X3D
- Einführung -

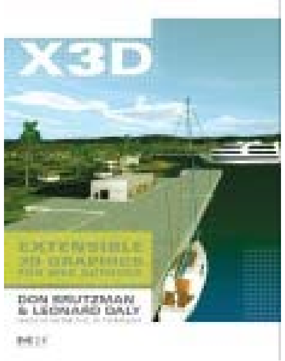
Überblick

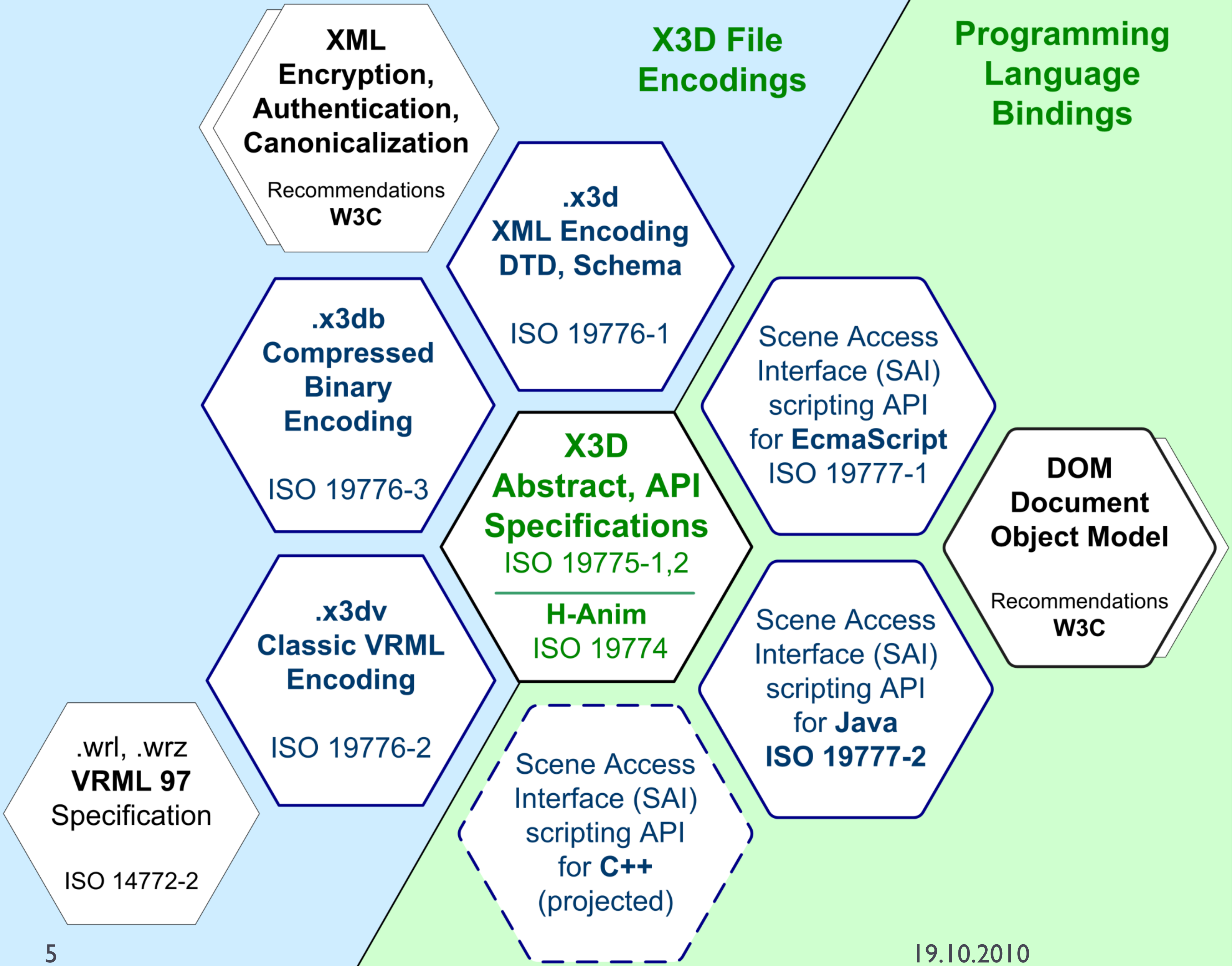
- Der X3D Standard
  - Überblick der Technologien
  - Profile
  - X3D Grundgerüst
- Objekte
  - Formen
  - Materialien
- Räumliche und logische Organisation
  - Gruppen
  - Koordinatensystem und Transformationen
  - Wiederverwendung mit DEF/USE
- Animation und Interaktion
  - Sensoren
  - Routen
  - Animation mit Interpolatoren und Timern
- Scripting
  - Wiederverwendung mit Prototypen
  - JavaScript

Der X3D Standard

Literatur

- **X3D: Extensible 3D Graphics for Web** Authors by Don Brutzman and Leonard Daly, Morgan Kaufmann Publishers, April 2007, 468 pages. http://x3dGraphics.com
- **X3D: Programmierung interaktiver 3D-Anwendungen für das Internet** von Jörg H. Kloss, Addison-Wesley Verlag, 2010
- stehen im Labor auf M4 und in der Uni Bibliothek


