Middleware Mechanisms for Interaction Interoperability in Collaborative Virtual Environments

Institute for High-Performance Computing and Networking
Italian National Research Council

University of Naples *“Parthenope”*

Mario Ciampi
Luigi Gallo
Antonio Coronato
Giuseppe De Pietro

Roadmap

- The problem
  - Interaction Interoperability in Collaborative Virtual Environments
- State of the art solutions
- The proposed approach
  - Architectural model
  - Middleware
- Implementation details
- Case study
- Conclusions and future work

The problem

Virtual Reality

- Virtual Reality (**VR**) technologies
  - make it possible to reproduce faithfully *real life events* in *computer-generated scenarios*

- Virtual Reality is a
  - visualization technology
  - communication interface
    - based on interactive and immersive visualization

Specificity

New VR applications = New interfaces

- Every domain requires a custom, *specific* design

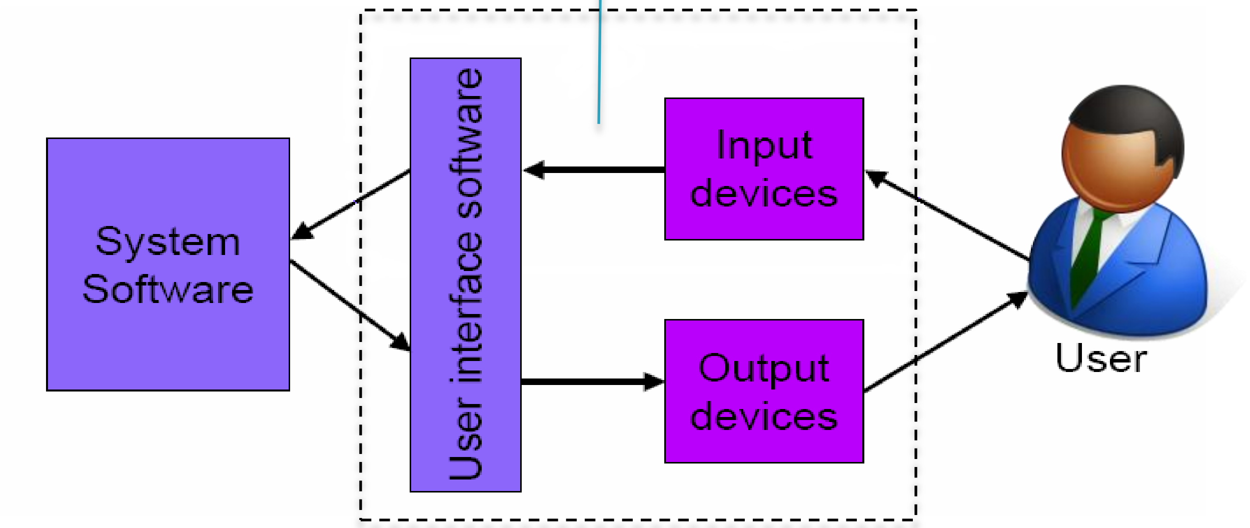
- Variables
  - Application
  - Domain
  - Task
  - Device
  - User

→ Interaction techniques

Interaction Techniques

- VR Interface components

Interaction Techniques
methods used to accomplish a given task via the interface



Collaborative Virtual Environments

- Collaborative Virtual Environments (**CVEs**)
  - make it possible the collaboration and interaction of many participants that may be spread over large distances
  - involve the crucial concept of *sharing*
    - knowledge of activities, environments, actions

The problem

Many participants = Many input devices = Many interaction techniques

how to share?

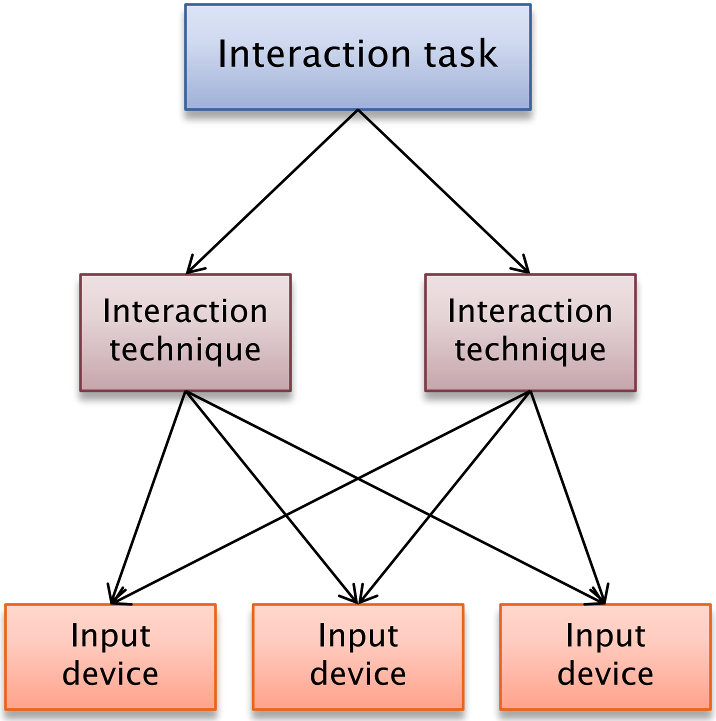
➡ **Heterogeneity** of *input devices* and *interaction techniques* is a problem to face

