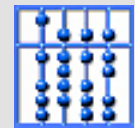


# Software and Tools for Hybrid User Interfaces

TU München

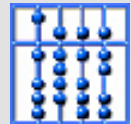
1. 2. 2005

Christian Sandor  
sandor@in.tum.de



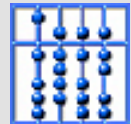
# Outline

- Introduction
  - What are Hybrid User Interfaces?
  - Challenges in Developing Hybrid UIs
  - Proposed Solution
  - Research Hypothesis
  - Approach
  - Research Contributions
- Problem Details
- My Approach
- Conclusion

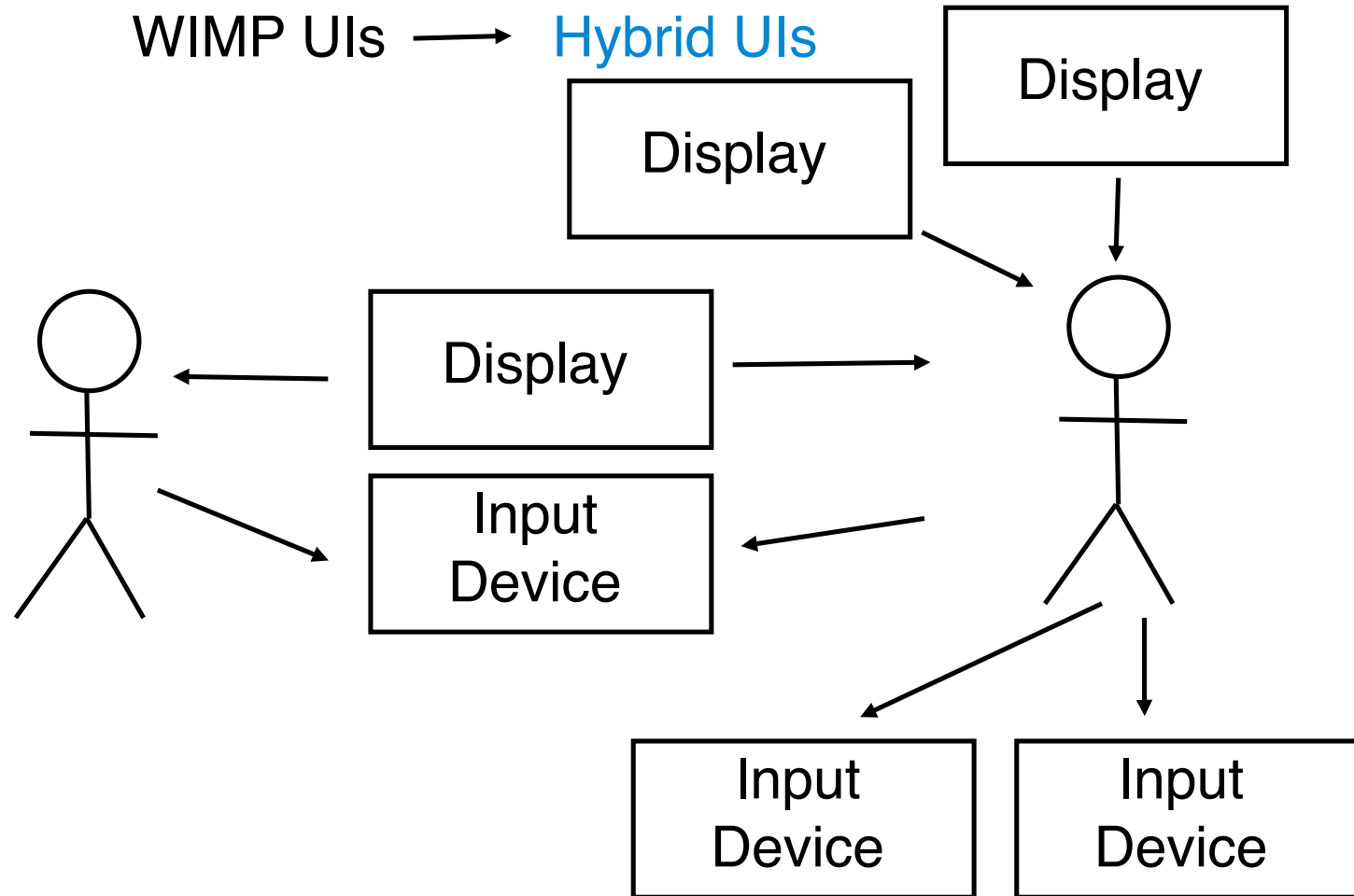


# What are Hybrid User Interfaces? (1/2)

- Originally defined by Feiner as
  - Mixing AR and 2D User Interfaces
  - Coordination of multiple screens
- I extend the definition:
  - Inherent multimodal input
  - Tangible input
  - Multiple users
  - Displaying 3D content

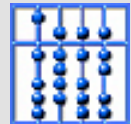


# What are Hybrid User Interfaces? (2/2)



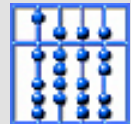
# Challenges in Hybrid UI Development

1. Missing software infrastructure:  
No out-of-the-box infrastructure is available
  
2. UI metaphors/idioms are unclear:  
Collaborative work is mandatory  
Tools for novices are necessary



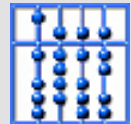
# Proposed Solution

1. Missing software infrastructure:  
Build infrastructure ("kernel")
2. UI metaphors/idioms are unclear:  
Build tools for experimentation on top of kernel
  - Authoring at runtime
  - Develop Hybrid User Interfaces **with** Hybrid User Interfaces



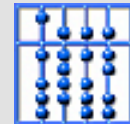
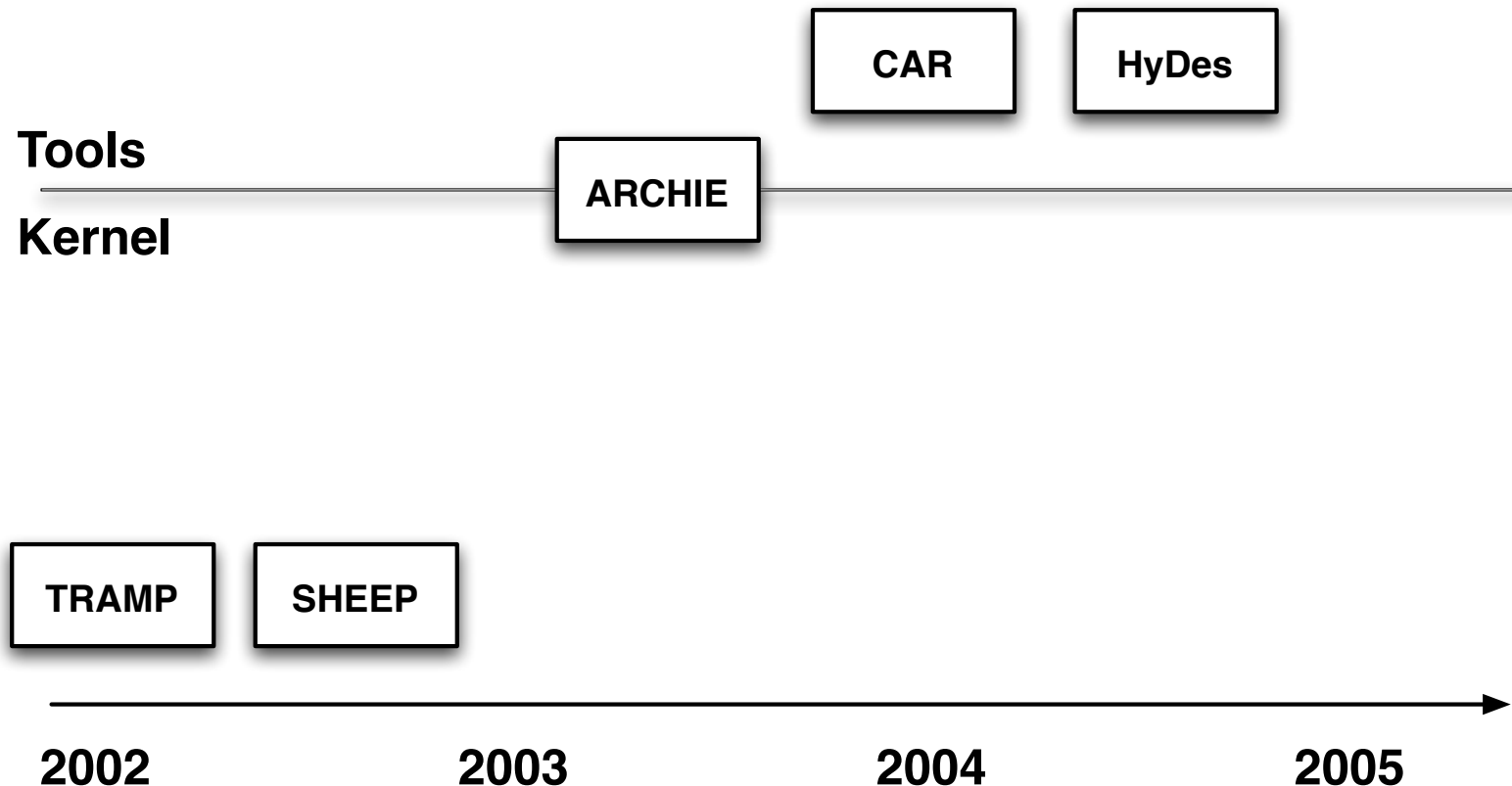
# Research Hypotheses

