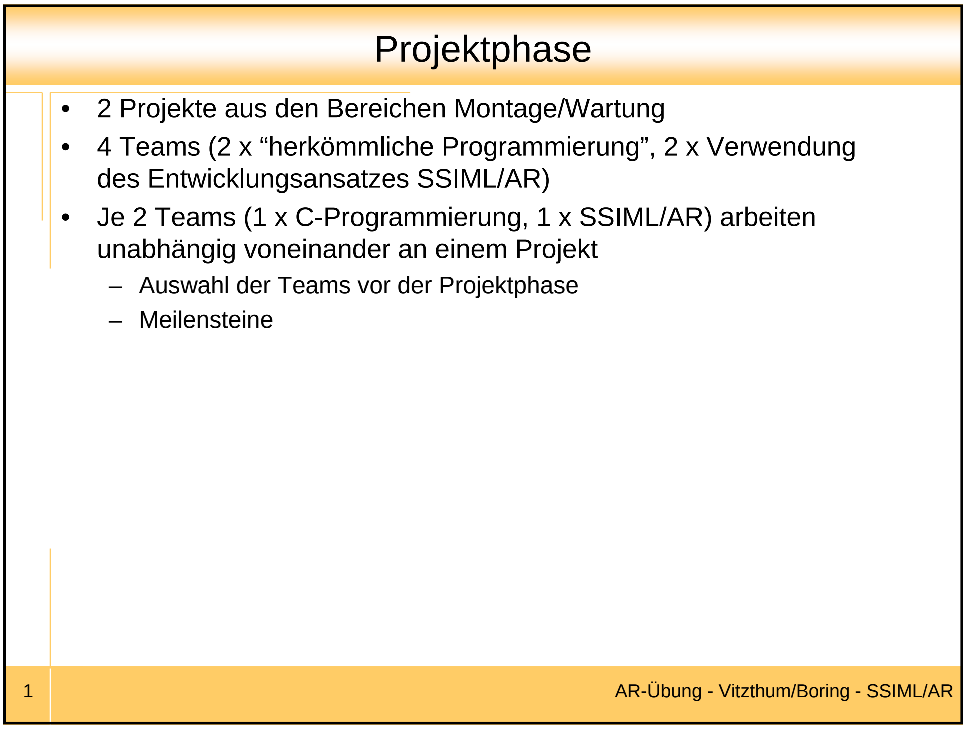
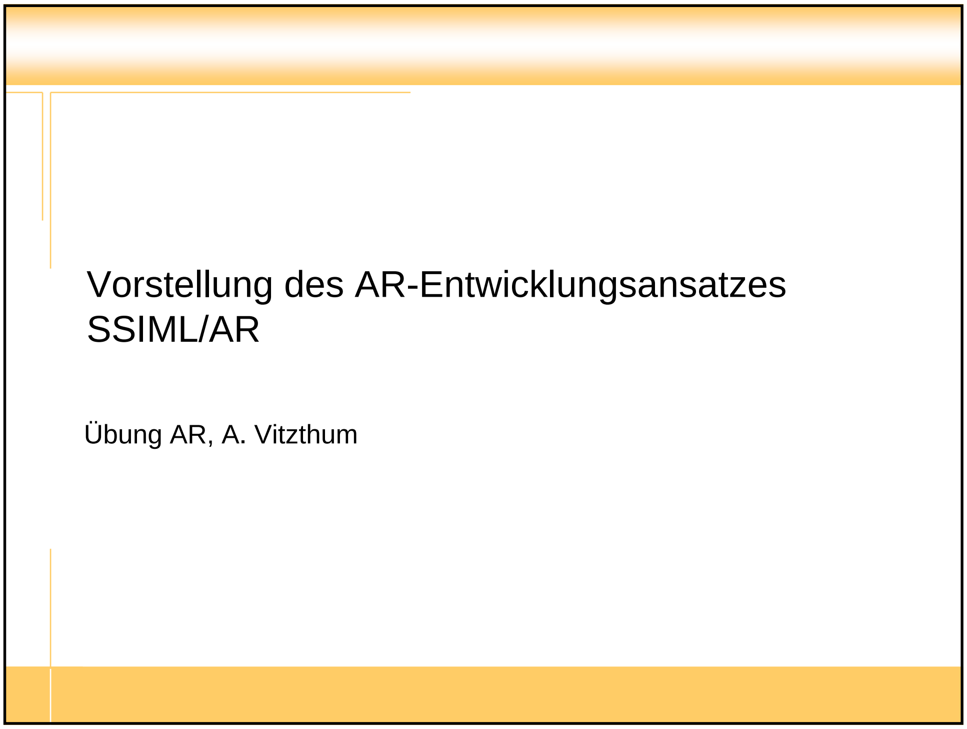


