Vrije Universiteit Brussel

VR-WISE: Conceptual Modeling for Virtual Reality

Prof. Dr. Olga De Troyer
WISE Research Group
Vrije Universiteit Brussel

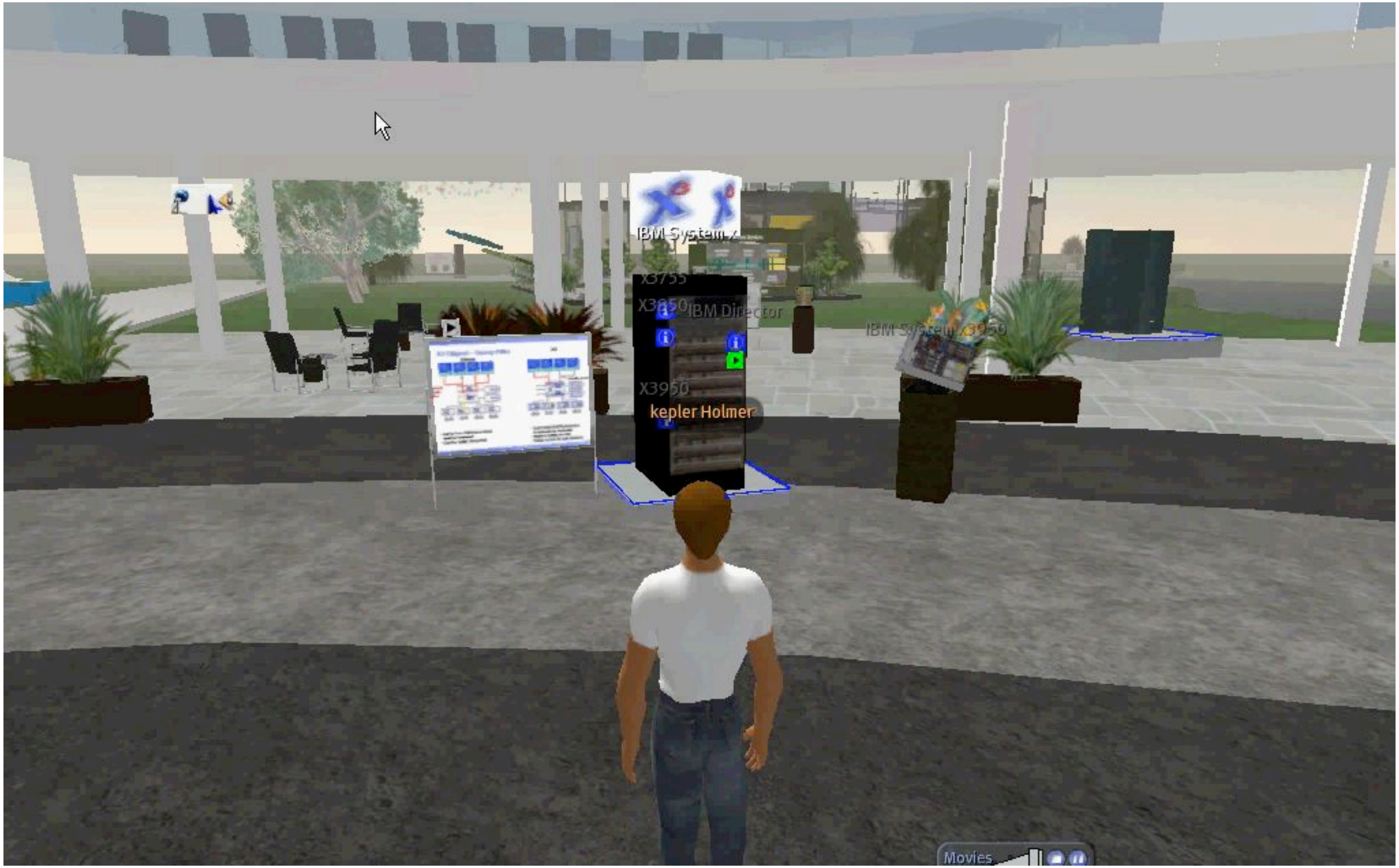
VR-WISE: Conceptual Modeling for Virtual Reality

- Context
- Motivation

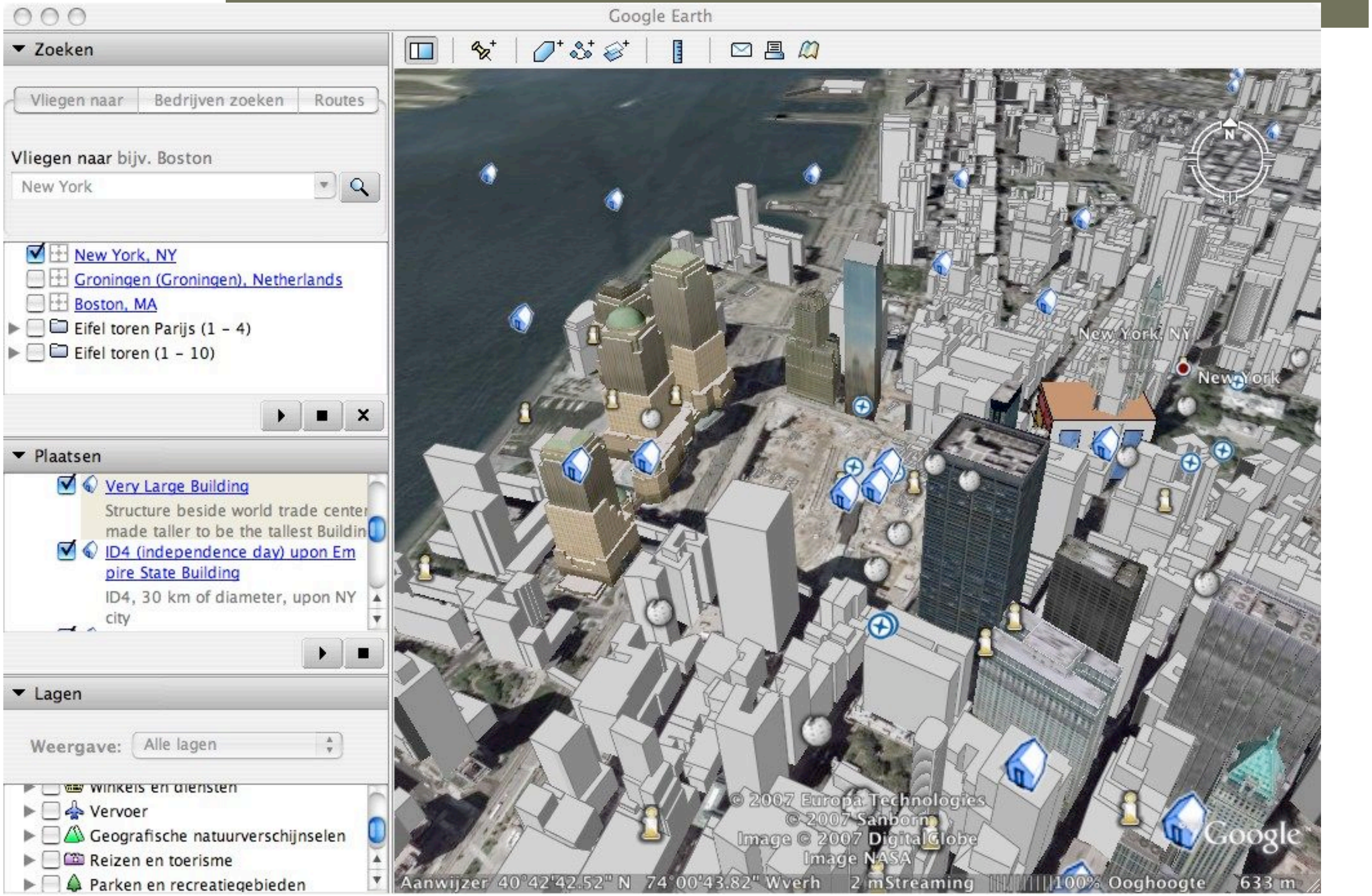
Context

- Desktop Virtual Reality
  - – Cheaper and faster hardware
  - – Dedicated tools
    - Authoring tools
      - – 3D Studio Max, Virtools, Google SketchUp, ...
    - Application toolkits (programming libraries)
      - – VR Juggler, Java3D, OpenSceneGraph, ...
    - Engines
      - – AVOK, Open Dynamic Engine (ODE), Panda 3D, ...
    - Players
      - – Octaga, Flux, ...

Second Life


IBM System x
X3755
X3850
IBM Director
X3950
kepler Holmer
IBM System x3950
Movies

Google Earth



Vliegen naar bijv. Boston

New York

- ☑ New York, NY
- □ Groningen (Groningen), Netherlands
- □ Boston, MA
- □ Eifel toren Parijs (1 – 4)
- □ Eifel toren (1 – 10)

- ☑ Very Large Building
  Structure beside world trade center made taller to be the tallest Buildin
- ☑ ID4 (independence day) upon Empire State Building
  ID4, 30 km of diameter, upon NY city

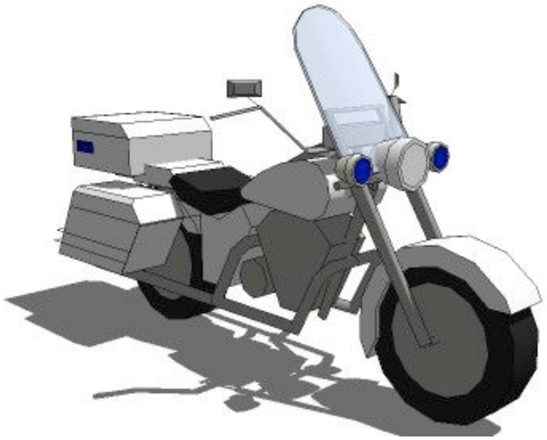
Weergave: Alle lagen

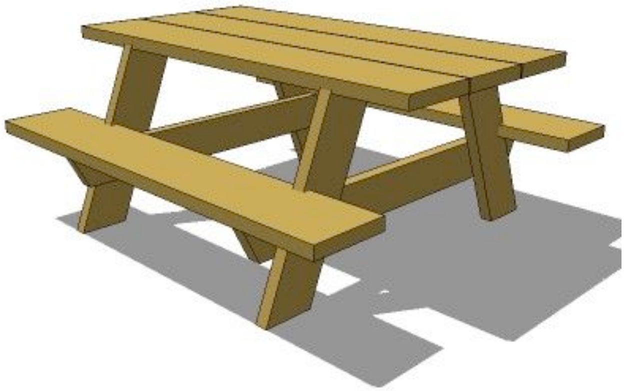
- □ Winkels en diensten
- □ Vervoer
- □ Geografische natuurverschijnselen
- □ Reizen en toerisme
- □ Parken en recreatiegebieden

Aanwijzer 40°42'42.52" N 74°00'43.82" Wverh 2 m Streaming ||||||||| 100% Ooghoogte 633 m

