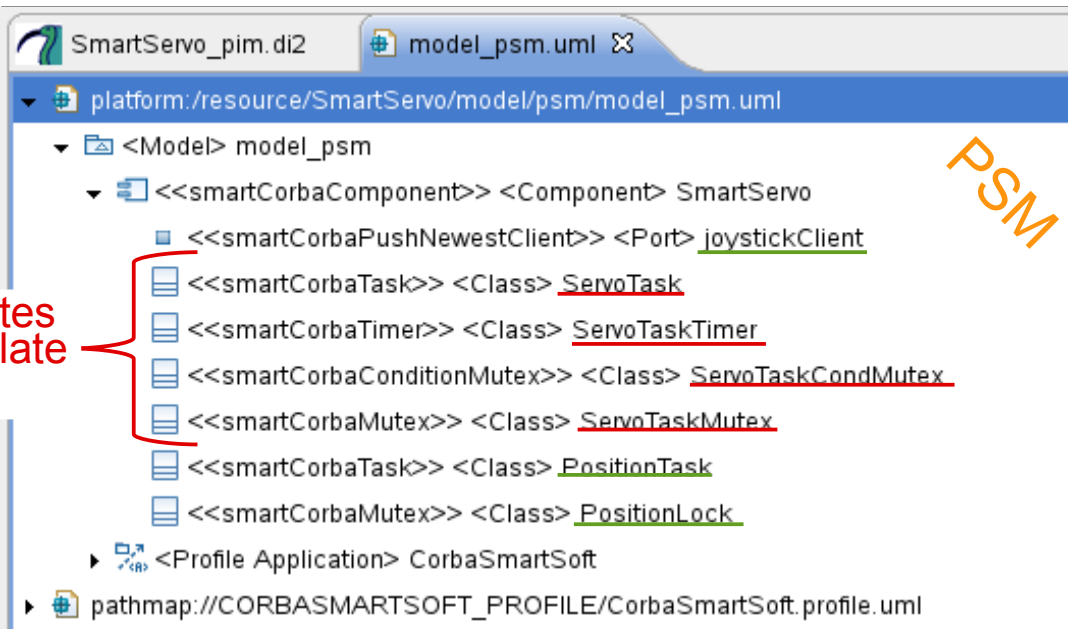
creating a component generate PSM



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



toolchain generates elements to emulate periodic tasks



```

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

#include <iostream>

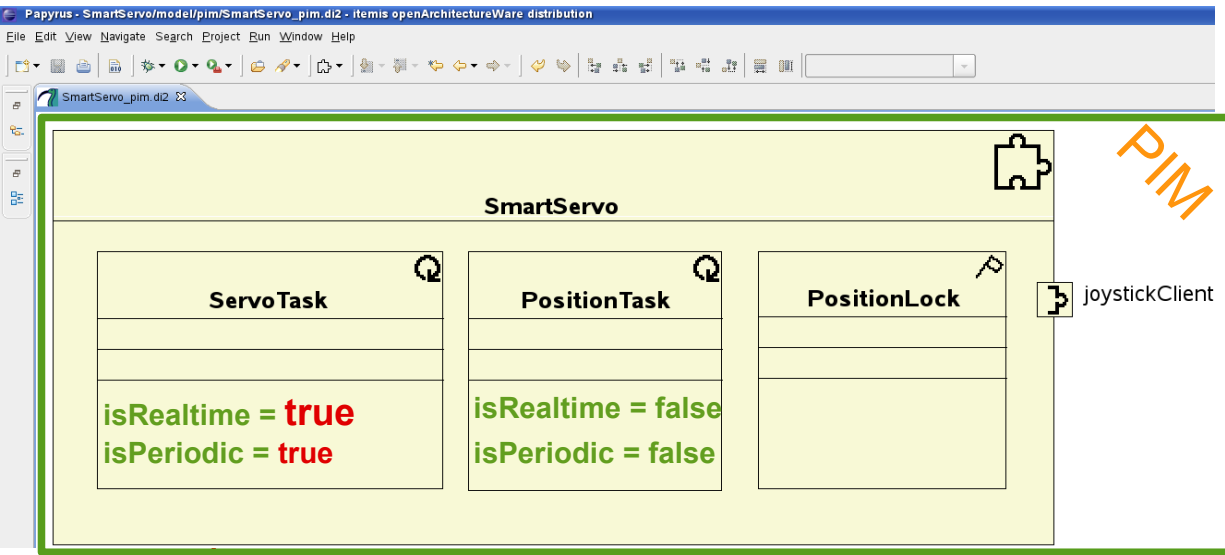
ServoTask::ServoTask()
{
    std::cout << "constructor ServoTask\n";
}

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

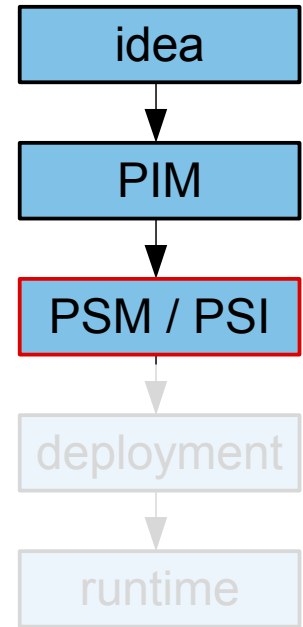
SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

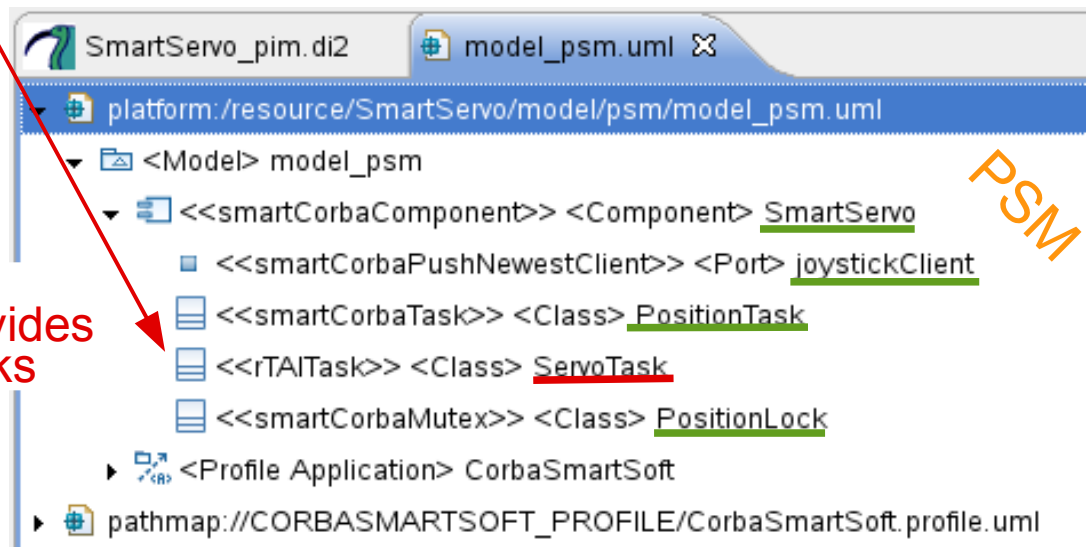
creating a component generate PSM



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



RTAI-Linux already provides periodic tasks



PSI

```

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

#include <iostream>

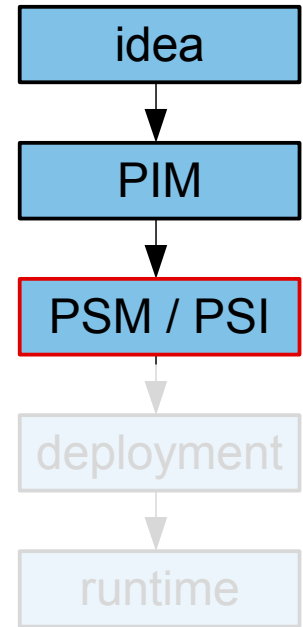
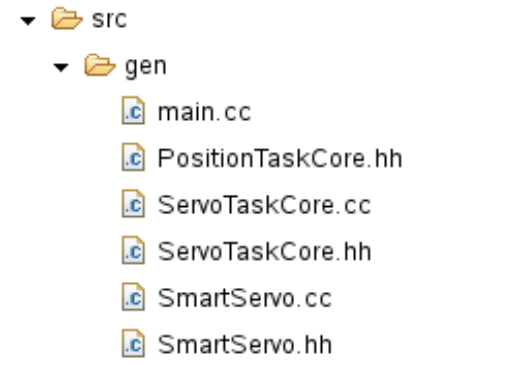
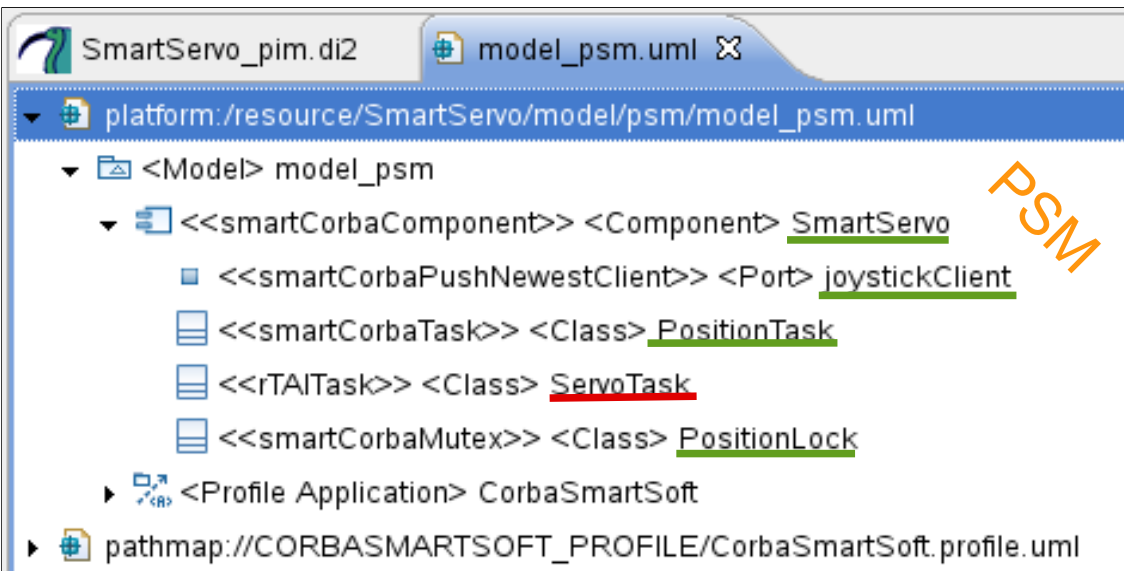
ServoTask::ServoTask()
{
    std::cout << "constructor ServoTask\n";
}