X3D
EXTENSIBLE 3D GRAPHICS FOR WEB AUTHORS
DON BRUTZMAN & LEONARD DALY



X3D File Encodings

Programming Language Bindings

XML Encryption, Authentication, Canonicalization
Recommendations
W3C

.x3d
XML Encoding
DTD, Schema
ISO 19776-1

.x3db
Compressed Binary Encoding
ISO 19776-3

X3D Abstract, API Specifications
ISO 19775-1,2

H-Anim
ISO 19774

.x3dv
Classic VRML Encoding
ISO 19776-2

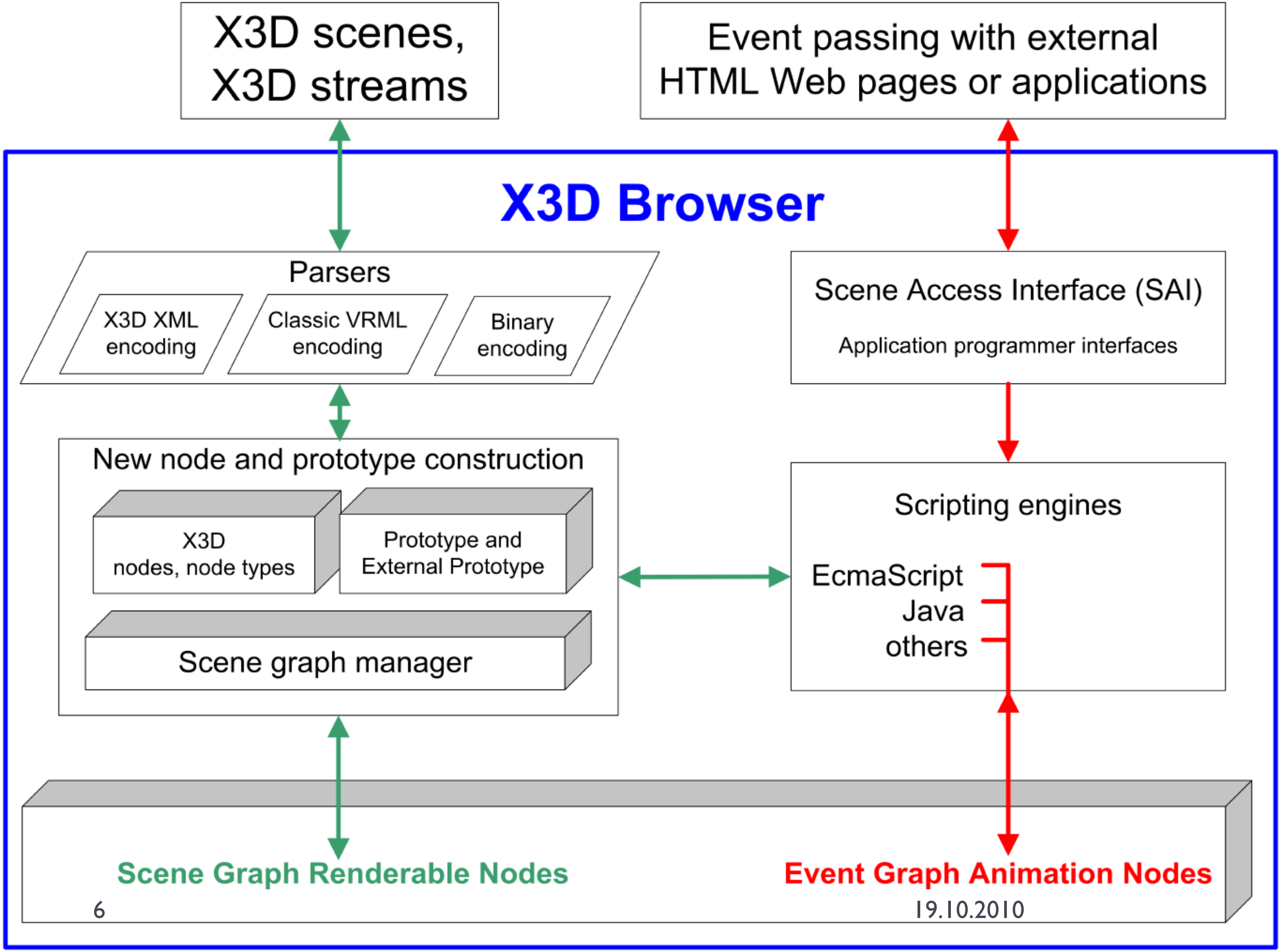
.wrl, .wrz
VRML 97
Specification
ISO 14772-2

Scene Access Interface (SAI) scripting API for **EcmaScript**
ISO 19777-1

DOM Document Object Model
Recommendations
W3C

Scene Access Interface (SAI) scripting API for **Java**
ISO 19777-2

Scene Access Interface (SAI) scripting API for **C++**
(projected)



X3D scenes, X3D streams

Event passing with external HTML Web pages or applications

X3D Browser

Parsers

X3D XML encoding

Classic VRML encoding

Binary encoding

Scene Access Interface (SAI)

