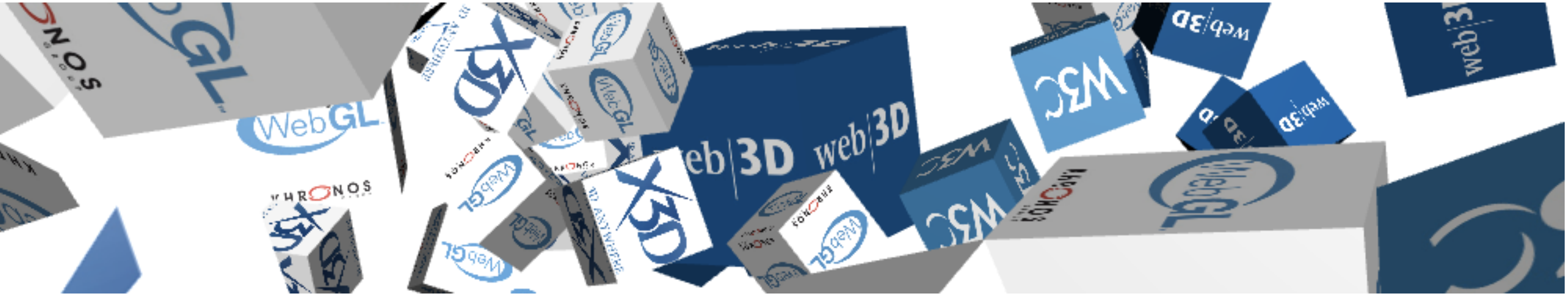
X3DOM

Getting declarative (X)3D into HTML



W3C TPAC 2010

Johannes Behr & Yvonne Jung

Fraunhofer IGD
Web3D Consortium
johannes.behr@igd.fraunhofer.de

Fraunhofer IGD

TECHNISCHE
UNIVERSITÄT
DARMSTADT

TU Graz

Qualitätsmanagement
zertifiziert nach
DIN ISO 9001:2000

IQNet CERTIFIED MANAGEMENT SYSTEM

Motivation

Imperative and declarative 3D

"Future of web3D" Panel 2008 (Web3D symposium)

Vladimir Vukicevic presented **Canvas3D/WebGL**

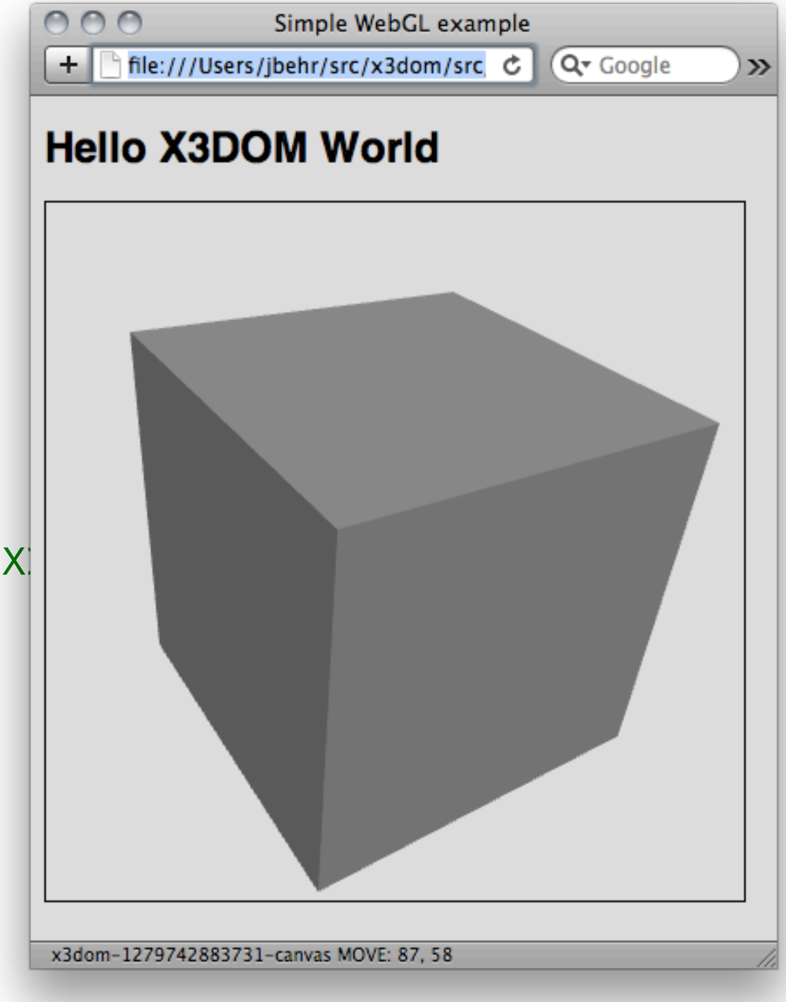
- **Pro:**
  - User-agent service => **Plugin-free** approach
  - OpenGL(-ES) proved itself as **excellent Graphics API**
- **Con:**
  - **Efficiency**: Spend too many (battery) resources to manage your scene?
  - **Concepts**: HTML Developer has to deal with GLSL and 4x4 matrices.
  - **Metadata**: Index and search "content" on WebGL-Apps?