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 PSI



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

```

SmartServo_pim.di2    model_psm.uml    ServoTasi
// include communication objects
#include "externalCommObjectHeaders.hh"

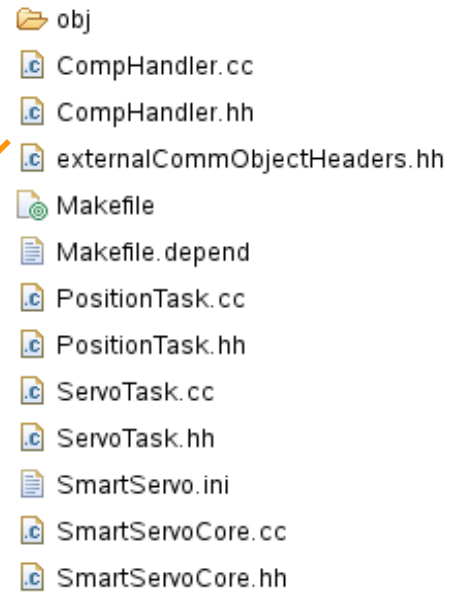
void CompHandler::onStartup()
{
  std::cout
    << "startup - put your startupCode

  CHS::StatusCode status;

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

  COMP->ioystickClient->subscribe();
  
```

PSI





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

```

SmartServoCore.hh
#ifndef _SMARTSERVOCORE_HH
#define _SMARTSERVOCORE_HH

#include <iostream>

class SmartServoCore
{
private:

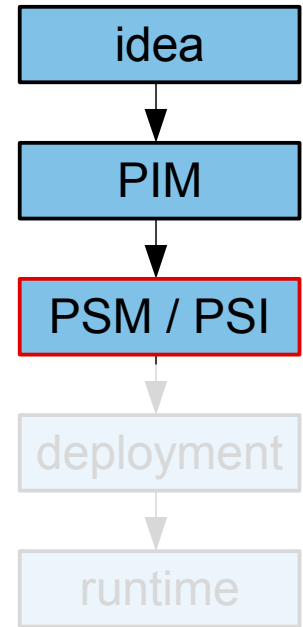
public:
  SmartServoCore();
  int servoOnTime;
};

#endif

```

code added
by developer

PSI





SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
integrate
user-code

```

SmartServo_pim.di2  model_psm.uml  PositionTask.cc  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;
}

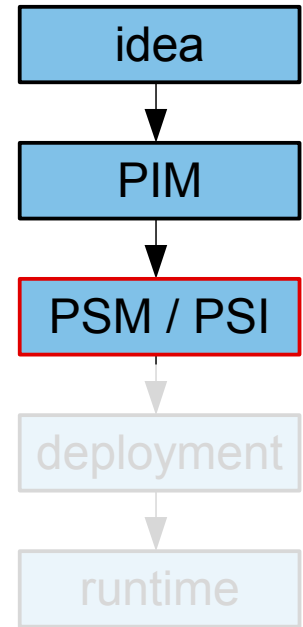
```

calculate servo position (servoOnTime)

wait for position update (joystickServer)

code added by developer

PSI





SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component
integrate
user-code

```

SmartServo_pim.di2 | model_psm.uml | PositionTask.cc | ServoTask.cc
#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();

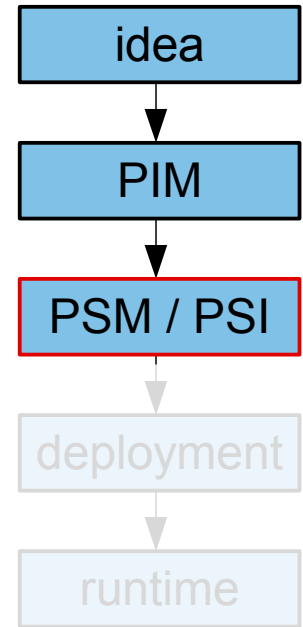
        // switch on
        outb(0xff, LPT_PORT);
        rt_sleep(on_time);

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

```

code added
by developer

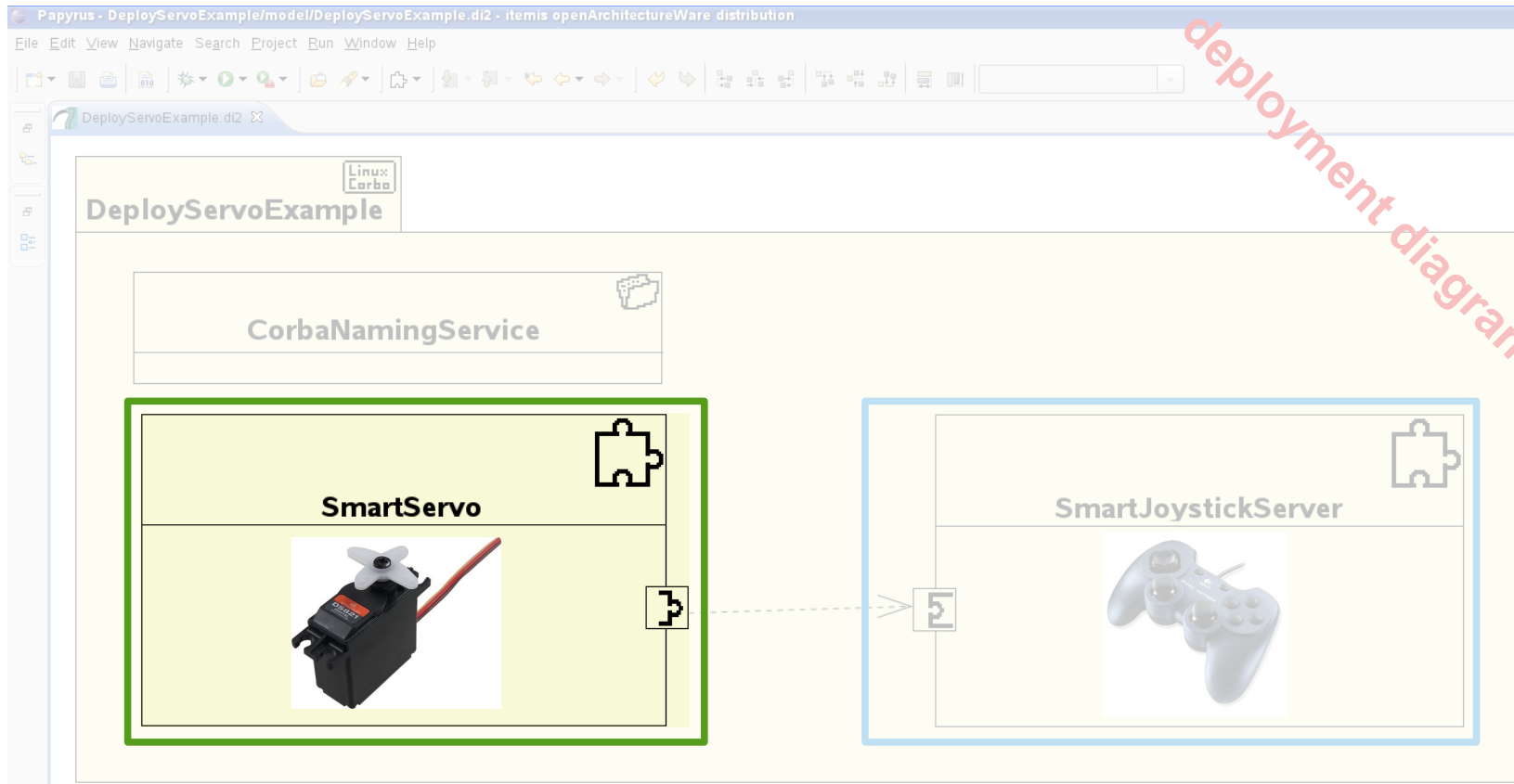
PSI



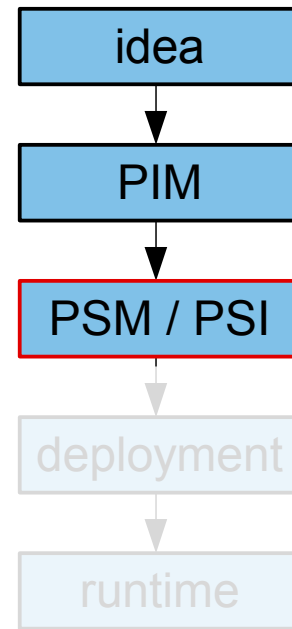


SmartSoft MDSD Toolchain Example 1: Servo (RTAI-Linux)