Application programmer interfaces

New node and prototype construction

X3D nodes, node types

Prototype and External Prototype

Scene graph manager

Scripting engines

EcmaScript

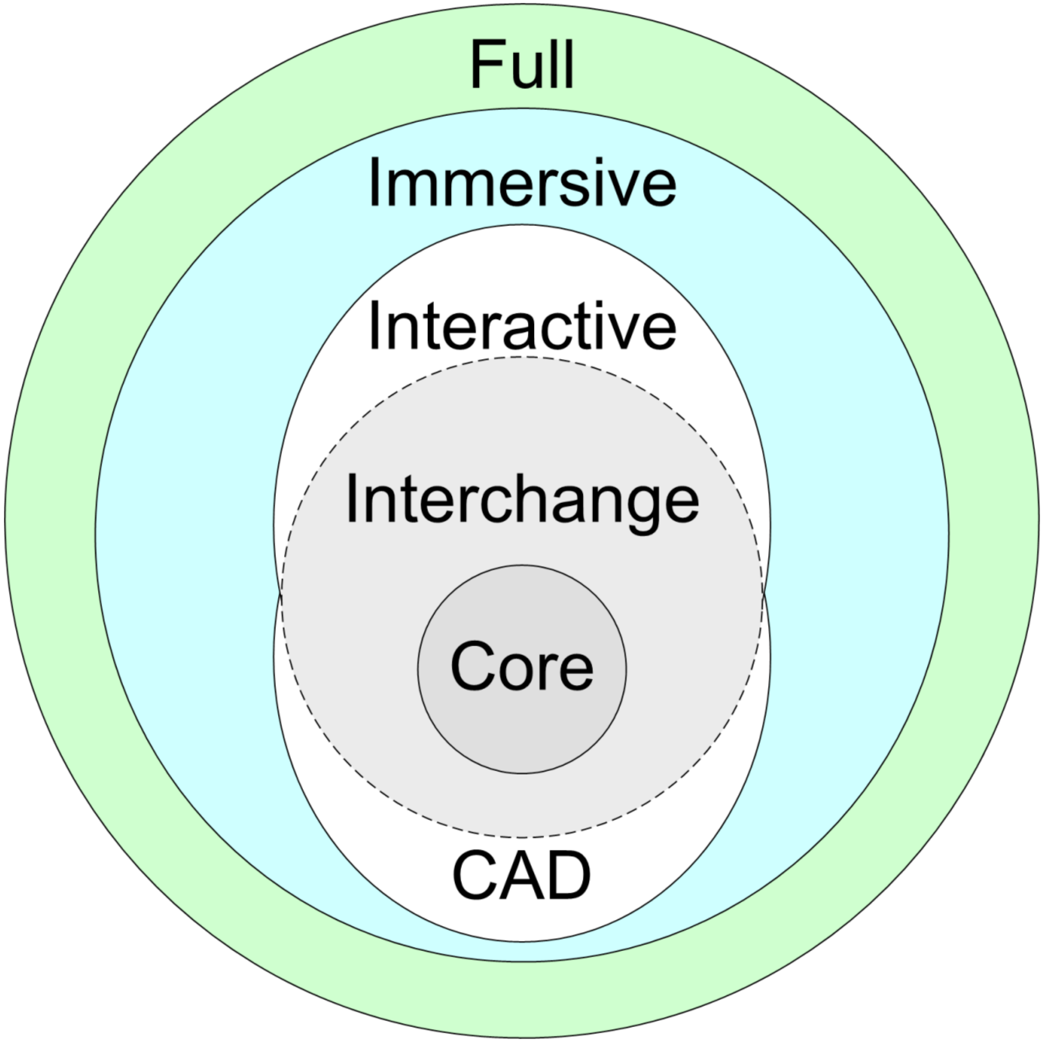
Java

others

Scene Graph Renderable Nodes

Event Graph Animation Nodes

X3D Profile



X3D Grundgerüst

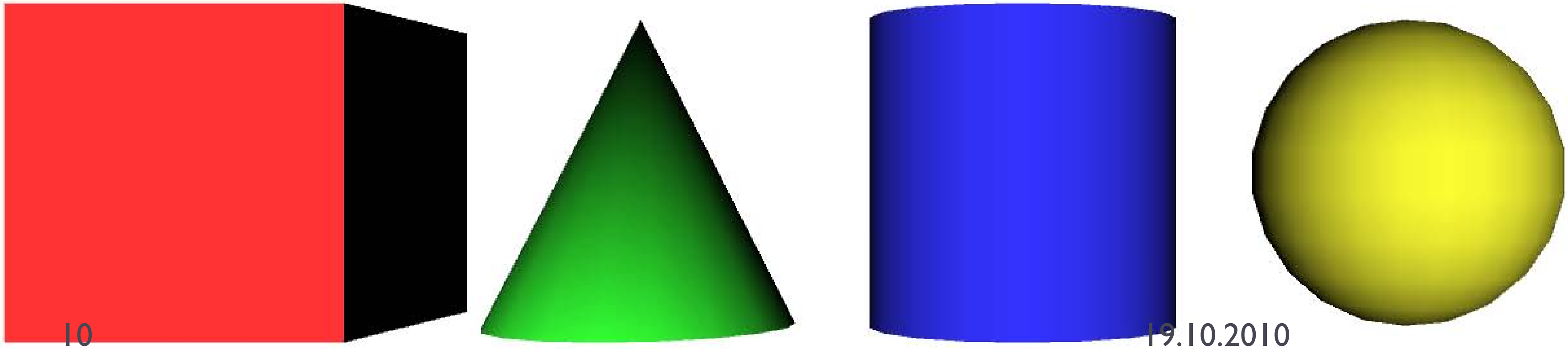
- <?xml version="1.0" encoding="UTF-8"?>
- <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
- <X3D xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' profile='Full' version='3.0' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.0.xsd'>
  - <head>
    - <meta name='title' content='Example.x3d'/>
  - </head>
  - <Scene>
  - </Scene>
- </X3D>