1. My UI kernel enables rapid prototyping of Hybrid User Interfaces
2. Interactive tools for novices that allow building Hybrid User Interfaces with Hybrid User Interfaces during runtime can help in developing metaphors



# Approach

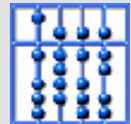
Build kernel and tools in 5 projects (each 1/2 year)





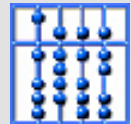
# Research Contributions

1. “Interaction Space” model as analysis model for Hybrid User Interfaces
2. Reusable, extensible infrastructure for Hybrid User Interfaces
3. Metric for tools to develop Hybrid User Interfaces
4. Implementation of prototypical tools:
  - a) CAR: Toolchain for developing User Interfaces in cars
  - b) HyDes: Building simple Hybrid User Interfaces immersively



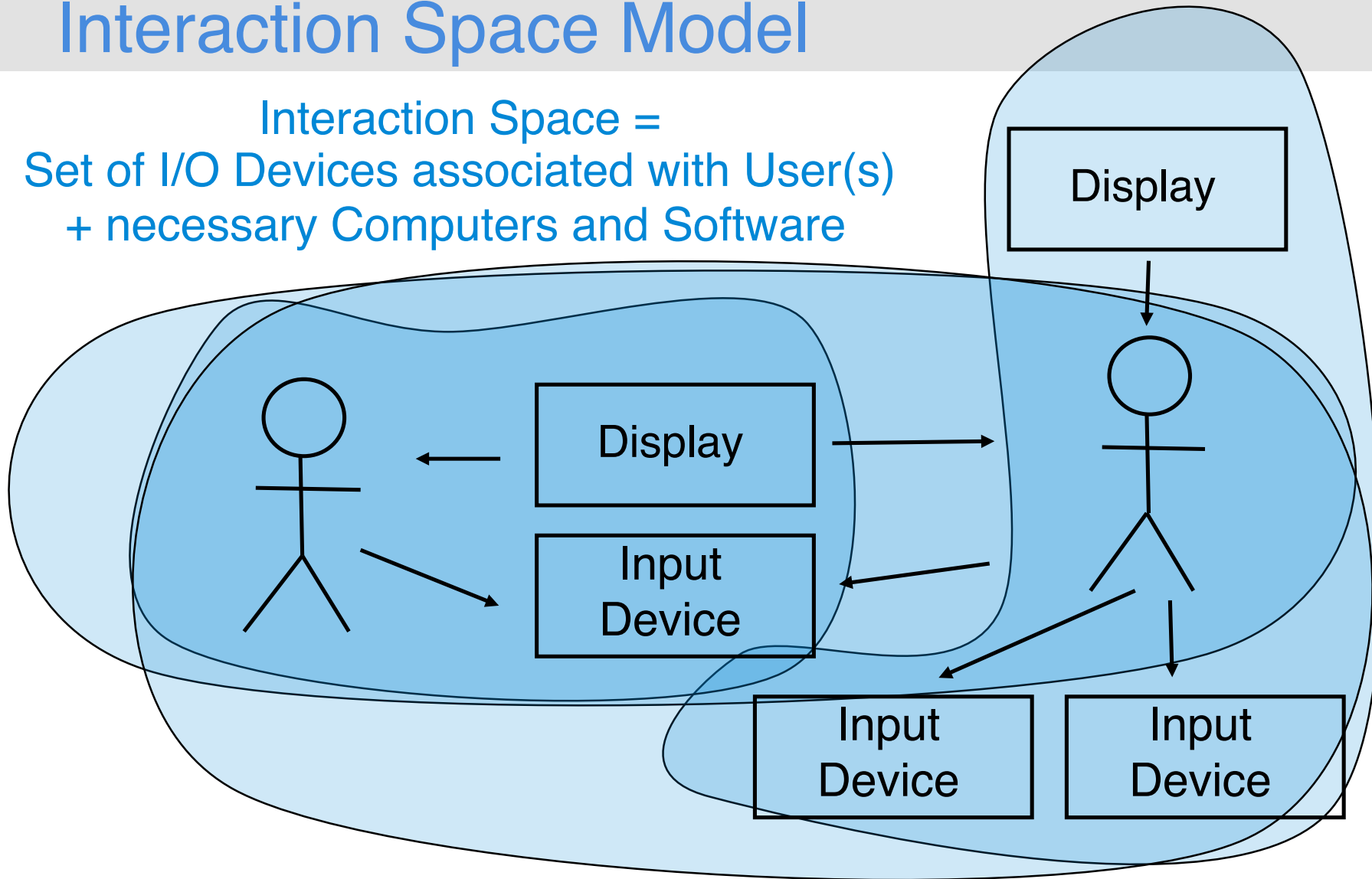
# Outline

- Introduction
- Problem Details
  - Kernel
    - Interaction Space model
    - Summary of Kernel Requirements
  - Tools
    - Why Tools For Novices?
    - Sample Tools
- My Approach
- Conclusion



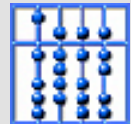
# Interaction Space Model

Interaction Space =  
Set of I/O Devices associated with User(s)  
+ necessary Computers and Software



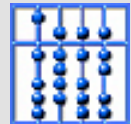
# Summary of Kernel Requirements

- Support for Interaction Spaces
  - Multiple shared input devices and displays for multiple users
  - Bidirectional multimodality
  - Distribution
  - Flexible reconfiguration (automatic or at least pleasant)
- Tangible AR:
  - Combining real world and virtual content
  - Efficiency
  - Support for non-linear interactions
  - Dataflow networks



# Why Tools for Novices?

- Exploring metaphors/paradigms requires participation of various researchers:
  - Human-factors
  - Psychology
  - Domain experts
  - Endusers
- These groups often do not have mathematical/computer science background
- Desired goals:
  - Extremely simple usage
  - Short turnaround times