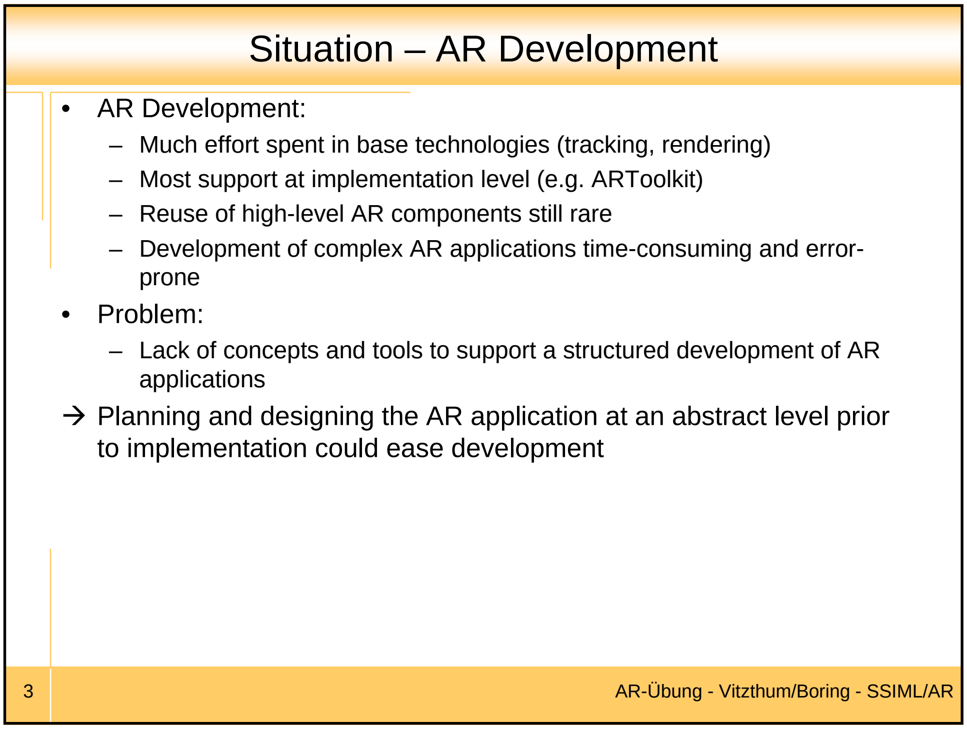
Projektphase

- 2 Projekte aus den Bereichen Montage/Wartung
- 4 Teams (2 x “herkömmliche Programmierung”, 2 x Verwendung des Entwicklungsansatzes SSIML/AR)
- Je 2 Teams (1 x C-Programmierung, 1 x SSIML/AR) arbeiten unabhängig voneinander an einem Projekt
  - Auswahl der Teams vor der Projektphase
  - Meilensteine



Vorstellung des AR-Entwicklungsansatzes SSIML/AR

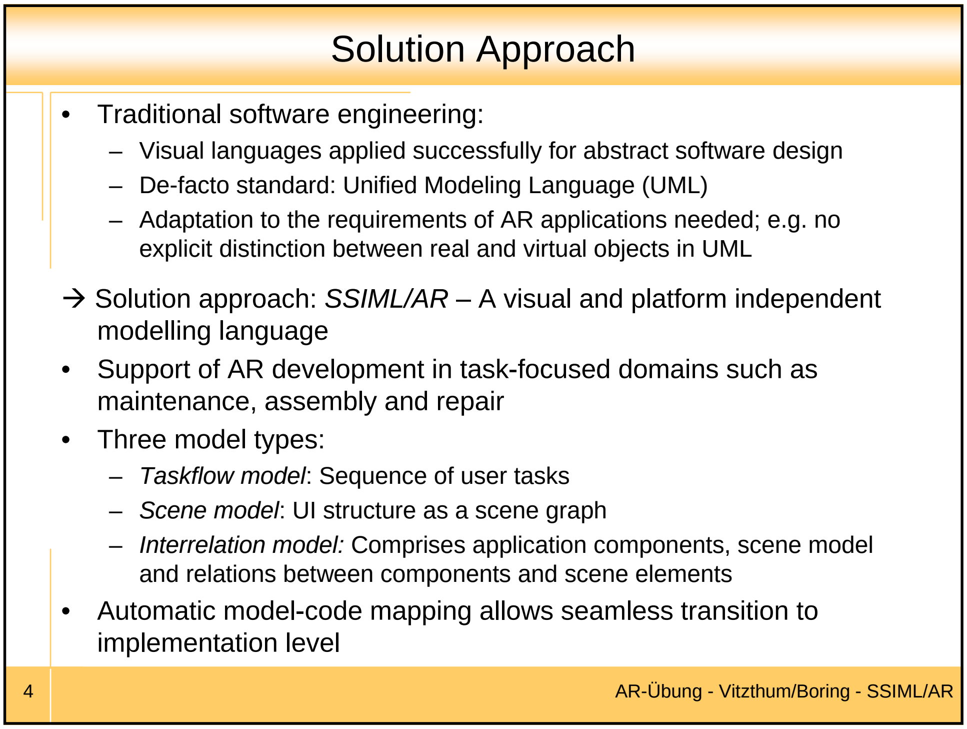
Übung AR, A. Vitzthum



Situation – AR Development

- AR Development:
  - Much effort spent in base technologies (tracking, rendering)
  - Most support at implementation level (e.g. ARToolkit)
  - Reuse of high-level AR components still rare
  - Development of complex AR applications time-consuming and error-prone
- Problem:
  - Lack of concepts and tools to support a structured development of AR applications