- Interaction interoperability
  - *The ability of two or more users to cooperate despite **differences** in language, interface, and execution platform*

State of the art solutions

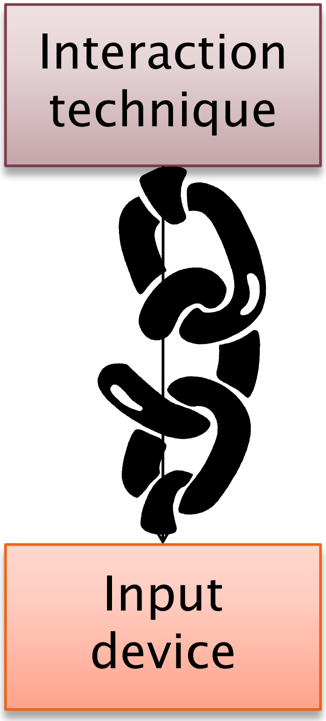
State of the art solutions

- **Decoupling**
  - interaction techniques from interaction tasks and input devices
- **Standardization**
  - ontologies to define interaction tasks and techniques
    - without specifying how tasks should be accomplished
- **Simulation**
  - mapping between many input devices to one interaction technique
    - bindings based on input device descriptors



State of the art solutions

- ## Usability problem
  - *Decoupling* interaction techniques from input devices and mixing them according to user patterns could reduce the *usability* of the interface
    - The implementation of a technique is strictly connected with the peculiar features of the input device used



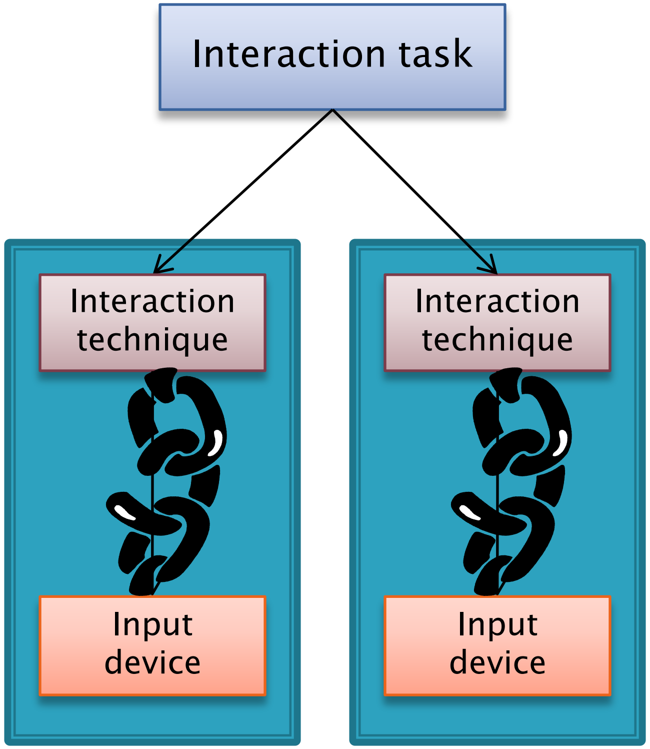
Interaction technique

Input device

The proposed approach

The proposed approach

- Strictly coupling
  - Interaction techniques and input devices are handled as black boxes
- Interaction standardization
  - Each user interface is mapped to a common representation