What approach could be **more efficient**, **independent of the visualisation method** and **better integrated** with todays W3C standards ?

Idea: Declarative (X)3D in HTML

Embed a live scenegraph in the DOM

```
<!DOCTYPE html >
<html >
    <body>
        <h1>Hello X3DOM World</h1>
            <x3d xmlns='…' profile='HTML' >
                <scene>
                    <shape>
                        <box></box
                    </shape>
                </scene>
            </x3d>
    </body>
</html>
```



Hello X3DOM World

x3dom-1279742883731-canvas MOVE: 87, 58

History of X3D in HTML

HTML5 Specification: http://www.w3.org/TR/html5/no.html#declarative-3d-scenes

*13.2 **Declarative** 3D scenes*
*Embedding 3D imagery into XHTML documents is the domain of X3D, or technologies based on **X3D** that are namespace aware.*

2007: First experiment by Philip Tayler, W3C: http://philip.html5.org

X3D in DOM, no DOM manipulation, Canvas3D for rendering

Used Canvas3D from Mozilla for rendering

2009: x3dom by Fraunhofer IGD, Based on code for Taylor: www.x3dom.org

Full DOM integration. **Native**, **X3D-Plugin** or **WebGL** for rendering

W3C TPAC 2009 in **Santa Clara**

http://web3d.org/x3d/presentations/X3D+HTML5.W3cTpac-20091106.pdf

2010: HTML/X3D IG incorporated with Web3D consortium

Develops HTML/X3D integration model based on x3dom

X3D Standard

State of the current integration model

Pro:

- **Non-profit Web3D Consortium** maintains and extends X3D
- **ISO standard** since 2004, (VRML pre
- Specification includes abstract **IDL** and **JavaScript-binding**
- **XML-Encoding** is **namespace** aware
- **Multiple implementations** are available (**open/closed source)**
- **Third-generation 3D** graphics language that extends predecessor
- **10 year W3C member** and contributor

Con:

- **Plugin** integration model
- Full Standard is **to complex**, Some concepts duplicate W3C techniques
- Simple Shape/Material assign **mechanism**. **No cascading** Material system

HTML/DOM Profile

Reduce X3D to 3D visualization component for HTML5

General Goal:

⇒ Utilizes **HTML/JS/CSS** for **scripting** and **interaction**

⇒ **Reduced complexity** and **implementation effort**

⇒ Reduces X3DOM to visualization component for 3D like SVG for 2D

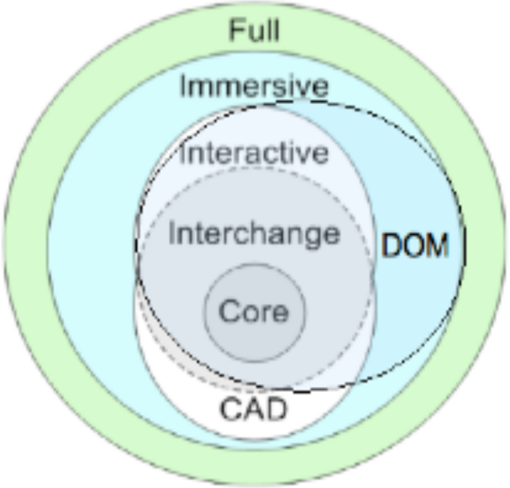
⇒2 Profiles: HTML and HTML-Tiny/WebSG

"**HTML**"-Profile (Extends "Interchange" Profile): **~ 80 nodes**

Full **runtime** with anim., navigation and **asynchronous data** fetching

No X3D-Script, Proto, High-Level Sensor-nodes

Declarative and explicit shader material



"**HTML-Tiny**" or "**WebSG**"-Profile: **~ 15 nodes**

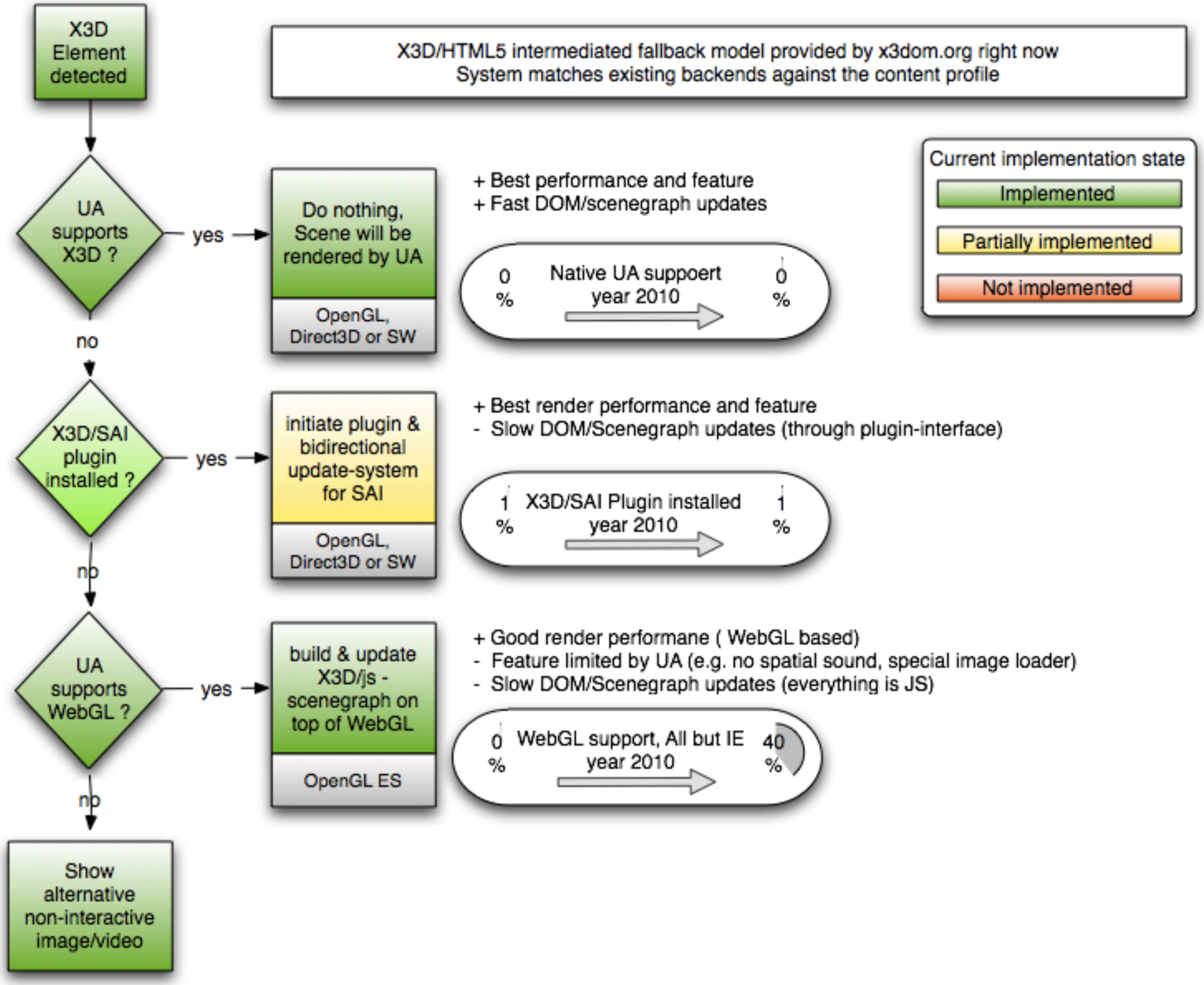
No Runtime at all: Just **redraw on changes**