Objekte in X3D

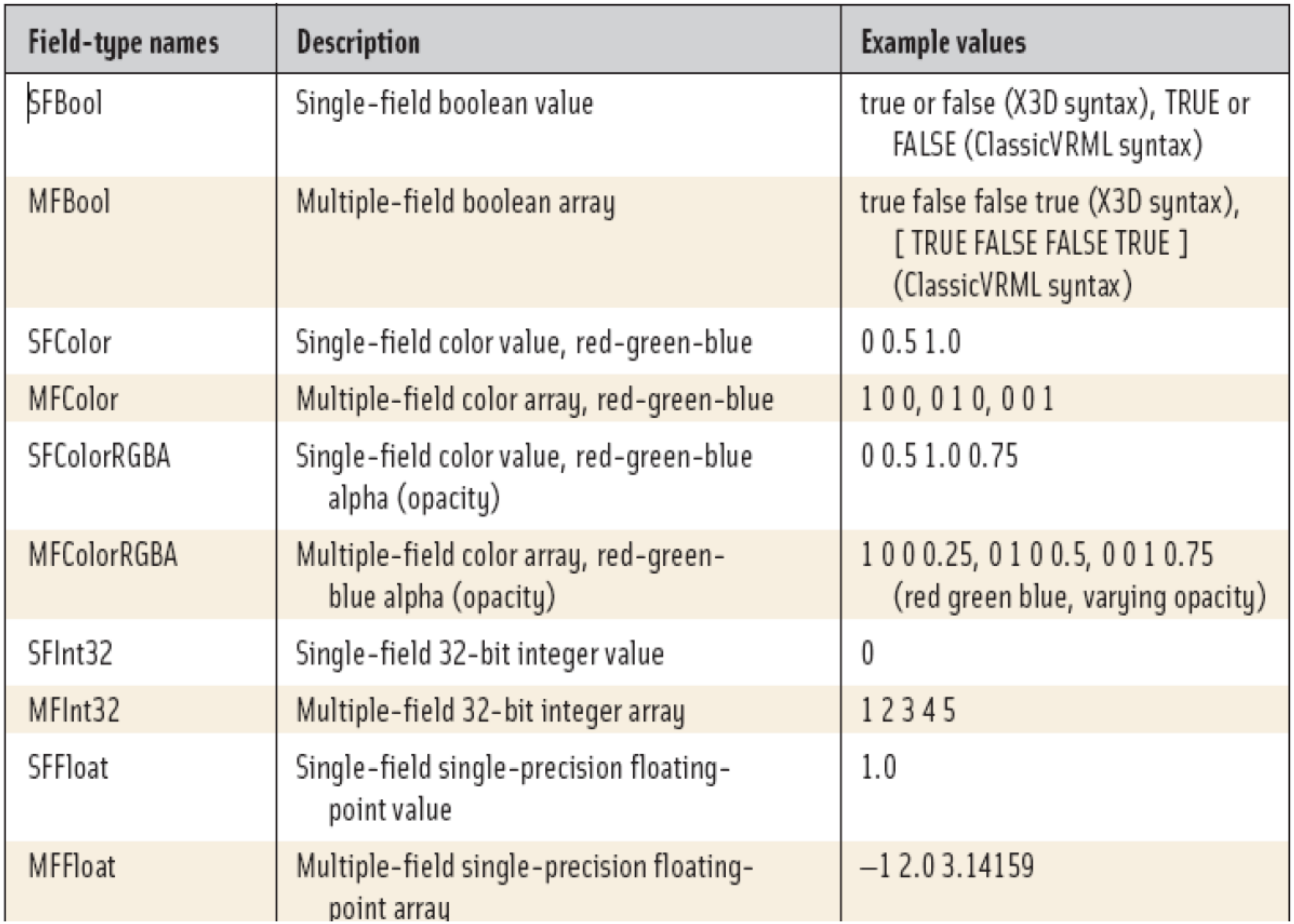
Formen

- <Shape>
  - <Box size='1 2 3'/>
  - <Appearance/>
- </Shape>

- <Shape>
  - <Sphere radius=‘1'/>
  - <Appearance/>
- </Shape>



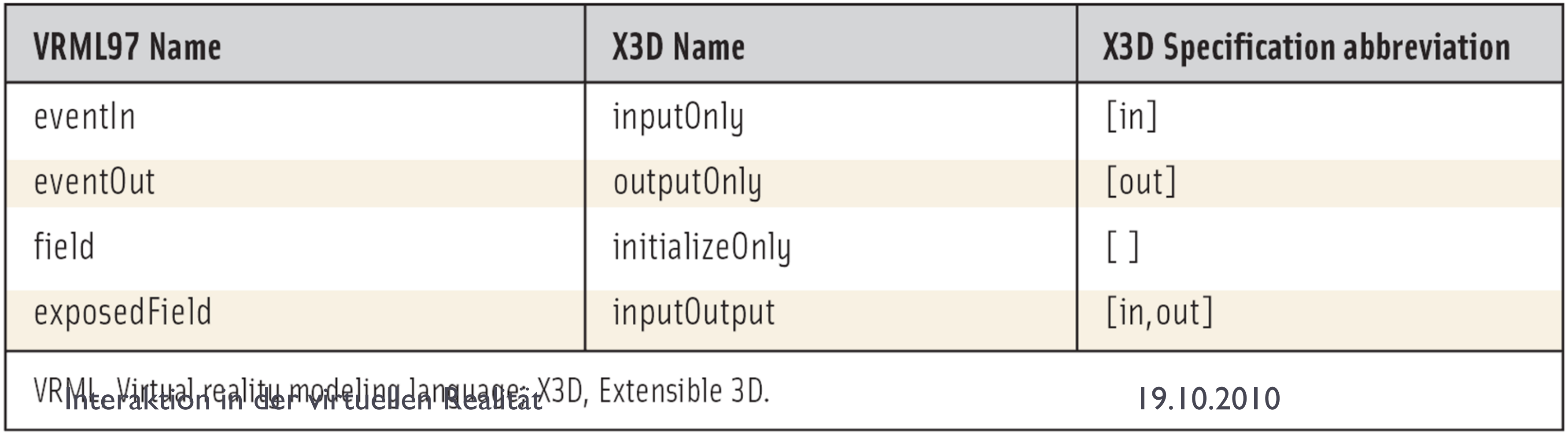
Felder



Field-type names	Description	Example values
SFBool	Single-field boolean value	true or false (X3D syntax), TRUE or FALSE (ClassicVRML syntax)
MFBool	Multiple-field boolean array	true false false true (X3D syntax), [TRUE FALSE FALSE TRUE] (ClassicVRML syntax)
SFColor	Single-field color value, red-green-blue	0 0.5 1.0
MFColor	Multiple-field color array, red-green-blue	1 0 0, 0 1 0, 0 0 1
SFColorRGBA	Single-field color value, red-green-blue alpha (opacity)	0 0.5 1.0 0.75
MFColorRGBA	Multiple-field color array, red-green-blue alpha (opacity)	1 0 0 0.25, 0 1 0 0.5, 0 0 1 0.75 (red green blue, varying opacity)
SFInt32	Single-field 32-bit integer value	0
MFInt32	Multiple-field 32-bit integer array	1 2 3 4 5
SFFloat	Single-field single-precision floating-point value	1.0
MFFloat	Multiple-field single-precision floating-point array	−1 2.0 3.14159

Feldzugriff

- accessType: input, output, initialize
- accessType determines if field is data sender,
- receiver, or holder
  - inputOnly: can only receive events
  - outputOnly: can only send events
  - initializeOnly: cannot send or receive
  - inputOutput: can send, receive and be initialized



VRML97 Name	X3D Name	X3D Specification abbreviation
eventIn	inputOnly	[in]
eventOut	outputOnly	[out]
field	initializeOnly	[ ]
exposedField	inputOutput	[in,out]

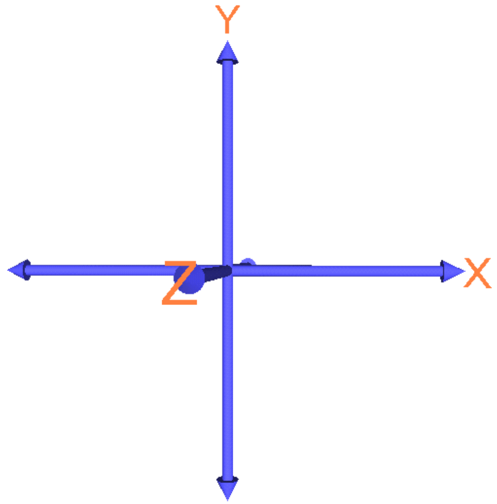
VRML, Virtual reality modeling language; X3D, Extensible 3D.