Context

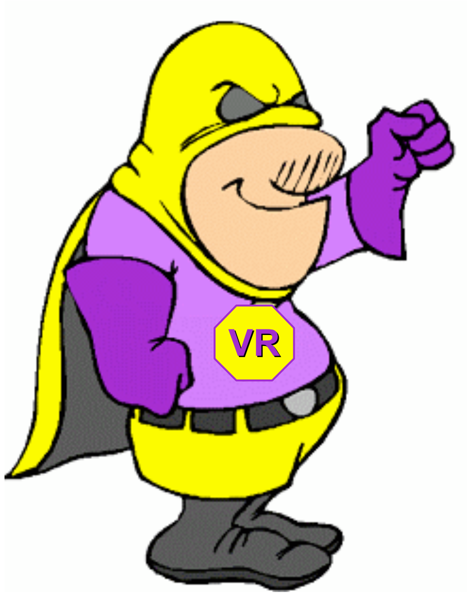
- Virtual Reality on the Internet/Web
  - Increased bandwidth
  - VR standards for the Web
    - VRML, X3D
  - New Web technology, e.g., Ajax3D
  - Increased availability of 3D content
    - e.g., Google 3D Warehouse





Developing a VR application

Developing a VR application



- Requires considerable VR background knowledge
- No systematic development process
  - Informal design phase
  - Lack of methods
  - Lack of abstraction mechanisms

↓

- Expensive
- Time consuming
- A lot of mismatches

VR-WISE: Conceptual Modeling for Virtual Reality

- Context
- Motivation
- Objectives of the research
- Approach

Objectives of the Research

- To open the development of VR-applications to a broader audience
- To allow a domain expert to be more involved in the design of a VR application
- To reduce the overall development time and cost

Solution

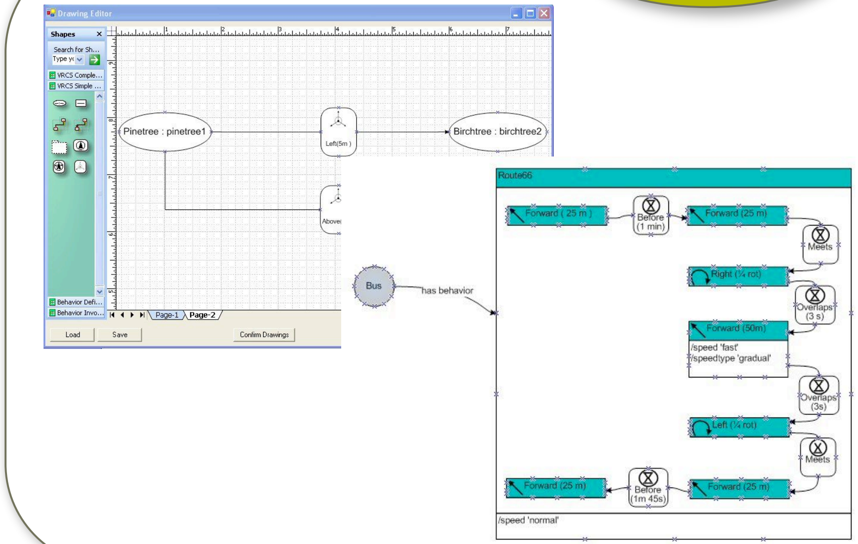
VR-WISE approach

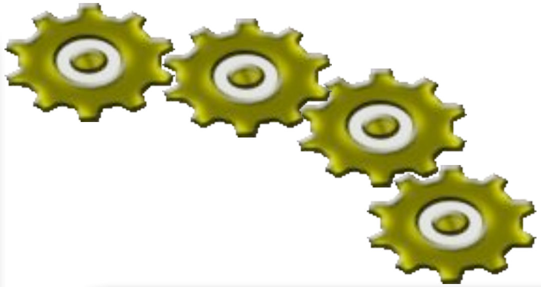
- Introduction of a Conceptual Design phase
  - High-level intuitive descriptions using the terminology of the application domain
  - Free from VR-implementation details
- Allows
  - Abstracting from implementation issues
    - Reduces the complexity
    - No deep VR knowledge needed
    - Domain experts may be involved
  - Easier and earlier communication with stakeholders
    - Earlier detection of design errors

VR-WISE

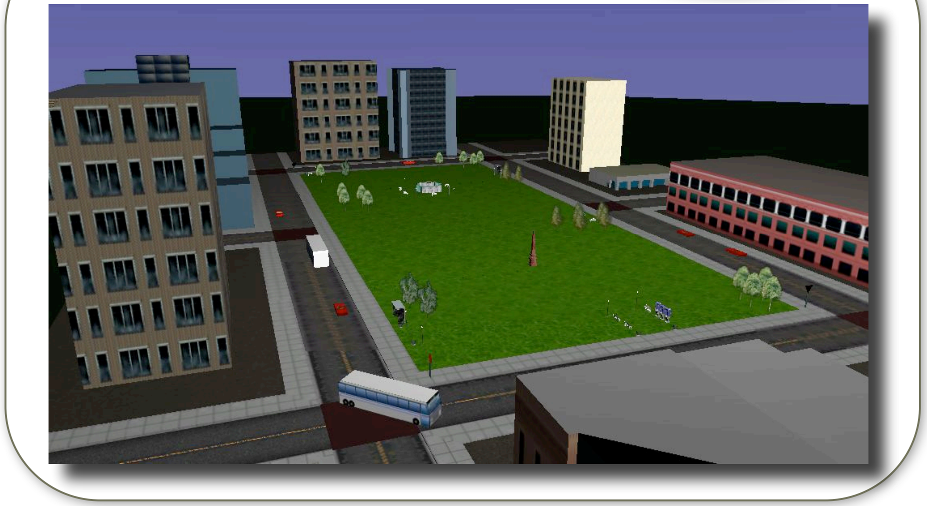
- Model-based development

Models



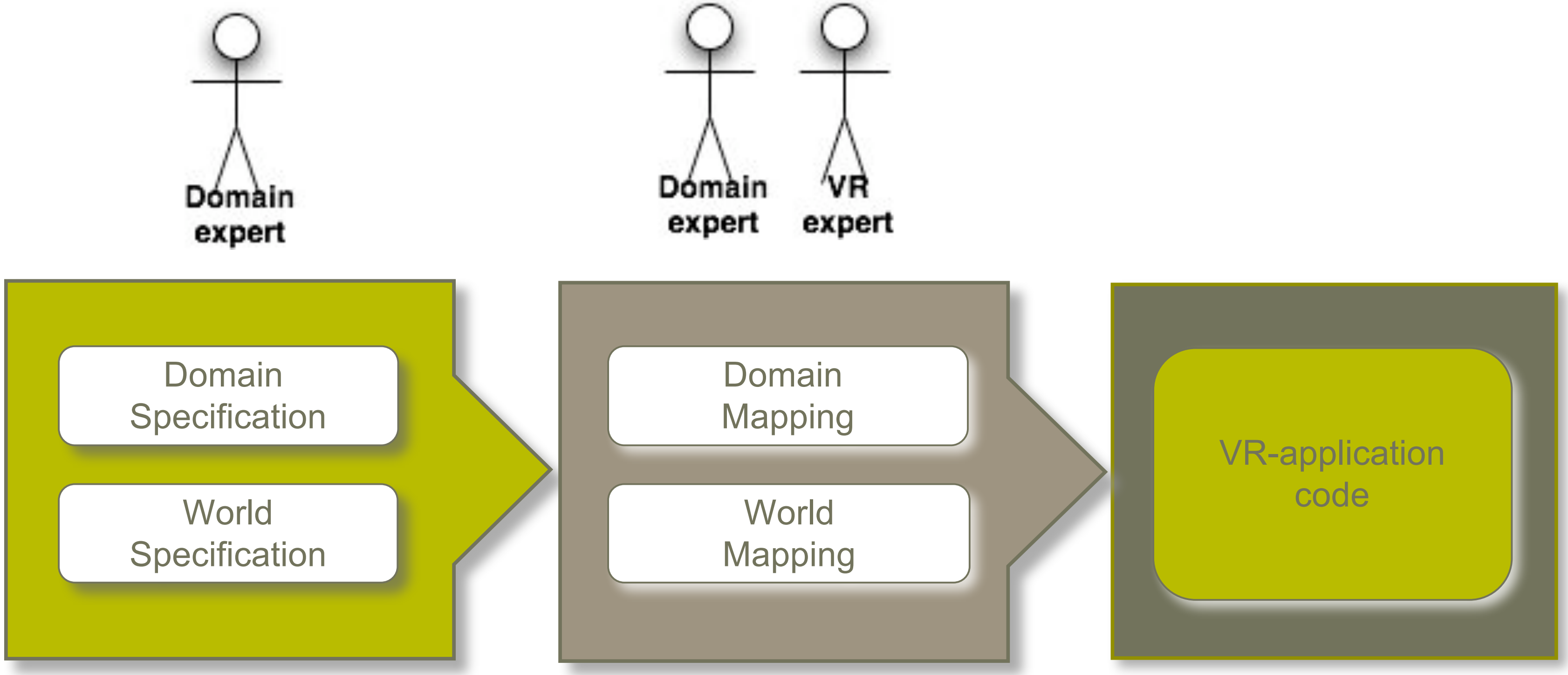


Code



Vrije Universiteit Brussel

VR-WISE Overview



Conceptual Specification Step

Mapping Step

Generation Step

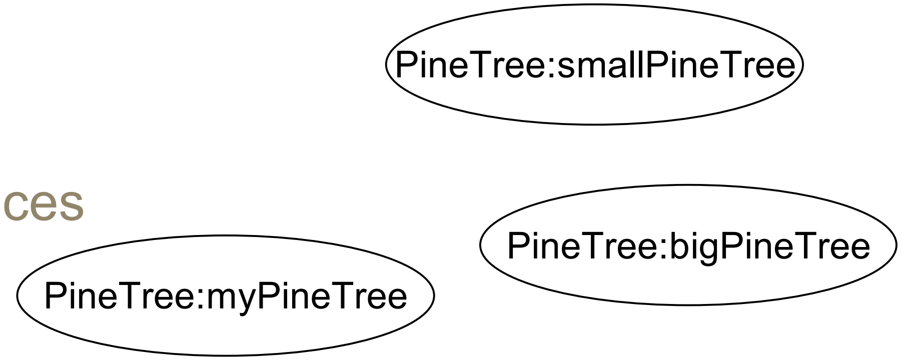
VR-WISE

- Graphical Conceptual Modeling Language
  - High-level concepts for modeling
    - Concepts and Objects
      - Including complex connected objects by means of joins
    - Positioning of concepts/objects
    - Behavior of concepts/objects
    - (Interaction)

Concepts and Objects

- Concepts
  - Domain concepts
    - Building, Pine Tree, Road Sign, Streetlight, Fountain, …
  - Properties
    - Visual: high, depth, material, ...
    - Non-visual: price, owner, ...