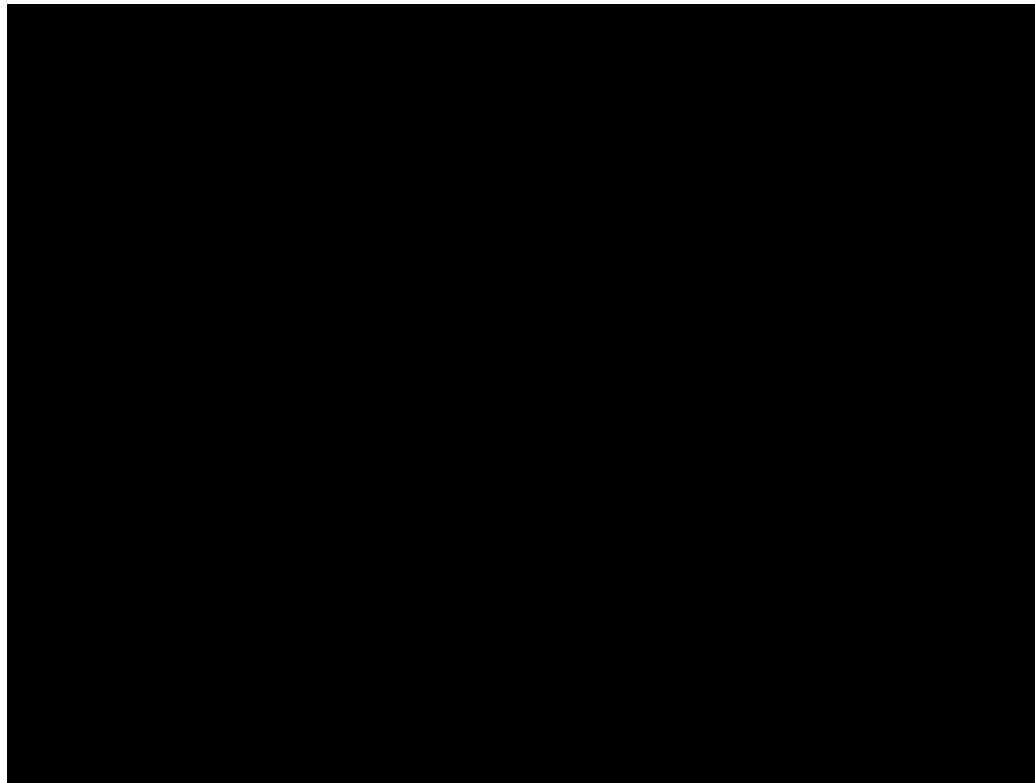
# Sample Tools (1/3)

- 3D Modelling: SmoothTeddy (Takeo Igarashi, 2003)



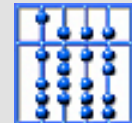
## Sample Tools (2/3)

- 3D Animation: MotionDoodles (Matthew Thorne, David Burke, and Michiel van de Panne. 2004)



## Sample Tools (3/3)

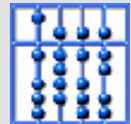
- AR Authoring: Immersive Authoring for Tangible Augmented Reality (Gerard Kim, 2004)





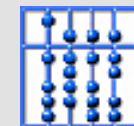
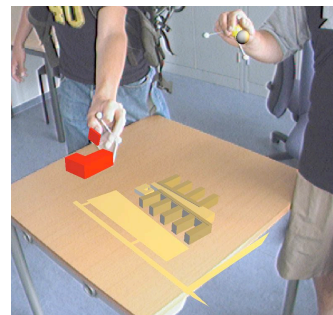
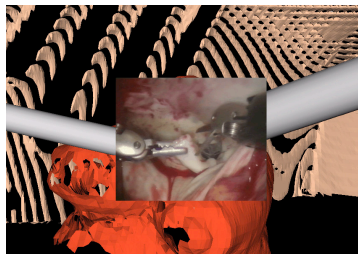
# Outline

- Introduction
- Problem Details
- My Approach
  - Kernel
    - DWARF
    - Component-based Hybrid UI Toolkit
    - Example: SHEEP
  - Tools
    - Viewing Volumes Visualization
    - Context-aware animations
    - Hybrid Designer
- Conclusion



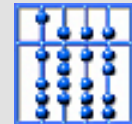
# DWARF in Brief

- Framework for mobile AR in ubiquitous computing environments
- Example scenarios:
  - Navigation (Pathfinder)
  - Maintenance (TRAMP)
  - Multi-player game (SHEEP)
  - Collaborative building design (ARCHIE)
  - Medical (HEART)



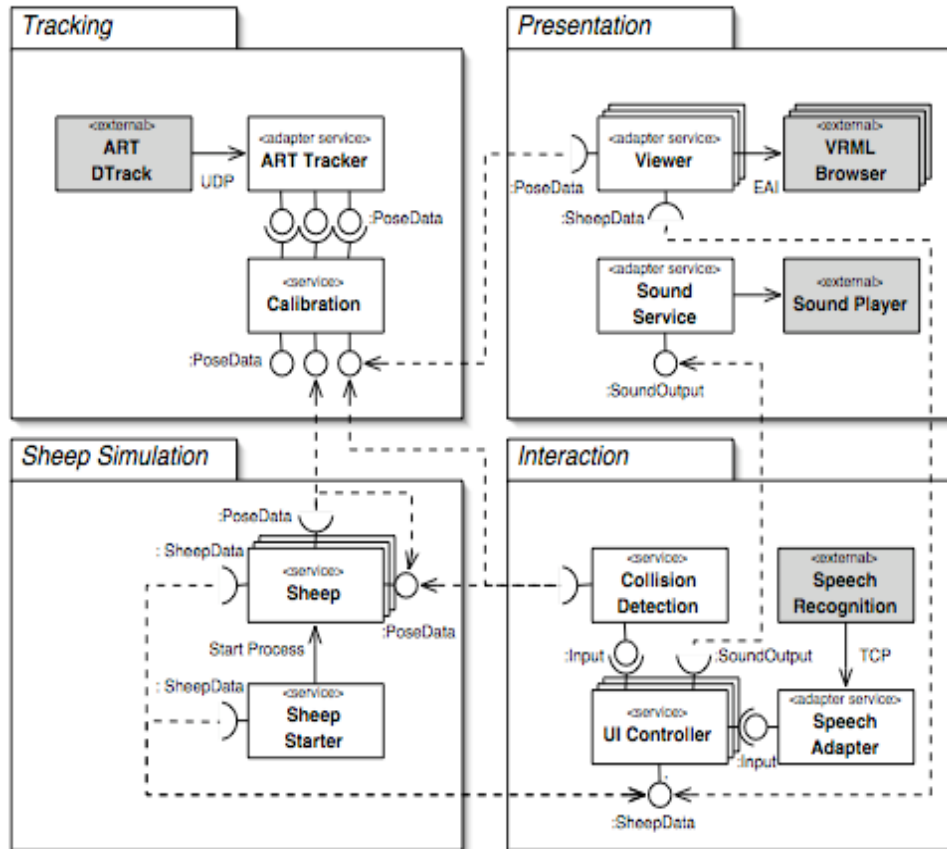
# DWARF Project Status

- Distributed Wearable Augmented Reality Framework (ISMAR01): Component-Based Peer-to-Peer Framework
- Design Goals:
  - Flexible: dynamic reconfiguration (context, component discovery)
  - Responsive: real-time communication (CORBA, shared memory)
  - Ad hoc distribution
  - Independent of: OS, programming language, hardware
- Relatively Mature:
  - Open Source (LGPL)
  - 10 students projects at TUM (3 years, team sizes: 5-50)
  - Used by 5 research groups world-wide (Japan, Germany, Singapore)

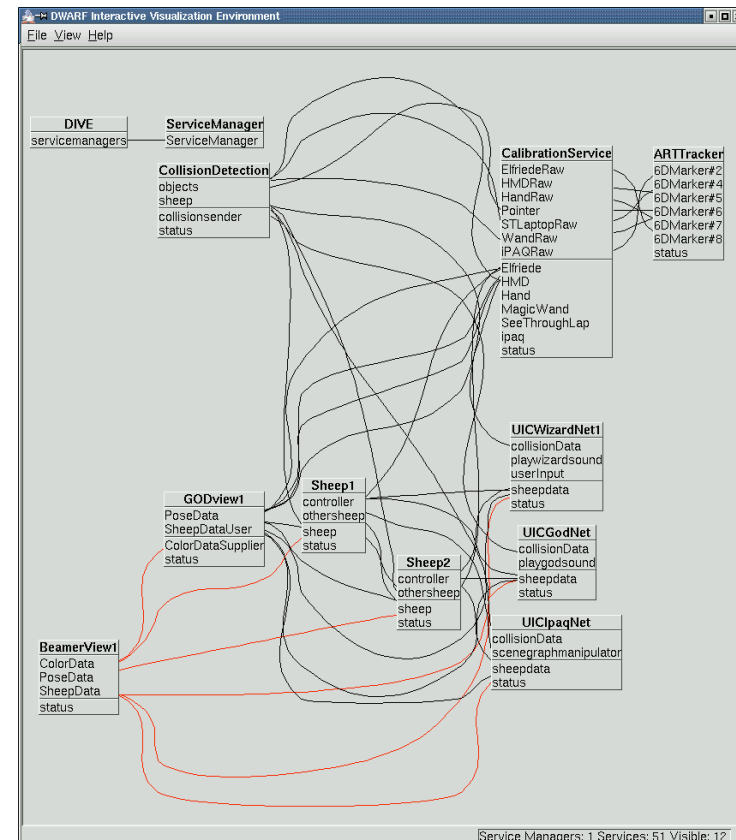


# DWARF Consists of Distributed Services

During design...