The proposed approach

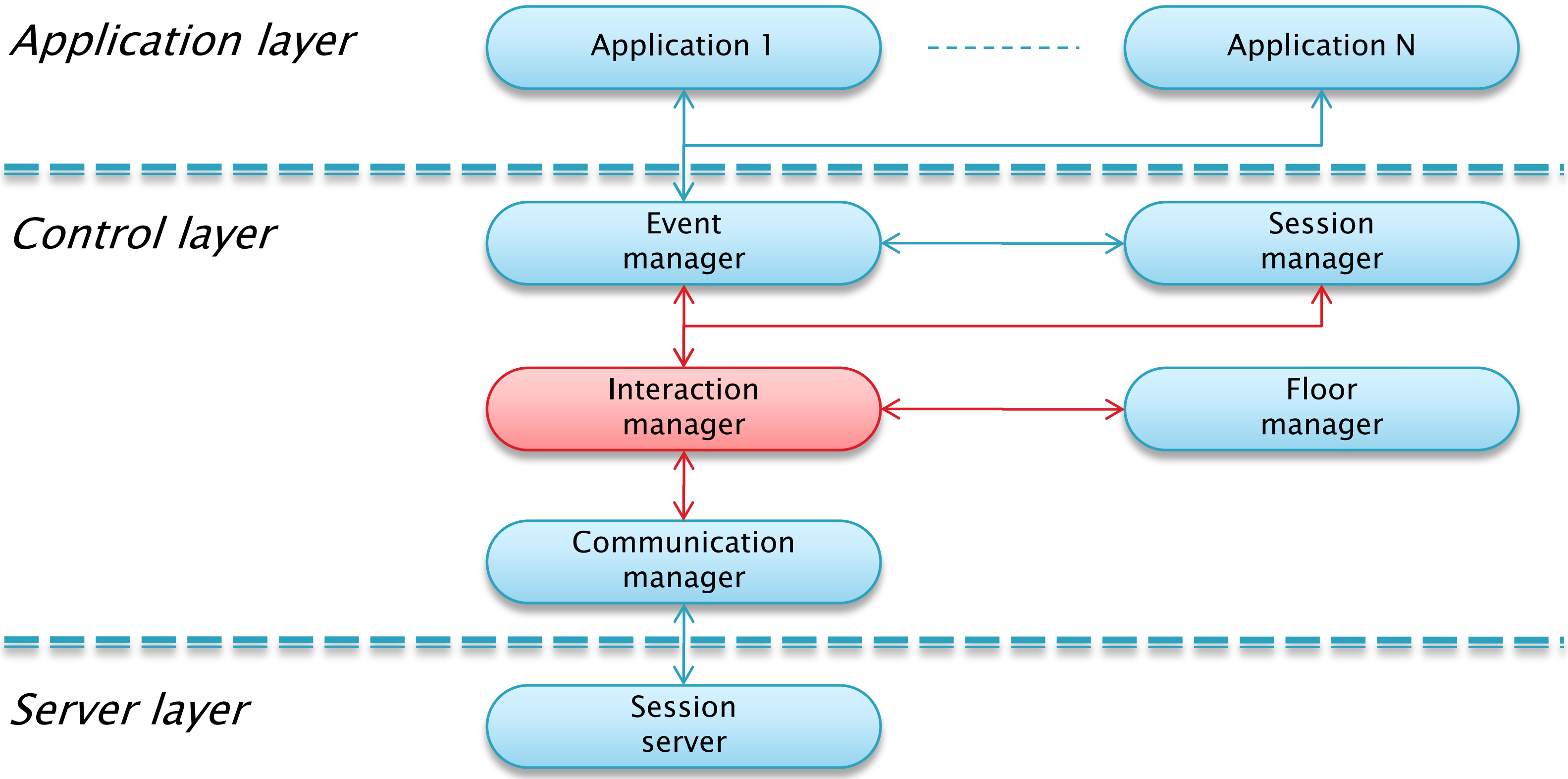
- The user who is in charge of creating the collaborative session
  - *implicitly* formalizes the common <u>language</u> that other user interfaces have to speak
    - by exposing her/his particular interface
  - *explicitly* specifies the peculiar <u>interaction rules</u>

The proposed approach

- Architectural model
  - able to handle differences in input devices and interaction tasks

- Middleware
  - that provides basic components to integrate heterogeneous user interfaces