- Objects
  - Instances of concepts
    - several Pine Tree instances

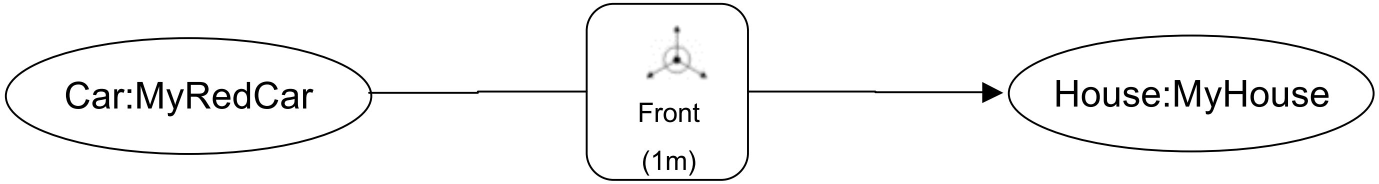


Positioning Objects

- Spatial relations
  - Objects can be positioned relative to each other by means of spatial relations instead of using exact coordinates
    - *in-front-of*, *above*, *left-of*, ...
  - More intuitive for non-VR-experts

Example:

my red car **is** 1 meter **in front** of my house

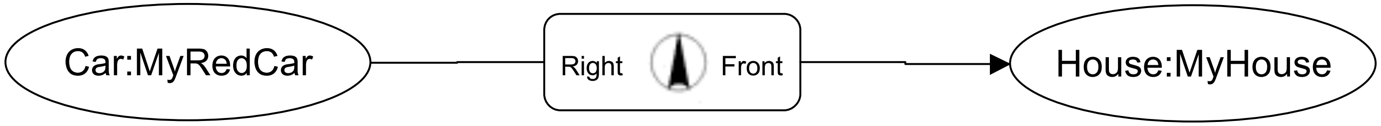


Positioning Objects

- Orientation relations
  - To orient objects relative to each other by means of their sides
    - *left*, *right*, *front*, *back*, *top*, *bottom*

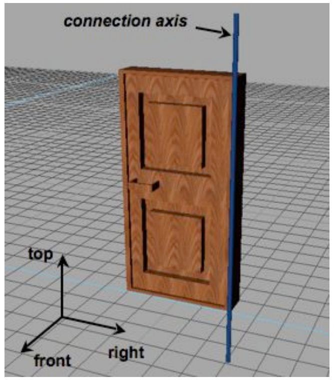
Example:

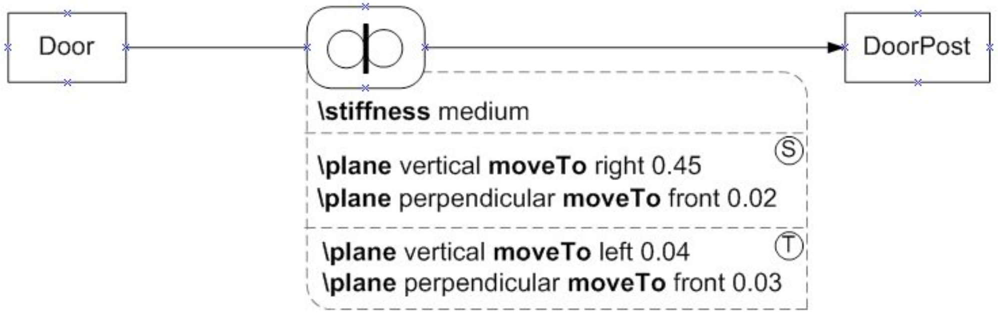
my red car's right side **is oriented towards** the front side of my house



Complex Connected Objects

- Objects can be connected in different ways
  - Connection axis relation




Door

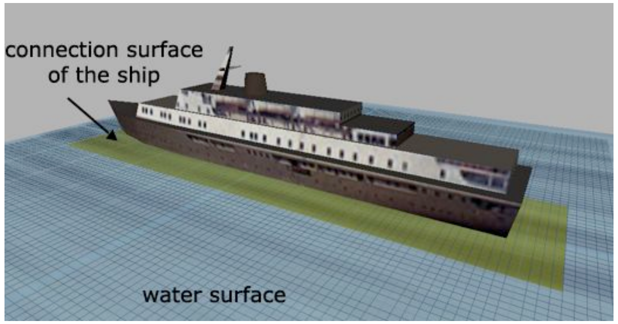
DoorPost

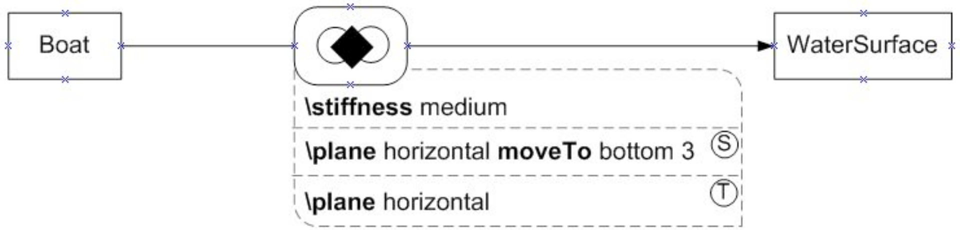
\stiffness medium

\plane vertical **moveTo** right 0.45
\plane perpendicular **moveTo** front 0.02

\plane vertical **moveTo** left 0.04
\plane perpendicular **moveTo** front 0.03

  - Connection surface relation




Boat

WaterSurface

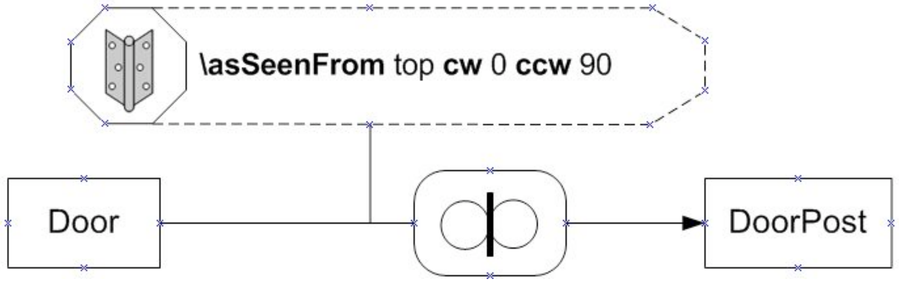
\stiffness medium

\plane horizontal **moveTo** bottom 3

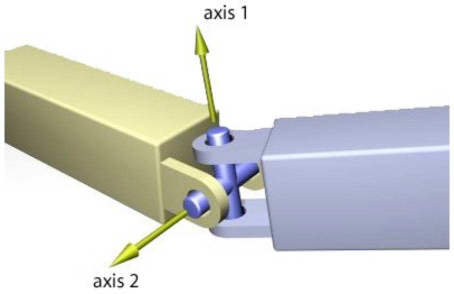
\plane horizontal

Complex Connected objects

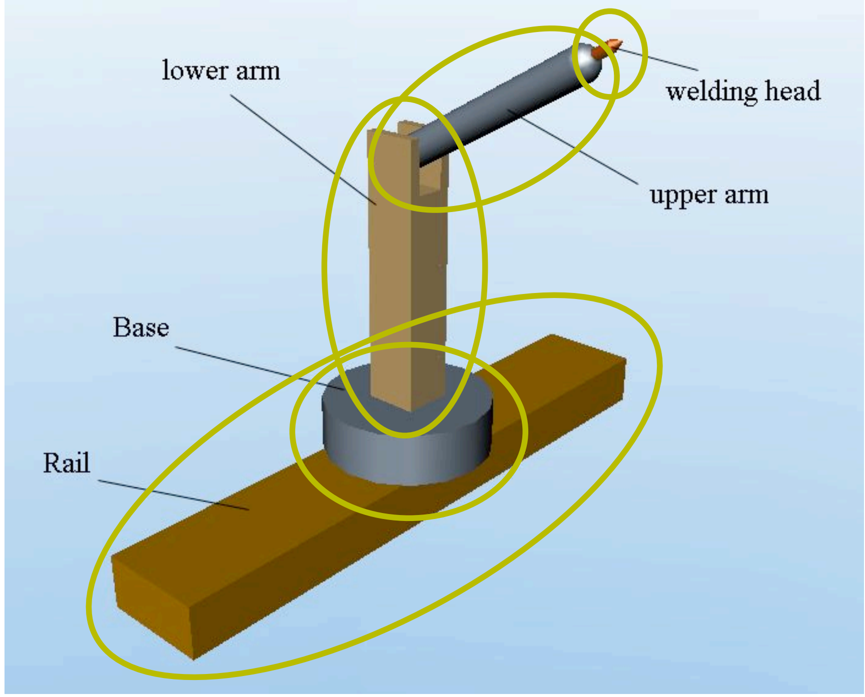
- Constraints may exist on connections
  - E.g., Hinge constraint for the door

