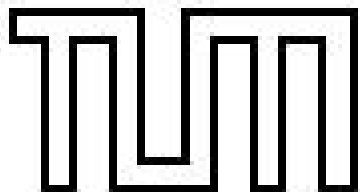


# CUIML: A Language For the Generation of Multimodal Human-Computer Interfaces

*Christian Sandor*

sandor@cs.tum.edu

*Technische Universität München  
Chair for Applied Software Engineering*



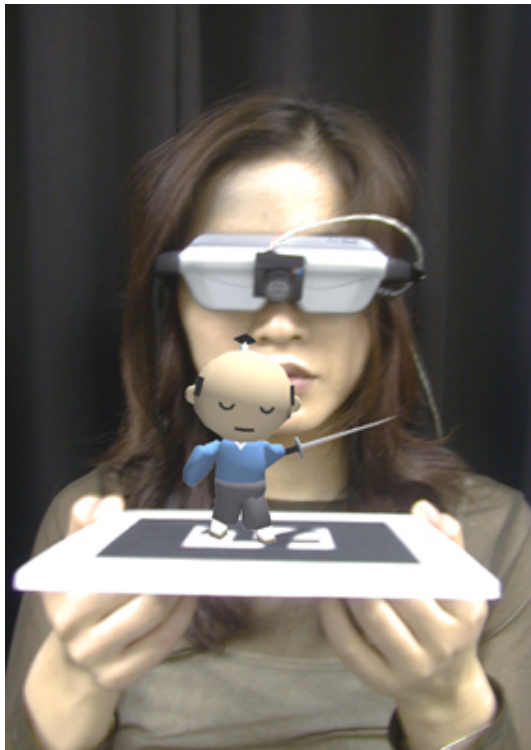
# Abstract

- *DWARF Project at the Technische Universität München*
- *UIML complied with some of our requirements*
- *Extension of UIML was developed and will now be presented*

# What is DWARF?

■ *Distributed Wearable Augmented Reality Framework*

■ DWARF movie 



Augmented Reality



Wearable Computer

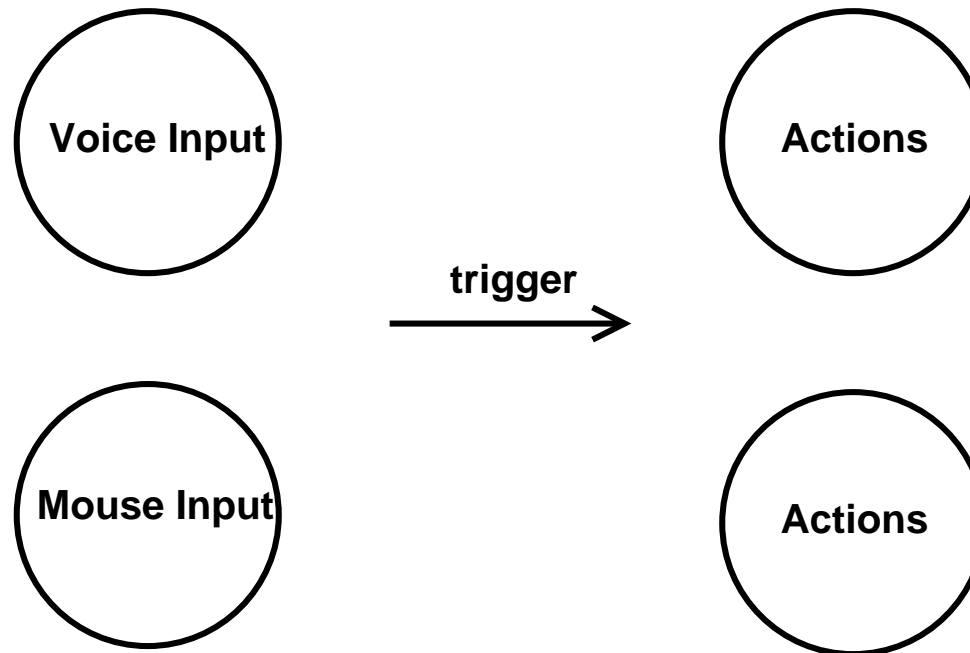
# The Problem

- *Framework for Presentation Layer of Wearable Computers*
  - An application should be platform independent

# The Problem

## ■ *Framework for Presentation Layer of Wearable Computers*

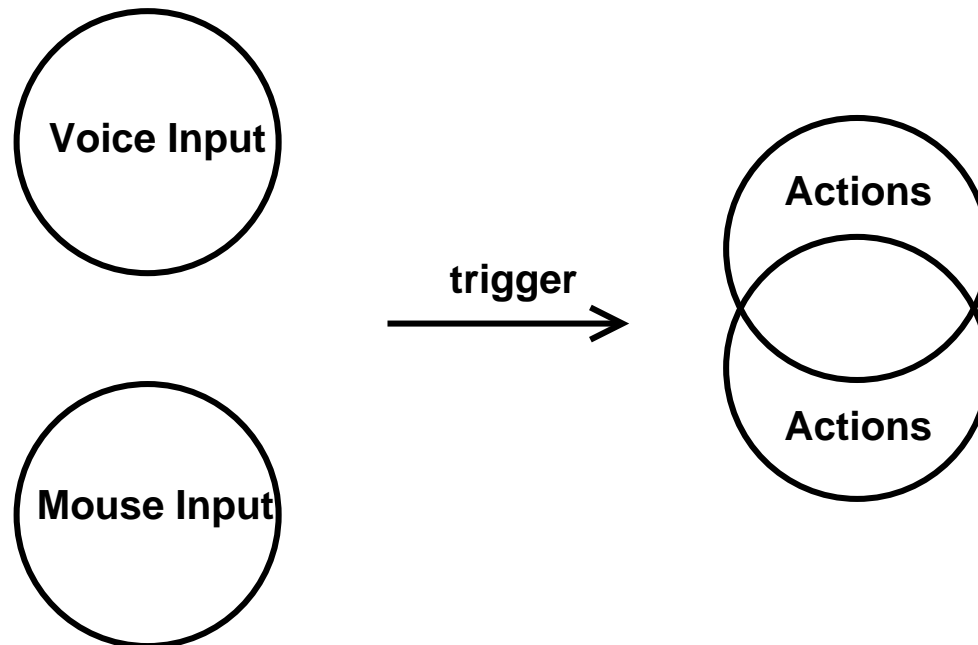
- An application should be platform independent
- Interaction with the user is multimodal