→ Planning and designing the AR application at an abstract level prior to implementation could ease development



Solution Approach

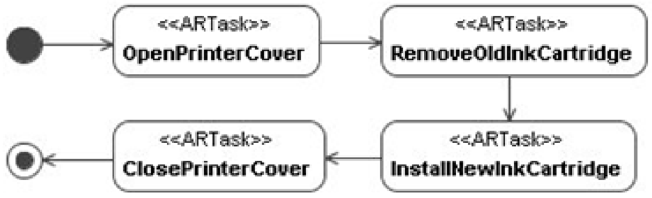
- Traditional software engineering:
  - Visual languages applied successfully for abstract software design
  - De-facto standard: Unified Modeling Language (UML)
  - Adaptation to the requirements of AR applications needed; e.g. no explicit distinction between real and virtual objects in UML

→ Solution approach: *SSIML/AR* – A visual and platform independent modelling language

- Support of AR development in task-focused domains such as maintenance, assembly and repair
- Three model types:
  - *Taskflow model*: Sequence of user tasks
  - *Scene model*: UI structure as a scene graph
  - *Interrelation model:* Comprises application components, scene model and relations between components and scene elements
- Automatic model-code mapping allows seamless transition to implementation level

Example Scenario

- The user is supported by an AR system in exchanging the cartridge of an inkjet printer
- The user has solve a sequence of tasks
- Taskflow model
  - Sequence of tasks is modelled with an UML activity diagram
  - Every task is represented by an UML action
  - Possibility to decompose tasks hierarchically or to model optional tasks



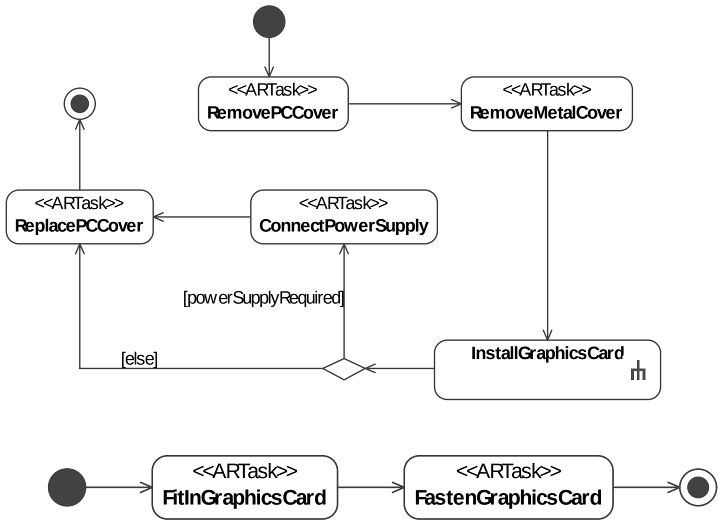
<<ARTask>>
OpenPrinterCover

<<ARTask>>
RemoveOldInkCartridge

<<ARTask>>
InstallNewInkCartridge

<<ARTask>>
ClosePrinterCover

Taskmodel – Example 2 – Graphics Card Installation



<<ARTask>>
RemovePCCover

<<ARTask>>
RemoveMetalCover

<<ARTask>>
ReplacePCCover

<<ARTask>>
ConnectPowerSupply

[powerSupplyRequired]

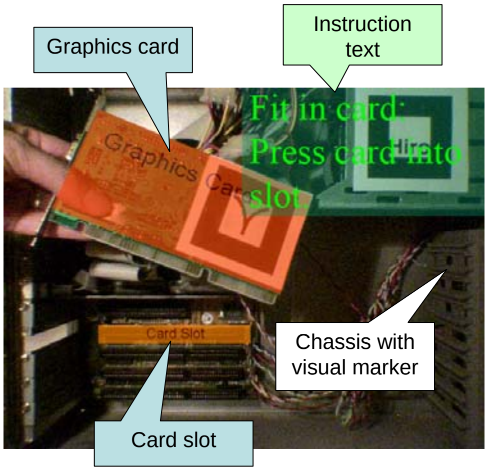
[else]