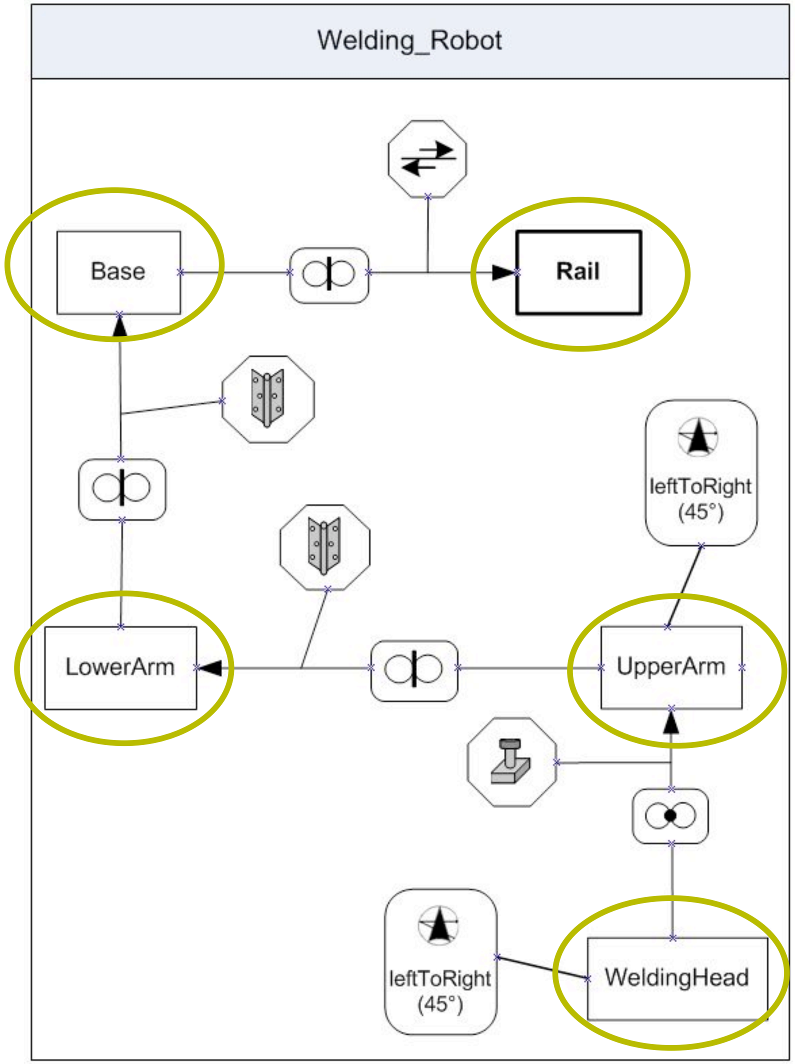


  - Or a Joystick constraint

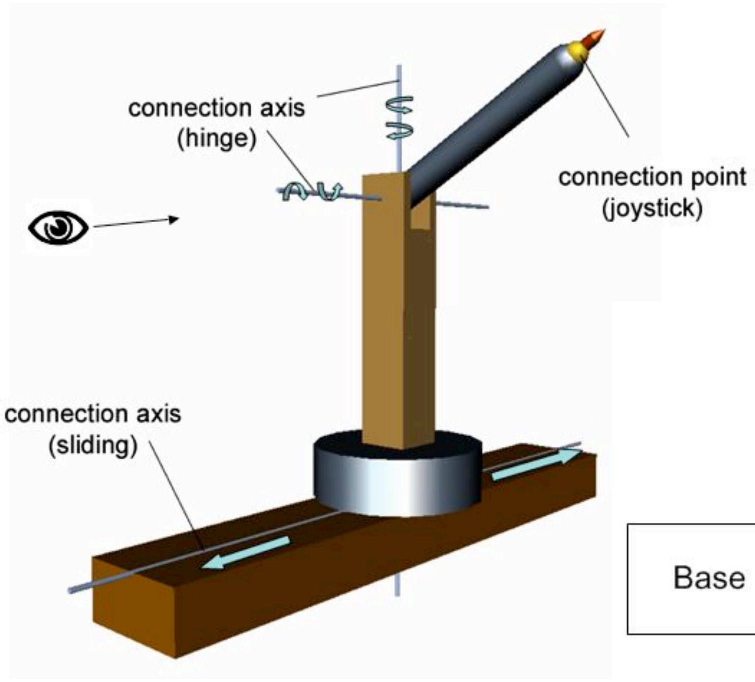


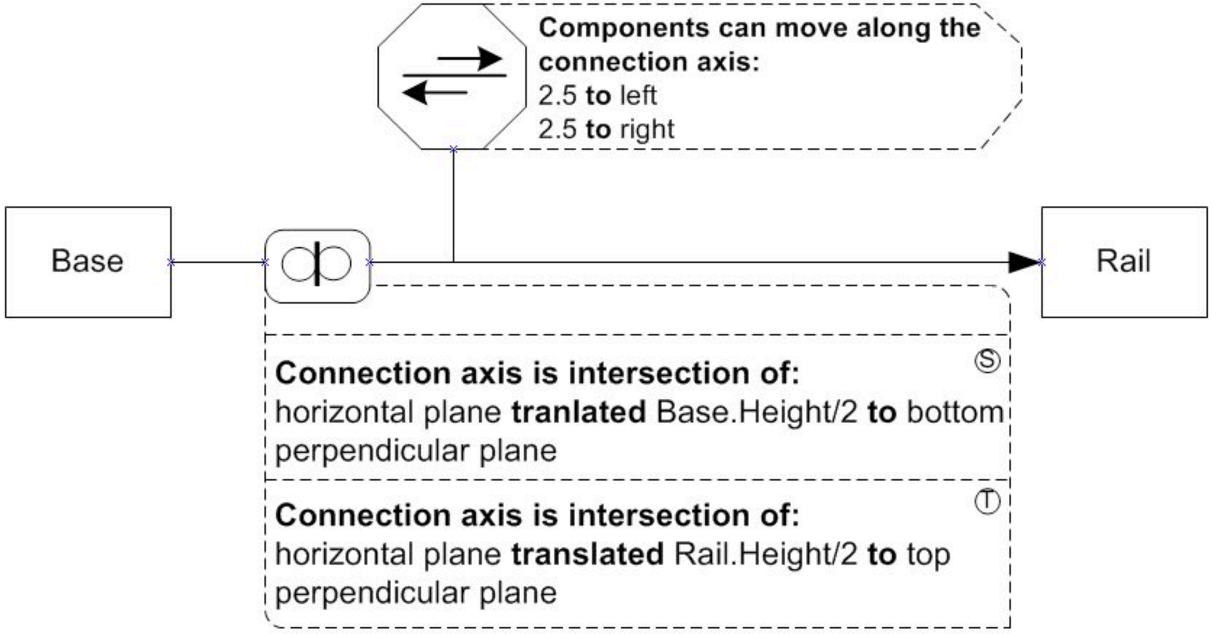
Complex Object - Example





Complex Object - Example (2)





Components can move along the connection axis:
2.5 **to** left
2.5 **to** right

Base

Rail

Ⓢ **Connection axis is intersection of:**
horizontal plane **tranlated** Base.Height/2 **to** bottom
perpendicular plane

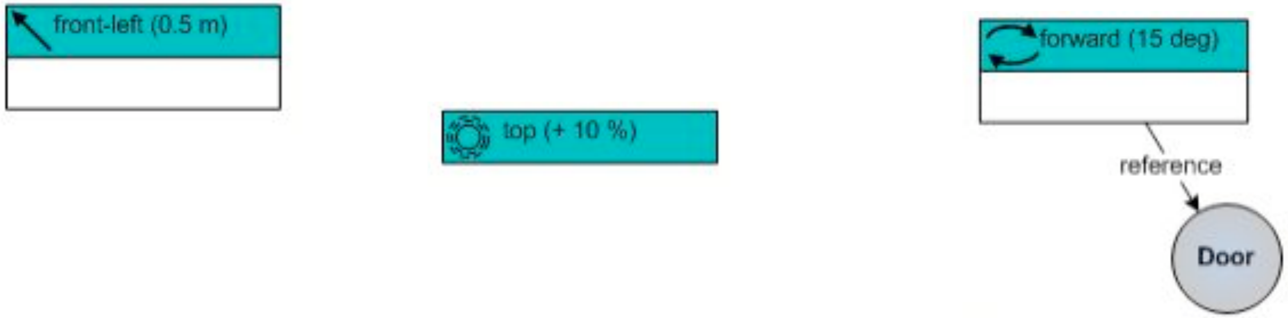
Ⓣ **Connection axis is intersection of:**
horizontal plane **translated** Rail.Height/2 **to** top
perpendicular plane

Behavior

- Specifying behaviors
  - Action-oriented approach
  - Independent from the static properties of the objects and independent of how the behavior is invoked
- Specifying the invocation of behaviors
  - Using events

Behavior

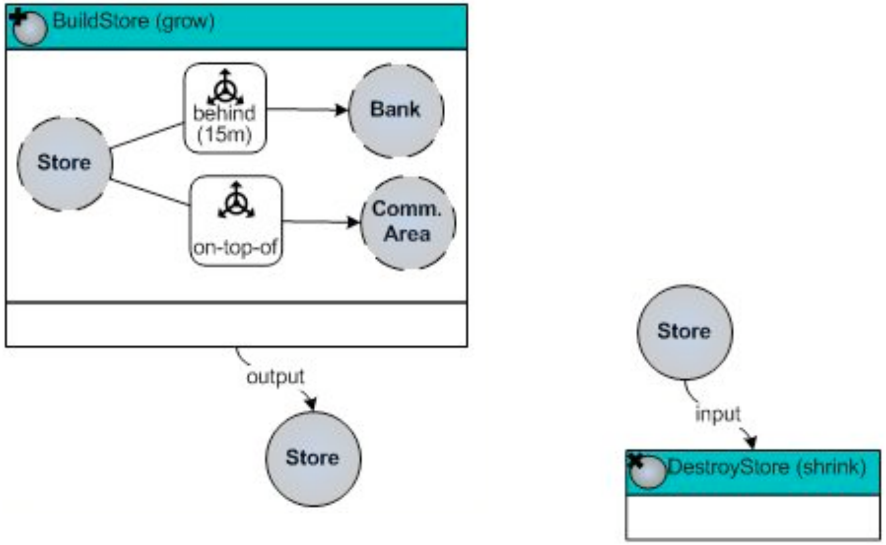
- Specifying behaviors
  - – Primitive behaviors (actions)
    - To change the position or the orientation of an object
      - – *move*, *turn*, *roll*, *resize*, *position*, *orientate* …



Behavior

– Primitive behaviors (actions)

- To change the appearance of an object at runtime
  - E.g., *transform*, *construct*, *destruct*, *group*, *ungroup*, *disperse*, *combine* …

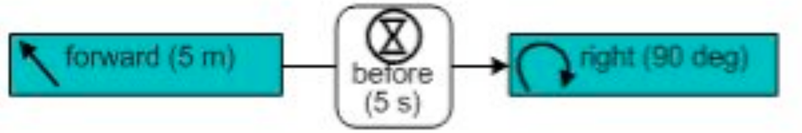


Behavior

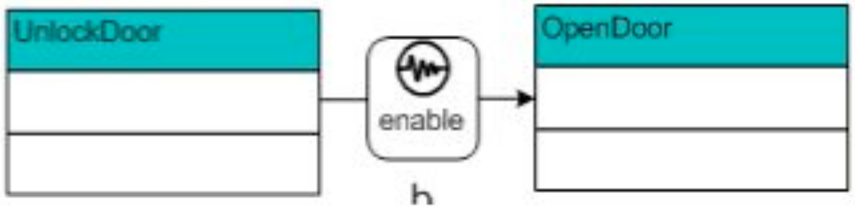
– Complex behaviors by combining behaviors by means of operators

Examples:

- Temporal operator for synchronizing behaviors

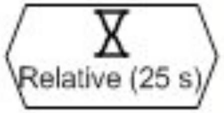


- Lifetime operator

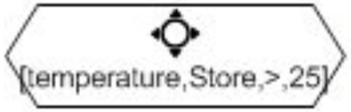


Behavior Invocation

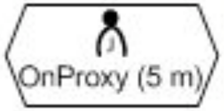
- Events are used to specify the triggering of behaviors
  - Time Event



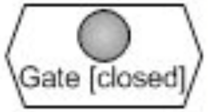
  - Context Event



  - User Event (user interaction)