# The Problem

## ■ *Framework for Presentation Layer of Wearable Computers*

- An application should be platform independent
- Interaction with the user is multimodal



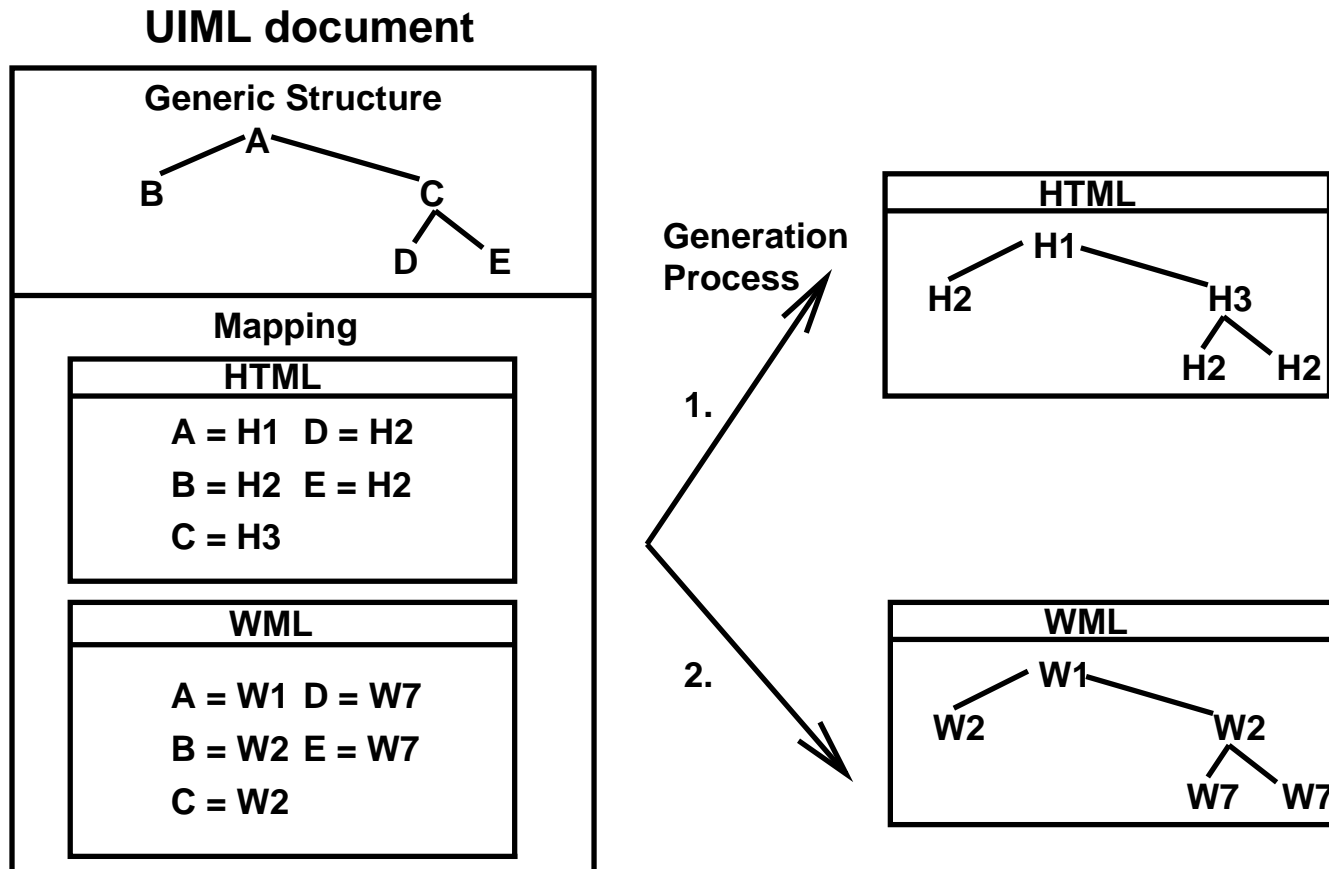
# CUIML – a solution

- *Cooperative User Interfaces Markup Language*
- *Based on UIML*
  - Separation of document structure and presentation
  - Transformation to markup languages
  - Display in browsers for VRML, VoiceXML, HTML (Views)
- *New concepts:*
  - Controller
  - Manipulators