overview
servo example



deployment diagram



SmartServo component
- created

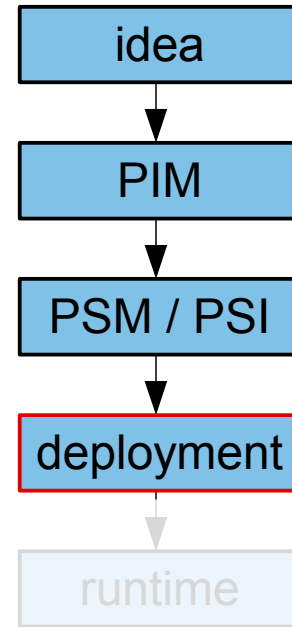
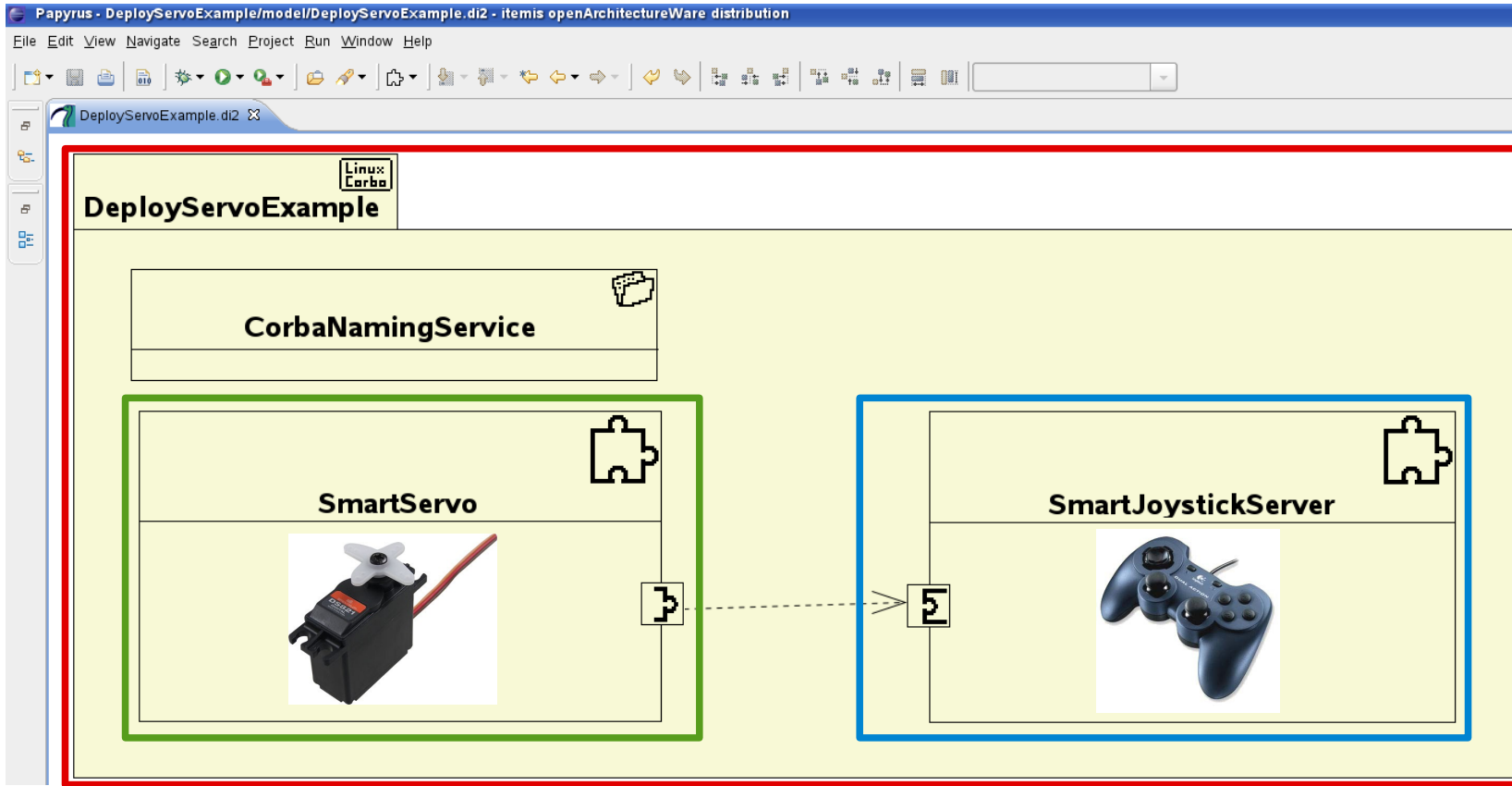
SmartJoystickServer component
- preexisting component (COTS)





SmartSoft MDSD Toolchain Example 1: Servo (RTAI-Linux)

create deployment



- Palette
- Select
- Marquee
- UML Links
- UML Elements
- SmartSoft Component
- SmartSoft Deployment
- CorbaNamingService
- Connection

use these elements to create deployment

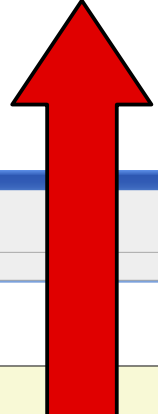
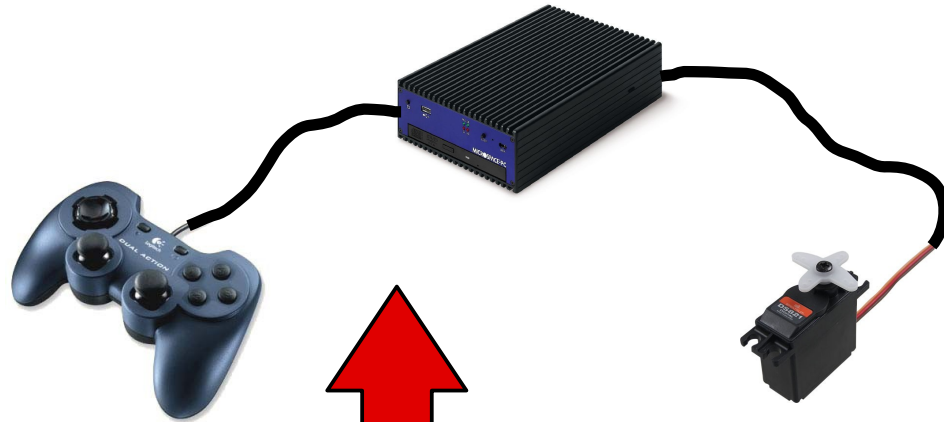




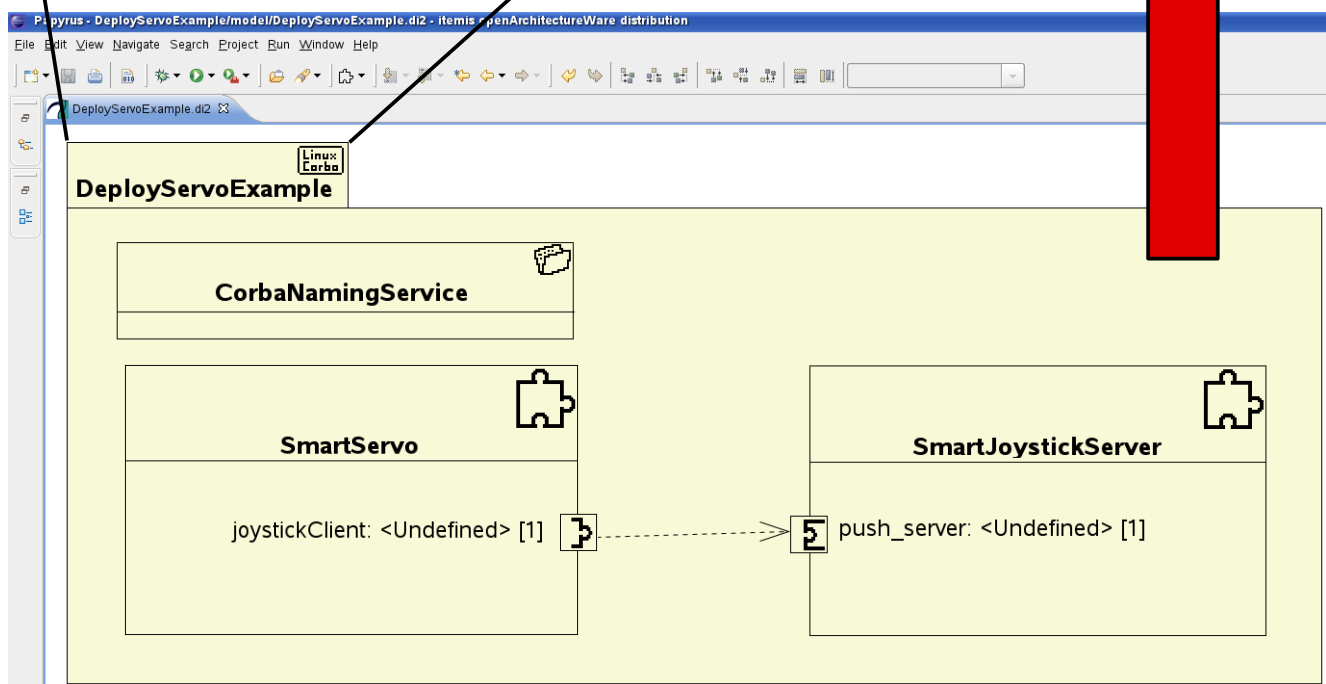
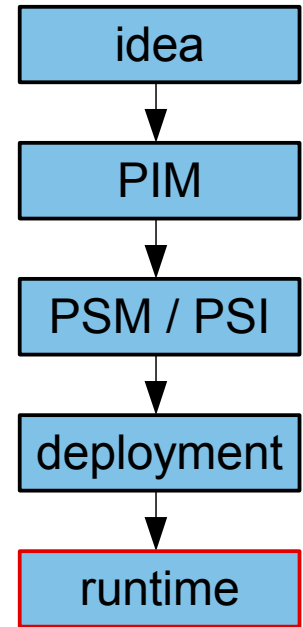
SmartSoft MDSD Toolchain Example 1: Servo (RTAI-Linux)

create deployment

- CorbaSmartSoftDevice (from CorbaSmartSoft)
 - ip: String [1..1] = 192.168.0.1
 - deployed: DeployType [1..1] = remote
 - username: String [1..1] = asteck
 - directory: String [1..1] = deployServo