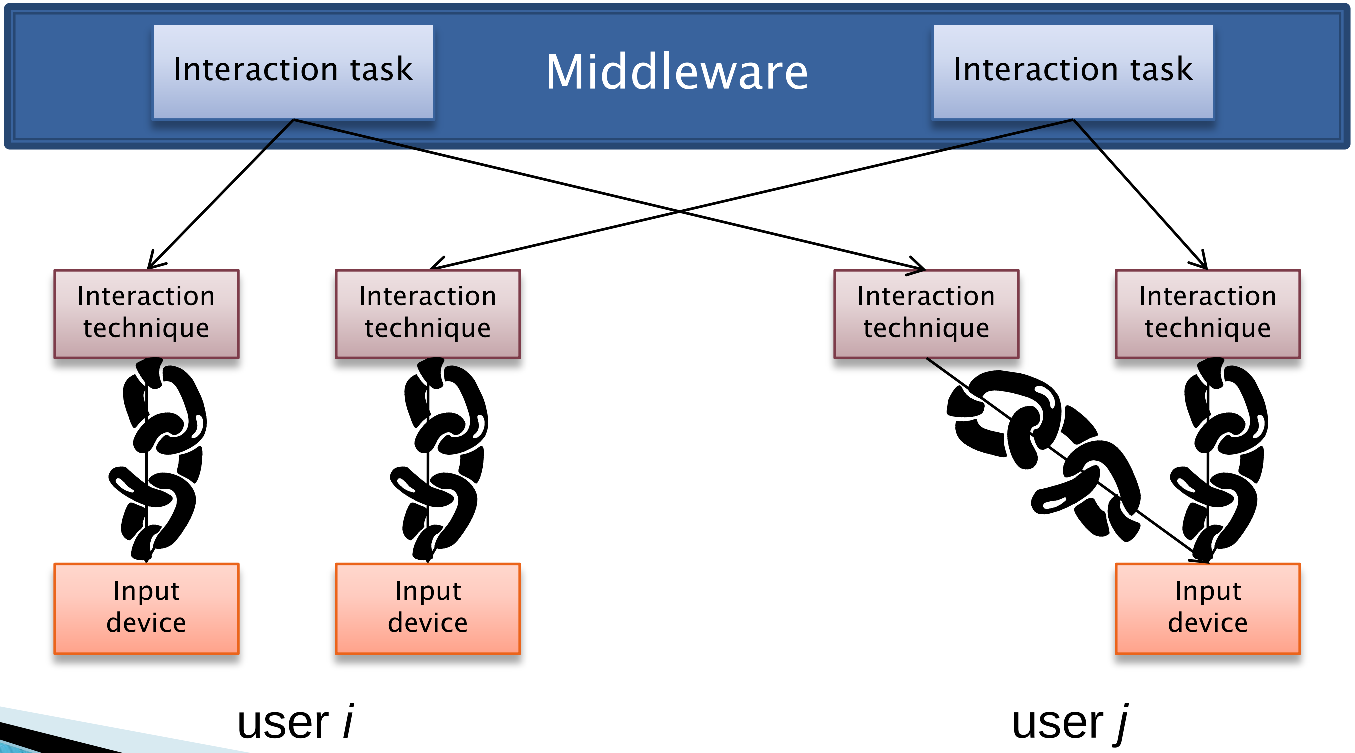
Architectural Model



- Interaction manager
  - enables the interaction standardization
    - handle transitions between interaction tasks