# System Design – Views

## ■ *UIML*:

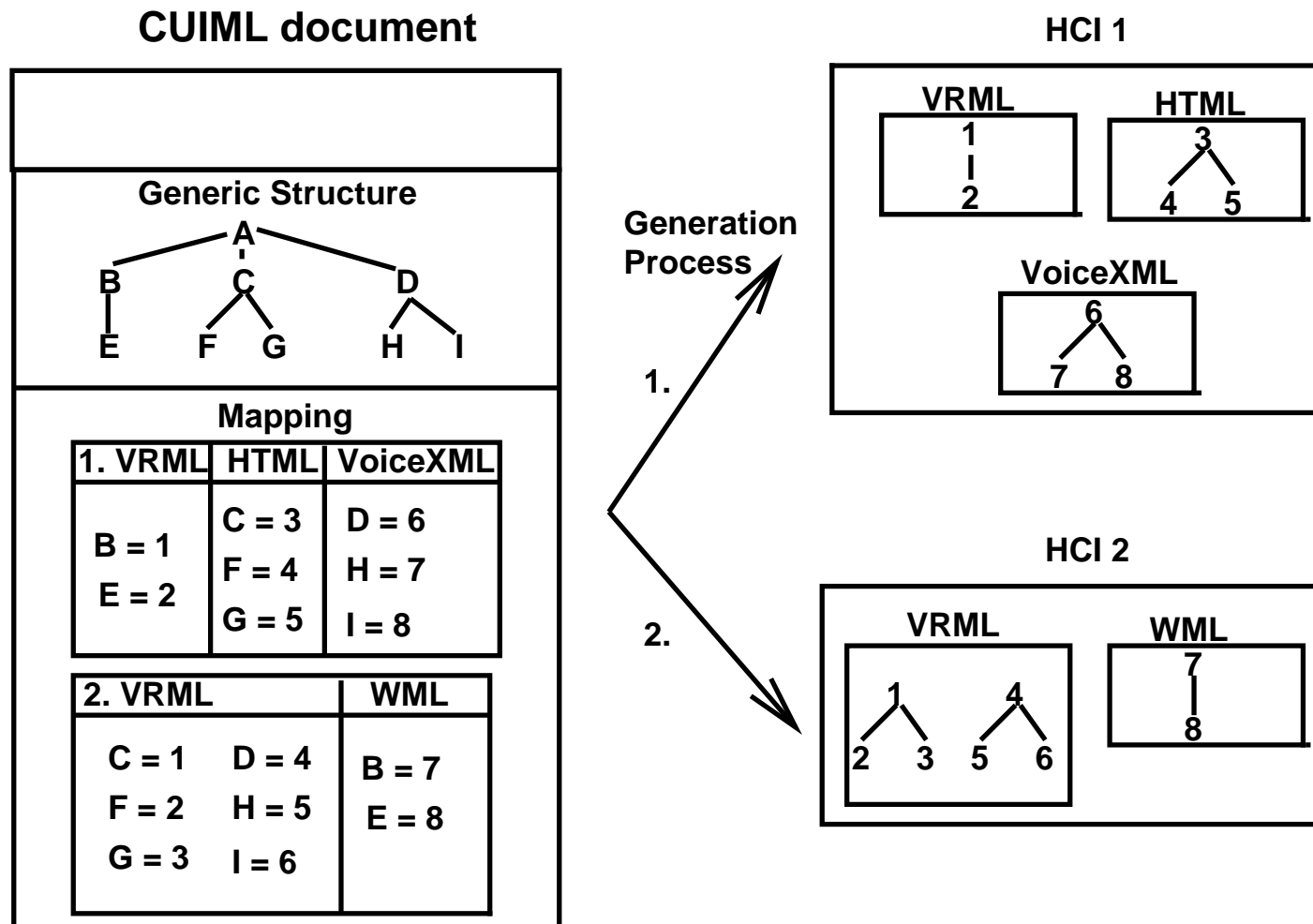
- Addresses requirement of platform independence
- No support for multiple cooperating Views





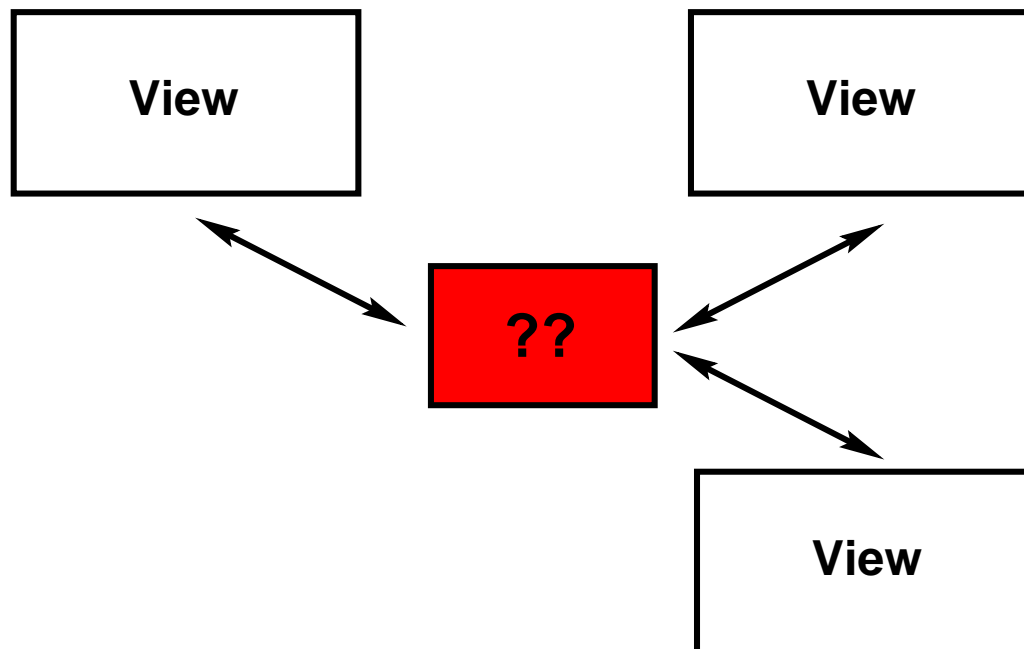
# System Design – Views (2)

## ■ *Multimodal Human-Computer Interfaces consist of multiple Views*



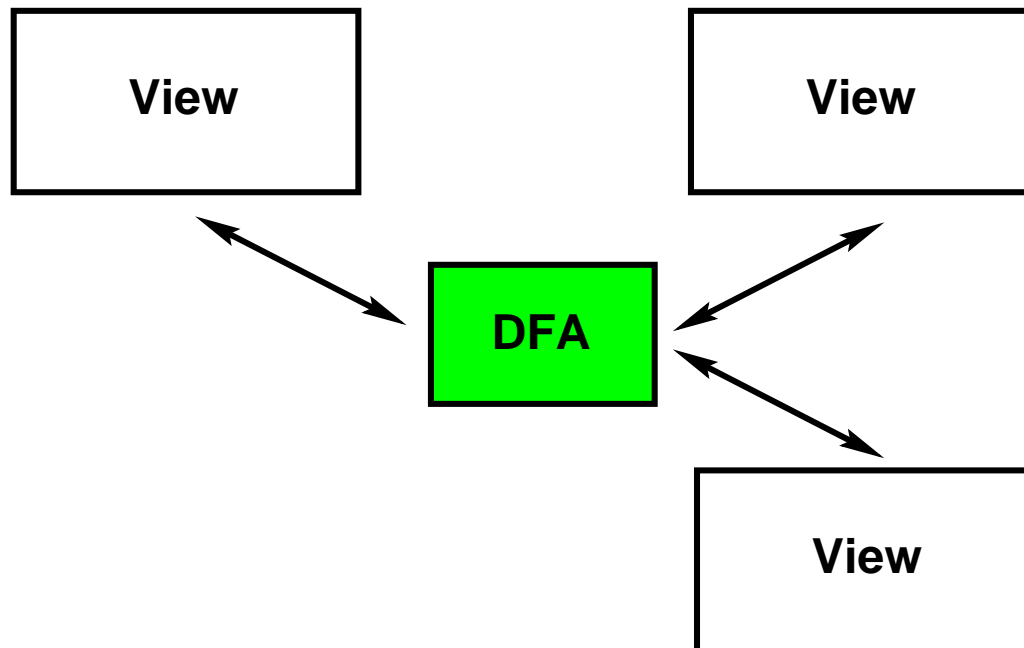
# System Design – Controller

- *A central component is needed*
  - Describes state of the HCI
  - Synchronizes views