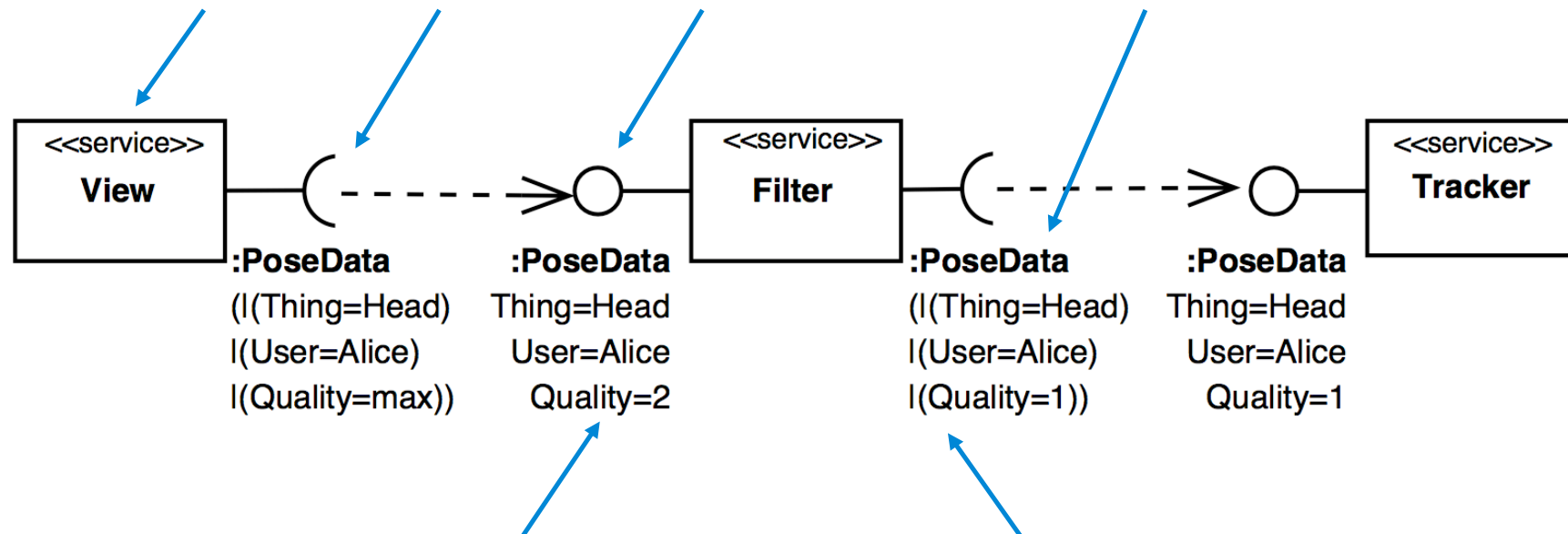


...and at run time



# Services in DWARF

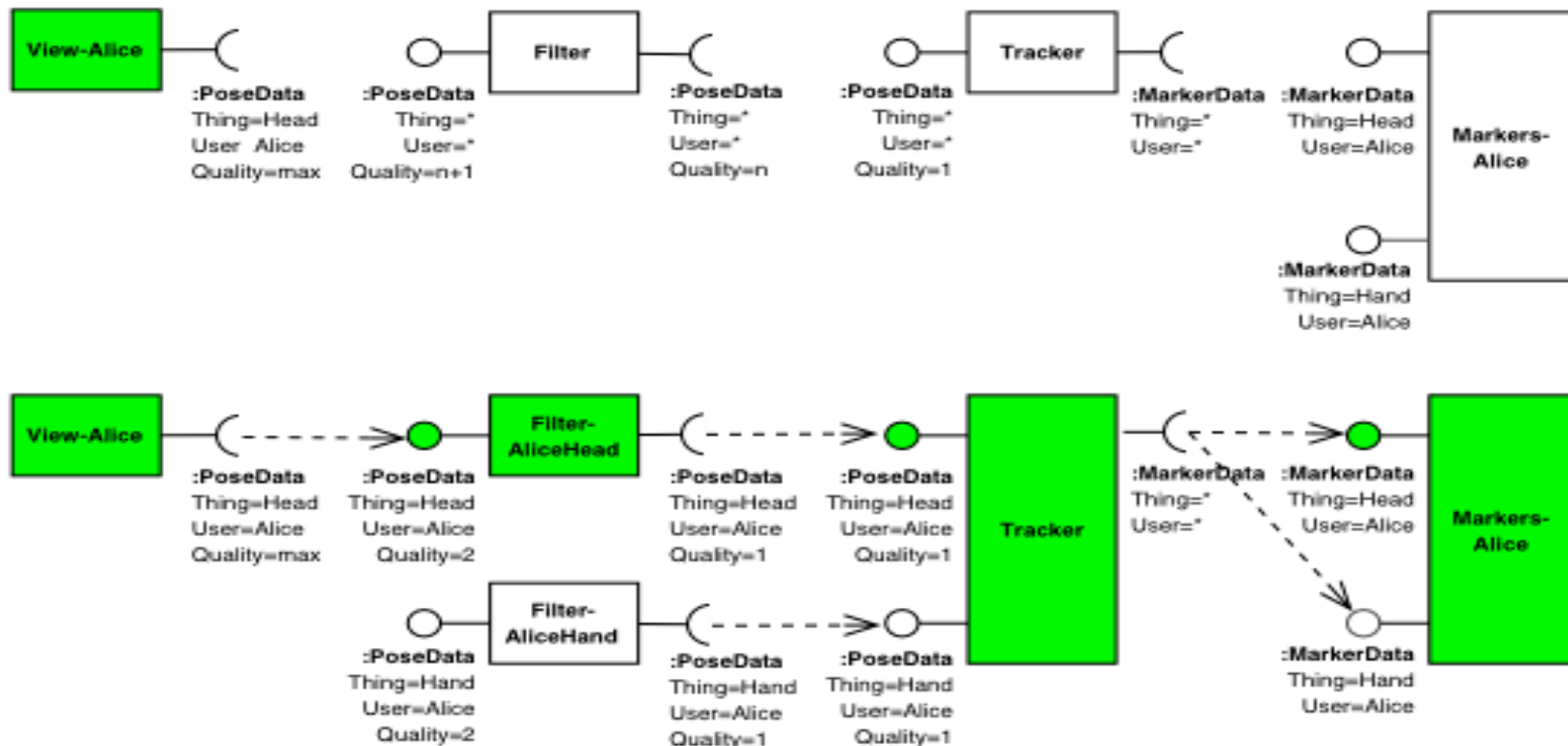
- Services have Needs and Abilities, which have types



- Abilities have Attributes, Needs have Predicates.
- These can be set at runtime.
- One service's Needs depend on other services' Abilities.
- Distributed CORBA-based Middleware establishes connections for communication between services (management, lookup, connection)

# Branching and Selection

- Middleware finds graphs of “potential” services in background
- When user (or application) requests a particular ability of a particular service, the appropriate chain is started up