Middleware

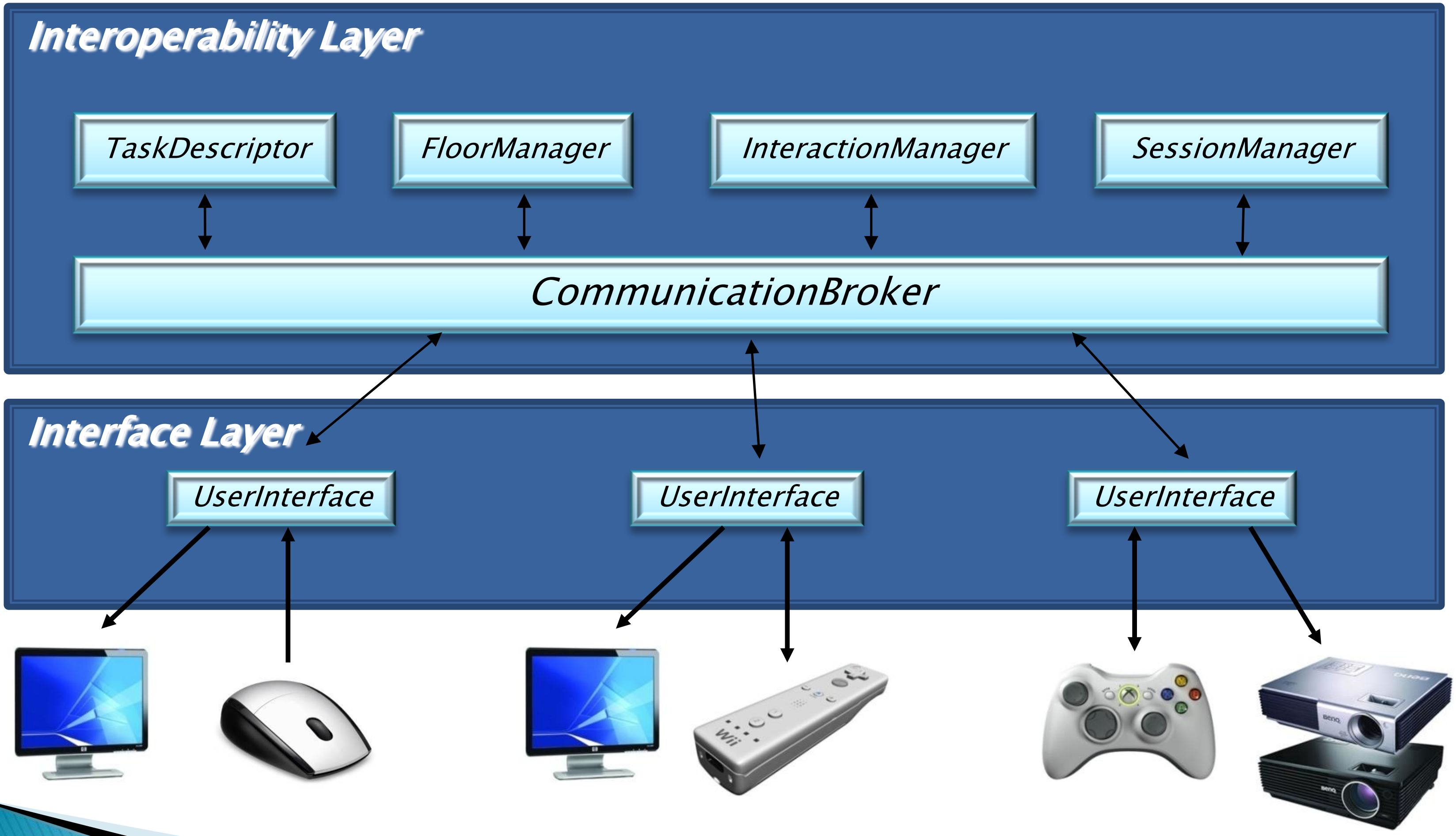
- The middleware platform let users
  - *share* interaction tasks
  - *communicate* with each other



Middleware Functionalities

- Sharing
  - a formal description of interaction tasks and interaction rules
- Handling
  - floor control
- Dispatching
  - event notifications to all participants

Middleware Architecture



Implementation details

The agents choice

- Software agents provide
  - easy mechanisms to describe their behaviors
  - effective communication mechanisms

Multi-Agent Systems

well suit to our needs

The Open Agent Architecture

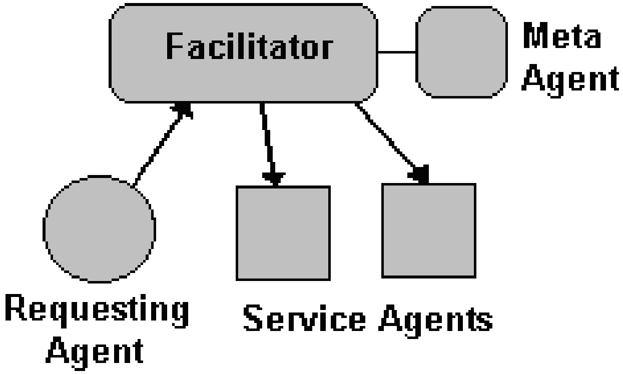
- The proof-of-concept middleware we developed relies on the SRI's Open Agent Architecture (OAA)

- OAA is an agent-based message-driven middleware architecture
  - enables agents to declare their capabilities by means of *solvables*:
    - *Procedure solvables* describe how agents perform actions
    - *Data solvables* provide access to a collection of data

