

Einführung in die Erweiterte
Realität
- 4. VRML -

Gudrun Klinker

Nov 11, 2003

Agenda

1. General information on VRML
2. Basic concepts: minimal VRML worlds
3. Combination of several objects
4. Model reusability
5. Levels of detail
6. Animation
7. Anchor points
8. Sensors
9. Scripts

1. General Information on VRML

- **The VRML repository:**
<http://www.web3d.org/vrml/vrml.htm>
- **Browsers und plug-ins:** e.g. Cosmo Player
<http://www.cai.com/cosmo/home.htm>
- **Book (excerpts only):**
VRML 2.0 Source Book, Andrea L. Ames, David R. Nadeau and John L. Moreland, 2nd Edition, Wiley & Sons, 1996. ISBN 0-471-16507-7

VRML-Models in the WWW

- <http://www.web3d.org/vrml/oblib.htm>
- <http://www.rdservice.de/german/produkte/VRCreator/library/vrlibrary.htm>
- <http://home.t-online.de/home/kiwano6/models.htm>
- <http://www.rccad.com/Gallery.htm>
- for recent demos and examples, see
<http://www.bruegge.in.tum.de/people/klinker/MMT/Einfuehrungen/VRML/>
<http://www.bruegge.in.tum.de/teaching/ss00/ar/vrml/>

2. Basic Concepts: Minimal VRML worlds

VRML Files

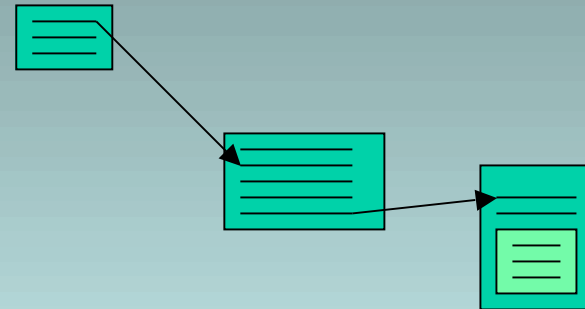
ASCII Files: `<my_world>.wrl`

- **VRML header:** `#VRML V2.0 utf8`
- **Comments:** `#`
- **Nodes:** Shapes, transformations, timers, interpolators, sensors, scripts, ...
- **Routes**

Object Descriptions in VRML

Nodes (nested, grouped) with

- **Fields** and **Field Values**
of specified **Field Value Types**
 - **Inputs** and **Outputs**
 - **Bounding Boxes**



Examples:

- Shape nodes

```
Shape {  
    appearance SFNode,  
    geometry SFNode  
}
```

- Other nodes:
transformations, timers, interpolators, sensors,
scripts, ...

<Chap. 2, 3>

Geometry Nodes

- Cylinder -

```
#VRML V2.0 utf8
```

```
# default cylinder
```

```
Shape {  
  geometry Cylinder {  
  } # Cylinder  
} # Shape
```

```
#VRML V2.0 utf8
```

```
# cylinder of specified size
```

```
Shape {  
  geometry Cylinder {  
    height 4.0  
    radius 2.0  
  } # Cylinder  
} # Shape
```


Geometry Nodes

- Box, Sphere -

```
#VRML V2.0 utf8

# box of specified size
Shape {
  geometry Box {
    size 2.0 1.0 4.0 # dx,dy,dz
  } # Box
} # Shape
```

```
#VRML V2.0 utf8

# sphere of specified radius
Shape {
  geometry Sphere {
    radius 2.5
  } # Sphere
} # Shape
```

Geometry Nodes

- Cone, Text -

```
#VRML V2.0 utf8

# cone of specified size
Shape {
  geometry Cone {
    bottomRadius 5.0
    height 1.5
  } # Cone
} # Shape
```

```
#VRML V2.0 utf8

# text of specified font
Shape {
  geometry Text {
    string ["Hello World", "Cheers"]
    fontStyle FontStyle {
      family "SERIF"
      style "BOLD"
      size 0.5
      justify "MIDDLE"
    }
  } # Text
} # Shape
```

Geometry Nodes

- Point Sets, Indexed Line Sets -

```
#VRML V2.0 utf8

# set of 3D points
Shape {
  geometry PointSet {
    coord Coordinate {
      point [
        0.0 0.0 0.0, # index 0
        0.0 1.0 0.0, # index 1
        1.0 1.0 0.0, # index 2
        1.0 0.0 0.0 # index 3
      ] # point
    } # Coordinate
  } # PointSet
} # Shape
```

```
#VRML V2.0 utf8

# set of line indices
Shape {
  geometry IndexedLineSet {
    coord Coordinate {...}
    coordIndex [
      0, 1, 2, -1,
      0, 2, 3
    ] # coordIndex
  } # IndexedLineSet
} # Shape
```