InstallGraphicsCard

<<ARTask>>
FitInGraphicsCard

<<ARTask>>
FastenGraphicsCard

Example Scenario: Real, virtual and hybrid objects



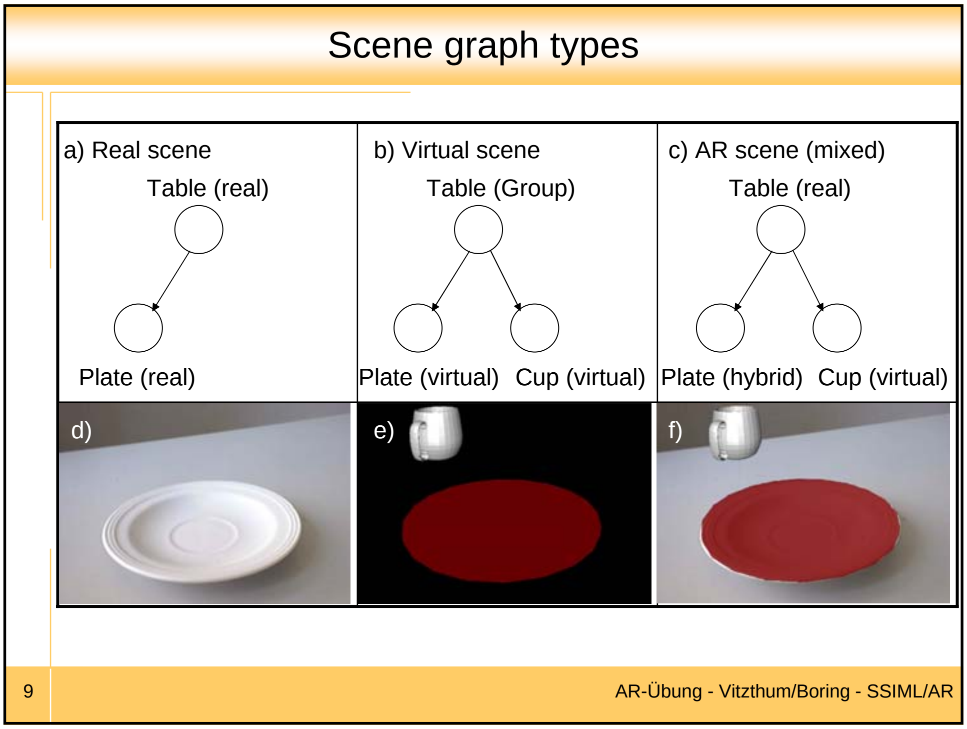
RealObject

VirtualObject

HybridObject

SSIML/AR Scene Model

- Basis: The Scene Structure and Integration Modelling Language (SSIML)
- Models 3D content structures (i.e. the 3D UI structure) using a scene graph-oriented notation
  - Encapsulation and reuse of subgraphs
- Important AR specific scene nodes:
  - *VirtualObject* node (V)
    - Exists only in the virtual world
  - *RealObject* node (R)
    - Physical object in the real world
    - Can be tracked by a tracking device
    - Has a non-visual representation in the virtual IS
    - Can serve as spatial reference point (i.e. a group element) for other objects in the virtual world
  - *HybridObject* node (H)
    - Has a real and a virtual component
    - The virtual component represents the real component in the virtual world (e.g. a 3D model of a printer represents a real printer)
    - Virtual component is aligned with the real component



Scene graph types

a) Real scene

Table (real)

Plate (real)

b) Virtual scene

Table (Group)

Plate (virtual) Cup (virtual)

c) AR scene (mixed)

Table (real)

Plate (hybrid) Cup (virtual)

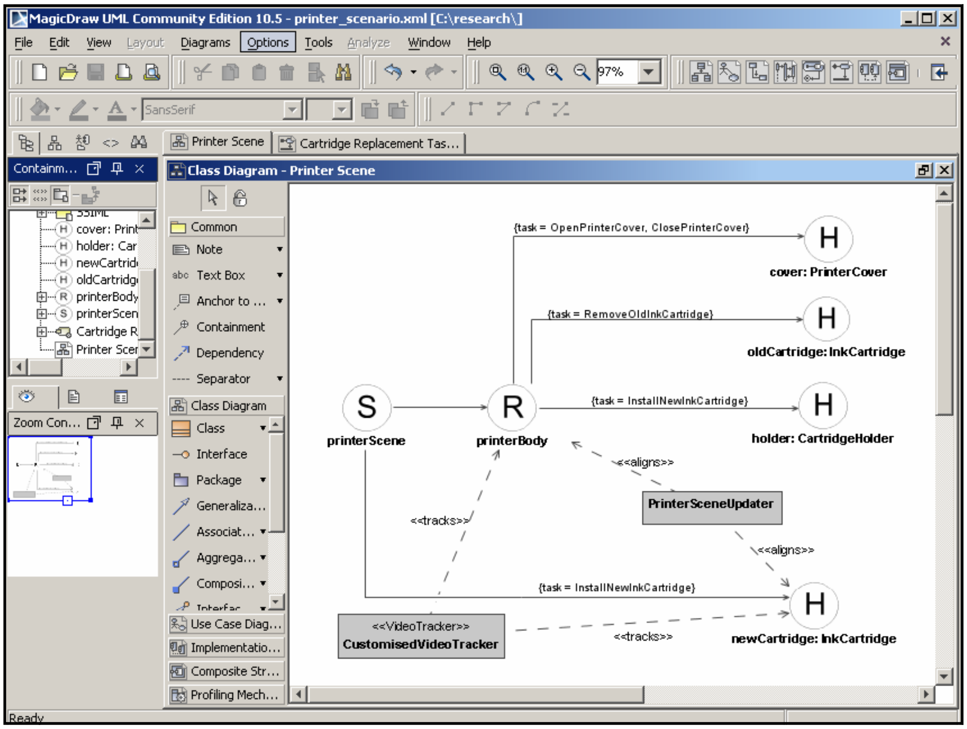
d)

e)

f)