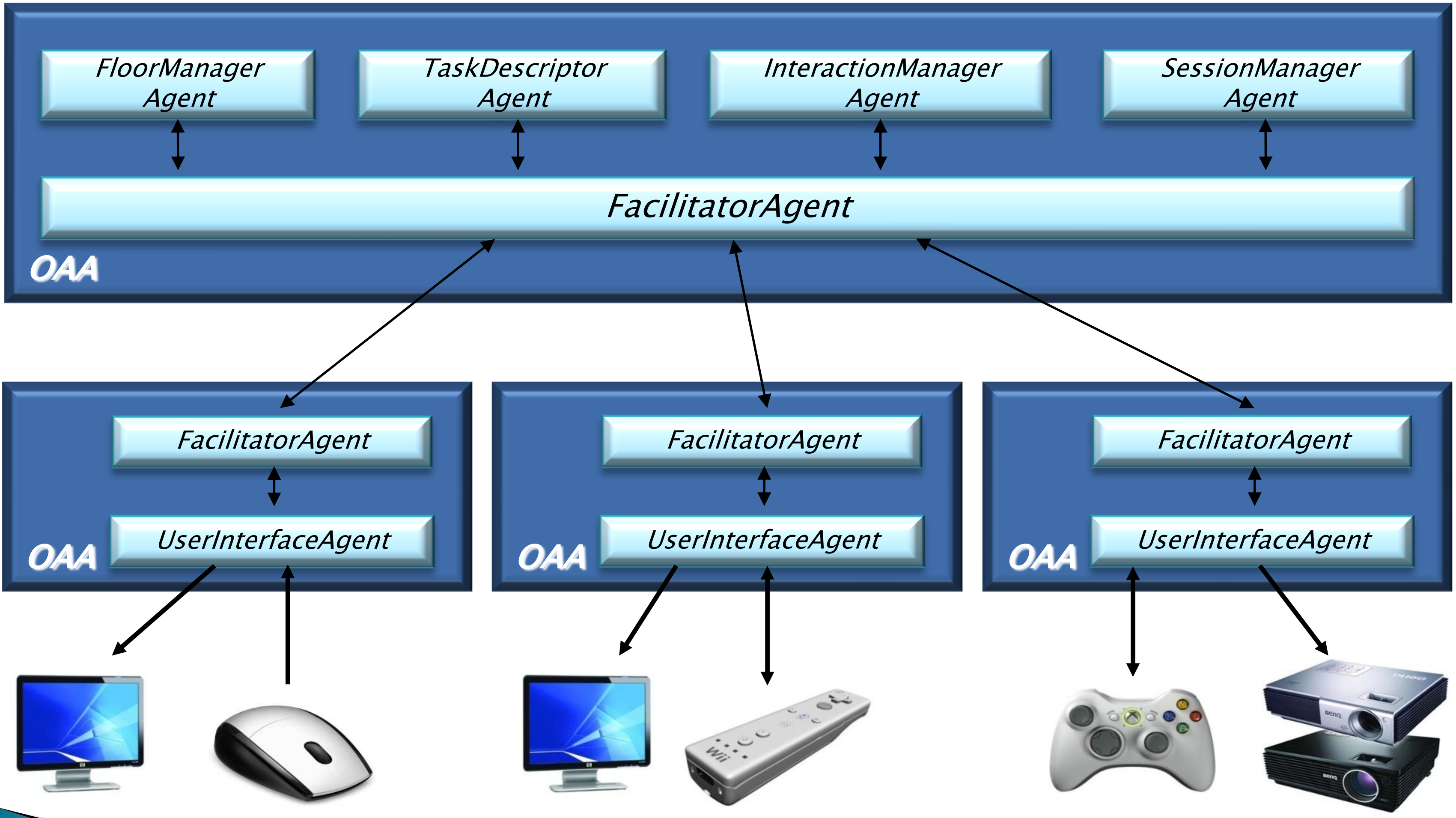
Event Notification

- Agent communications are coordinated by a *facilitator agent*
  - it maintains a list of what each agent is able to do (*solvables*)

- Delegated Computing Model
  - The facilitator agent delegates service requests to appropriate agents by implementing the publish/subscribe paradigm



Middleware Implementation



Formalization of interaction

- Formal descriptions of interaction tasks
  - written using the *Interagent Communication Language (ICL)*
    - OAA's interface and communication language