deploy to target system and run startscript

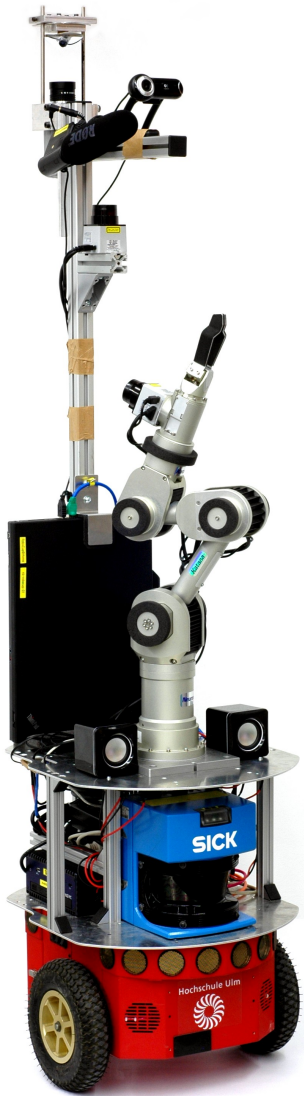




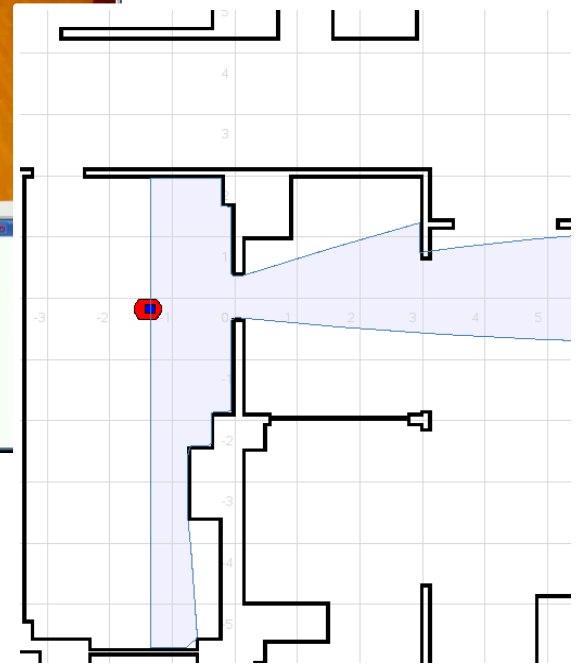
SmartSoft DSD Toolchain

Example 2: Navigation Task

overview navigation task



The screenshot displays a ROS environment with several terminal windows. The top-left window shows the deployment of a task. The middle-left window shows the configuration of a SmartMapperGridMap. The middle-right window shows the configuration of a SmartPlannerBreadthFirstSearch. The bottom-left window shows the configuration of a SmartPlannerBaseServer. The bottom-middle window shows the configuration of a SmartCtdServer. The bottom-right window shows the SmartRobotConsole interface. A camera view of the robot in a room is also visible in the top-right corner of the screenshot.



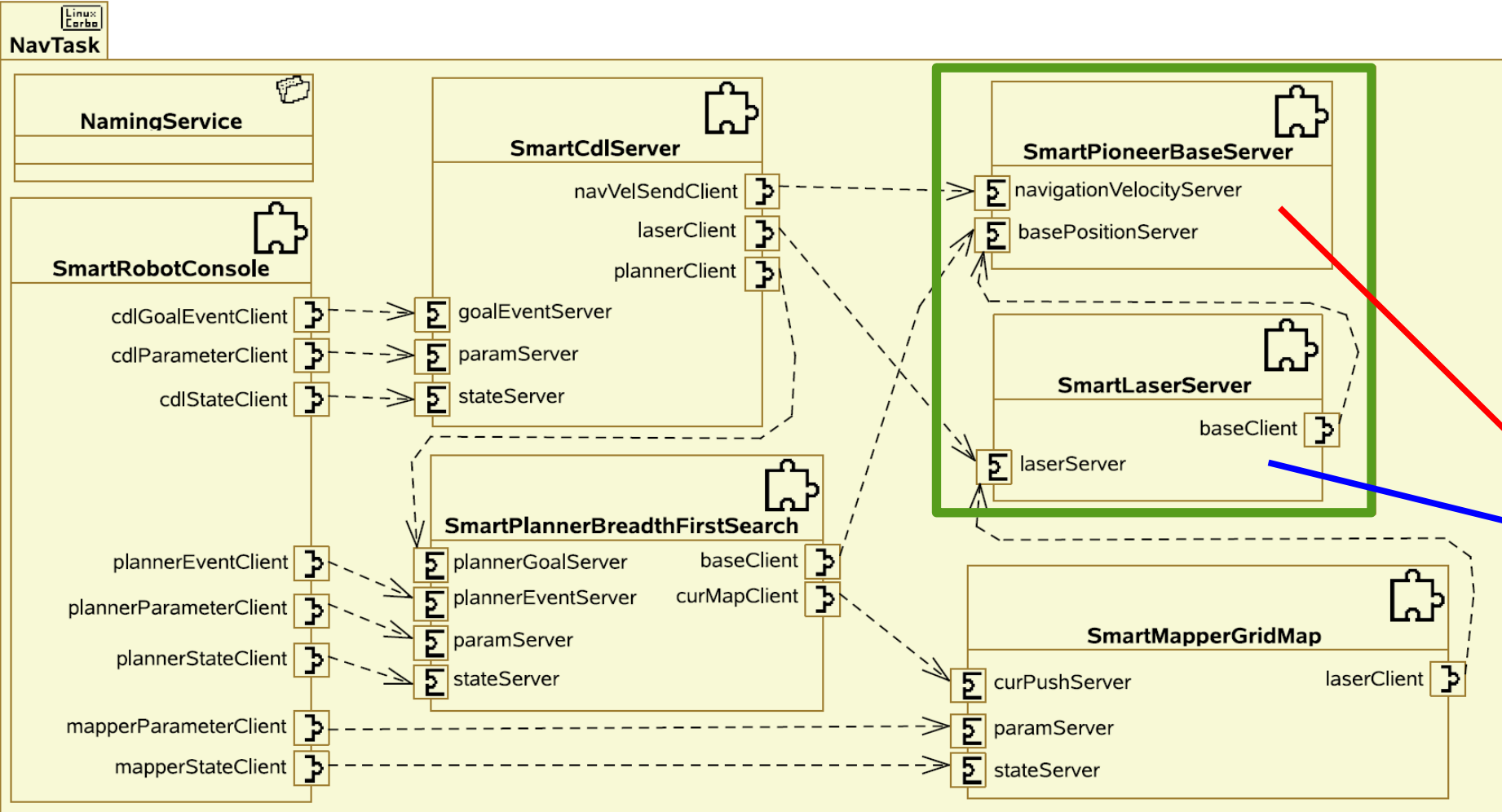
- deployment of COTS components
- simple replacement of components (P3DX/SICK ↔ Player/Stage)
- robot drives to positions given by operator
- collision avoidance, map building, path planning, etc.





SmartSoft MDSD Toolchain Example 2: Navigation Task

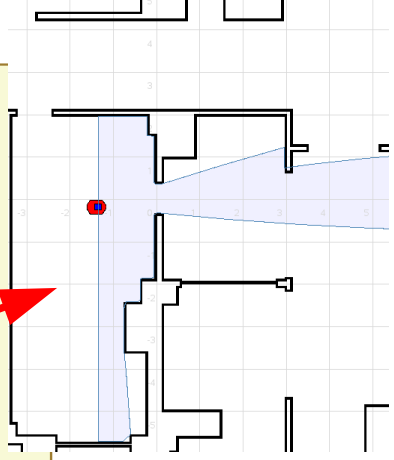
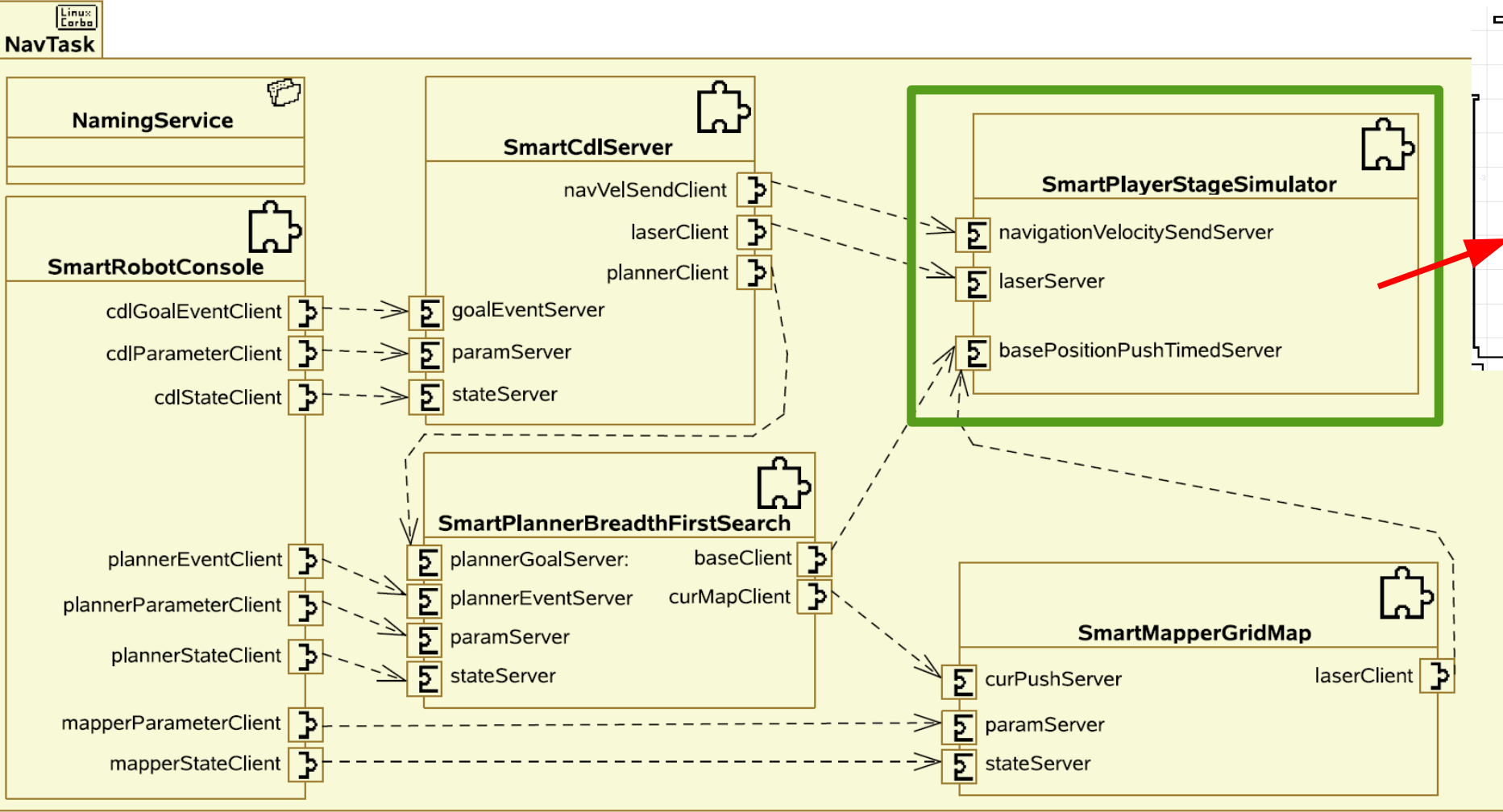
deployment
navigation task
P3DX/SICK