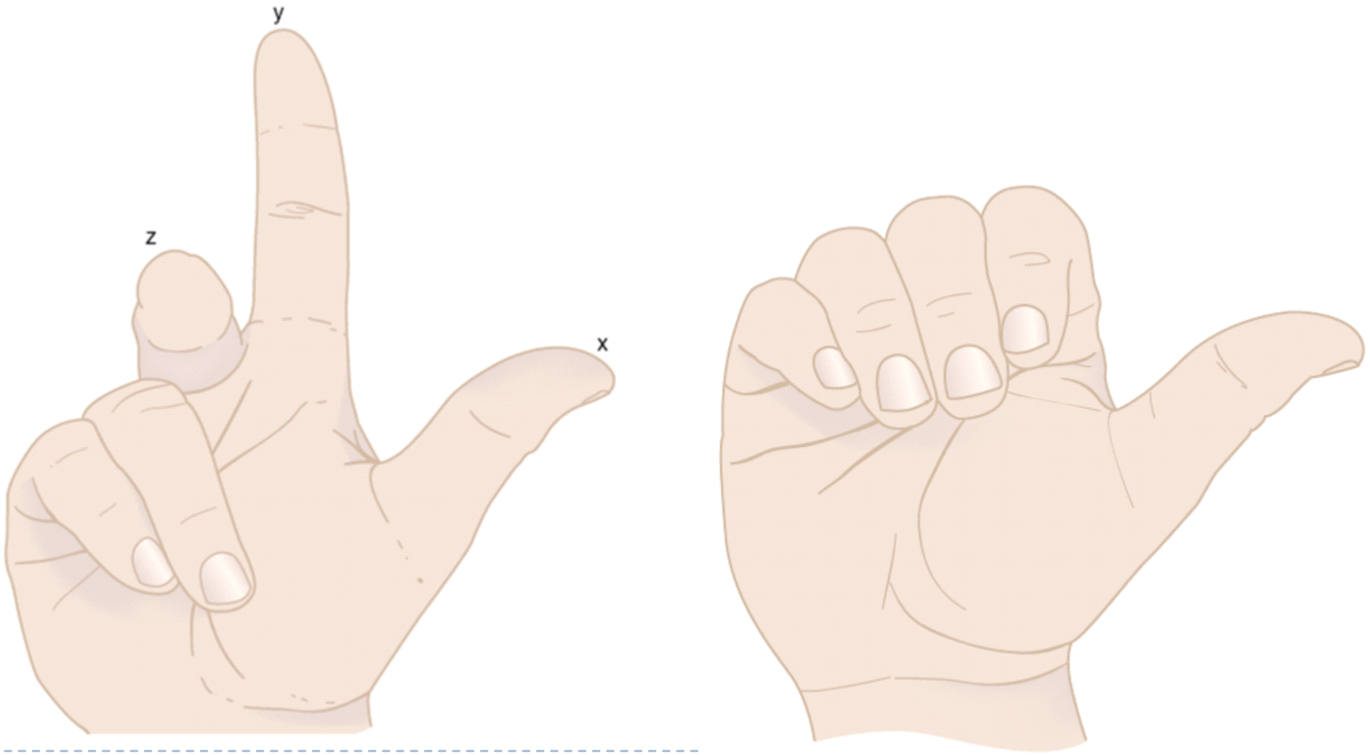
Praxis

- Shapes
- Materials

Räumliche und logische Organisation

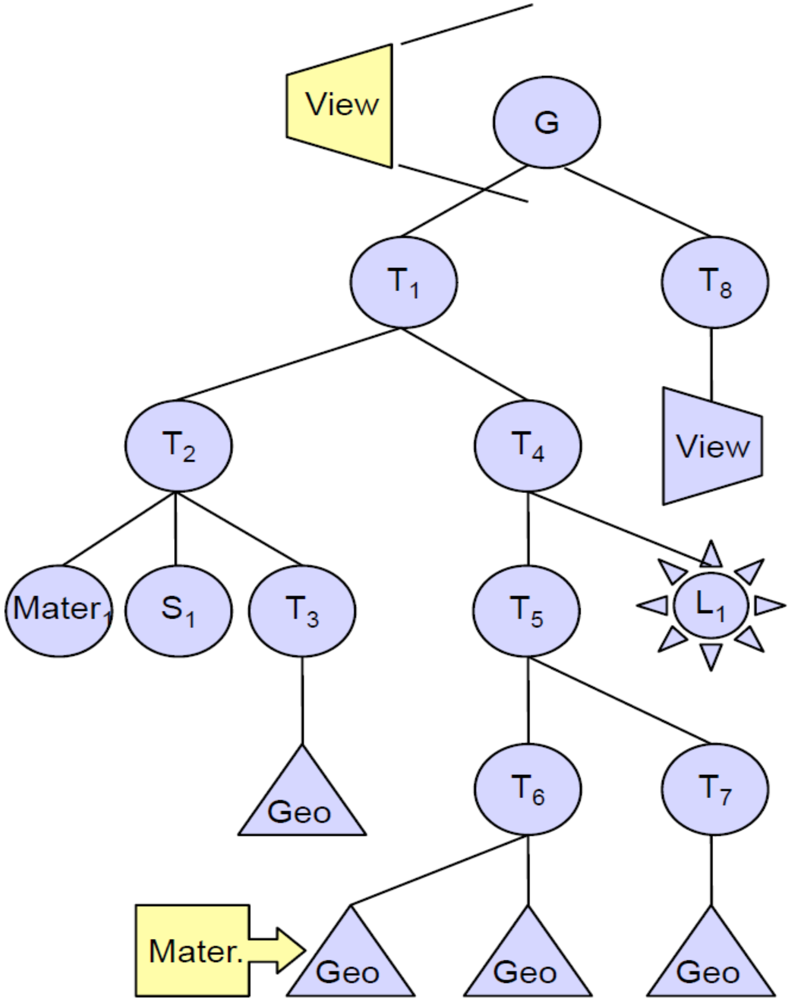
Koordinatensystem



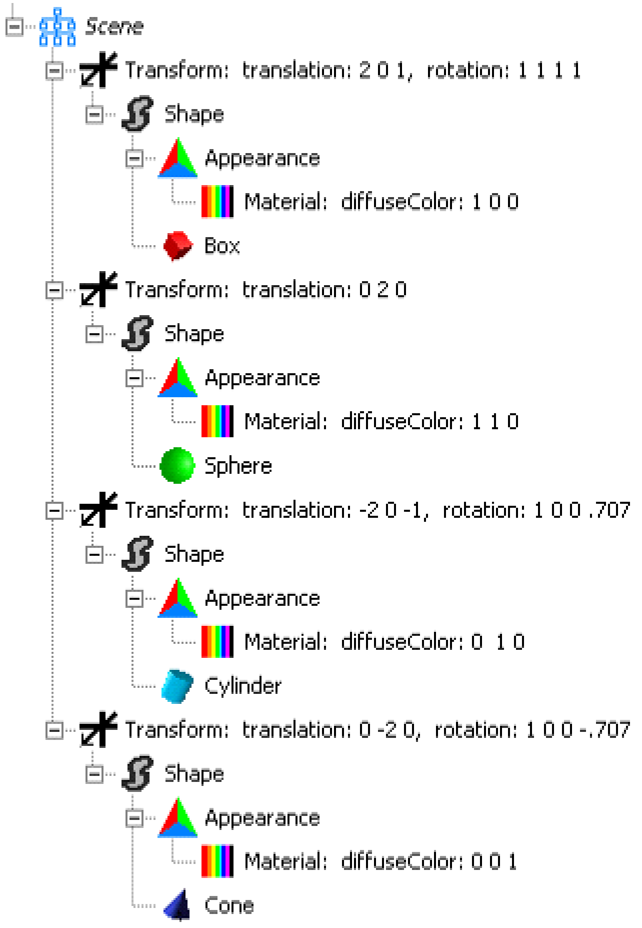