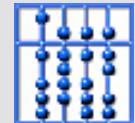


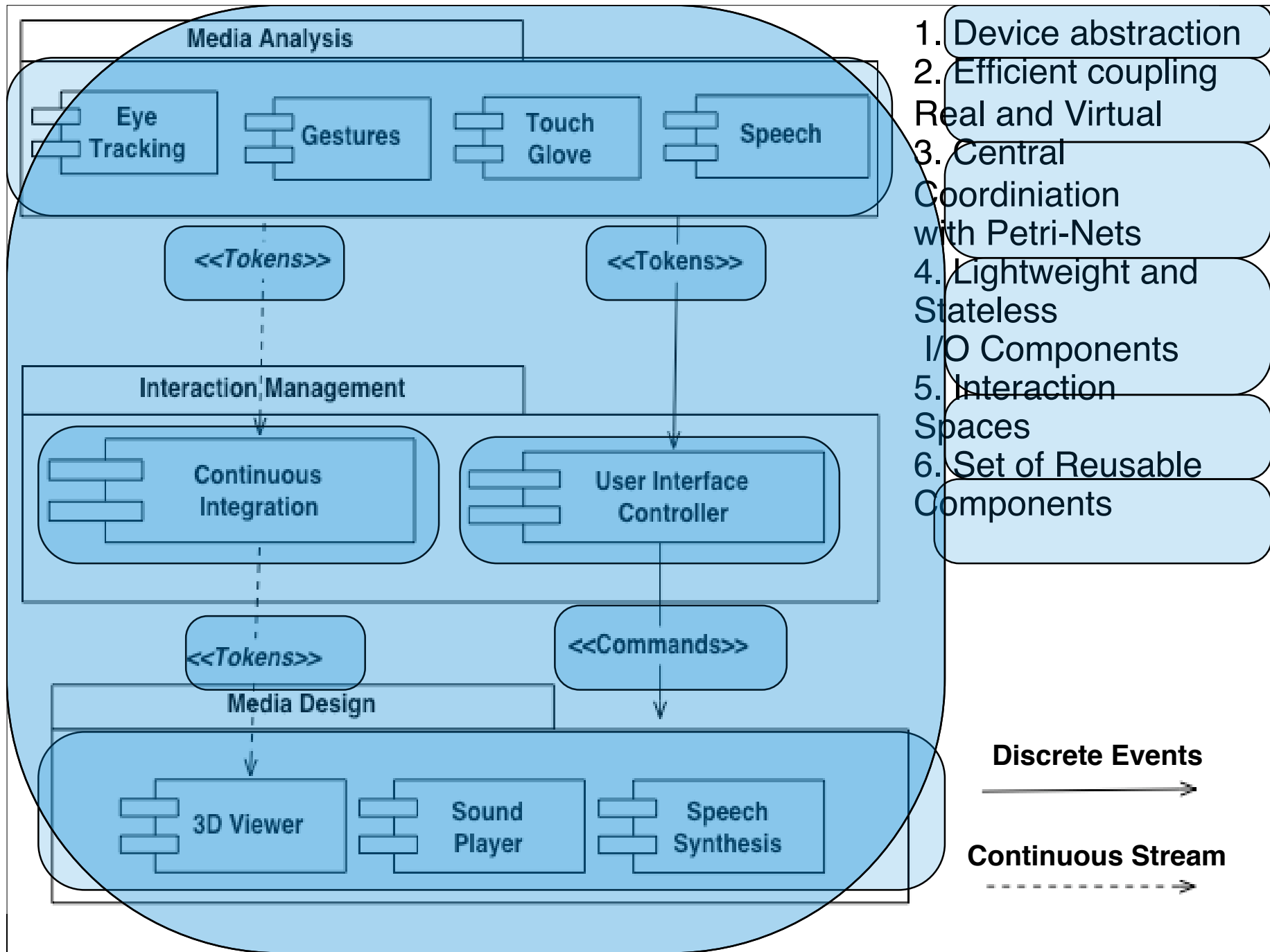
# Component-based Hybrid UI Toolkit



Christian Sandor  
(sandor@in.tum.de)

Software and Tools for Hybrid User Interfaces  
TU München



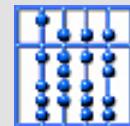
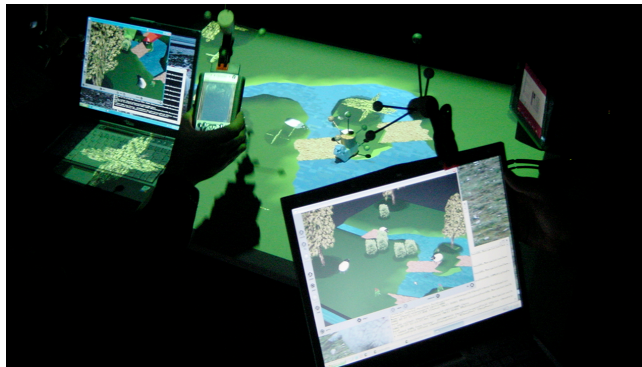
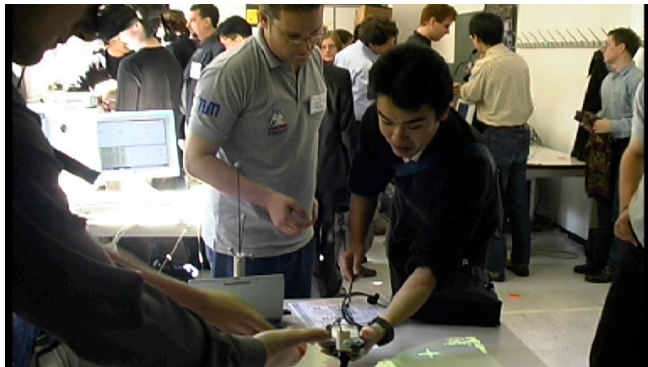


1. Device abstraction
2. Efficient coupling Real and Virtual
3. Central Coordination with Petri-Nets
4. Lightweight and Stateless I/O Components
5. Interaction Spaces
6. Set of Reusable Components

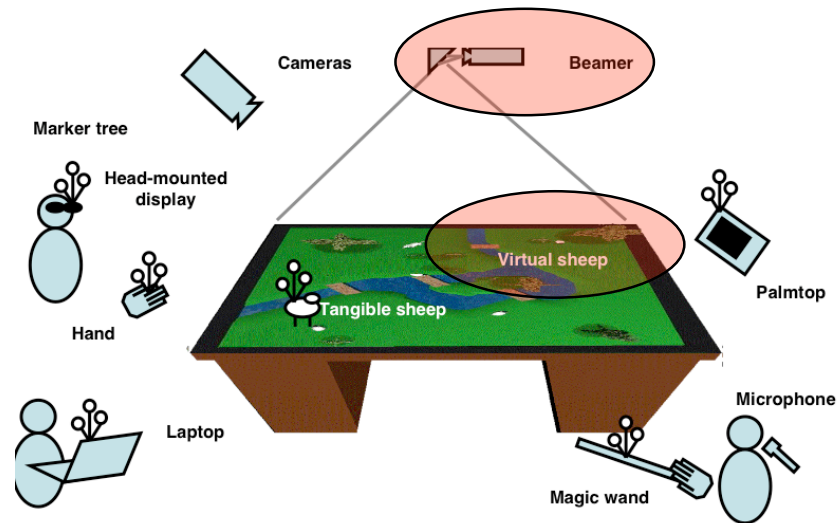


# Example: SHEEP

- The Shared Environment Entertainment Pasture
- Demo at ISMAR02, paper at ISMAR03
- Shown on lots of occasions afterwards
- Multi-user game employing Hybrid UIs