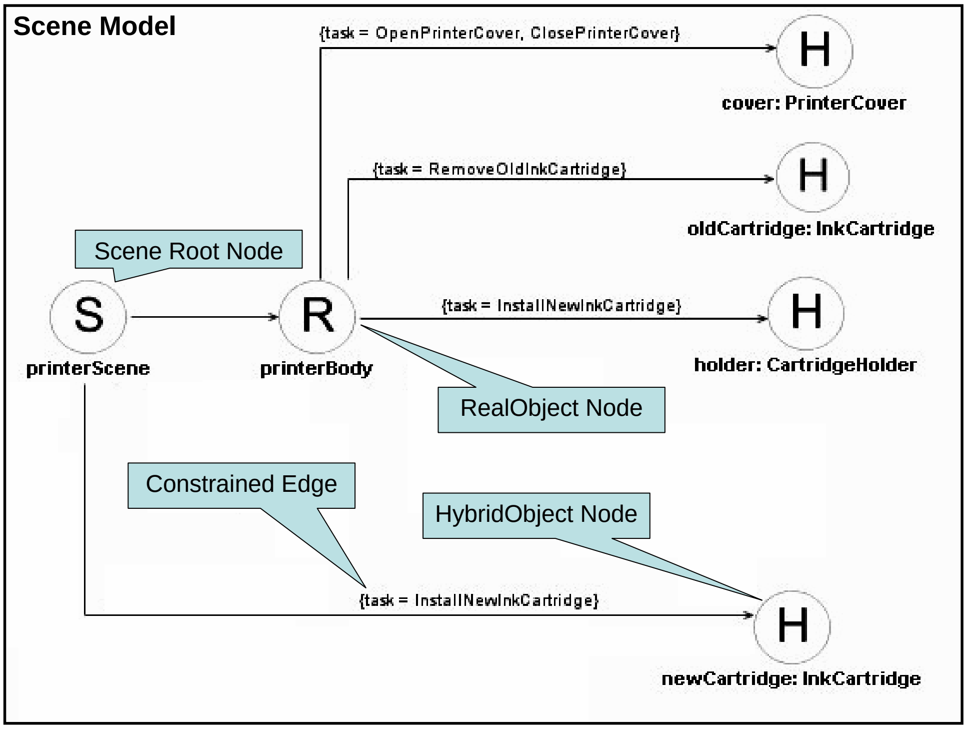
9

AR-Übung - Vitzthum/Boring - SSIML/AR



MagicDraw UML Community Edition 10.5 - printer_scenario.xml [C:\research\]

Class Diagram - Printer Scene



Scene Model

{task = OpenPrinterCover, ClosePrinterCover}
H
cover: PrinterCover
{task = RemoveOldInkCartridge}
H
oldCartridge: InkCartridge
S
printerScene
R
printerBody
{task = InstallNewInkCartridge}
H
holder: CartridgeHolder
{task = InstallNewInkCartridge}
H
newCartridge: InkCartridge