Transformation

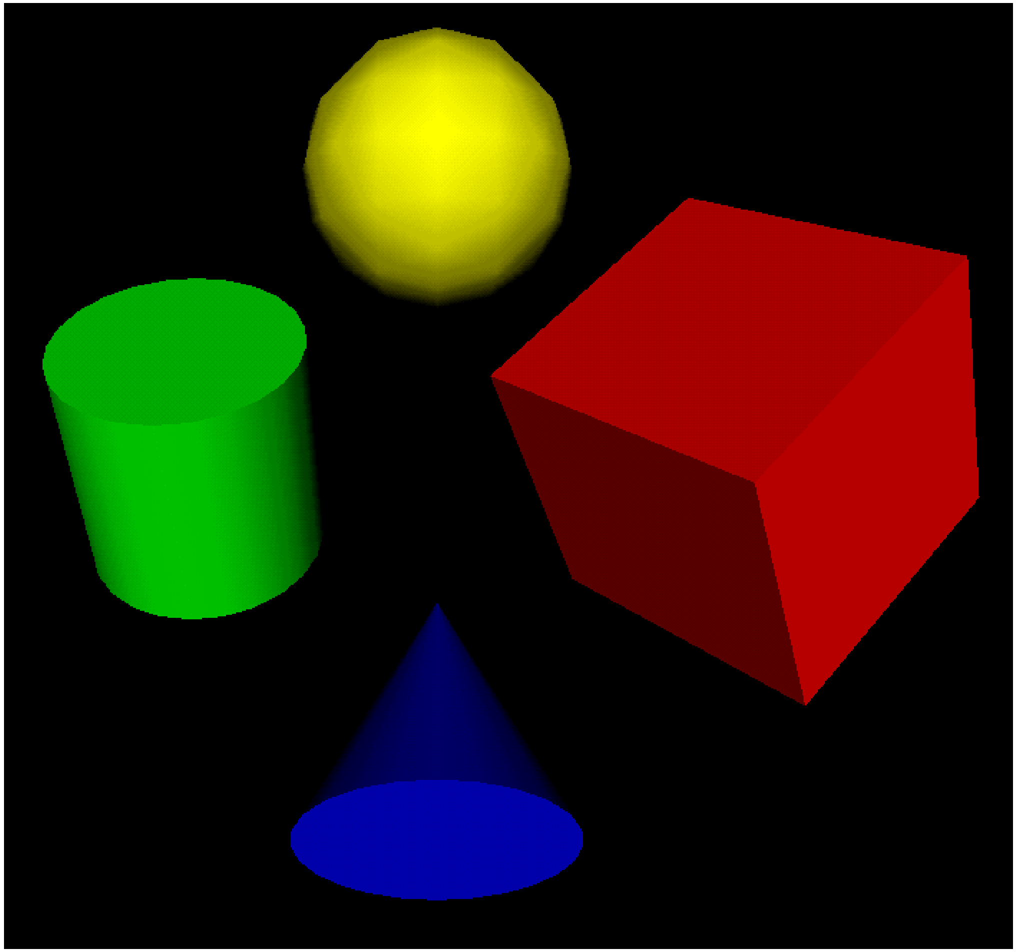
- <Transform
  - translation=“0 0 0“
  - rotation=“1 0 0 3.141“
  - scale=“1 1 1“
  - >
    - …
  - </Transform>