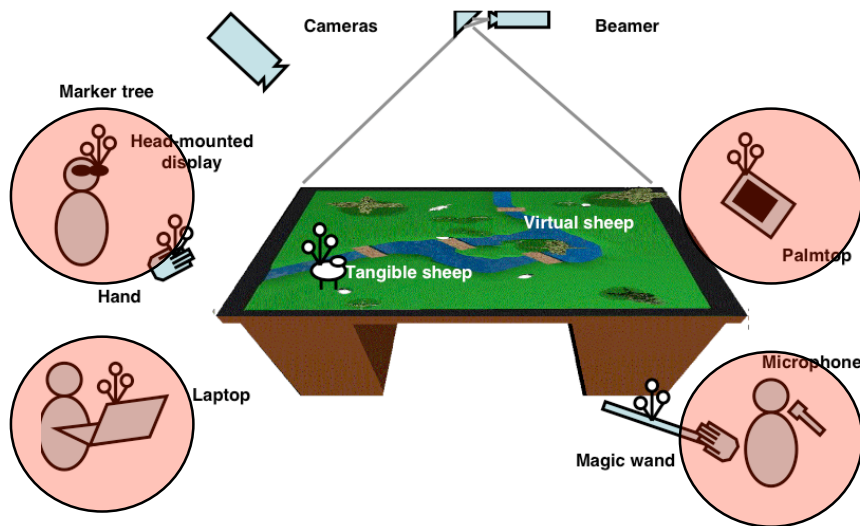


# SHEEP - User's View (1)



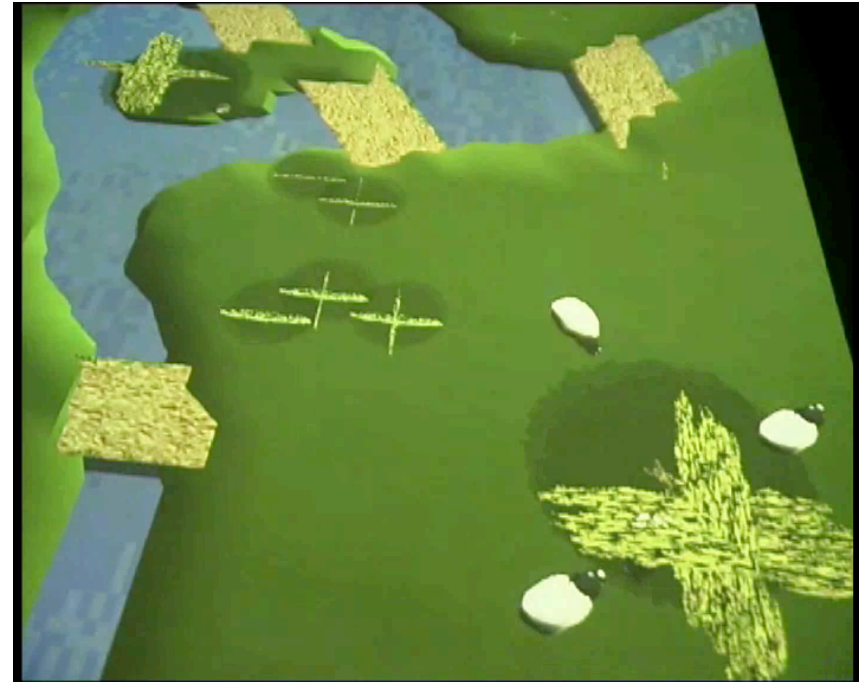
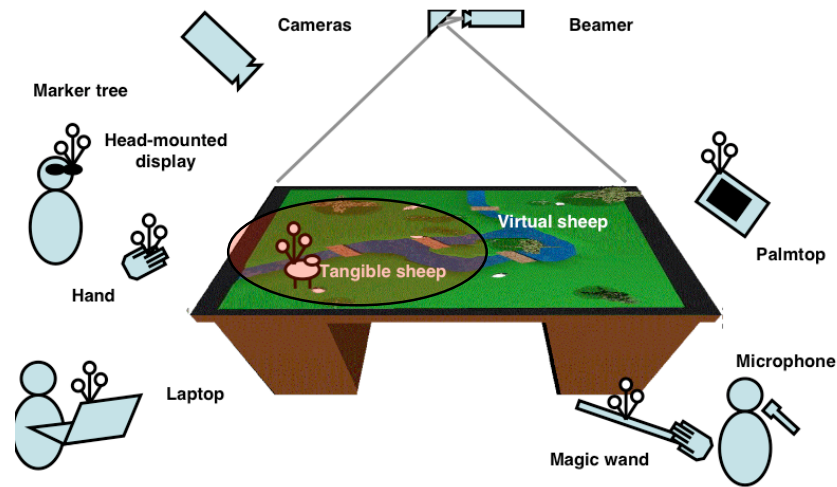
Landscape containing virtual sheep is projected onto a table

# SHEEP - User's View (2)



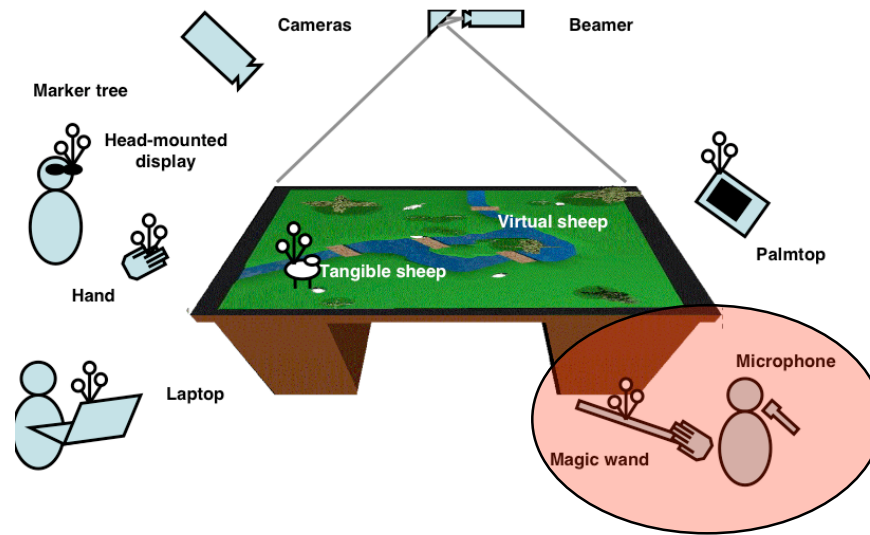
Players have a variety of devices for interaction

# SHEEP - User's View (3)



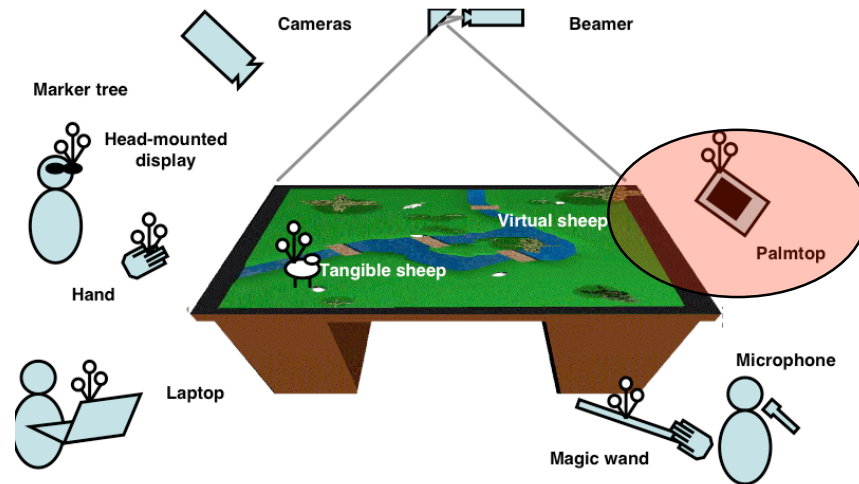
A tangible plastic sheep attracts the virtual sheep