SmartSoft MDSD Toolchain Example 2: Navigation Task

deployment
navigation task
Player/Stage



SmartSoft MDSD Toolchain

Example 2: Navigation Task

Live Demo

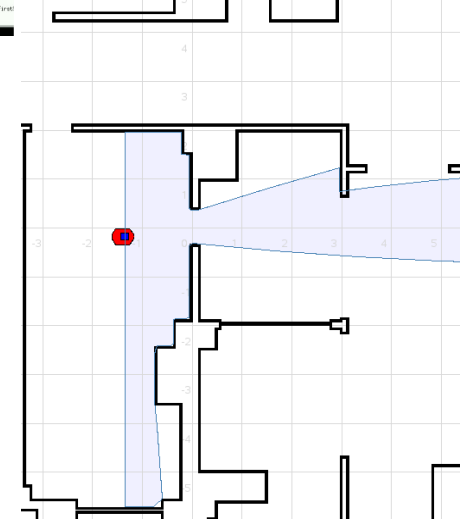
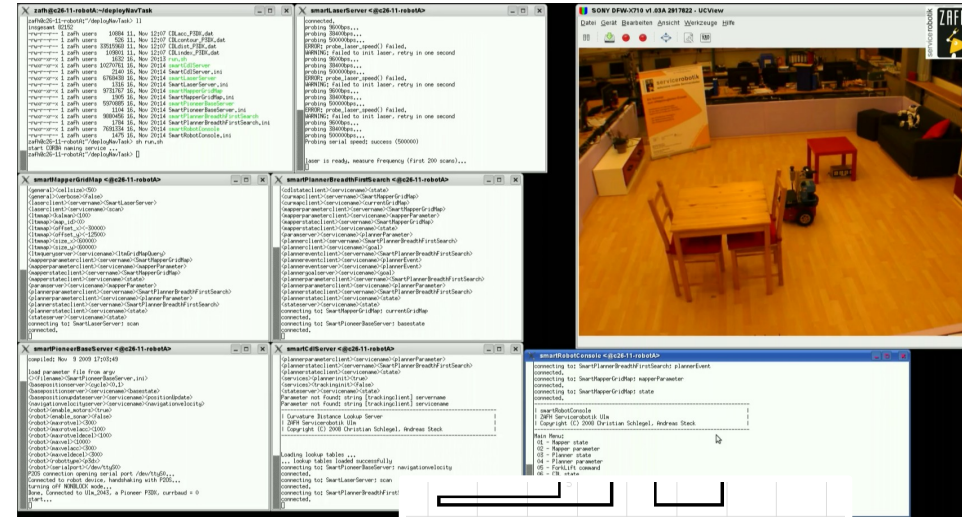
- how to create a deployment in SmartSoft MDSD Toolchain
- define the initial wiring between the components
- add platform specific details
- add target platform settings
- deploy scenario to target
- run scenario in Player/Stage simulator

watch on youtube:

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

attached file:

2009-12-02-deploy-toolchain.mp4



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

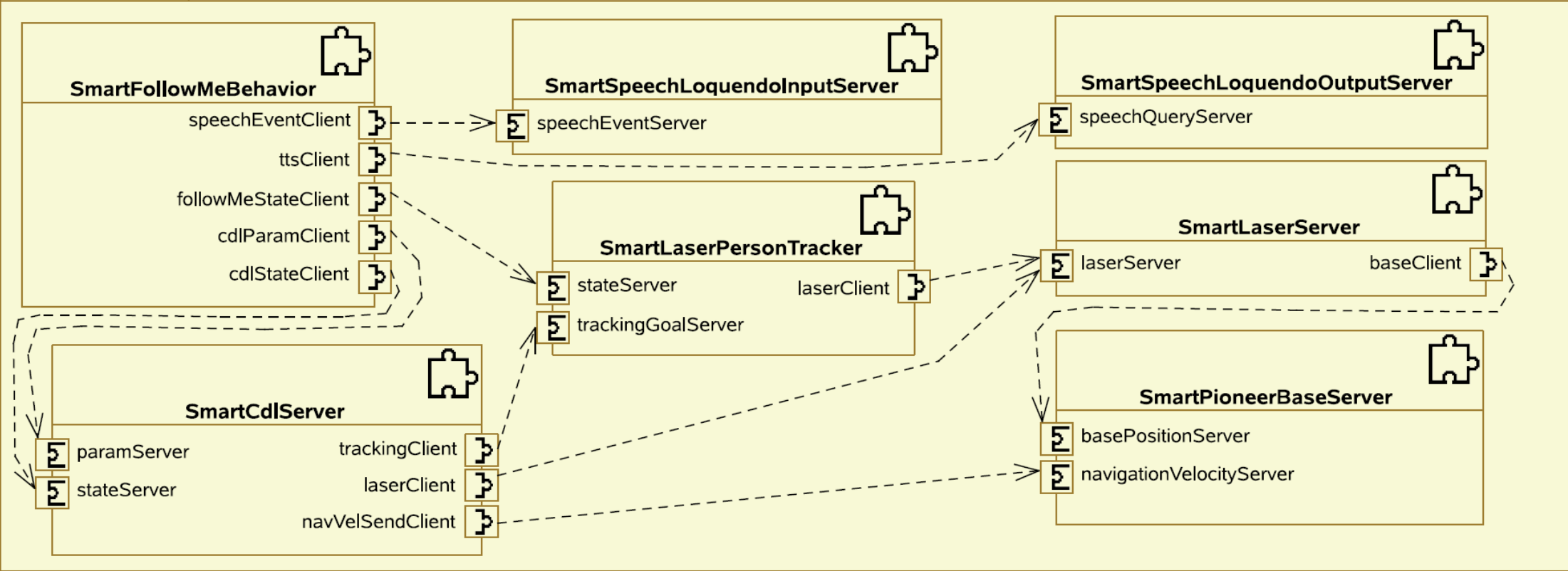


SmartSoft MDSD Toolchain

Example 3: "Follow Me" - RoboCup@Home

deployment

Linux Corba
DeployFollowMe



SmartSoft MDSD Toolchain

Example 3: “Follow Me” - RoboCup@Home

overview



[watch on youtube:](#)

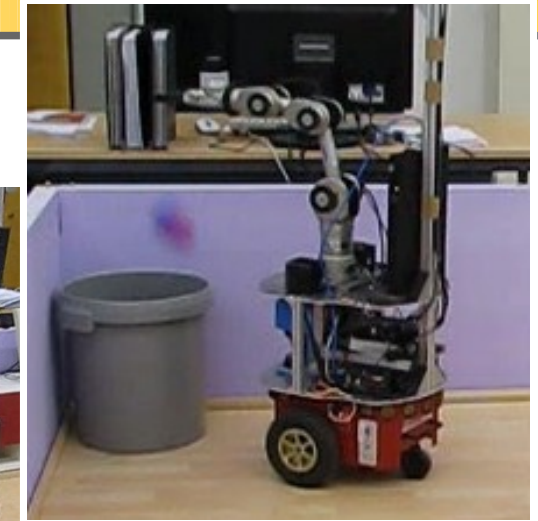
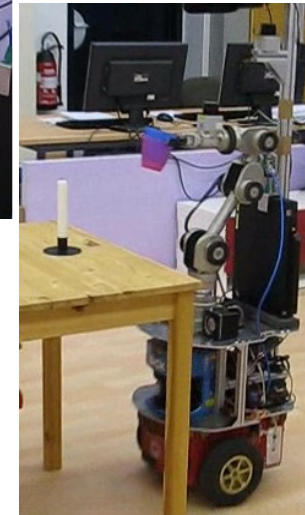
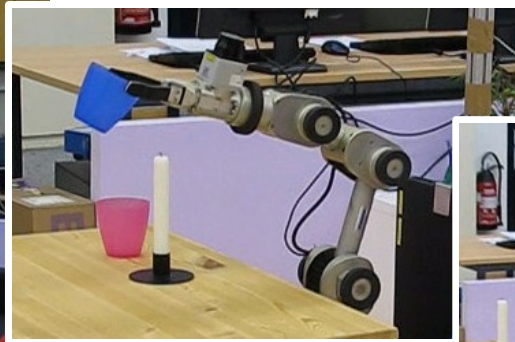
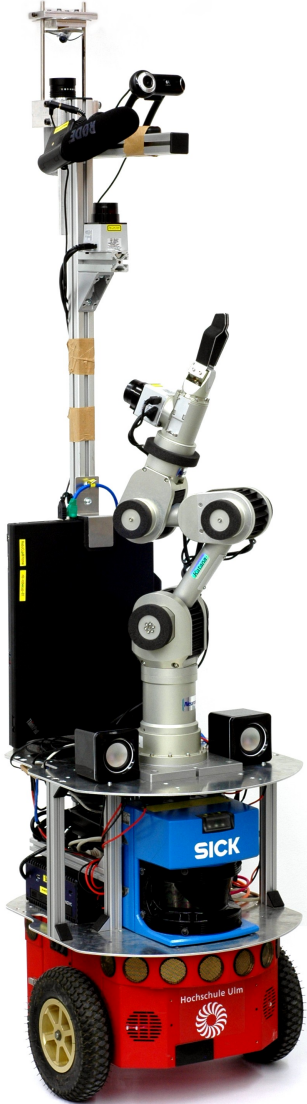
http://www.youtube.com/watch?v=-xvO_bbrb-l
http://www.youtube.com/watch?v=grf-_-32oY0