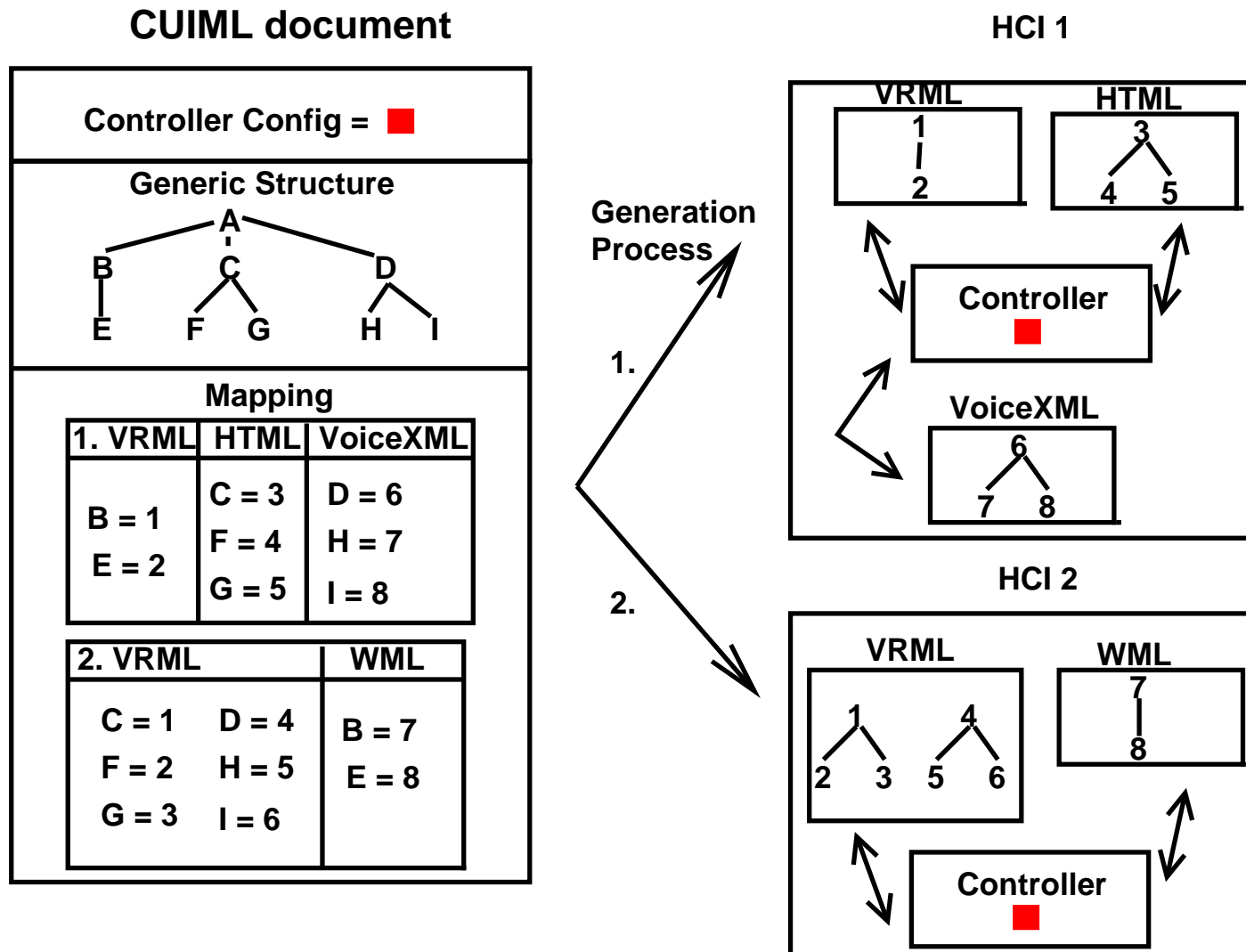
# System Design – Controller

- *A central component is needed*
  - Describes state of the HCI
  - Synchronizes views
- *Deterministic Finite Automaton with XML based Configuration*