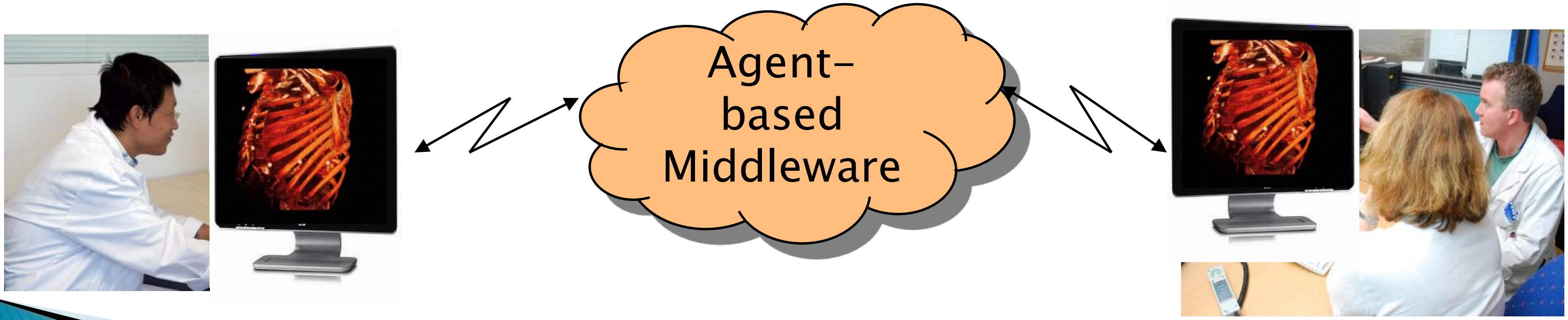
- Interaction rules
  - specified by using the *Interactive Cooperative Objects (ICO)* formalism*
    - ICO extends object Petri nets with the notion of *interaction events*

*D. Navarre, R. Bastide, A. Schyn, L. P. Nedel, and C. M. D. S. Freitas, *"A formal description of multimodal interaction techniques for immersive virtual reality applications"*, in *Proc. of the Tenth IFIP TC13 International Conference on Human-Computer Interaction. Springer,* 2005, pp. 170-183.

Case study

A medical imaging application

- MITO: Medical Imaging Toolkit
  - cross-platform application (soon open-source) developed by ICAR and IBB institutes of the Italian CNR
  - Goal
    - allowing two users to cooperate in a semi-immersive virtual environment for volumetric medical data exploration