# SHEEP - User's View (4)



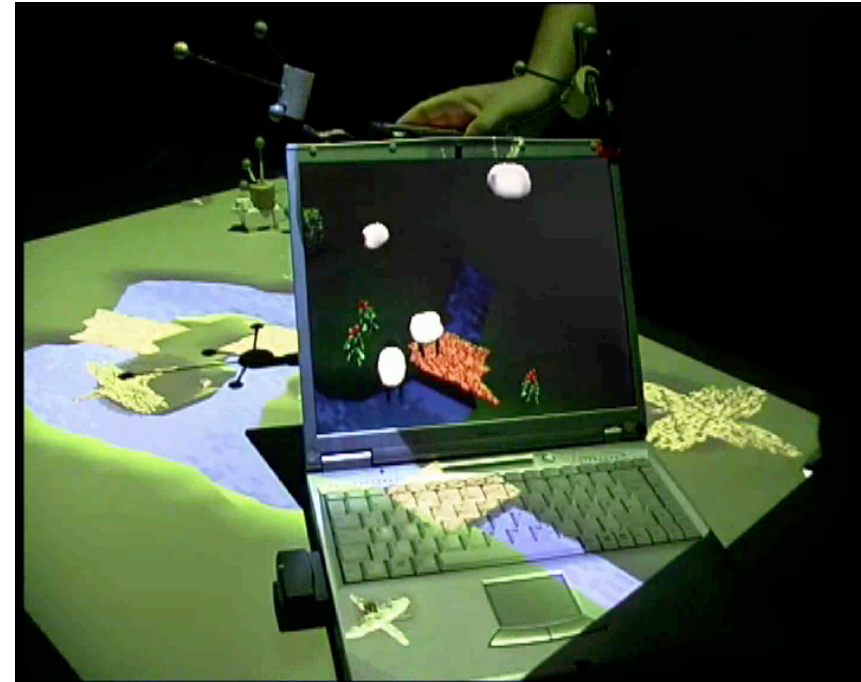
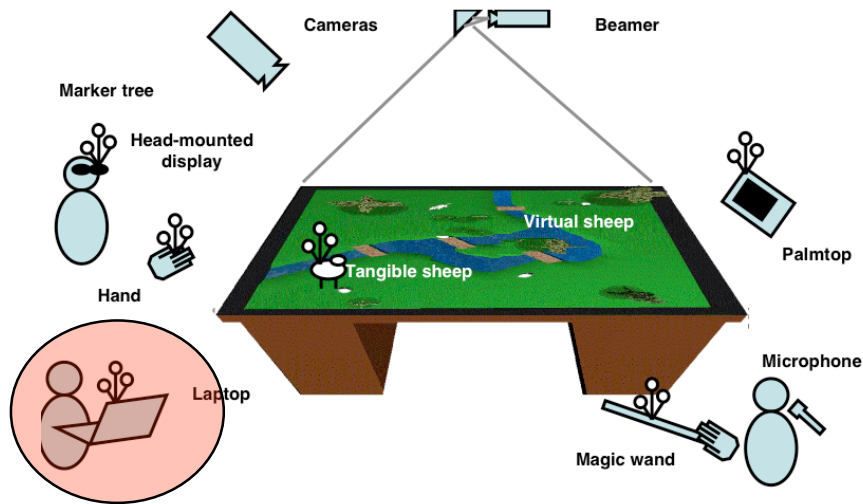
Point-and-Speak is used to create and remove sheep

# SHEEP - User's View (5)



Sheep can be scooped and dropped with an iPAQ

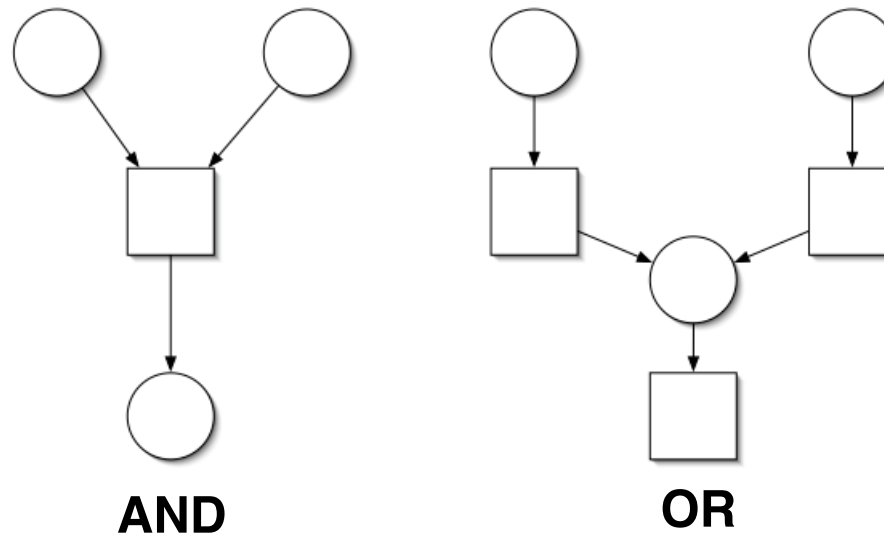
# SHEEP - User's View (6)



A tracked laptop as window to the virtual world

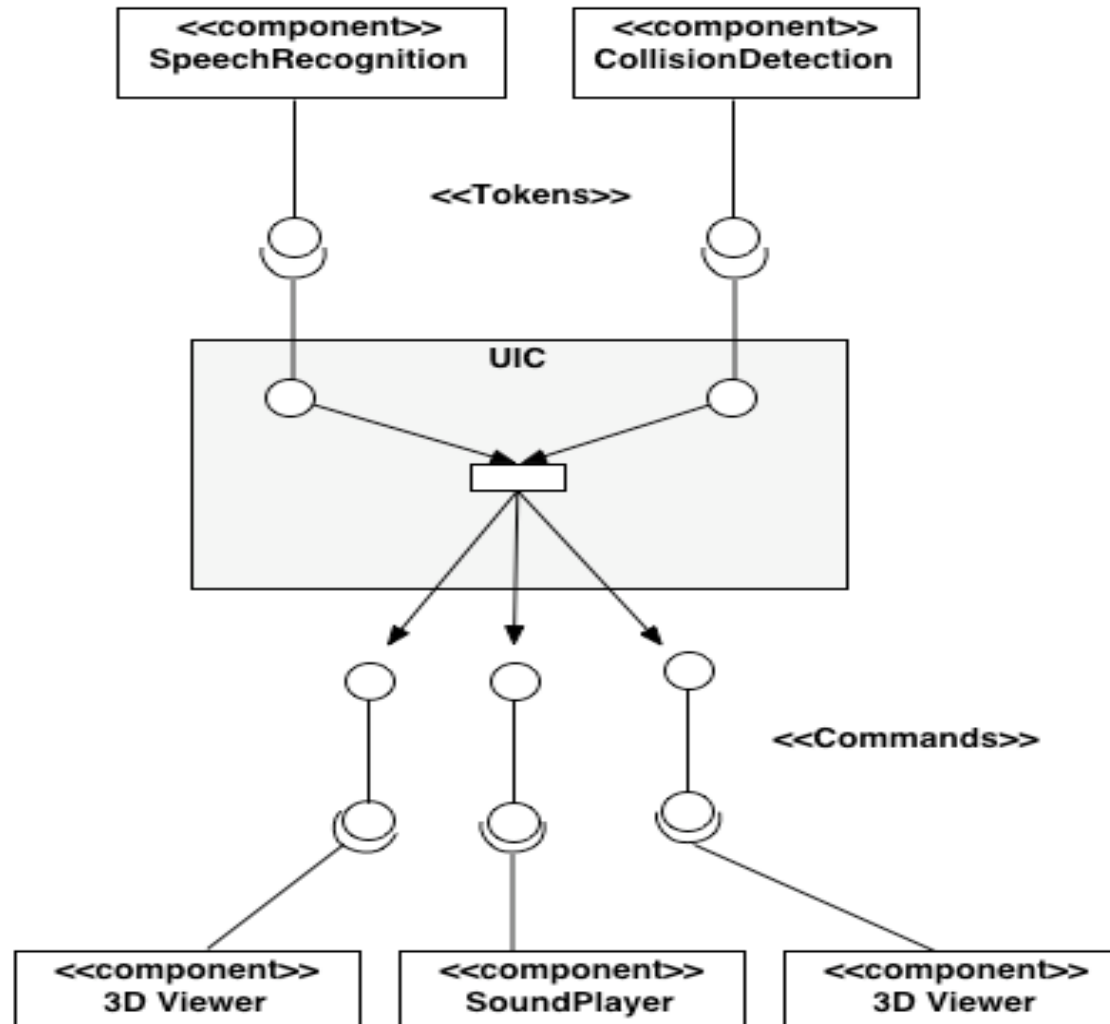
# Modelling an Interaction Space (1)