[attached files:](#)

[2009-09-10-followMe.mp4](#)
[2010-02-25-followMe.mp4](#)

SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario



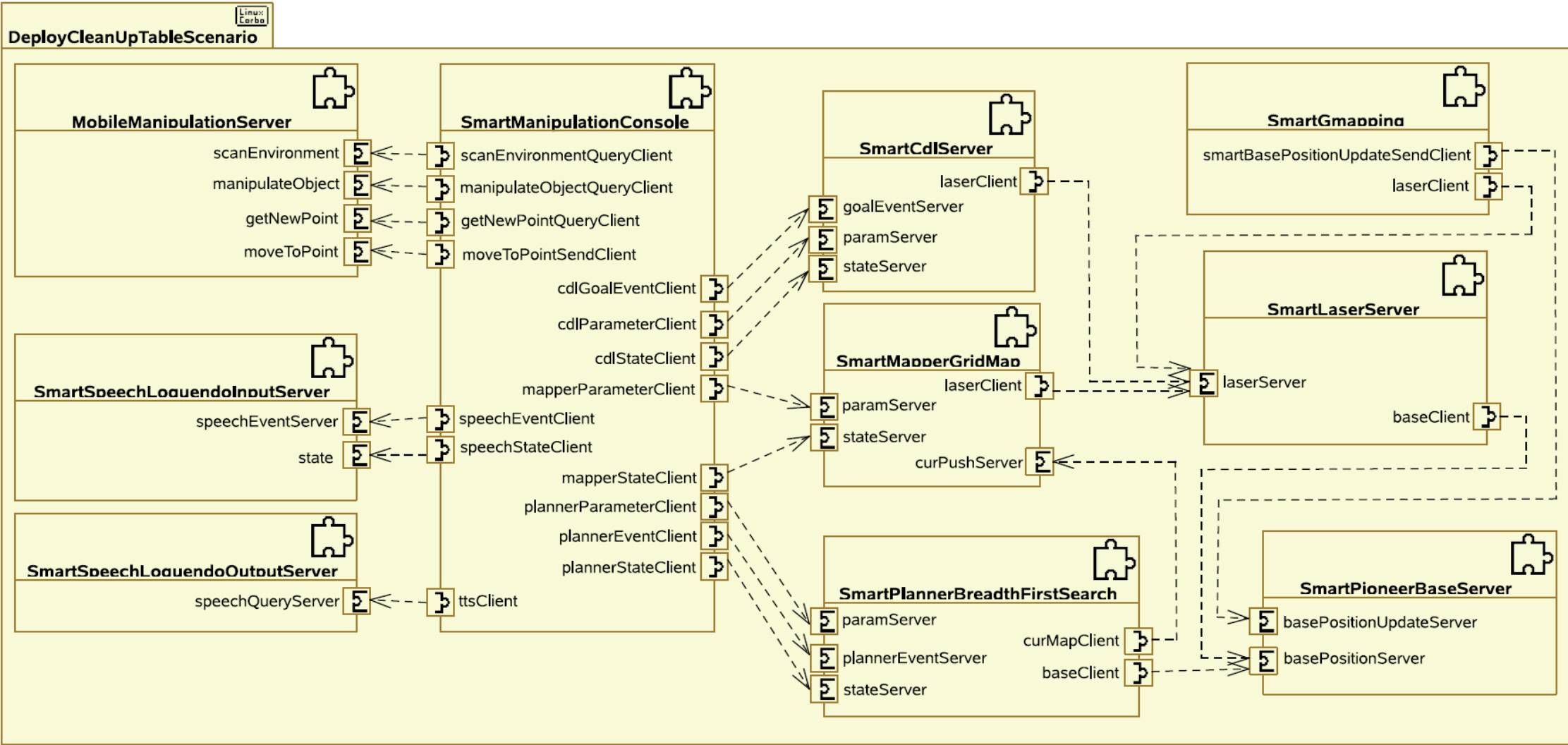
- deployment of COTS components
- complex real world scenario in everyday environment
- robot cleans up the table



SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario

deployment

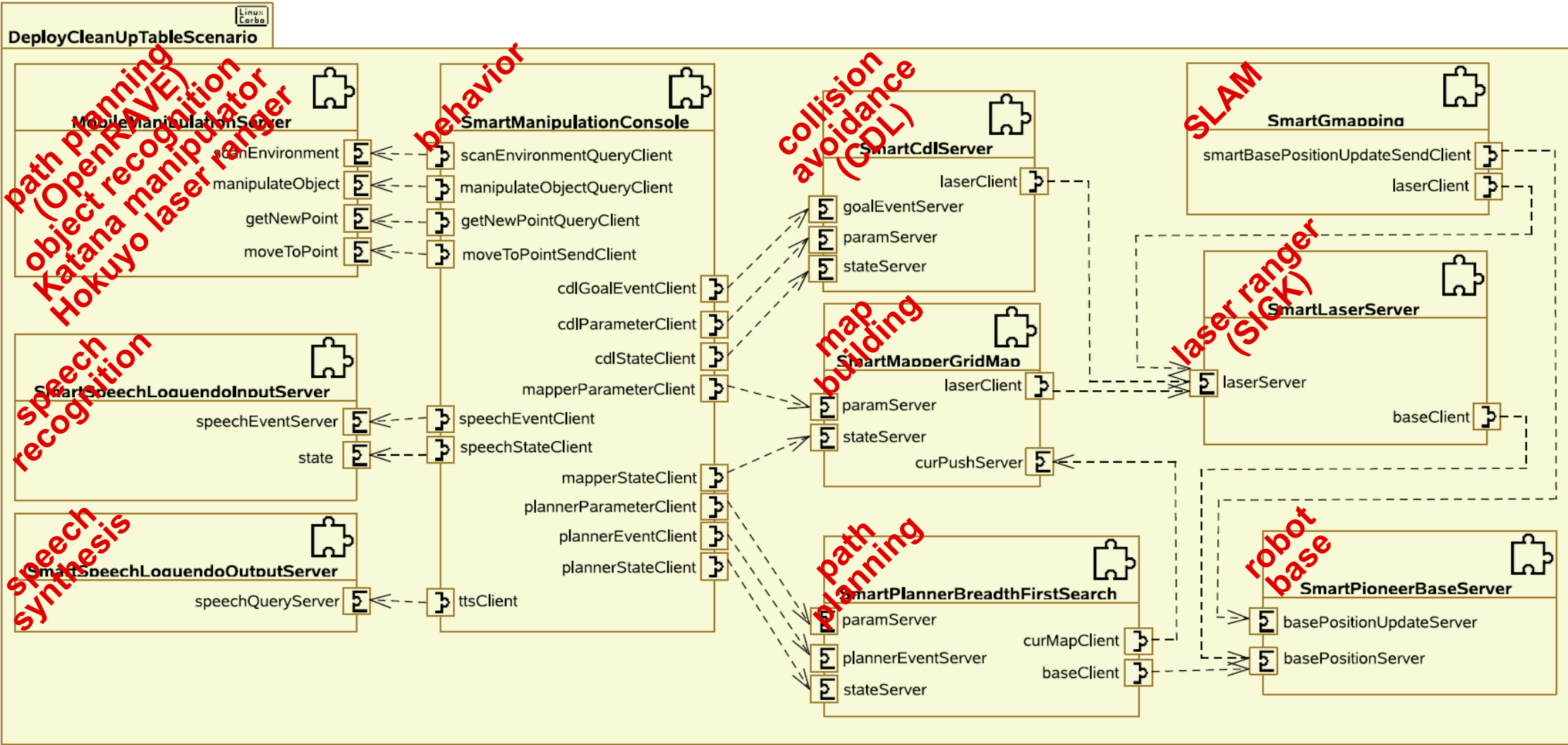




SmartSoft MDSD Toolchain

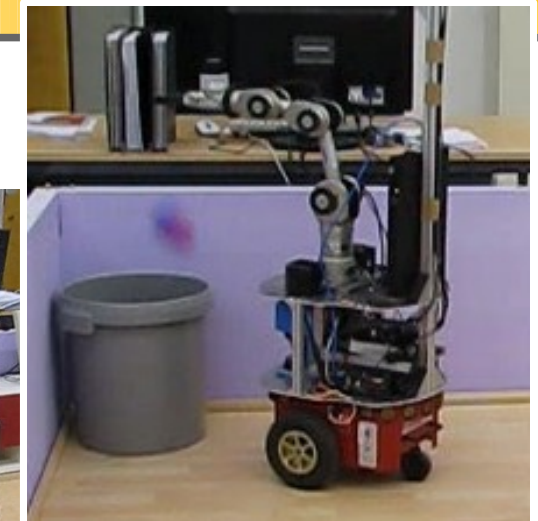
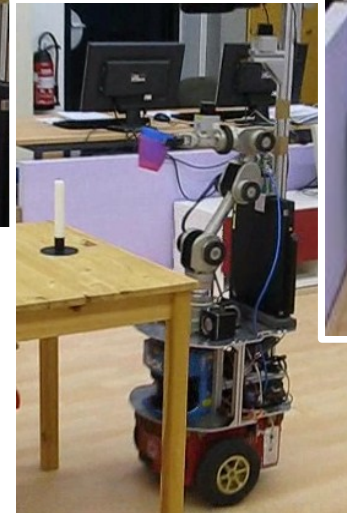
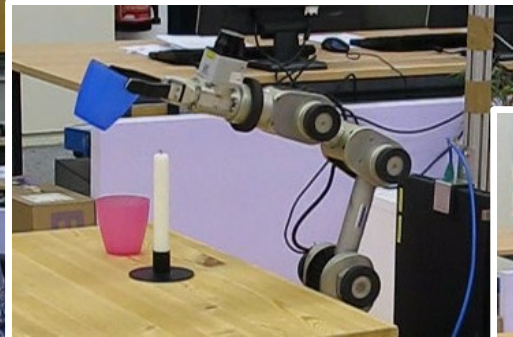
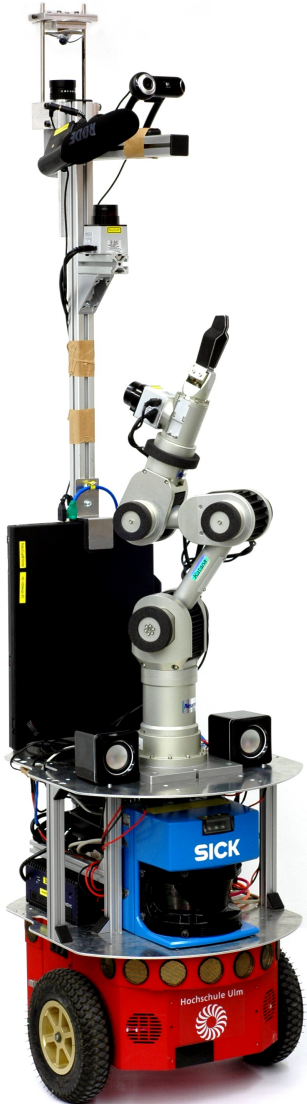
Example 4: Cleanup Table Scenario