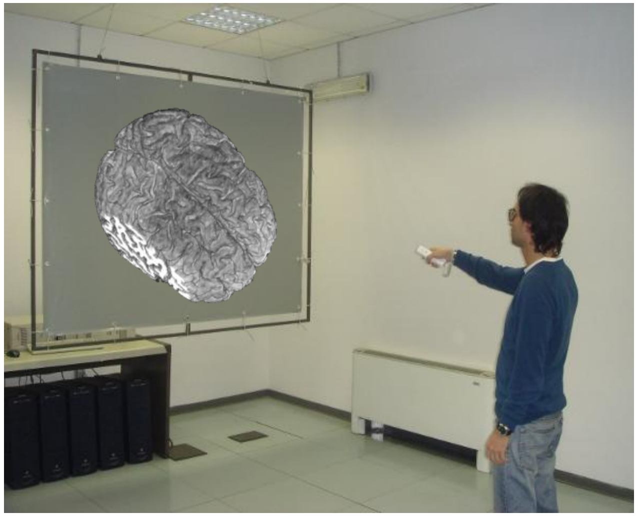


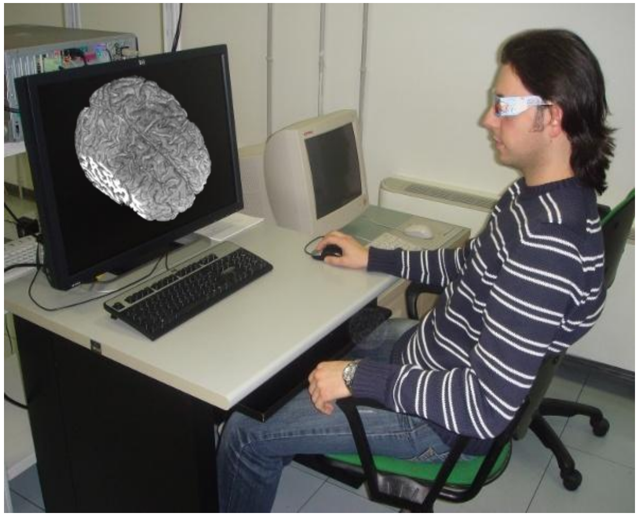
Agent-based Middleware

Input devices and interaction techniques

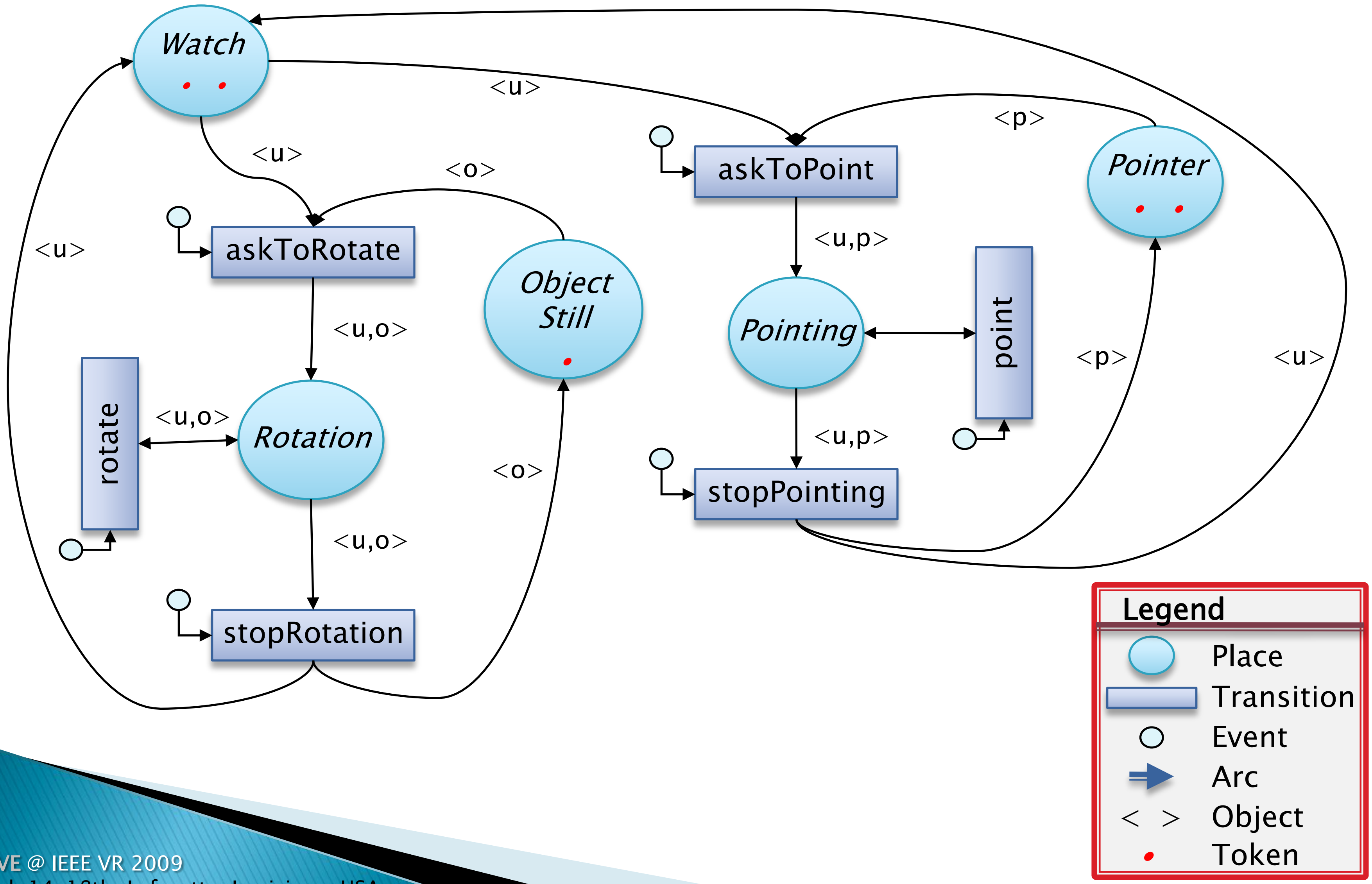
- The two users use distinct input devices
  - Wiimote
  - Mouse

- The interaction techniques are different as well





Interaction management



Conclusions and future work

Conclusions and future work

Conclusions

- We have described an architectural model and an agent-based middleware aimed at handling interaction interoperability in CVEs
- A proof-of-concept implementation has been applied to a medical scenario

Future work

- Building tools to simplify the modeling of interaction rules
- Applying this model to more complex interaction tasks

Thank You

- Thanks
  - For listening
  - To the anonymous reviewers

- Questions?

MMVE @ IEEE VR 2009
March 14-18th, Lafayette, Louisiana, USA