Scene Root Node

RealObject Node

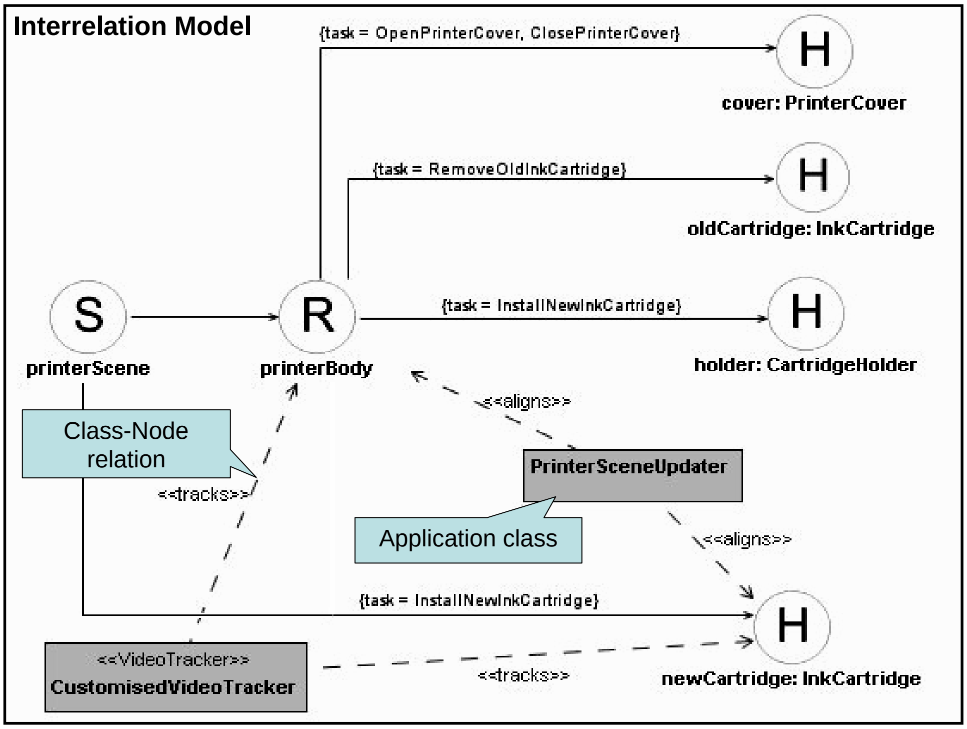
Constrained Edge

HybridObject Node

SSIML/AR – Interrelation Model

- Interrelation model:
  - *<<tracks>>* - relation between tracking software component and real or hybrid object
  - *<<aligns>>* - relation between scene updating component and virtual or hybrid object

Interrelation Model



{task = OpenPrinterCover, ClosePrinterCover}

H

cover: PrinterCover

{task = RemoveOldInkCartridge}

H

oldCartridge: InkCartridge

S

printerScene

R

printerBody

{task = InstallNewInkCartridge}

H

holder: CartridgeHolder

<<aligns>>

Class-Node relation

PrinterSceneUpdater

<<tracks>>

Application class

<<aligns>>

{task = InstallNewInkCartridge}

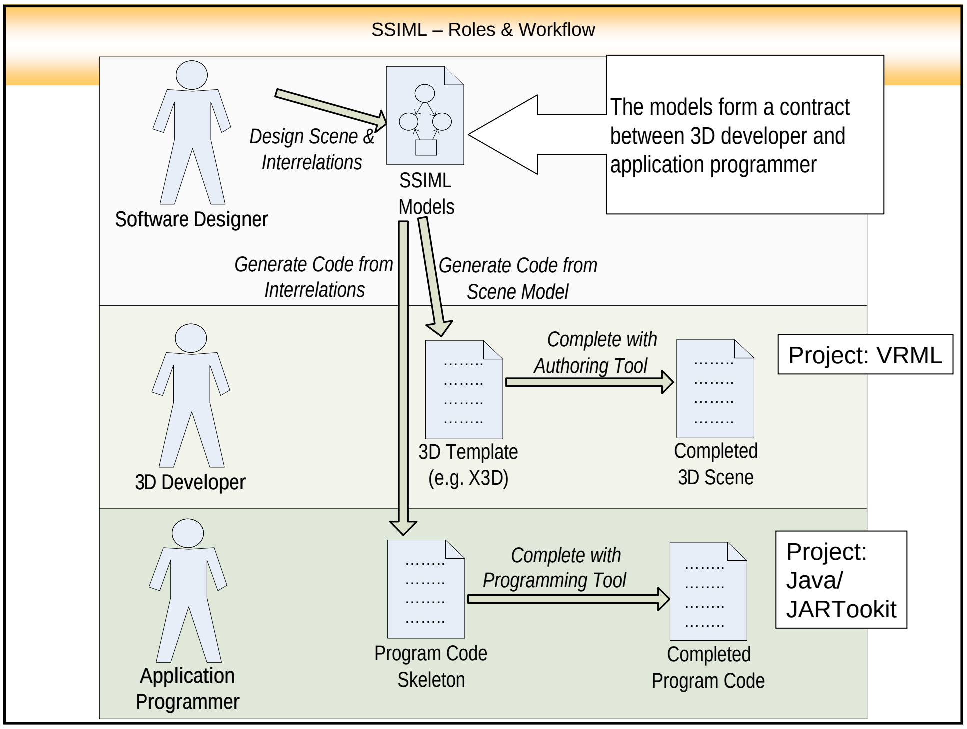
H

<<VideoTracker>>
CustomisedVideoTracker

<<tracks>>

newCartridge: InkCartridge

SSIML – Roles & Workflow



Software Designer

Design Scene & Interrelations

SSIML Models

The models form a contract between 3D developer and application programmer

Generate Code from Interrelations

Generate Code from Scene Model

3D Developer

3D Template (e.g. X3D)