Generic **<mesh>** node without vertex semantics

Only explicit shader material

Implementation

JS-Layer: Supports **native**, X3D/SAI-**Plugin** or **WebGL**



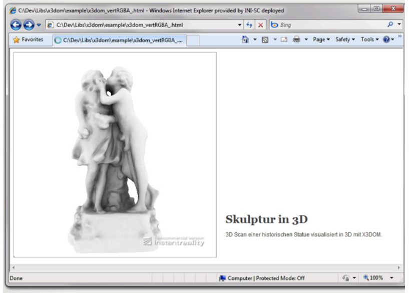
X3D/HTML5 intermediated fallback model provided by x3dom.org right now
System matches existing backends against the content profile

X3D Element detected
UA supports X3D ?
yes
Do nothing, Scene will be rendered by UA
OpenGL, Direct3D or SW
no
X3D/SAI plugin installed ?
yes
initiate plugin & bidirectional update-system for SAI
OpenGL, Direct3D or SW
no
UA supports WebGL ?
yes
build & update X3D/js - scenegraph on top of WebGL
OpenGL ES
no
Show alternative non-interactive image/video
0 %
Native UA suppoert year 2010
0 %
1 %
X3D/SAI Plugin installed year 2010
1 %
0 %
WebGL support, All but IE year 2010
40 %
Current implementation state
Implemented
Partially implemented
Not implemented

+ Best performance and feature
+ Fast DOM/scenegraph updates

+ Best render performance and feature
- Slow DOM/Scenegraph updates (through plugin-interface)

+ Good render performane (WebGL based)
- Feature limited by UA (e.g. no spatial sound, special image loader)
- Slow DOM/Scenegraph updates (everything is JS)



C:\Dev\Libs\x3dom\example\x3dom_vertRGBA_.html - Windows Internet Explorer provided by INI-SC deployed

C:\Dev\Libs\x3dom\example\x3dom_vertRGBA_.html

Skulptur in 3D

3D Scan einer historischen Statue visualisiert in 3D mit X3DOM.

Done

Computer | Protected Mode: Off

x3dom.org/x3dom/release/x3dom.js

JavaScript-based layer, supports native implementation

```
<!DOCTYPE html >
<html >
    <head>
          <link rel="stylesheet" type="text/css" href="x3dom.css” >
          <script type="text/javascript" src="x3dom.js"></script>
    </head>
    <body>
       <h1>HTML5 Hello World</h1>
          <x3d xmlns=”…” profile=‘…’ backend=‘…’ >
                    <scene>
                              …
                    </scene>
          </x3d>
    </body>
</html>
```