Geometry Nodes

- Indexed Face Sets -

```
#VRML V2.0 utf8

# set of line indices
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {...}
    coordIndex [
      0, 1, 2, -1,
      0, 2, 3
    ] # coordIndex
  } # IndexedLineSet
} # Shape
```

Appearance Nodes

- Materials/Colors-

```
#VRML V2.0 utf8
```

```
Shape {  
  geometry SFNode,  
  appearance Appearance {  
    material Material {  
      ambient Intensity 0.2  
      diffuse Color    0.8 0.8 0.8  
      specular Color   0.0 0.0 0.0  
      ....  
    } # Material  
    texture SFNode  
    textureTransform SFNode  
  } # Appearance  
} # Shape
```

3. Combination of Several Objects

- **Inlining** of other files <Chap. 12>
`Inline {url [...], ...}`
- **Geometric Transformations** <Chap. 5-7>
`Transform {
 children [...]
 translation
 rotation
 scale
}`

Geometric Transformations

- Box, Sphere -

```
#VRML V2.0 utf8

# cross in xz-plane; translated in x-direction, rotated 45 deg around y-axis
Transform {
  children [
    Shape { geometry Box { size 1.0 1.0 4.0 } } # Box1
    Shape { geometry Box { size 4.0 1.0 1.0 } } # Box 2
  ]
  translation 10.0 0.0 0.0
  rotation 0.0 0.0 1.0 0.785 # in radians; pi/4= 45 degrees
  scale 1.0 0.1 1.0
} #Transform
```

Example: Sphere plus Cylinder

```
#VRML V2.0 utf8
```

```
# 1. Object
```

```
Inline {url "sphere.wrl"}
```

```
# 2. Object
```

```
Transform {
```

```
  translation 0.0 2.0 0.0
```

```
  rotation 0.0 0.0 1.0 -0.524 # radians; 30 degrees
```

```
  children [ Inline {url "cylinder-c.wrl"} ]
```

```
} # Transform
```


Example: Blocks World (Baufix)

- Used as library for further demos

```
#VRML V2.0 utf8
Transform {
  translation -6.0 0.5 0.0
  children [ Inline {url "yellow-box1.wrl"} ]
} # Transform
Transform {
  translation -4.5 1.0 0.0
  children [ Inline {url "blue-box2.wrl"} ]
} # Transform
...
```

4. Model Reusability

Reuse of Objects

- Inlining
- DEF and USE

```
DEF Green-Box3
  Inline {url "../blocks-
    world/green-box3.wrl"
  } # Inline cylinder
...
USE Green-Box3
```

<Chap. 2>

Multiple Use of Groups of Objects

- Groups

```
Group {
  children [
    ...
  ]
  ...
}
```

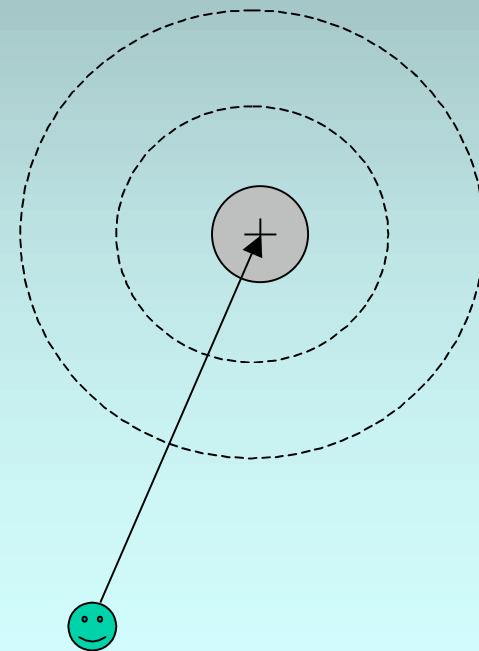
<Chap. 3, 11>

Example: Replicated Model

```
DEF Rep-Structure Group {
  children [
    DEF Baufix-Structure Inline {url "structure-group.wrl"} # 1. structure
    Transform {                                     # replicated structure
      translation 0.0 0.0 -2.0
      children [USE Baufix-Structure]
    } # Transform 2. Structure
    # ... Further structures ...
  ] # children
} # group
Transform {
  translation 5.0 0.0 0.0
  children [USE Rep-Structure]                     # replicated group
}
```