Complete with Authoring Tool

Completed 3D Scene

Project: VRML

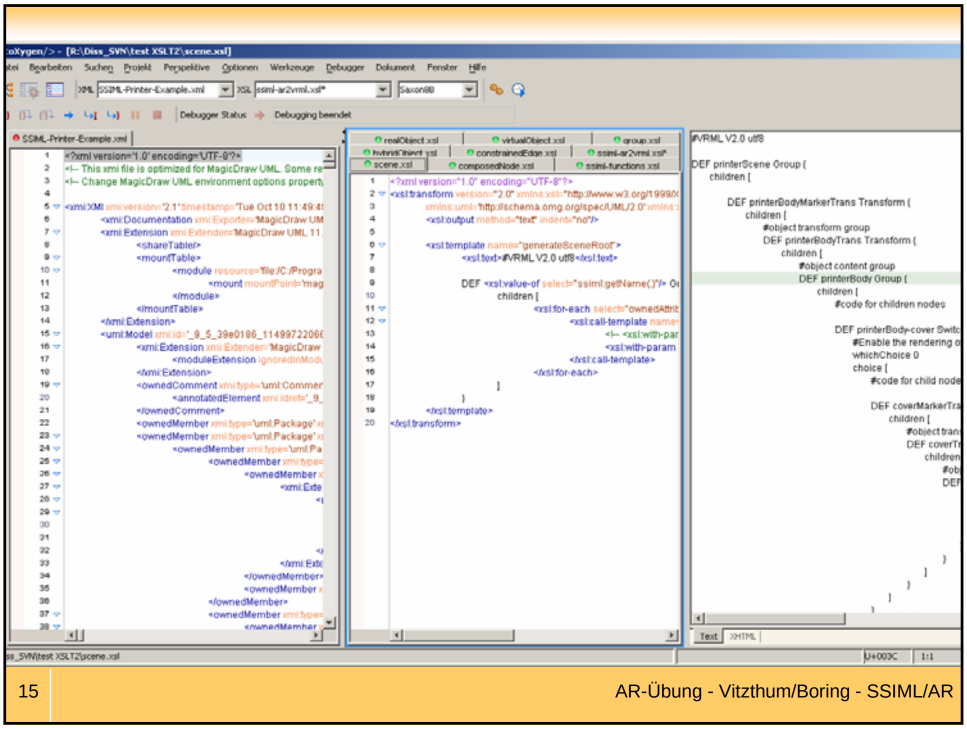
Application Programmer

Program Code Skeleton

Complete with Programming Tool

Completed Program Code

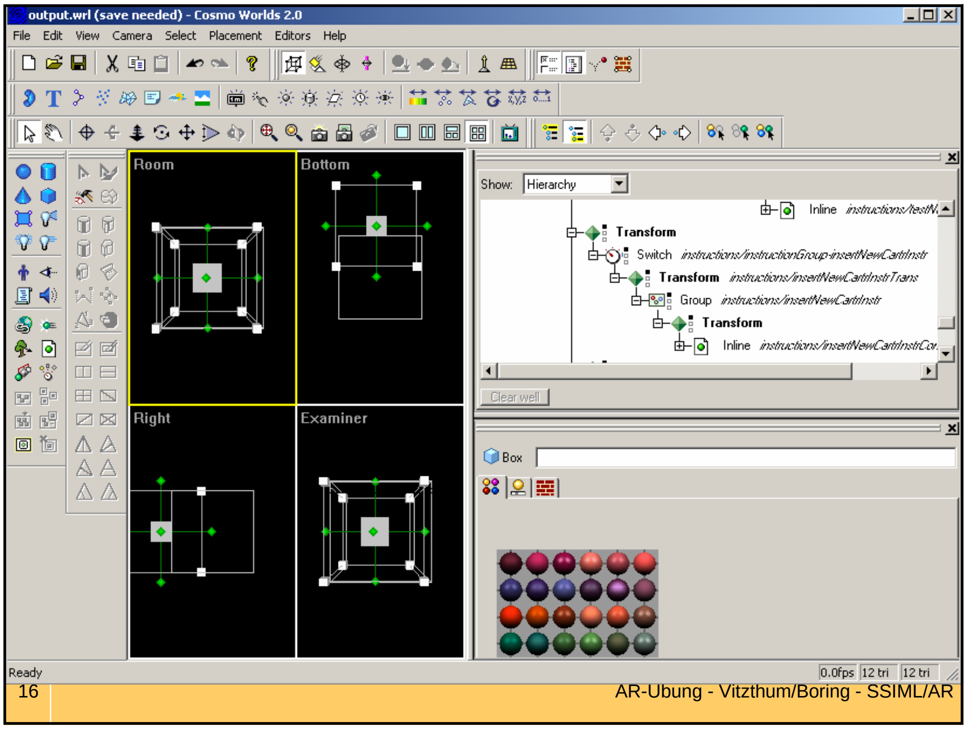
Project: Java/ JARTookit


:oXygen/> - [R:\Diss_SVN\test XSLT2\scene.xsl]
tei Bearbeiten Suchen Projekt Perspektive Optionen Werkzeuge Debugger Dokument Fenster Hilfe
XML SSIML-Printer-Example.xml
XSL ssiml-ar2vrml.xsl*
Saxon8B
Debugger Status
Debugging beendet
SSIML-Printer-Example.xml
realObject.xsl
virtualObject.xsl
group.xsl
hybridObject.xsl
constrainedEdge.xsl
ssiml-ar2vrml.xsl*
scene.xsl
composedNode.xsl
ssiml-functions.xsl
Text
XHTML
ss_SVN\test XSLT2\scene.xsl
U+003C
1:1

```
<?xml version='1.0' encoding='UTF-8'?>
<!-- This xmi file is optimized for MagicDraw UML. Some re
<!-- Change MagicDraw UML environment options property

<xmi:XMI xmi:version='2.1' timestamp='Tue Oct 10 11:49:4
        <xmi:Documentation xmi:Exporter='MagicDraw UM
        <xmi:Extension xmi:Extender='MagicDraw UML 11.
                <shareTable/>
                <mountTable>
                        <module resource='file:/C:/Progra
                                <mount mountPoint='mag
                        </module>
                </mountTable>
        </xmi:Extension>
        <uml:Model xmi:id='_9_5_39e0186_1149972206
                <xmi:Extension xmi:Extender='MagicDraw
                        <moduleExtension ignoredInModu
                </xmi:Extension>
                <ownedComment xmi:type='uml:Commer
                        <annotatedElement xmi:idref='_9_
                </ownedComment>
                <ownedMember xmi:type='uml:Package' x
                <ownedMember xmi:type='uml:Package' x
                        <ownedMember xmi:type='uml:Pa
                                <ownedMember xmi:type=
                                        <ownedMember x
                                                <xmi:Exte
                                                        <





                                                        <
                                                </xmi:Exte
                                        </ownedMember>
                                        <ownedMember x
                                </ownedMember>
                                <ownedMember xmi:type=
                                        <ownedMember x
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/X
        xmlns:uml="http://schema.omg.org/spec/UML/2.0" xmlns:
        <xsl:output method="text" indent="no"/>

        <xsl:template name="generateSceneRoot">
                <xsl:text>#VRML V2.0 utf8</xsl:text>

                DEF <xsl:value-of select="ssiml:getName(.)"/> G
                        children [
                                <xsl:for-each select="ownedAttrib