DOM Manipulation

Node appending and removal

HTML/X3D code:

```
…
<group id=‘root’></group>
.   …
```

HTML-Script to add nodes:

```
trans = document.createElement('Transform');
trans.setAttribute(‘translation’, ‘1 2 3’ );
document.getElementById(‘root’).appendChild(trans);
```

HTML-Script to remove nodes:

```
document.getElementById(‘root’).removeChild(trans);
```

DOM Manipulation

Field updates with setAttribute() or SAI-Field interfaces

HTML/X3D code:

```
    …
    <material id=‘mat’></material>
    …
    <coordinate id=‘coord’ point=‘5.6 3 87, 8.8 8.4 3.2, …’ ></coordinate>
.   …
```

Generic HTML-Script with **setAttribute()**: also useful for libs like **jQuery**

```
    document.getElementByid(‘mat’).setAttribute(‘diffuseColor’,’red’);
```

HTML-Script using SAI-Field interface: X3D JS-binding for more efficiency

```
    var saiField = document.getElementById(‘coord).getField(‘point’);
    saiField[4711].x = 0.815;
```

HTML Events

User Interaction through DOM Events

Supports interaction with standard HTML-Events. Supports **ancient** (Netscape2) and **addEventListener()** interfaces.

```
<x3d xmlns=”…">
<Scene>
  <Shape>
    <Appearance>
      <Material id=‘mat’ diffuseColor=‘red’ />
     </Appearance>
     <Box onclick=“document.getElementById(‘mat’).diffuseColor=‘green’” />
   </Shape>
</Scene>
</x3d>
```

HTML Events

3DPickEvent extends DOM Leven 3 MouseEvent

```
interface 3DPickEvent : MouseEvent {
  readonly attribute float  worldX;        // 3d world coordinates
  readonly attribute float  worldY;
  readonly attribute float  worldZ;
  readonly attribute float  localX;        // element local 3d coordinates
  readonly attribute float  localY;
  readonly attribute float  localZ;
  readonly attribute float  normalX;       // surface normal
  readonly attribute float  normalY;
  readonly attribute float  normalZ;
  readonly attribute float  colorRed;      // surface color (with alpha)
  readonly attribute float  colorGreen;
  readonly attribute float  colorBlue;
  readonly attribute float  colorAlpha
  readonly attribute float  texCoordS;     // surface texture coordinates
  readonly attribute float  texCoordT;
  readonly attribute float  texCoord;
  object                                   getMeshPickData (in DOMString vertexProperty);
};
```

HTML Events

Additional 3D Events extend Dom Level 3 UIEvent

Inspired by the X3D-Sensors Environment Sensors

```
// Visibitlity of sub-trees
interface 3DVisibilityEvent : UIEvent {
  ...;
}

// Object-Camera Transformation sensor
interface 3DProximityEvent : UIEvent {
  …;
}

// Object-Transformation
Interfac 3DTransformEvent : Event {
  …;
}
```

DOM Manipulation
CSS 3D Transforms

CSS 3D Transforms Module Level 3; W3C Draft

Utilized to transform and update **<transform>** nodes

```
<style type="text/css">
    #trans {
           -webkit-animation: spin 8s infinite linear;
    }
    @-webkit-keyframes spin {
           from { -webkit-transform: rotateY(0); }
           to   { -webkit-transform: rotateY(-360deg); }
    }
</style>
    …
<transform id="trans" >
    <transform style="-webkit-transform: rotateY(45deg);">
    …
```