5. Level of Detail

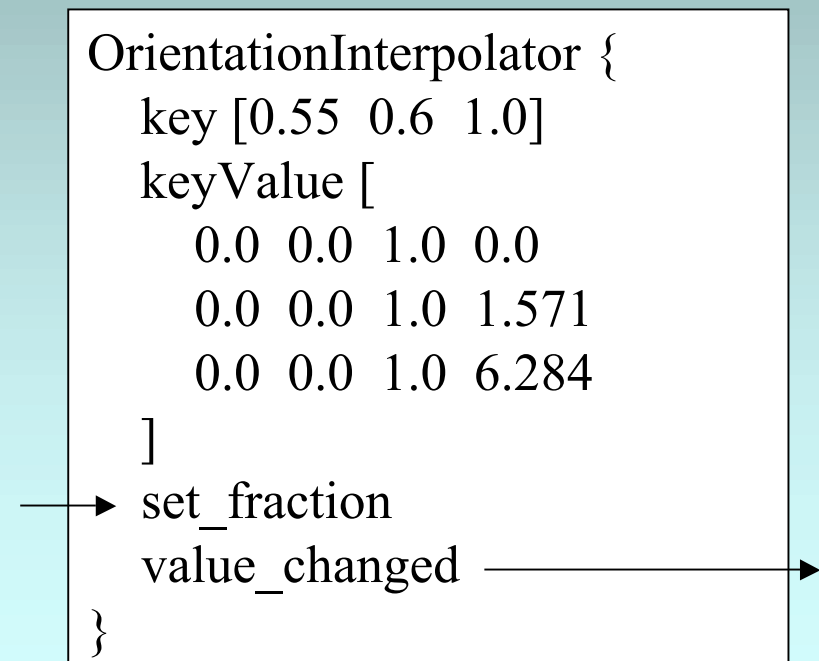
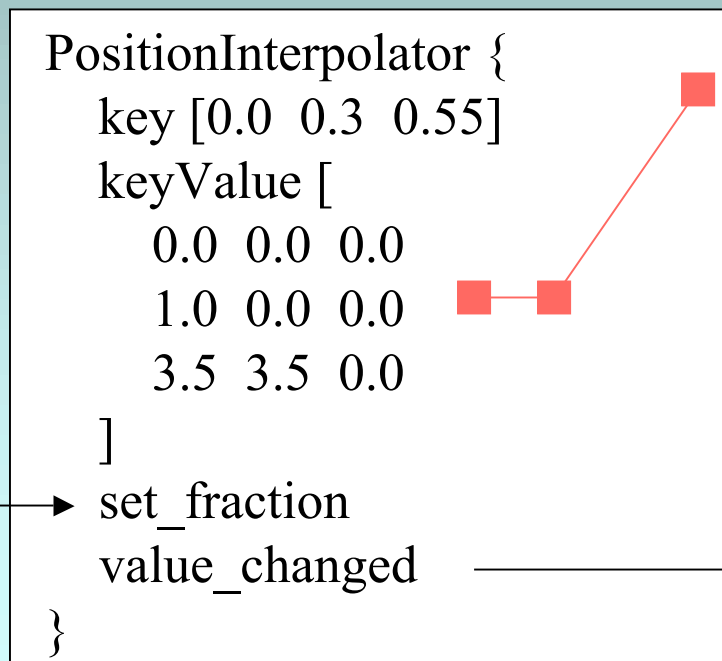
- Geometric objects with varying amounts of complexity
 - selective presentation depending on the distance between the viewer and the object center
 - LOD {
 center 0.0 0.0 0.0
 range [7.5 15.0]
 level [
 Inline {url “obj-highdetail.wrl”}
 Inline {url “obj-mediumdetail.wrl”}
 Inline {url “obj-lowdetail.wrl”}
]
}