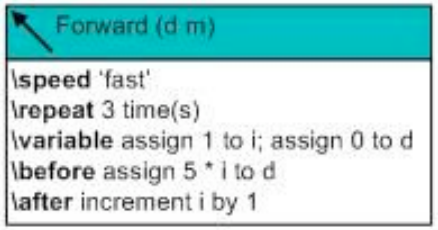


  - Collision Event (inter-object interaction)
  - Constraint Event



More complex behaviors

– More complexity can be expressed by means of a scripting language

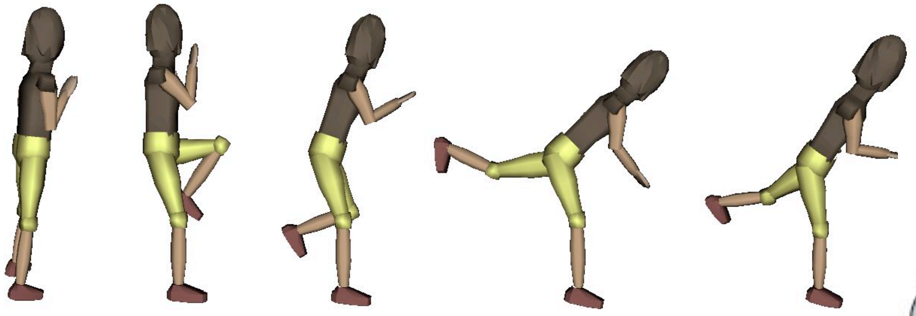


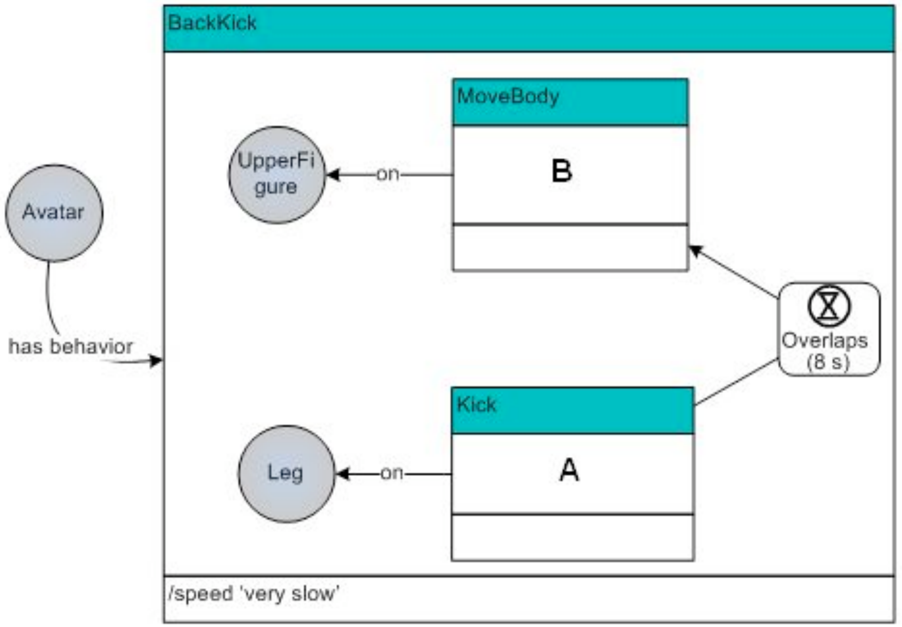
Forward (d m)

```
\speed 'fast'
\repeat 3 time(s)
\variable assign 1 to i; assign 0 to d
\before assign 5 * i to d
\after increment i by 1
```

Behavior Definition - Example

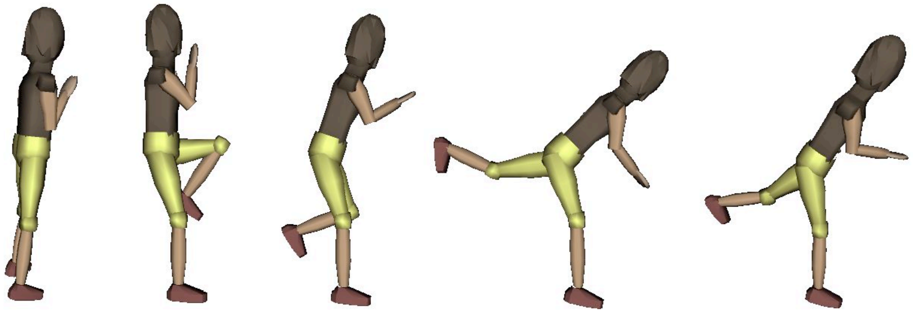
Mr. Phillip’s famous back kick.

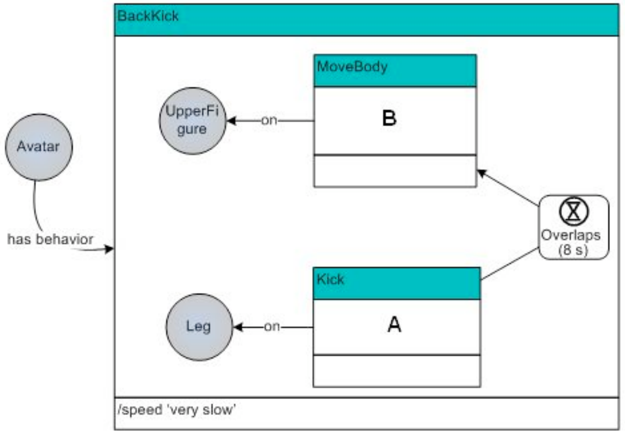


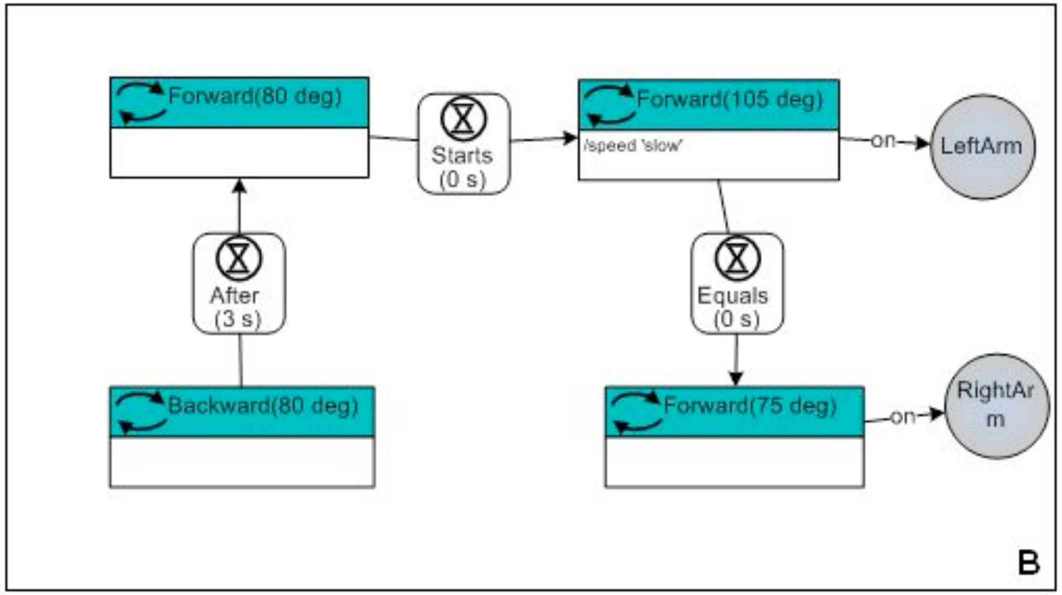


Behavior Definition - Example

Mr. Phillip's famous back kick.

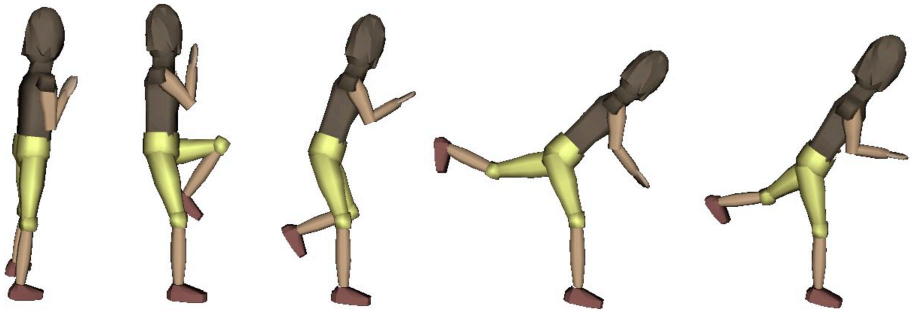


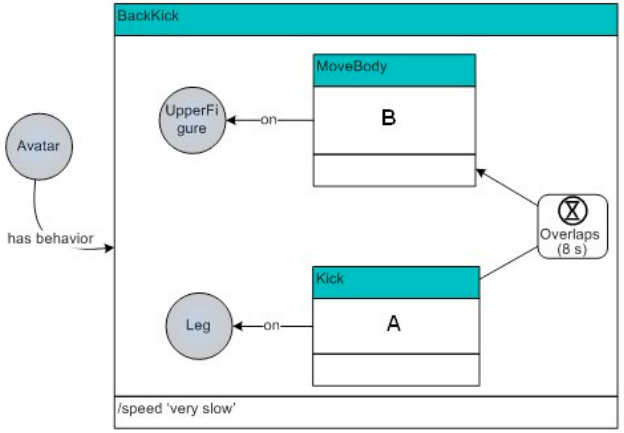


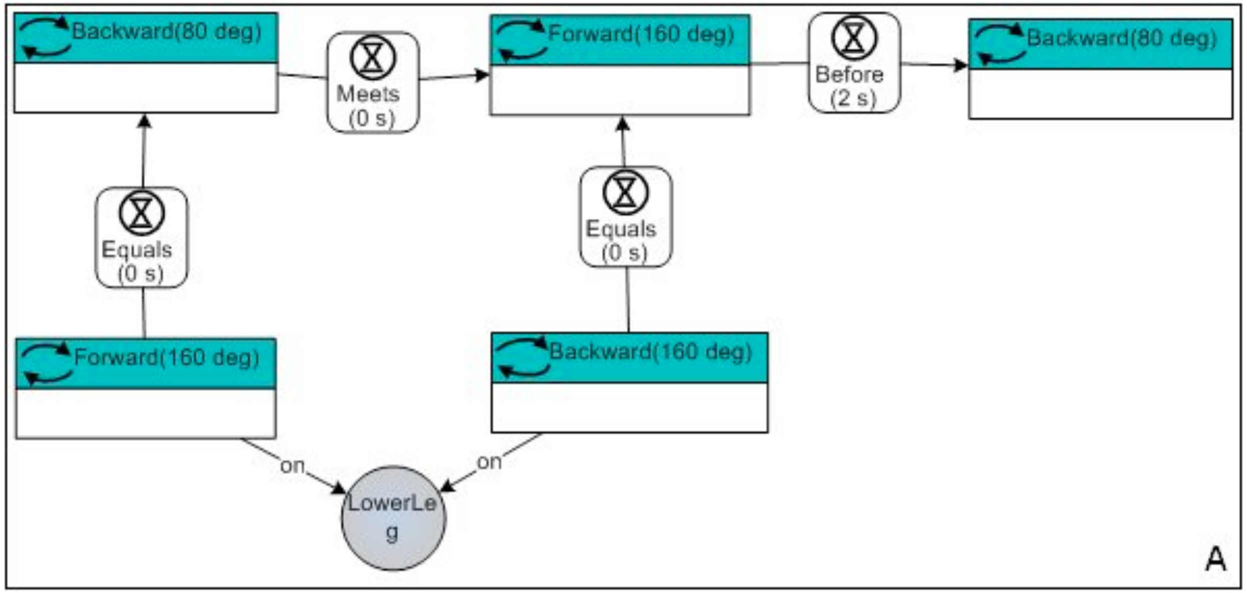


Behavior Definition - Example

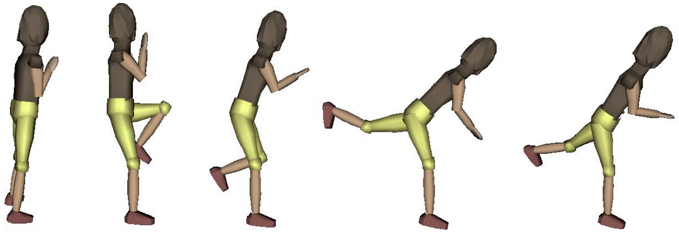
Mr. Phillip's famous back kick.