- Formal Model of the User Interface Controller:  
Petri-nets [Petri, 1962]
  - Structure: Places, transitions and arcs
  - Inputs: Tokens
  - Transitions fire, when all incoming places have tokens

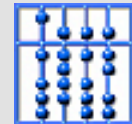
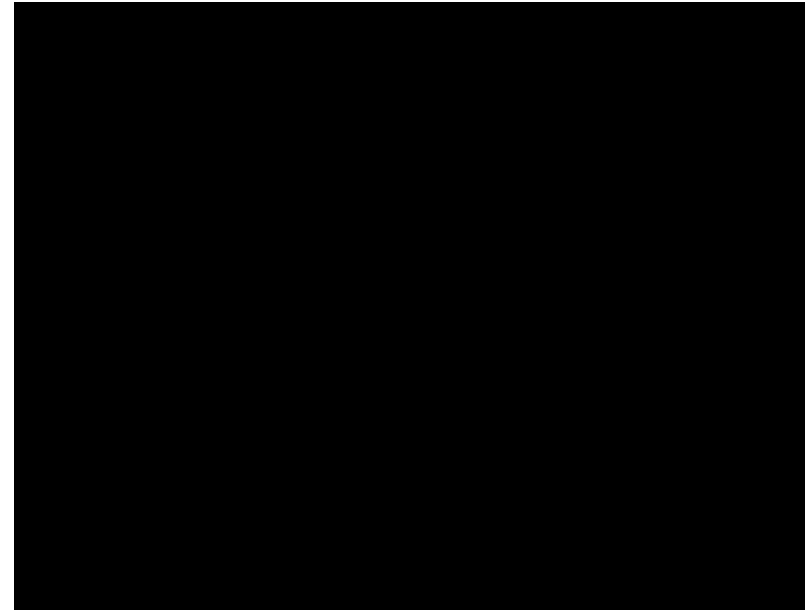
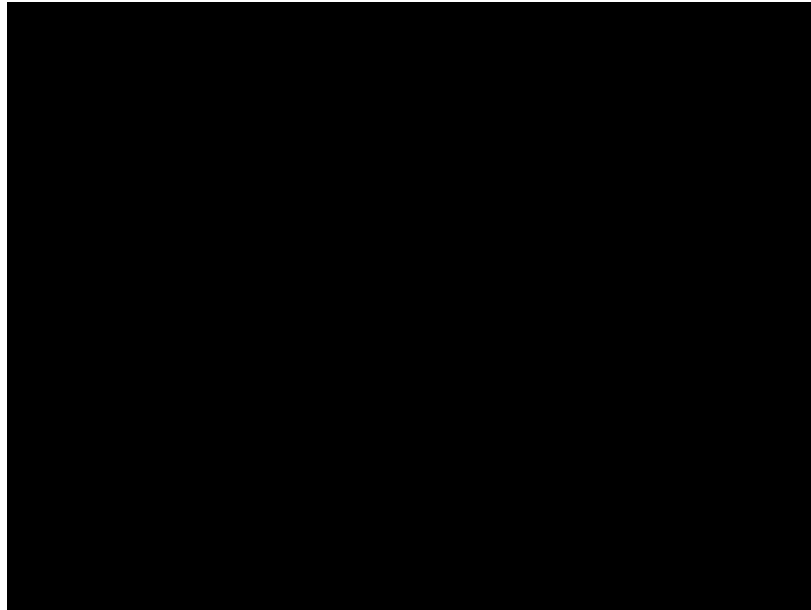




# Modelling an Interaction Space (2)

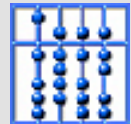


# Modelling an Interaction Space (3)



# Outline

- Introduction
- Problem Details
- My Approach
  - Kernel
    - DWARF
    - Component-based Hybrid UI Toolkit
    - Example: SHEEP
  - Tools
    - Viewing Volumes Visualization
    - Context-aware animations
    - Hybrid Designer
- Conclusion



# Tool 1: Viewing Volumes Visualization

- AUIs respond to eye gaze
  - Change Blindness / Preattentive processing (Rensink)
  - Context for input interpretation
- How to visualize eye gaze?



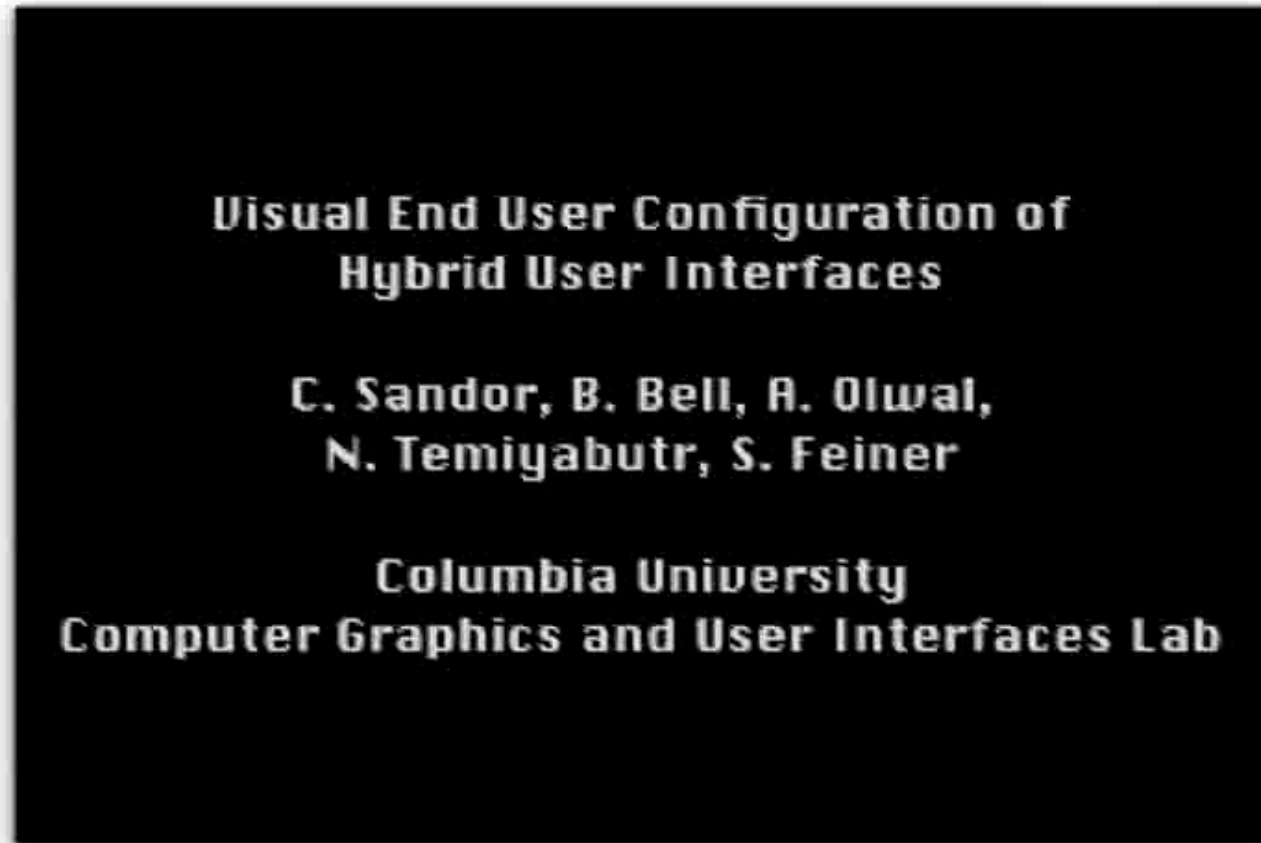
## Tool 2: Context-aware animations

- WIM that changes according to context
- Tilting and zooming with proxy object
- Context coupling by sketching



## Tool 3: Hybrid Designer

- How to couple input devices to graphics?



# End of Talk

Thank you for your attention.  
Please ask your questions!



Christian Sandor  
(sandor@in.tum.de)

Software and Tools for Hybrid User Interfaces  
TU München

