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. Vitzhum
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 to 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

![Taskflow Diagram](image)

Taskmodel – Example 2 – Graphics Card Installation

![Taskmodel Diagram](image)
Example Scenario: Real, virtual and hybrid objects

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)
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

SSIML – Roles & Workflow

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

1. **Software Designer**
   - Design Scene & Interrelations
   - Generate Code from Interrelations

2. **3D Developer**
   - Complete with Authoring Tool
   - 3D Template (e.g. X3D)
   - Completed 3D Scene

3. **Application Programmer**
   - Complete with Programming Tool
   - Program Code Skeleton
   - Completed Program Code

Project: VRML

Project: Java/ JARTookit
Extensions

**Behaviour**

- **HoodIsClosed**
  - MessageEvent [message: "hoodTS clicked"]

- **CloseHood**
  - $<<\text{KFTransformAnimationState}}>>$
  - rotations $= 0 \ 0 \ 1 \ 50^\circ, \ 0 \ 0 \ 1 \ 25^\circ, \ 0 \ 0 \ 1 \ 0^\circ$
  - keyTimes $= 0, \ 0.5, \ 1$

- **OpenHood**
  - $<<\text{KFTransformAnimationState}}>>$
  - rotationsZ $= 0^\circ, \ 25^\circ, \ 50^\circ$

- **HoodIsOpen**

**3D Components**

- **<<Component>>**
  - **ToggleLamp**
    - description : String (A lamp which can be switched on and off)
      - **<<ComponentInstance>>**
        - theLamp : Lamp
          - geometryURL
          - lightColour

- **<<BehaviourInstance>>**
  - toggleLight : ToggleLightBehaviour

AR-Übung - Vitzthum/Boring - SSIML/AR