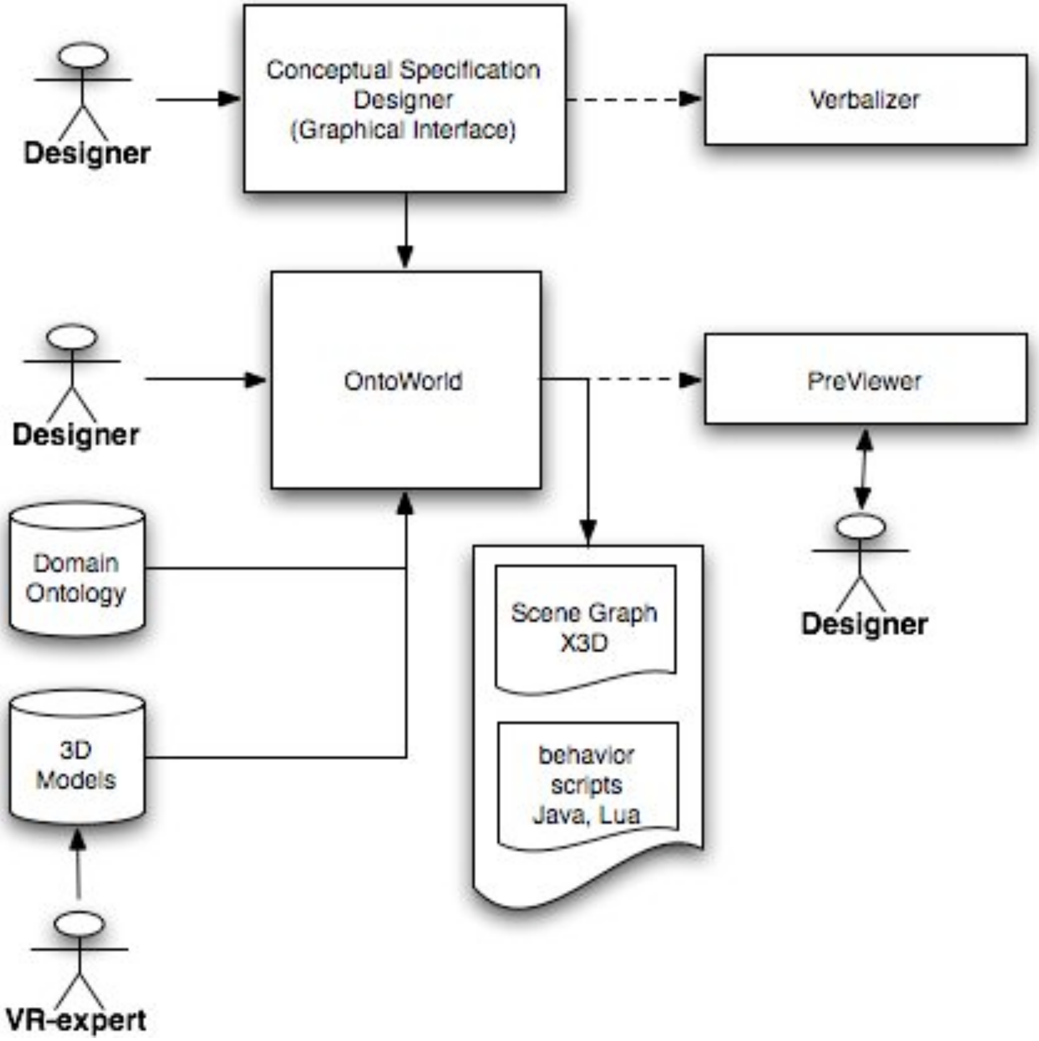
Generated Behavior



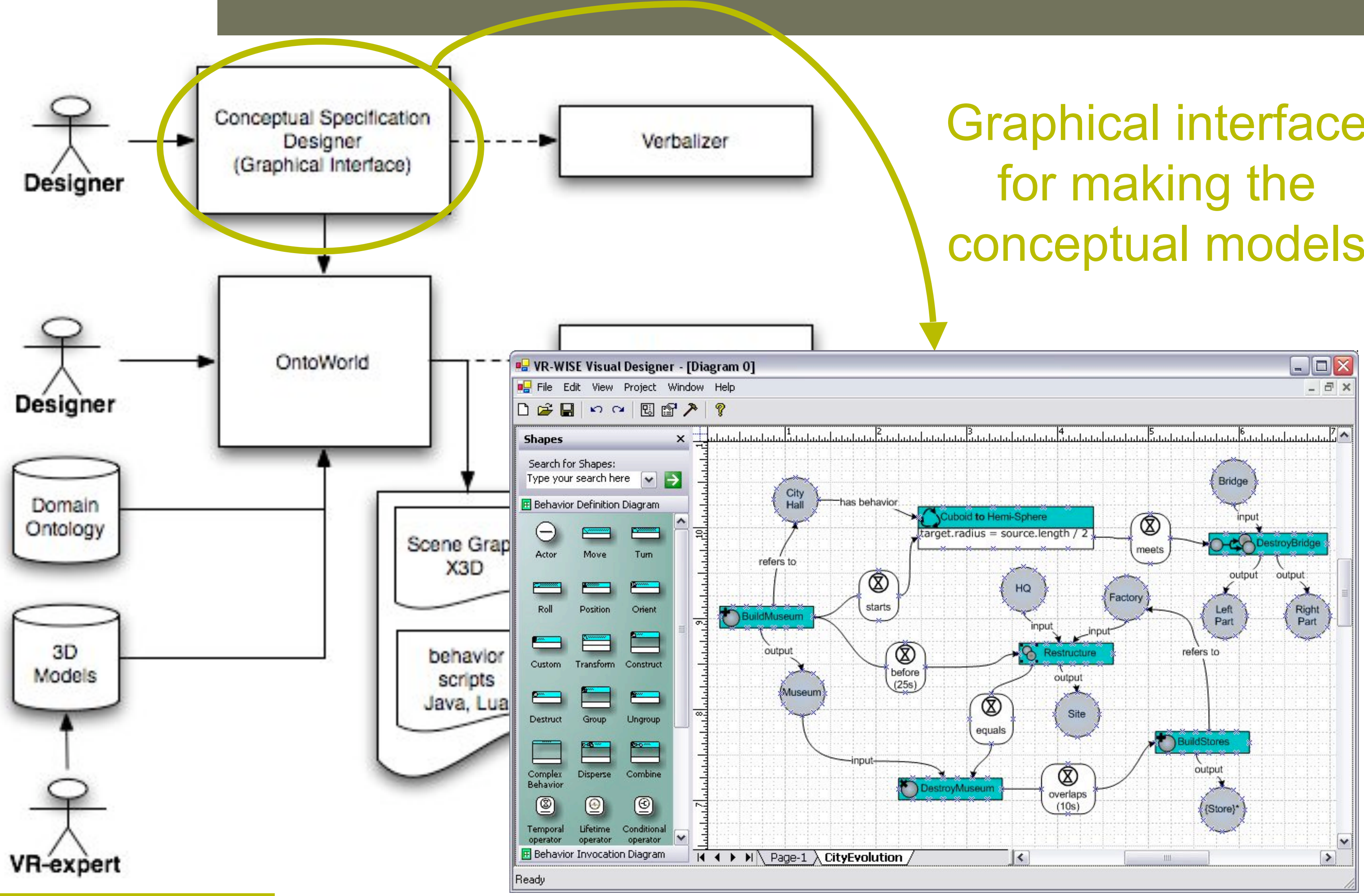
VR-WISE: Conceptual Modeling for Virtual Reality