<Kap. 25>

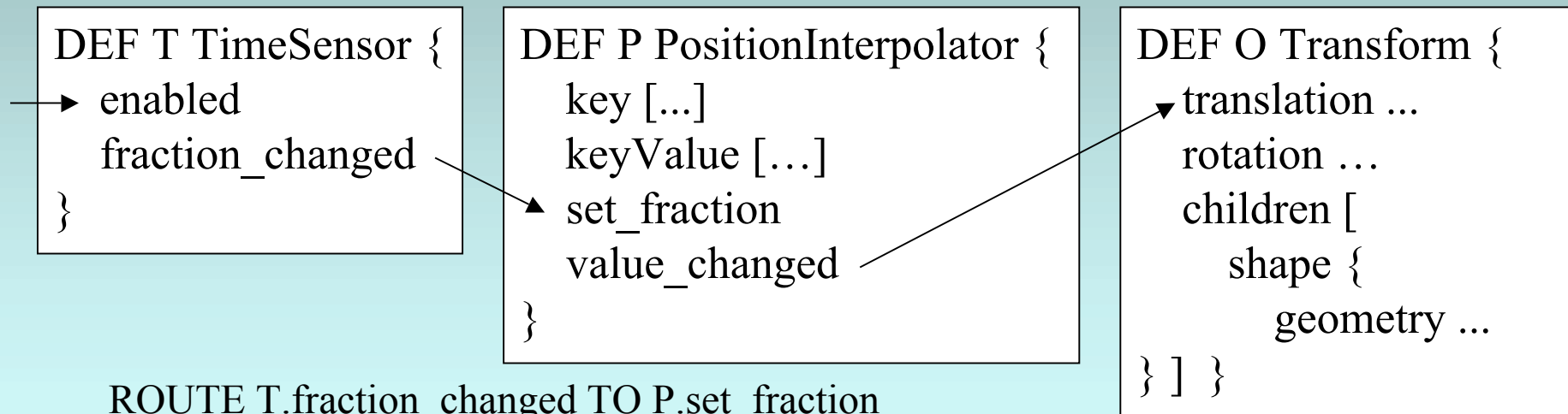
6. Keyframe Animation

- VRML: linear Interpolation between n “key frames”



ROUTEs

- Connections between nodes in order to forward events



ROUTE T.fraction_changed TO P.set_fraction

ROUTE P.value_changed TO O.set_translation

7. Anchors

- Transition from one world into another (and back) via [Anchors](#).
- (e.g.: Dungeon Worlds, different levels)

```
Anchor {  
  url ["...", "...", ...]  
  children [...]  
  description "..."  
  ...  
}
```

Anchors for AR

- “hot-spots”
 - as a 3D menu (e.g.: next to a real object) supplying dynamic, stepwise construction information
 - contained within the virtual models, providing additional information for specific object parts
 - contained within the virtual models, providing alternate object presentations
 - embedded as virtual camera symbols within the scene, representing alternate (video-based) views of the scene (--> tele-presence)

8. Sensors

- Sensors for various interaction modes:
 - cursor position (TOUCH): isOver, isActive
DEF Touch-Green TouchSensor {}
ROUTE Touch-Green.touchTime TO Clock.set_startTime
 - cursor motion (MOTION)
 - CylinderSensor
 - PlaneSensor
 - SphereSensor
DEF Spin-Cylinder SphereSensor {}
ROUTE Spin-Cylinder.rotation_changed TO Red-Cylinder2-
b.set_rotation
 - when nested sensors are used, the innermost is selected
 - several modi for progress (Offsets)

More on Sensors

- Viewer proximity (for cursor motion)
 - visibility
 - VisitivitySensor {}
 - viewer proximity
 - ProximitySensor {}
 - viewer collision
 - Collision node

```
DEF MyCollision Collision {
  children []
  collide TRUE
}
ROUTE MyCollision.collideTime TO OuchSound.set_startTime
...
```

9. Skripts

- Connection VRML -> Java for generating more complex behavior
 - complex motion paths of objects (e.g., modelling a realistic walking motion)
 - intelligent reactions of objects
 - communication in distributed systems (e.g., for multi-player games)

Skripts

- Example: Motion Interpolator

VRML:

```
DEF Mover Script {
  url "move1.js"
  eventIn  SFFloat  set_fraction
  eventOut SFVec3f value_changed
}
...
ROUTE Clock.fraction_changed TO Mover.set_fraction
ROUTE Mover.value_changed TO Red-Cylinder2-b.set_translation
```

Java file (move1.js):

```
// move a shape in a sinusoidal path
function set_fraction (fraction,eventTime) {
  value_changed[0] = 6.0 + Math.sin (fraction * 6.28); // x component
  value_changed[1] = -2.0; // y component
  value_changed[2] = -2 +fraction*4; // z component
}
```