                                        <xsl:call-template name=
                                                <!-- <xsl:with-par
                                                <xsl:with-param
                                        </xsl:call-template>
                                </xsl:for-each>
                        ]
                }
        </xsl:template>
</xsl:transform>
```

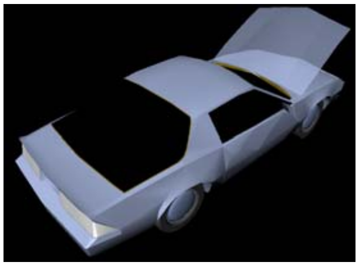
```
#VRML V2.0 utf8

DEF printerScene Group {
    children [

        DEF printerBodyMarkerTrans Transform {
            children [
                #object transform group
                DEF printerBodyTrans Transform {
                    children [
                        #object content group
                        DEF printerBody Group {
                            children [
                                #code for children nodes

                                DEF printerBody-cover Switc
                                    #Enable the rendering o
                                    whichChoice 0
                                    choice [
                                        #code for child node

                                        DEF coverMarkerTra
                                            children [
                                                #object tran
                                                DEF coverTr
                                                    children
                                                        #obj
                                                        DEF






                                                }
                                            ]
                                        }
                                    ]
                                }
```

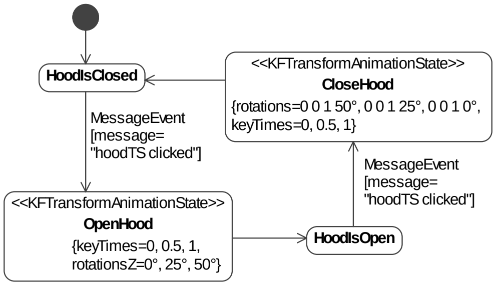

output.wrl (save needed) - Cosmo Worlds 2.0
File Edit View Camera Select Placement Editors Help
Room
Bottom
Right
Examiner
Show: Hierarchy
Clear well
Box
Ready
0.0fps
12 tri
12 tri

```
Inline  instructions/testN
Transform
    Switch  instructions/instructionGroup-insertNewCartInstr
        Transform  instructions/insertNewCartInstrTrans
            Group  instructions/insertNewCartInstr
                Transform
                    Inline  instructions/insertNewCartInstrCo
```

Extensions

Behaviour





HoodIsClosed

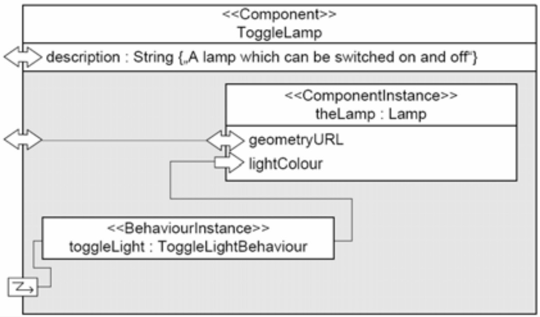
MessageEvent [message= "hoodTS clicked"]

<<KFTransformAnimationState>>
OpenHood
{keyTimes=0, 0.5, 1, rotationsZ=0°, 25°, 50°}

HoodIsOpen

MessageEvent [message= "hoodTS clicked"]

<<KFTransformAnimationState>>
CloseHood
{rotations=0 0 1 50°, 0 0 1 25°, 0 0 1 0°, keyTimes=0, 0.5, 1}



<<Component>>
ToggleLamp

description : String {„A lamp which can be switched on and off"}

<<ComponentInstance>>
theLamp : Lamp

geometryURL

lightColour

<<BehaviourInstance>>
toggleLight : ToggleLightBehaviour

3D Components