Gruppen und Transformationen



```
Scene
  Transform:  translation: 2 0 1,  rotation: 1 1 1 1
    Shape
      Appearance
        Material:  diffuseColor: 1 0 0
      Box
  Transform:  translation: 0 2 0
    Shape
      Appearance
        Material:  diffuseColor: 1 1 0
      Sphere
  Transform:  translation: -2 0 -1,  rotation: 1 0 0 .707
    Shape
      Appearance
        Material:  diffuseColor: 0  1 0
      Cylinder
  Transform:  translation: 0 -2 0,  rotation: 1 0 0 -.707
    Shape
      Appearance
        Material:  diffuseColor: 0 0 1
      Cone
```



Wiederverwendung mit DEF/USE

- <Transform translation=“0 0 0“>
  - <Shape DEF=“BOX“>
    - <Box size=“0 0 0“/>
    - <Appearance/>
  - </Shape>
- </Transform>
- <Transform translation=“1 0 0“>
  - <Shape USE=“BOX“/>
- </Transform>
- <Transform translation=“2 0 0“>
  - <Shape USE=“BOX“/>
- </Transform>

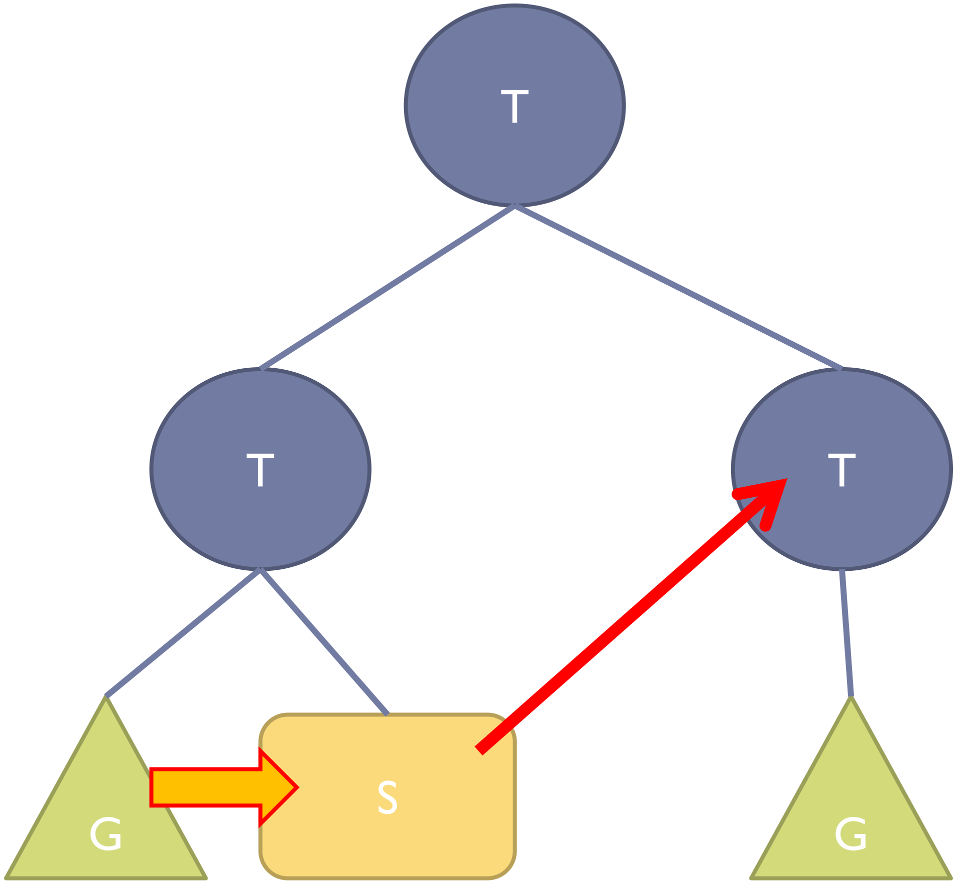
Praxis

- Transformationen
- Szene im InstantReality zeigen
  - Web-Interface

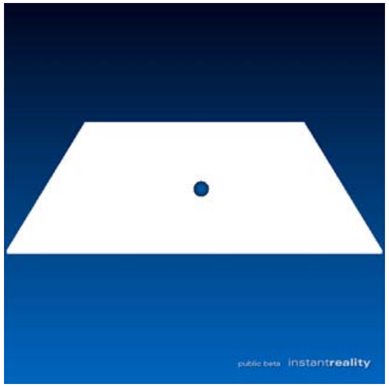
Animation und Interaktion

Sensoren

- TouchSensor
- PlaneSensor
- CylinderSensor
- KeySensor, StringSensor



Beispiel



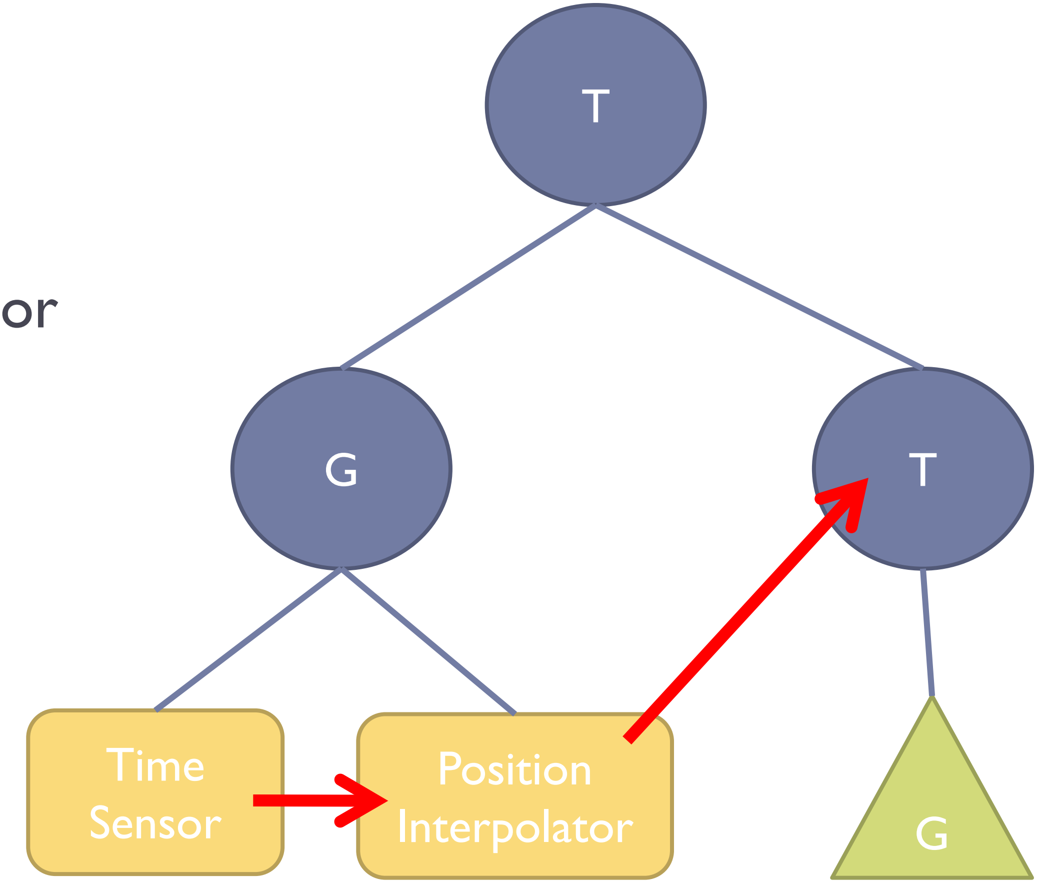
- <Transform DEF='trans_sphere'>
  - <Shape>
    - <Sphere radius='0.25'/>
  - </Shape>
- </Transform>
- <Transform DEF='trans_plane' translation='0 -0.25 0'>
  - <TouchSensor DEF='ts' />
  - <Shape>
    - <Box size='10 0.1 10'/>
  - </Shape>
- </Transform>
- <ROUTE fromNode='ts' fromField='hitPoint_changed' toNode='trans_sphere' toField='set_translation'/>

Praxis

- TouchSensor Beispiel

Animationen

- TimeSensor
- Interpolatoren
  - PositionInterpolator