deployment



SmartSoft MDSD Toolchain

Example 4: Cleanup Table Scenario



[watch on youtube:](#)

<http://www.youtube.com/watch?v=40d4DIk5LCQ>

[attached file:](#)

[2010-02-23-cleanUpTable.mp4](#)



SmartSoft MDSD Toolchain Links

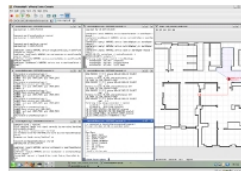
SmartSoft - Mozilla Firefox <2>

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

SmartSoft
Components and Toolchain for Robotics

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

What is SmartSoft?



standardized components whose inter-requirements.

YouTube - Kanal von RoboticsAtHsUlm - Mozilla Firefox

http://www.youtube.com/roboticsAtHsUlm

YouTube
Broadcast Yourself™

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

Info Kommentare Favorit Weiterleiten Playlists Melden





Addendum

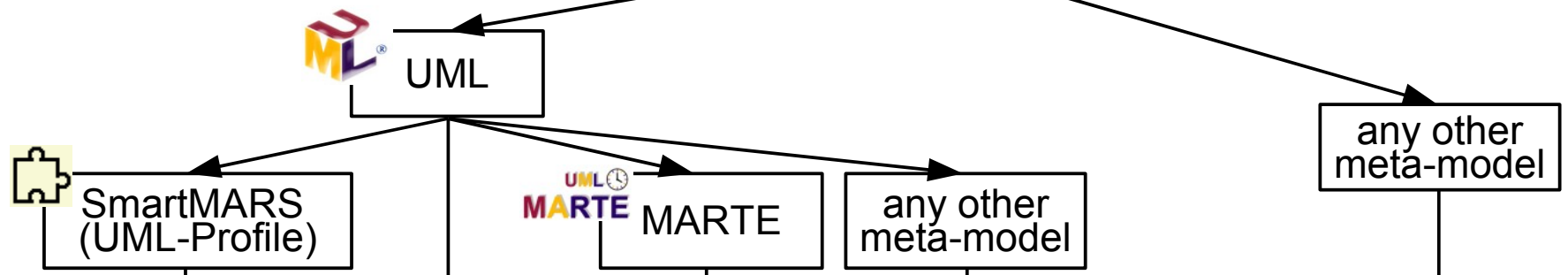


Eclipse Modeling Framework Meta-Modeling

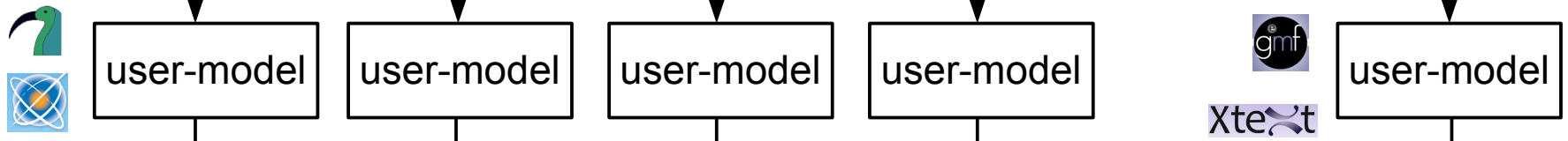
M3
meta-meta-model



M2
meta-model



modeling tool