- Context
- Motivation
- Objectives of the research
- Approach
- Tool support

Tool Support

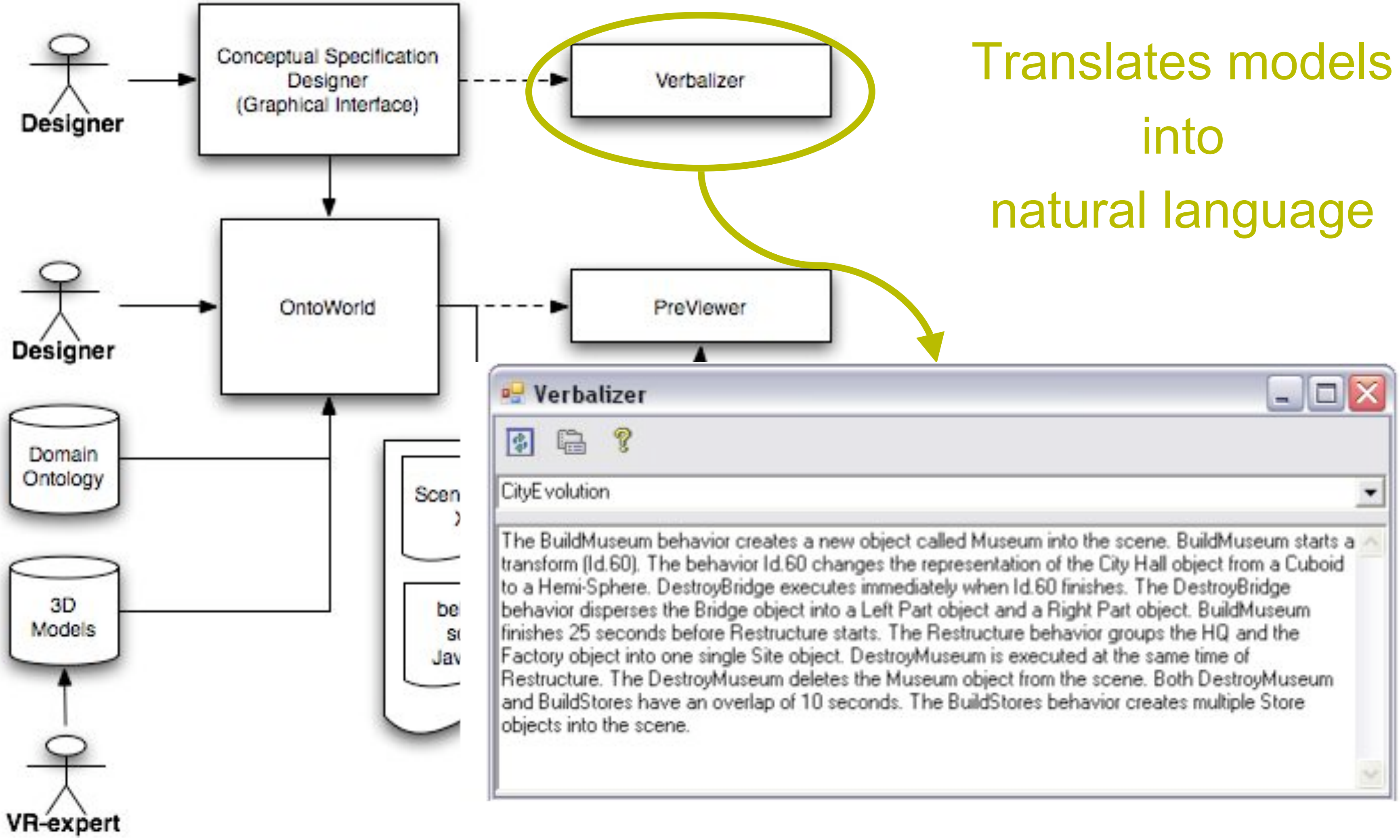


Tool Support



Graphical interface for making the conceptual models

Tool Support



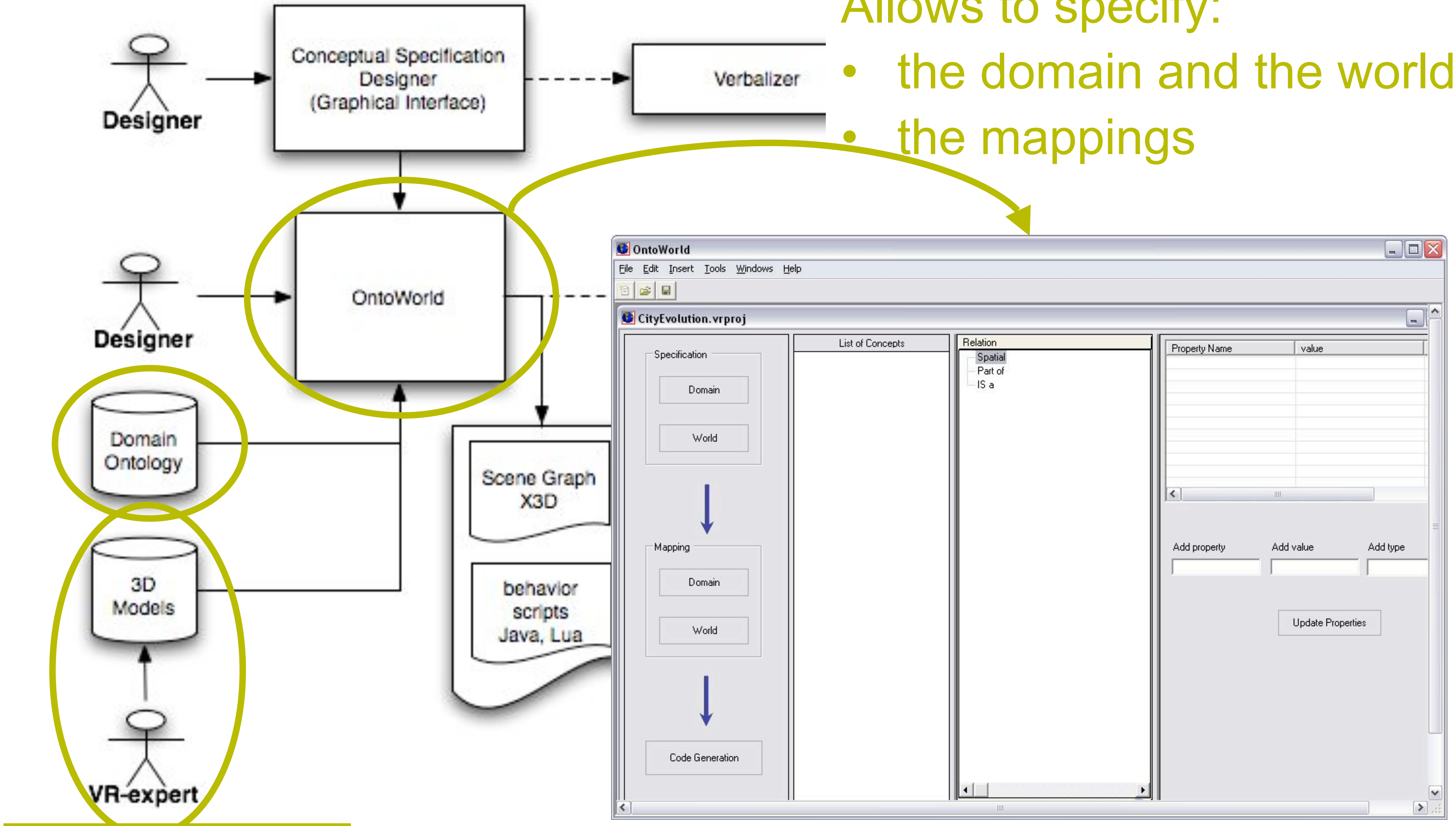
The BuildMuseum behavior creates a new object called Museum into the scene. BuildMuseum starts a transform (Id.60). The behavior Id.60 changes the representation of the City Hall object from a Cuboid to a Hemi-Sphere. DestroyBridge executes immediately when Id.60 finishes. The DestroyBridge behavior disperses the Bridge object into a Left Part object and a Right Part object. BuildMuseum finishes 25 seconds before Restructure starts. The Restructure behavior groups the HQ and the Factory object into one single Site object. DestroyMuseum is executed at the same time of Restructure. The DestroyMuseum deletes the Museum object from the scene. Both DestroyMuseum and BuildStores have an overlap of 10 seconds. The BuildStores behavior creates multiple Store objects into the scene.

Translates models into natural language

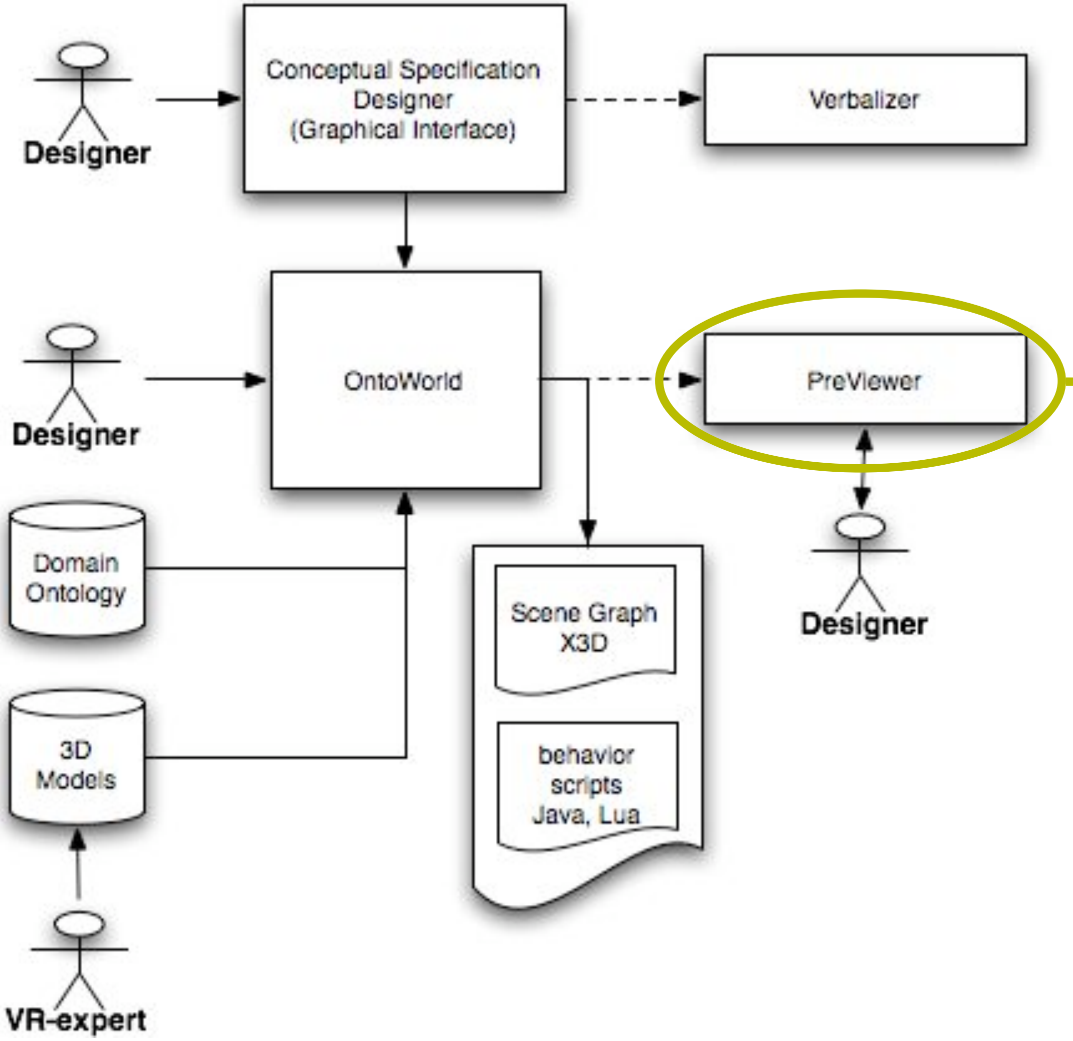
Tool Support

Allows to specify:

- the domain and the world
- the mappings

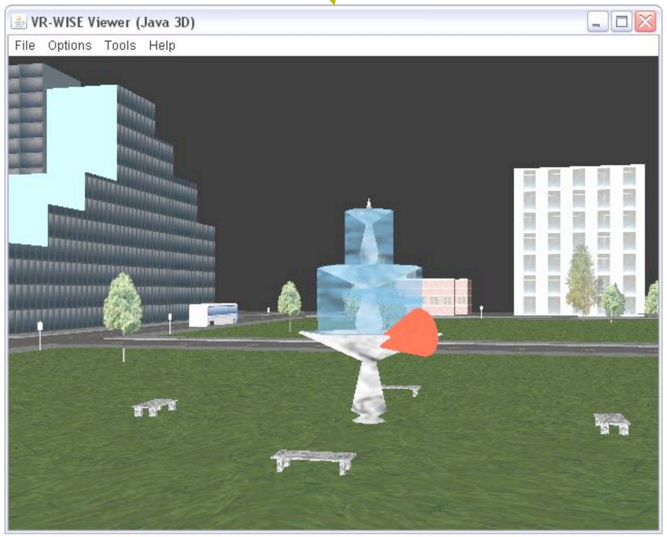


Tool Support



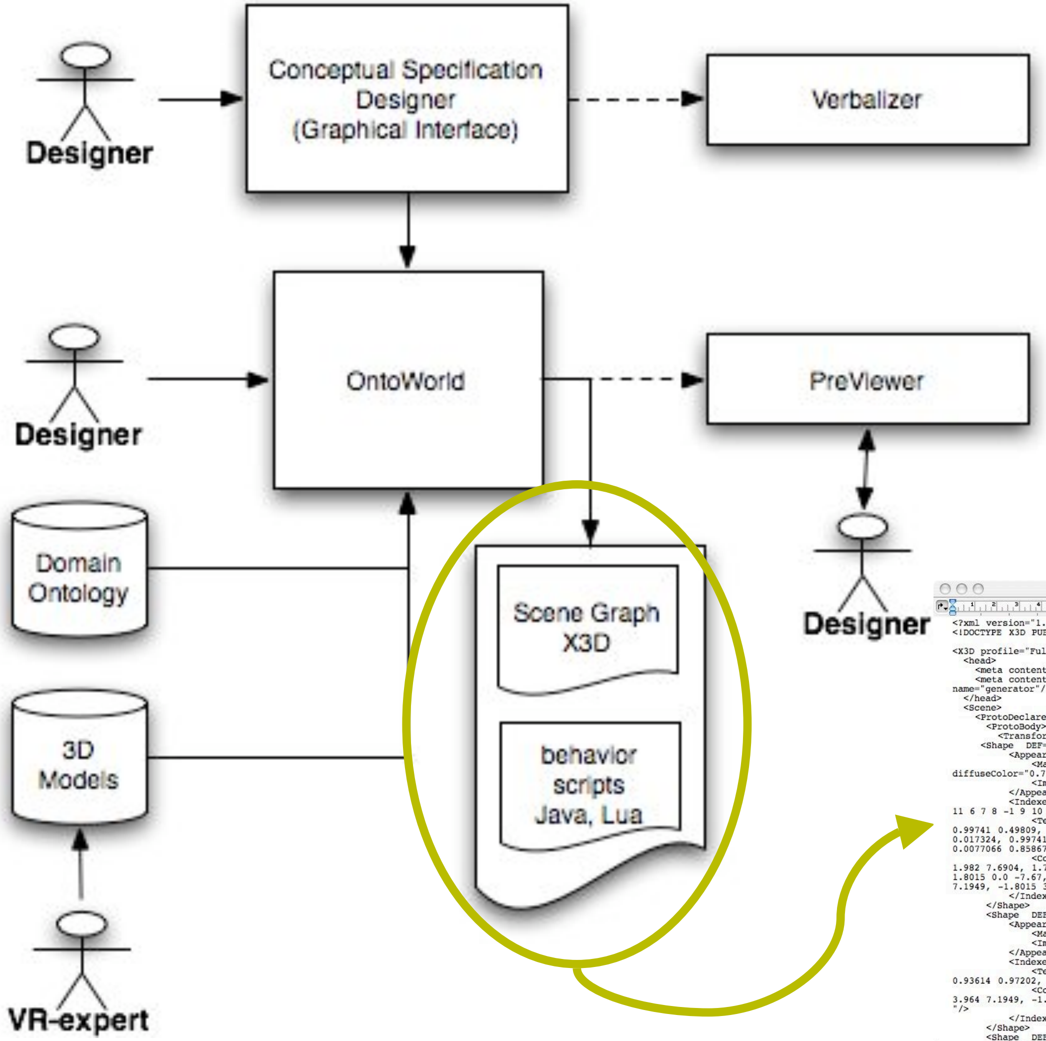
Allows to preview the world

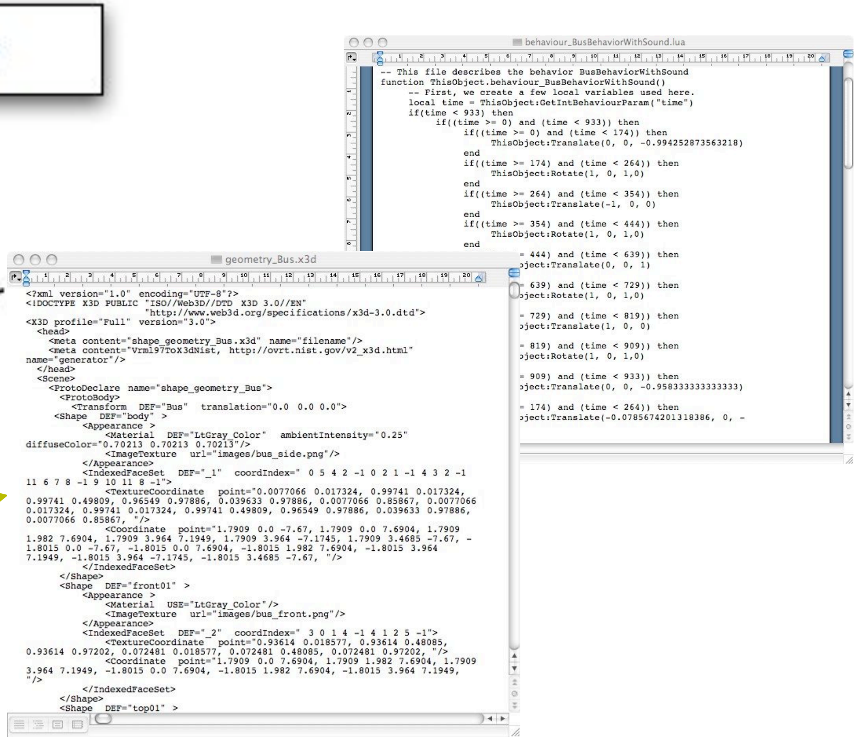
Fast prototyping



Tool Support

Code generation





```
-- This file describes the behavior BusBehaviorWithSound
function ThisObject.behaviour_BusBehaviorWithSound()
    -- First, we create a few local variables used here.
    local time = ThisObject:GetIntBehaviourParam("time")
    if(time < 933) then
        if((time >= 0) and (time < 933)) then
            if((time >= 0) and (time < 174)) then
                ThisObject:Translate(0, 0, -0.994252873563218)
            end
            if((time >= 174) and (time < 264)) then
                ThisObject:Rotate(1, 0, 1,0)
            end
            if((time >= 264) and (time < 354)) then
                ThisObject:Translate(-1, 0, 0)
            end
            if((time >= 354) and (time < 444)) then
                ThisObject:Rotate(1, 0, 1,0)
            end
                = 444) and (time < 639)) then
              oject:Translate(0, 0, 1)

                = 639) and (time < 729)) then
              oject:Rotate(1, 0, 1,0)

                = 729) and (time < 819)) then
              oject:Translate(1, 0, 0)

                = 819) and (time < 909)) then
              oject:Rotate(1, 0, 1,0)

                = 909) and (time < 933)) then
              oject:Translate(0, 0, -0.958333333333333)

                = 174) and (time < 264)) then
              oject:Translate(-0.0785674201318386, 0, -
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
                     "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Full" version="3.0">
  <head>
    <meta content="shape_geometry_Bus.x3d" name="filename"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html"
name="generator"/>
  </head>
  <Scene>
    <ProtoDeclare name="shape_geometry_Bus">
      <ProtoBody>
        <Transform DEF="Bus" translation="0.0 0.0 0.0">
     <Shape DEF="body" >
          <Appearance >
              <Material DEF="LtGray_Color" ambientIntensity="0.25"
diffuseColor="0.70213 0.70213 0.70213"/>
              <ImageTexture url="images/bus_side.png"/>
          </Appearance>
          <IndexedFaceSet DEF="_1" coordIndex=" 0 5 4 2 -1 0 2 1 -1 4 3 2 -1
11 6 7 8 -1 9 10 11 8 -1">
              <TextureCoordinate point="0.0077066 0.017324, 0.99741 0.017324,
0.99741 0.49809, 0.96549 0.97886, 0.039633 0.97886, 0.0077066 0.85867, 0.0077066
0.017324, 0.99741 0.017324, 0.99741 0.49809, 0.96549 0.97886, 0.039633 0.97886,
0.0077066 0.85867, "/>
              <Coordinate point="1.7909 0.0 -7.67, 1.7909 0.0 7.6904, 1.7909
1.982 7.6904, 1.7909 3.964 7.1949, 1.7909 3.964 -7.1745, 1.7909 3.4685 -7.67, -
1.8015 0.0 -7.67, -1.8015 0.0 7.6904, -1.8015 1.982 7.6904, -1.8015 3.964
7.1949, -1.8015 3.964 -7.1745, -1.8015 3.4685 -7.67, "/>
          </IndexedFaceSet>
      </Shape>
      <Shape DEF="front01" >
          <Appearance >
              <Material USE="LtGray_Color"/>
              <ImageTexture url="images/bus_front.png"/>
          </Appearance>
          <IndexedFaceSet DEF="_2" coordIndex=" 3 0 1 4 -1 4 1 2 5 -1">
              <TextureCoordinate point="0.93614 0.018577, 0.93614 0.48085,
0.93614 0.97202, 0.072481 0.018577, 0.072481 0.48085, 0.072481 0.97202, "/>
              <Coordinate point="1.7909 0.0 7.6904, 1.7909 1.982 7.6904, 1.7909
3.964 7.1949, -1.8015 0.0 7.6904, -1.8015 1.982 7.6904, -1.8015 3.964 7.1949,
"/>
          </IndexedFaceSet>
      </Shape>
      <Shape DEF="top01" >
```

VR-WISE

- Ontology-based
  - Domain ontology to describe the world
    - Allows to use terminology of application domain
  - Ontologies as internal knowledge representation
- Semantic Virtual Worlds
  - Use of domain ontologies allows to capture real world semantics
    - Semantic search engines
    - Semantic annotations

VR-WISE: Conceptual Modeling for Virtual Reality

- Context
- Motivation
- Objectives of the research
- Approach
- Tool support
- Conclusions
- Current & future work

Conclusions

- Conceptual design phase in the development process of VR application
  - More people can create a VR application
  - More people can be involved
    - Domain experts; other stakeholders
    - Easier to satisfy the requirements and the expectations of stakeholders
  - Model-based
    - Code can be generated
    - Less expensive and faster
  - Ontology-based
    - Easier to incorporate semantics
    - More usable, allows for semantic search, semantic annotations

Vrije Universiteit Brussel

Current and future work

- Extending the set of modeling concepts:
  - More primitive behaviors, e.g., coloring, sound, ...
  - Mechanism for combining connections
  - More constraints, like constraints on behavior
  - Cameras, viewpoints, light sources, shadows, ...
  - Interaction-controlled behavior
  - Avatars
- Current work
  - Scenarios
  - Patterns for modeling behavior/the scene
  - Semantic annotations for existing worlds

More information

Part of this work has been done in the context of the OntoBasis project and the VR-DeMo project (IWT)

See http://vr-wise.vub.ac.be