  - OrientationInterpolator
  - ColorInterpolator
  - …



Praxis

- Animation

Scripting

Prototypes – Wie definiere ich eigene Knoten?

- Einbinden mit
  - <ExternProtoDeclare name='MyNode' url='MyNode.x3d' />
- Verwenden mit
  - <MyNode/>
- Spezifikation in der Datei MyNode.x3d:
  - <ProtoDeclare name='MyNode'>
    - <ProtoInterface>
      - <field name='translation' accessType='inputOutput' type='SFVec3f' value='0 0 0'/>
    - </ProtoInterface>
    - <ProtoBody>
      - <Transform>
        - <IS>
          - <connect protoField='translation' nodeField=‘translation' />
        - </IS>
      - </Transform>
    - </ProtoBody>
  - </ProtoDeclare>

Scripting

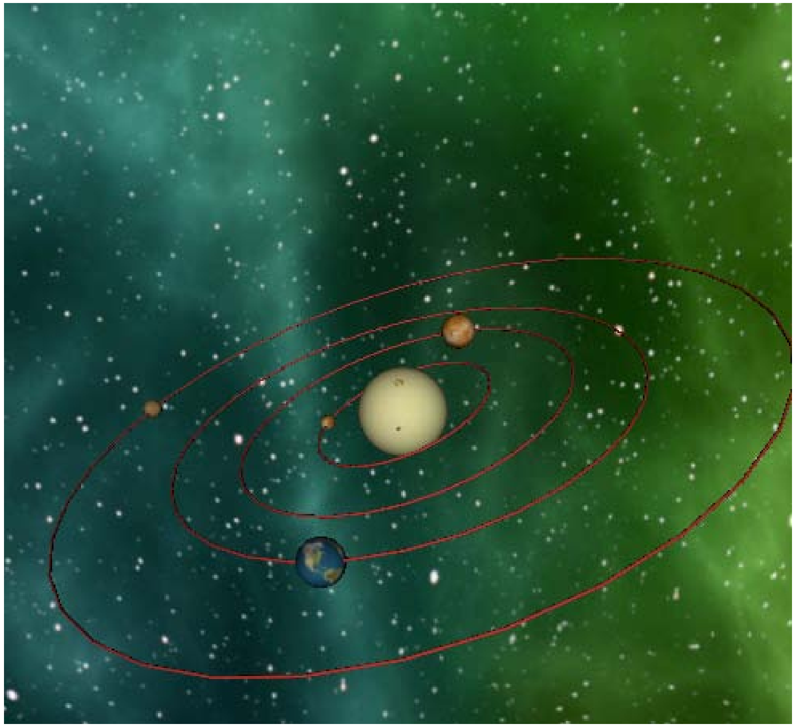
```
<Script DEF='my_script' mustEvaluate='true'>
    <field name='my_time' accessType='inputOnly' type='SFTime' />
    <field name='my_target' accessType='inputOnly' type='SFVec3f'/>
    <field name='my_translation' accessType='outputOnly' type='SFVec3f' />
    <![CDATA[javascript:
        var target = new SFVec3f(0,0,0);
        function my_target( value ) { target = value; };
        function my_time(value)
        {
            diff = target.subtract( my_translation );
            diff = diff.multiply(0.1);
            my_translation = my_translation.add(diff);
        }
     ]]>
</Script>
```

Praxis

- Scripting: PositionChaser

Aufgabe

Sonnensystem



- Sonne + 4 Planeten
- Planeten bewegen sich um die Sonne!
- Planeten bewegen sich um die eigene Achse!
- Texturen gibt es hier:
  - http://planetpixelemporium.com/planets.html
- Weiteres im Wiki unter
  - https://hiro.techfak.uni-bielefeld.de/twiki/bin/view/Main/InstantRealityEinfuehrung