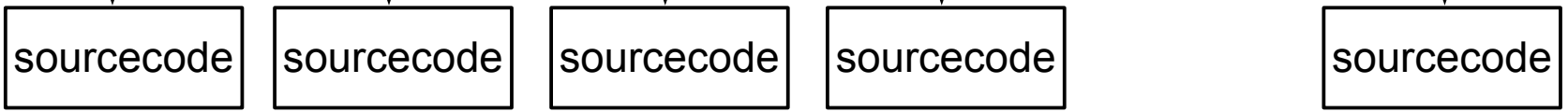


M1
model

code generation

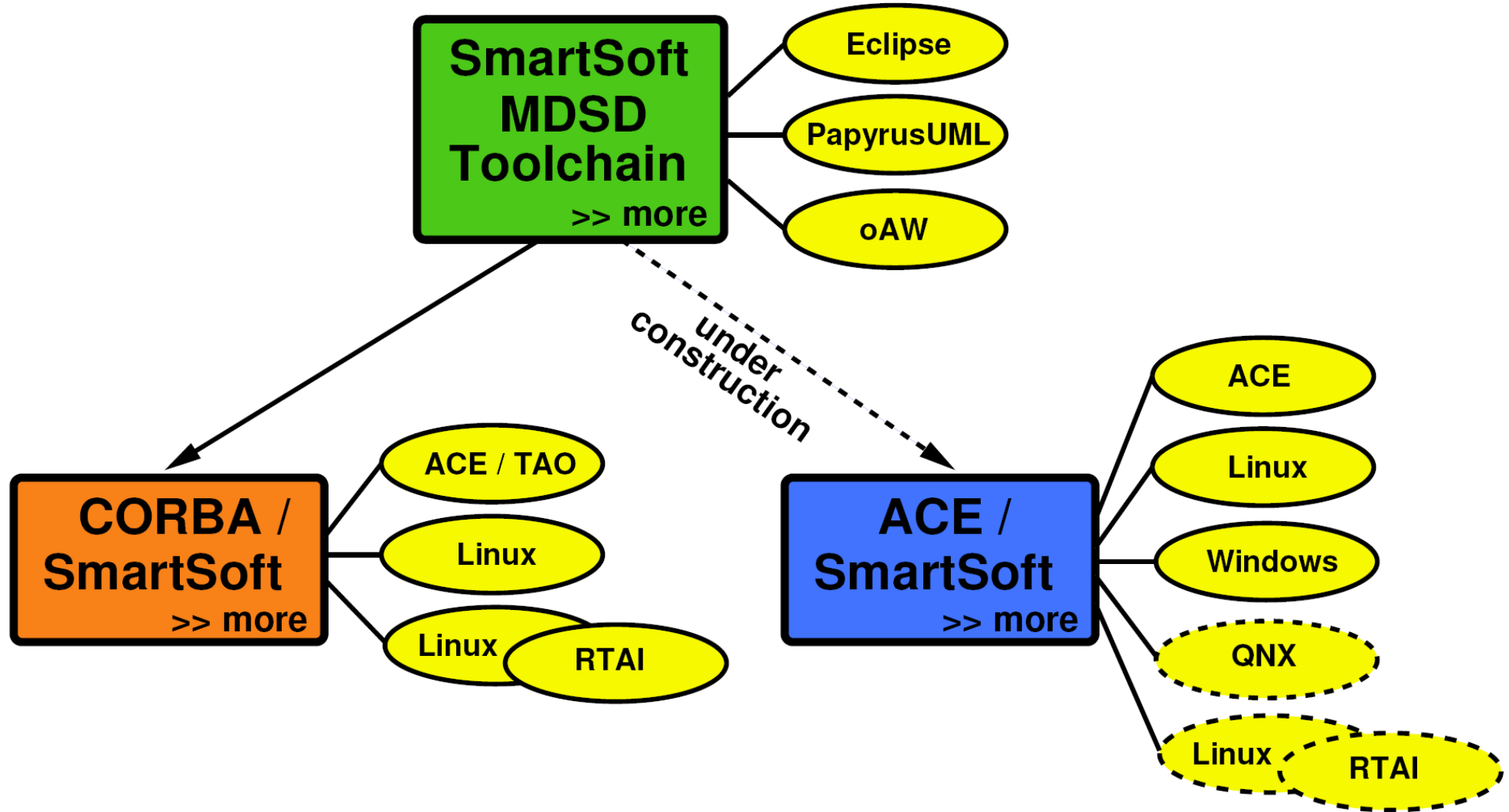


M0
sourcecode/
text



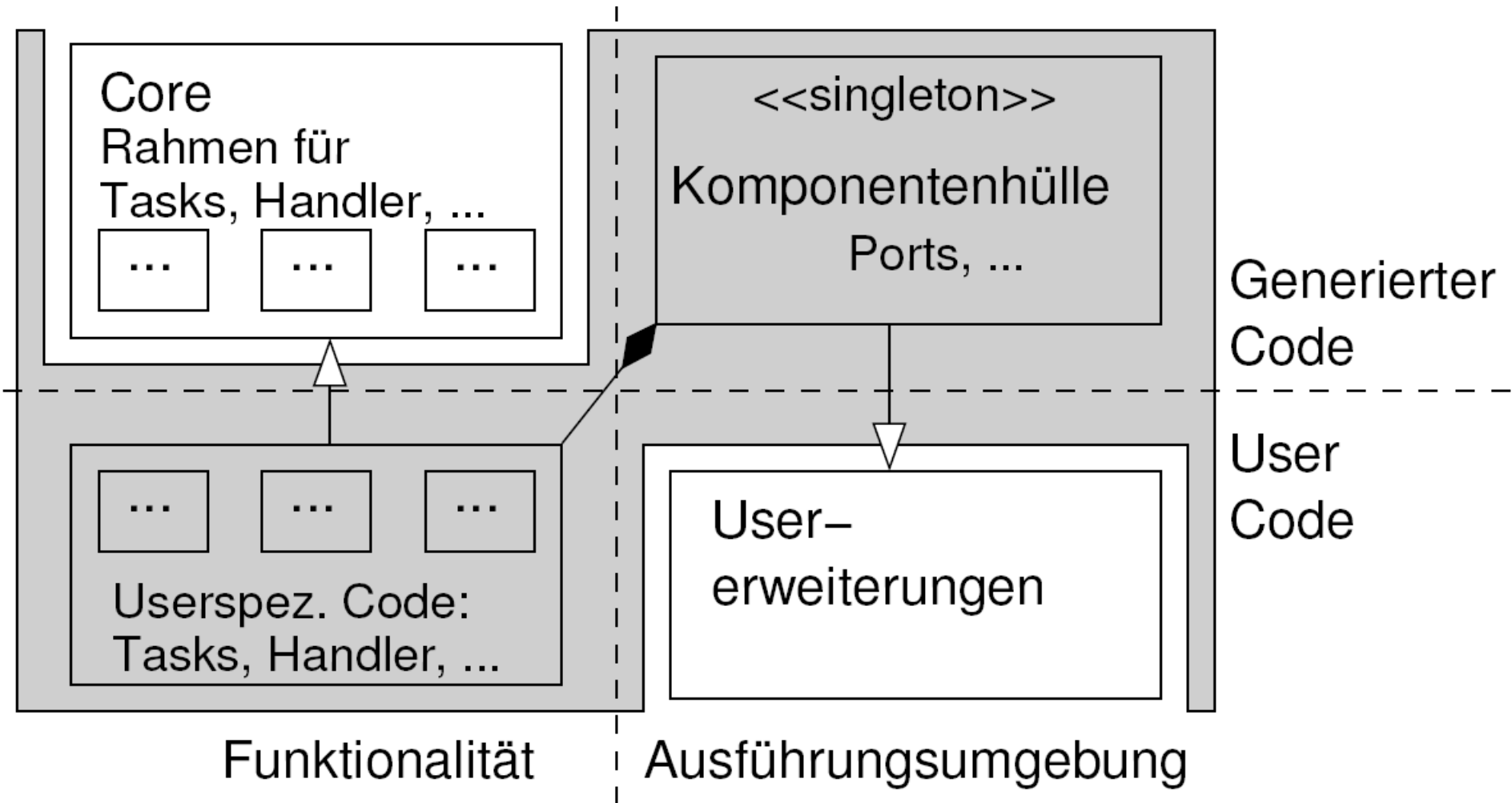


SmartSoft MDSD Toolchain





SmartSoft MDSD Toolchain Generation Gap Pattern

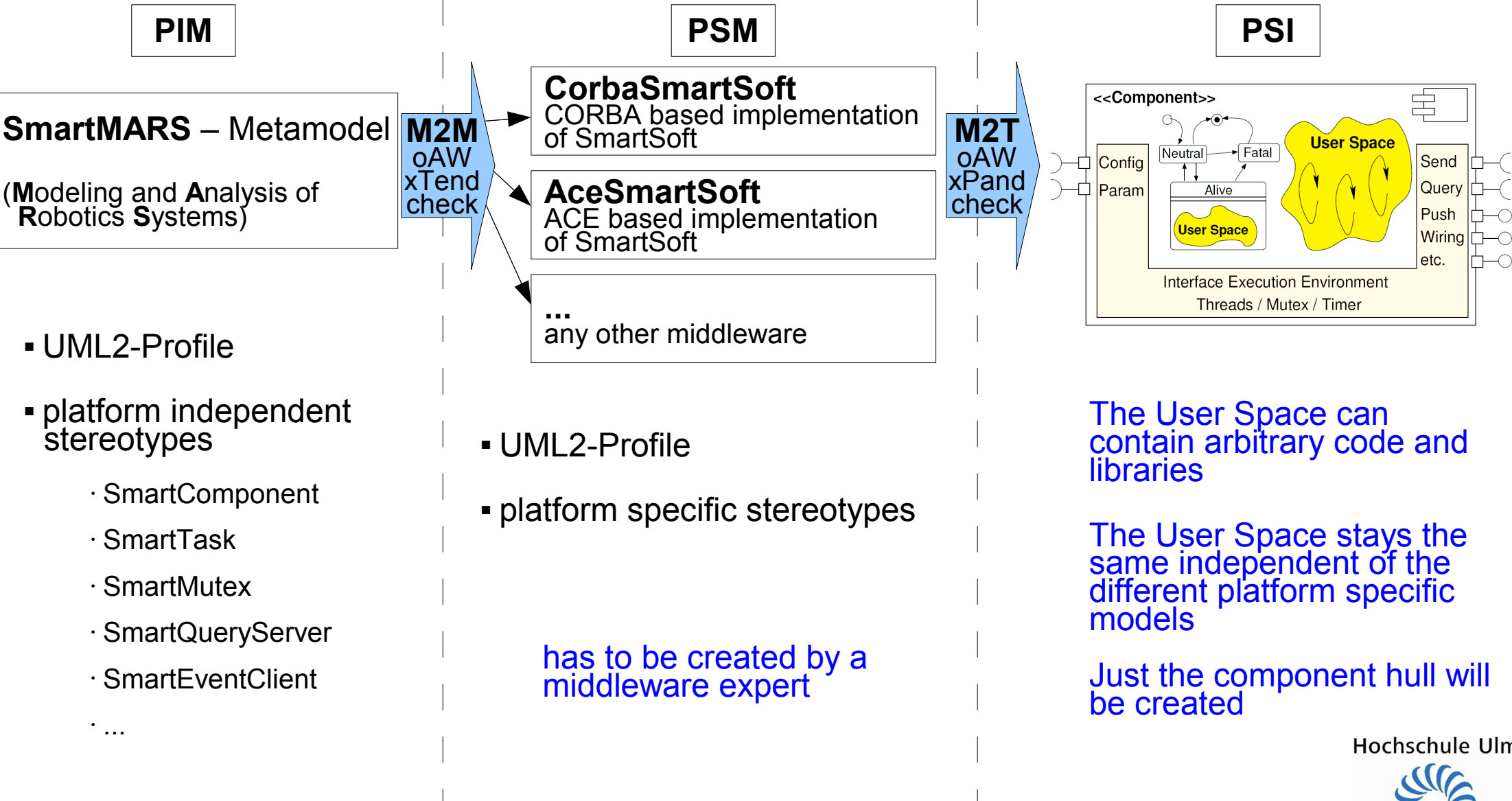




SmartSoft MDSD Toolchain

Example 1: Servo (RTAI-Linux)

creating a component workflow



SmartSoft MDSD Toolchain

Creating a component workflow

PIM

PSM

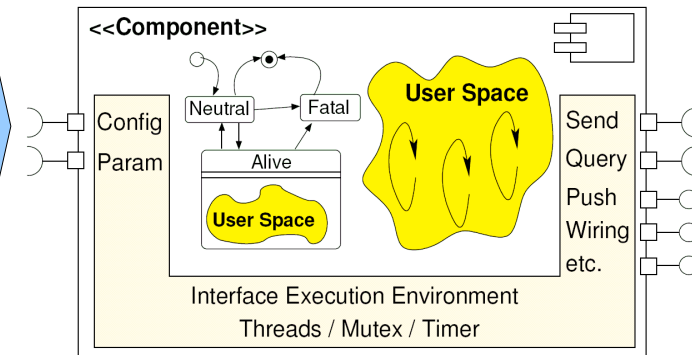
PSI

SmartMARS – Metamodel
(Modeling and Analysis of Robotics Systems)

M2M
oAW
xTend
check

CorbaSmartSoft
CORBA based implementation of SmartSoft

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

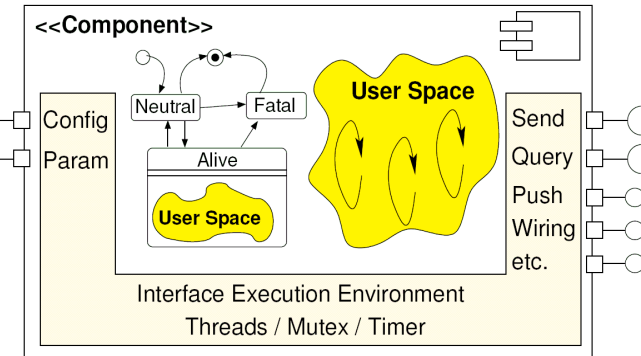
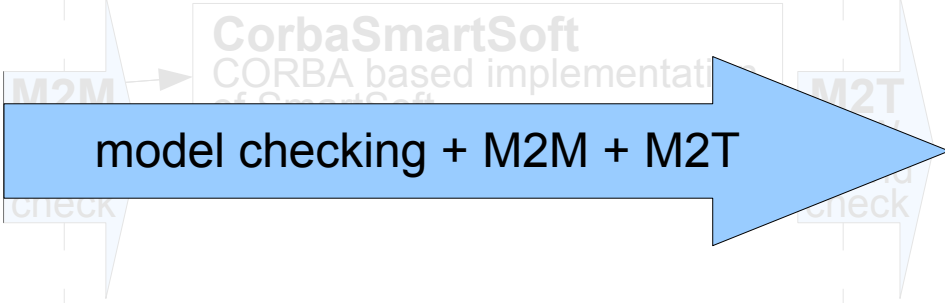
Creating a component workflow

PIM

PSM

PSI

SmartMARS – Metamodel
(Modeling and Analysis of Robotics Systems)



- 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