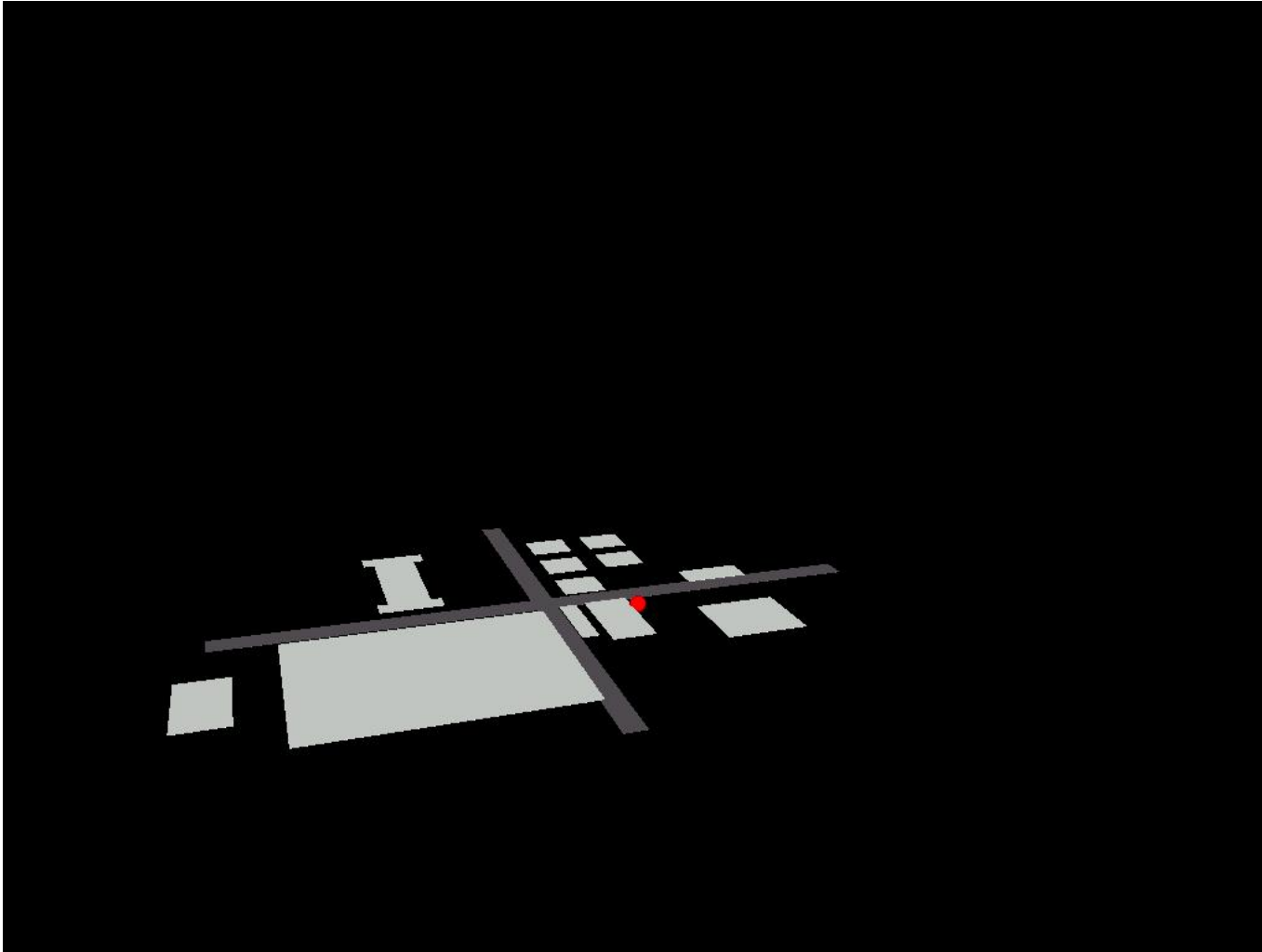
# System Design – Controller (2)

