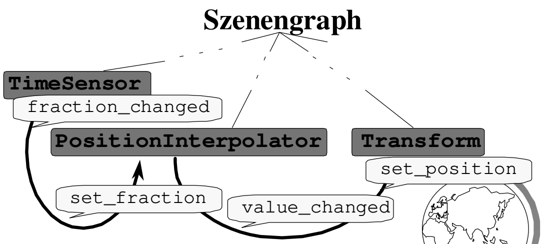
VRML

Dreidimensionale, virtuelle Welten im Internet

Interaktive Animationen

- Events, Routes, Sensoren und Interpolatoren
- `Script`-Knoten
  - JavaScript/VRMLScript
  - Java
- External Authoring Interface EAI

Events, Routes, Sensoren und Interpolatoren

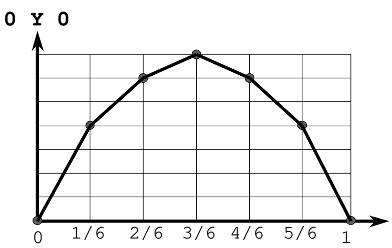


Beispiel: Hüpfender Ball

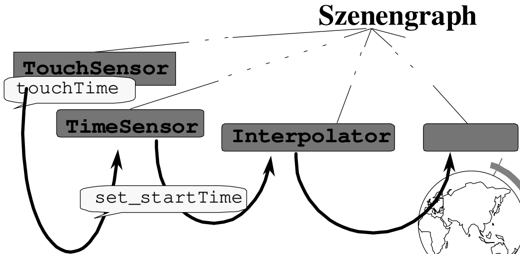
WRL Datei

```
DEF BALL
 Transform { children Shape { appearance USE APP
                              geometry Sphere { radius 1 } }
           }

DEF CLOCK TimeSensor { loop TRUE cycleInterval 3.5 }

DEF POSITION
 PositionInterpolator
  {  key [ 0, 0.166, 0.333, 0.499, 0.666, 0.833, 1 ]
     keyValue [ 0 0 0,   0 4 0,   0 6 0,   0 7 0,
                0 6 0,   0 4 0,   0 0 0  ]  }

ROUTE CLOCK.fraction_changed TO POSITION.set_fraction
ROUTE POSITION.value_changed TO BALL.set_translation
```

PositionInterpolator



Starten einer Animation per TouchSensor



Beispiel: Hüpfender Ball 2

WRL Datei

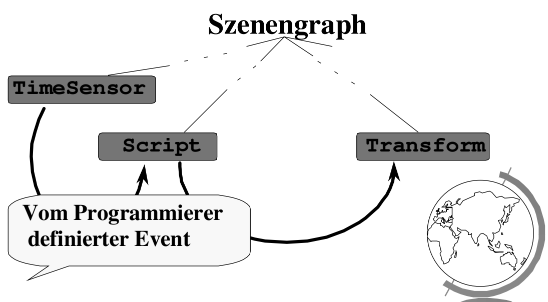
```
DEF BALL
 Transform { children [ DEF SCHALTER TouchSensor { }
                        Shape { appearance USE APP
                                geometry Sphere { radius 1 } }
                      ] }

DEF CLOCK TimeSensor { loop FALSE cycleInterval 3.5 }

DEF POSITION
 PositionInterpolator
  {  key [ 0, 0.166, 0.333, 0.499, 0.666, 0.833, 1 ]
     keyValue [ 0 0 0,   0 4 0,   0 6 0,   0 7 0,
                0 6 0,   0 4 0,   0 0 0  ]  }

ROUTE CLOCK.fraction_changed TO POSITION.set_fraction
ROUTE POSITION.value_changed TO BALL.set_translation
ROUTE SCHALTER.touchTime TO CLOCK.set_startTime
```

Sensoren und Interpolatoren

- CylinderSensor
- PlaneSensor
- ProximitySensor
- SphereSensor
- TimeSensor
- TouchSensor
- VisibilitySensor

- ColorInterpolator (SFColor)
- CoordinateInterpolator (MFVec3f)
- NormalInterpolator (MFVec3f)
- OrientationInterpolator (SFRotation)
- PositionInterpolator (SFVec3f)
- ScalarInterpolator (SFFloat)

Animationen mit `Script`-Knoten



`Script`

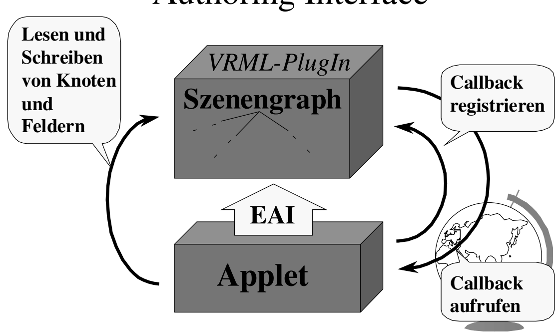
```
Script
  { <Feld und Eventdeklarationen>
    directOutput TRUE
    mustEvaluate FALSE
    url <Programm>
  }
```

Beispiel: Kreisender Ball

WRL Datei

```
DEF ANIMATOR
 Script {
  eventIn SFFloat tick
  eventOut SFVec3f newPosition
  field SFFloat rho 0.0
  url
   "javascript: function tick(val,ts)
    { newPosition[0]=Math.cos(rho)+2*Math.sin(rho/2);
      newPosition[1]=Math.sin(rho);
      newPosition[2]=Math.cos(rho*2)
                    +Math.sin(rho)*Math.cos(rho);
      rho = rho+0.3; }" }

ROUTE CLOCK.fraction_changed TO ANIMATOR.tick
ROUTE ANIMATOR.newPosition TO BALL.set_translation
```

Animationen per External Authoring Interface



Beispiel: `createVRMLFromString`

```
<HTML><HEAD><TITLE> Test EAI </TITLE></HEAD>
<BODY>
<SCRIPT>
 function createSrc()
  { var vstr=document.Formular.VRMLSrc.value.toString());
    document.APP.addNode(vstr);
    return; }
</SCRIPT>
<EMBED SRC="root.wrl" BORDER=0 HEIGHT=250 WIDTH=600>
<APPLET NAME="APP" CODE="EAIApplet.class" MAYSCRIPT
        WIDTH=10 HEIGHT=10> </APPLET>
<FORM NAME="Formular">
Enter a VRML 2.0 Definition of a Children Node:<BR>
<TEXTAREA NAME="VRMLSrc" ROWS=10 COLS=80> </TEXTAREA><BR>
<INPUT TYPE=BUTTON VALUE="Create" ONCLICK="createSrc();” >
</FORM>
</BODY>
</HTML>
```

Beispiel: `createVRMLFromString`

Netscape Hypertext Document

```
import java.applet.*;
import vrml.external.field.*;
import vrml.external.Node;
import vrml.external.Browser;

public class EAIApplet extends Applet
{ Browser browser;
  Node root;

  public void init()
   { browser = Browser.getBrowser(this);
     root = browser.getNode("ROOT"); }

  public void addNode(String str)
   { EventInMFNode addChildren =
            (EventInMFNode) root.getEventIn("addChildren");
     Node nodes[] = browser.createVrmlFromString(str);
     addChildren.setValue(nodes); }
 }
```