## ■ *CUIML extended with Controller*

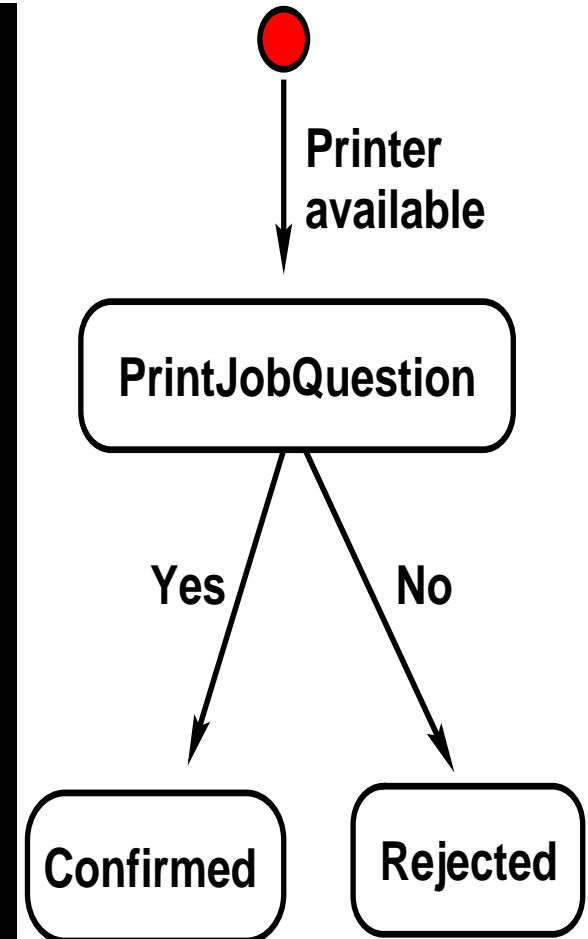


# Controller Example

## ■ *State transitions of a View*



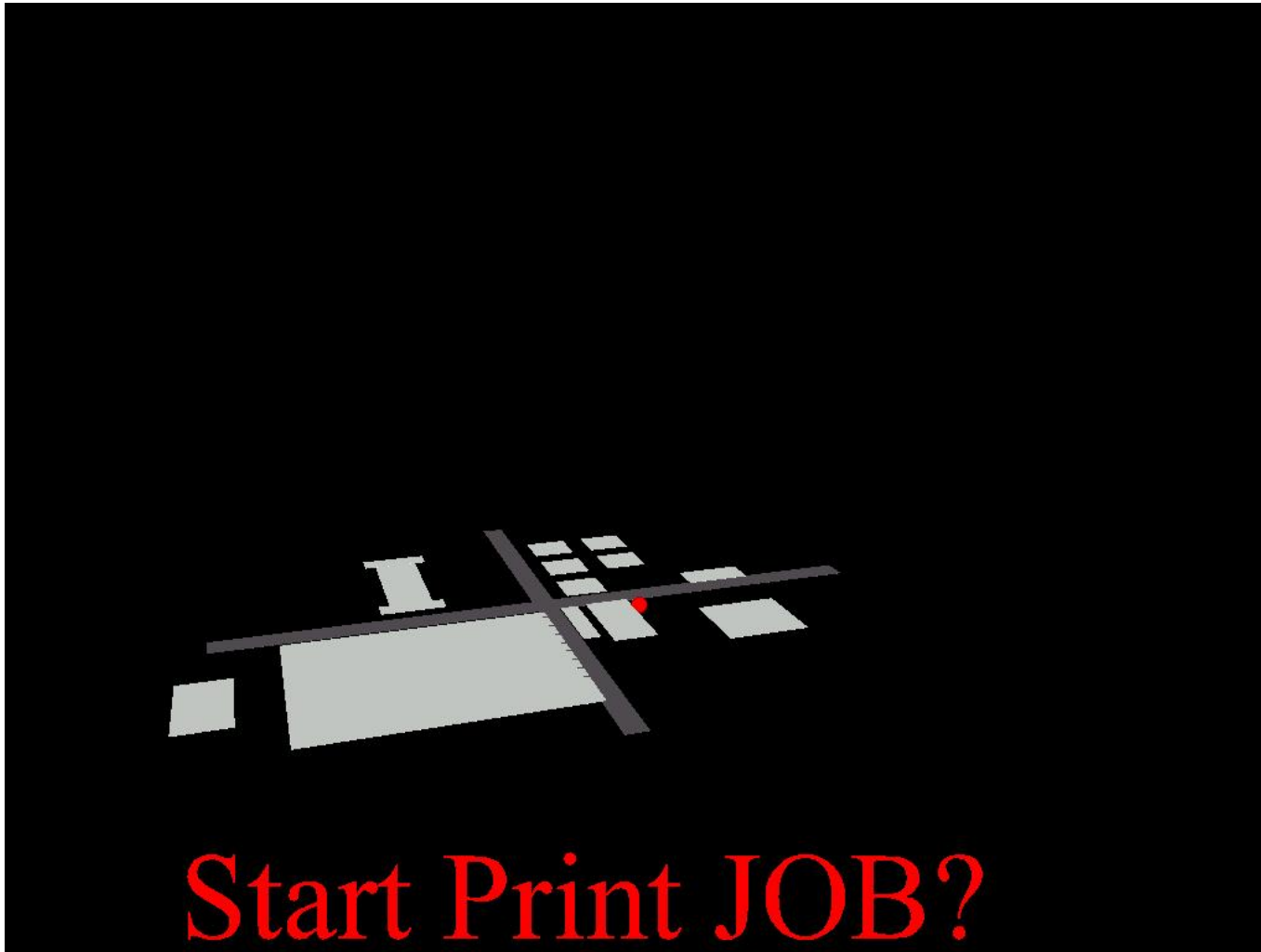
VRML View



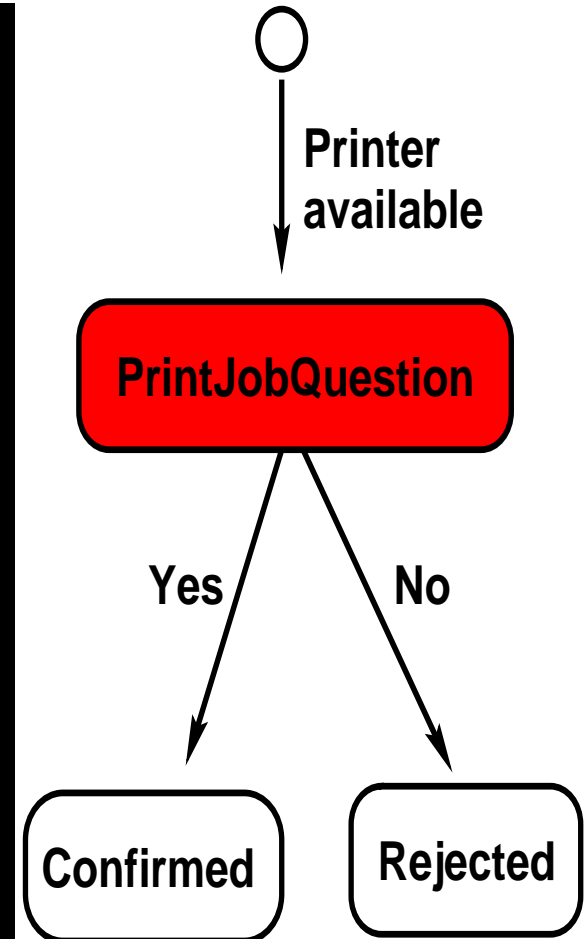
Controller

# Controller Example

## ■ *State transitions of a View*



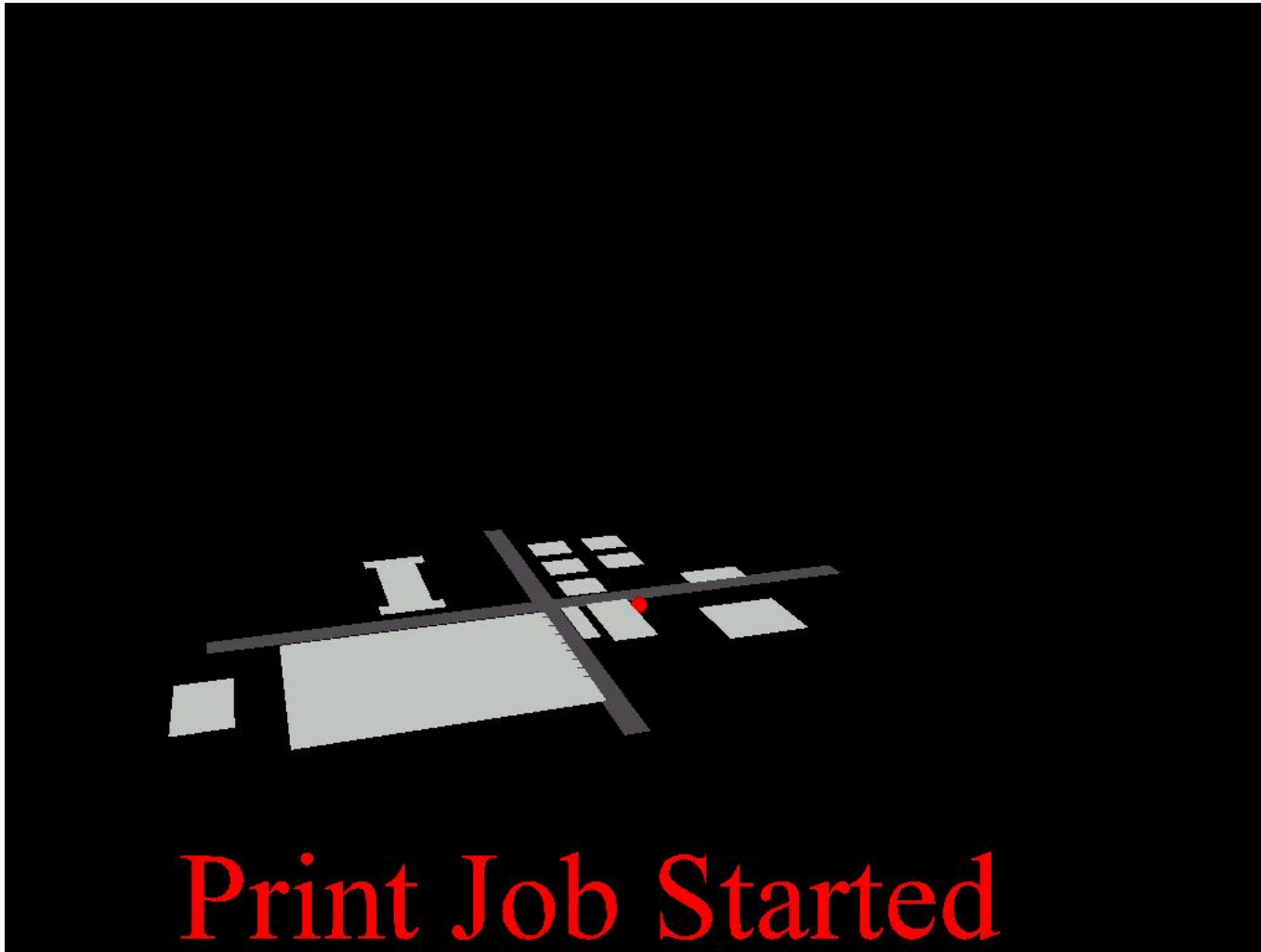
VRML View



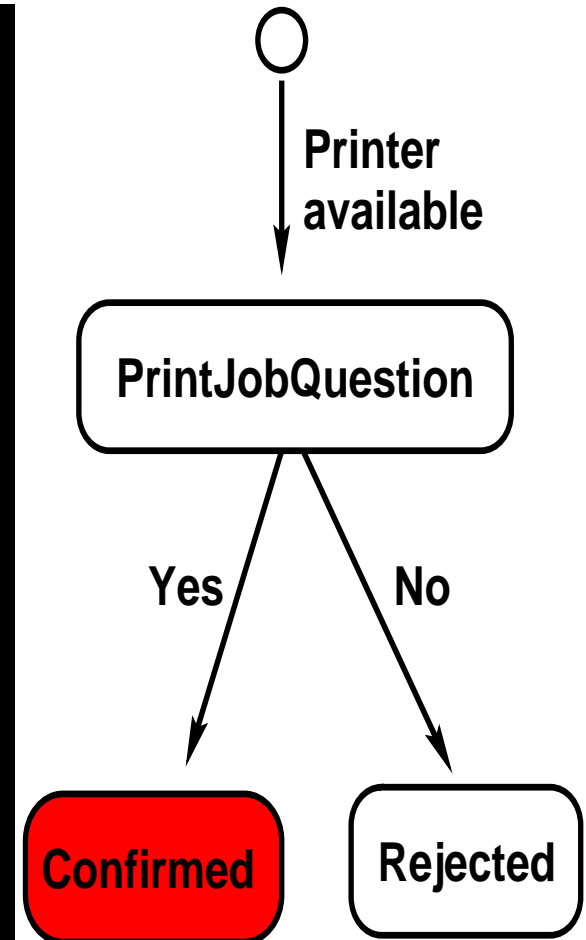
Controller

# Controller Example

## ■ *State transitions of a View*



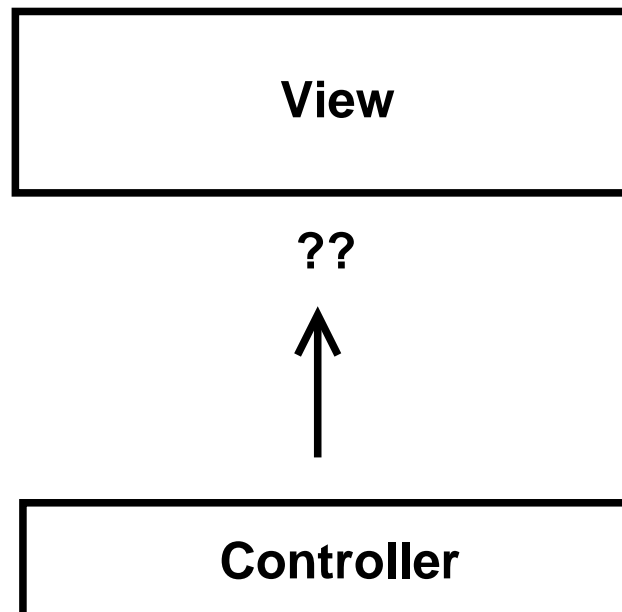
VRML View



Controller

# System Design – Manipulators

- *Views have to be modified by the Controller*
  - Access mechanisms for Views differ

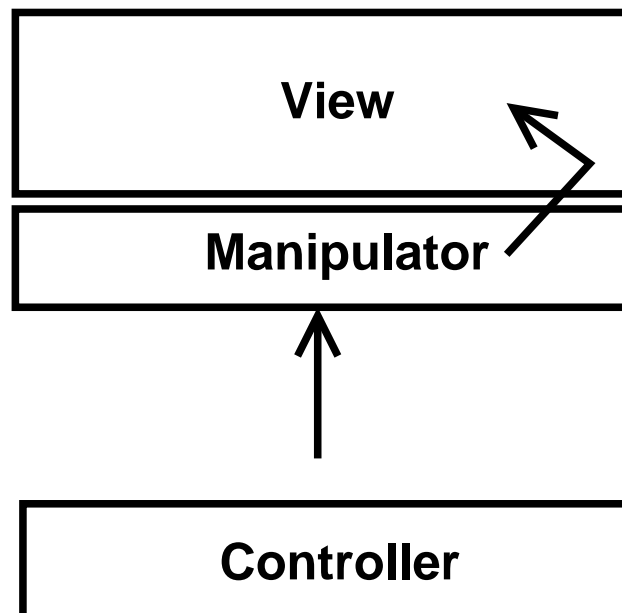




# System Design – Manipulators

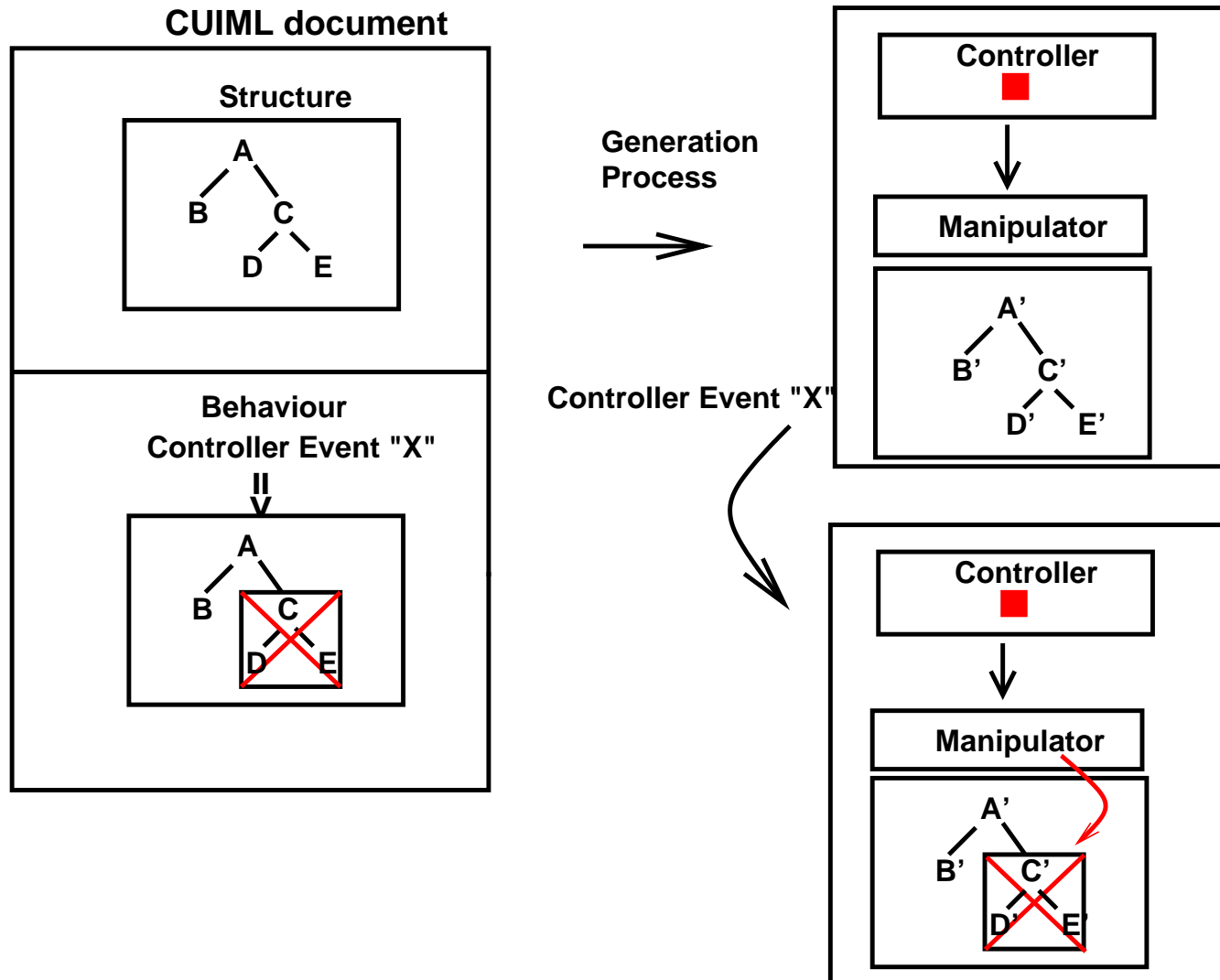
## ■ *Views have to be modified by the Controller*

- Access mechanisms for Views differ
- Every View needs an appropriate Manipulator
- Described in behaviour section



# System Design – Manipulators

## ■ *Example of a Manipulator*



# Benefits of CUIML

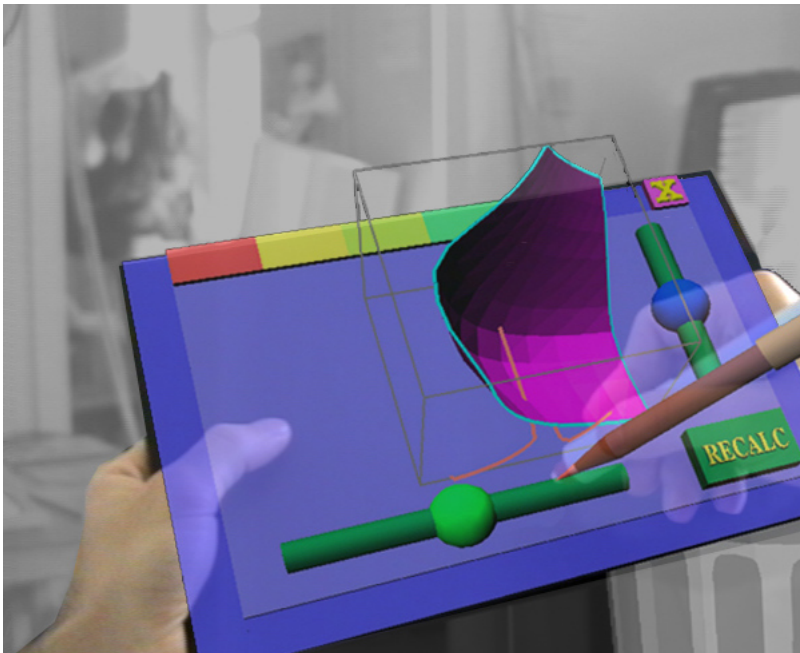
- *Eases the task of developing multimodal HCIs*
- *Dynamic reconfiguration of the HCI to preserve functionality*
- *Incorporates the advantages of UIML*
  - Separation of development tasks
  - Platform independence

# Future Work

- *Rewrite renderers with JavaCC*
- *Adapt the Human-Computer Interface at runtime*

# Future Work

- *Rewrite renderers with JavaCC*
- *Adapt the Human-Computer Interface at runtime*
- *Establish standard metaphors*



<http://www.cg.tuwien.ac.at/research/vr/pip/>



<http://www.csl.sony.co.jp/person/rekimoto/cube/>