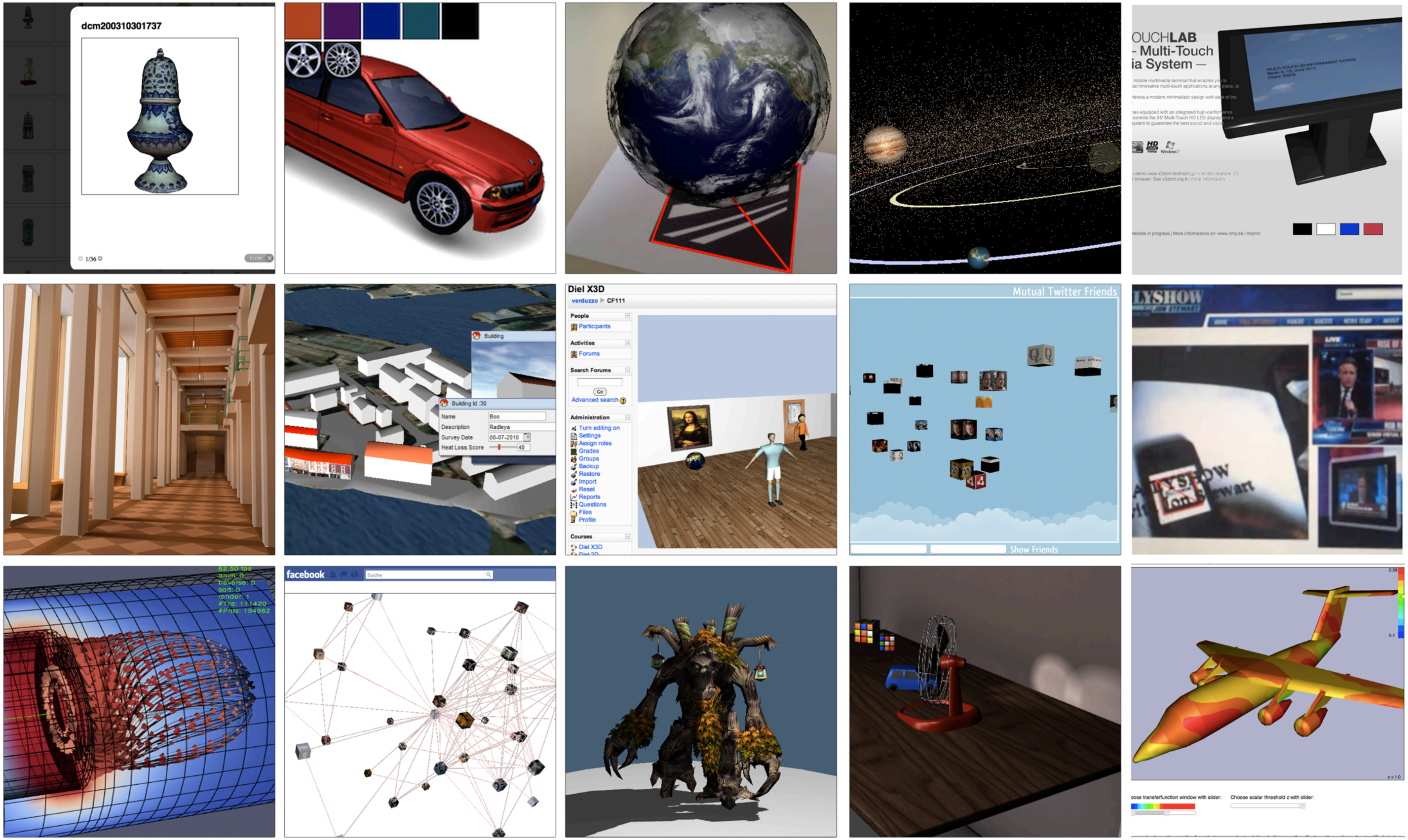
HTML Events

Events from the X3D subsystem

```
<x3d xmlns="http://www.web3d.org/specifications/x3d-3.0.xsd">
    <Viewpoint id=‘cam01’ />
    <Viewpoint id=‘cam02’ />
    <Scene>
        <Shape><Box size="4 4 4" /></Shape>
   </Scene>
</x3d>
<script type="text/javascript">
     var cam = document.getElementsById(‘cam01’);
        cam.addEventListener(”active",
        function() { alert(”Viewpoint 01 is active!"); },  false);
</script>
```

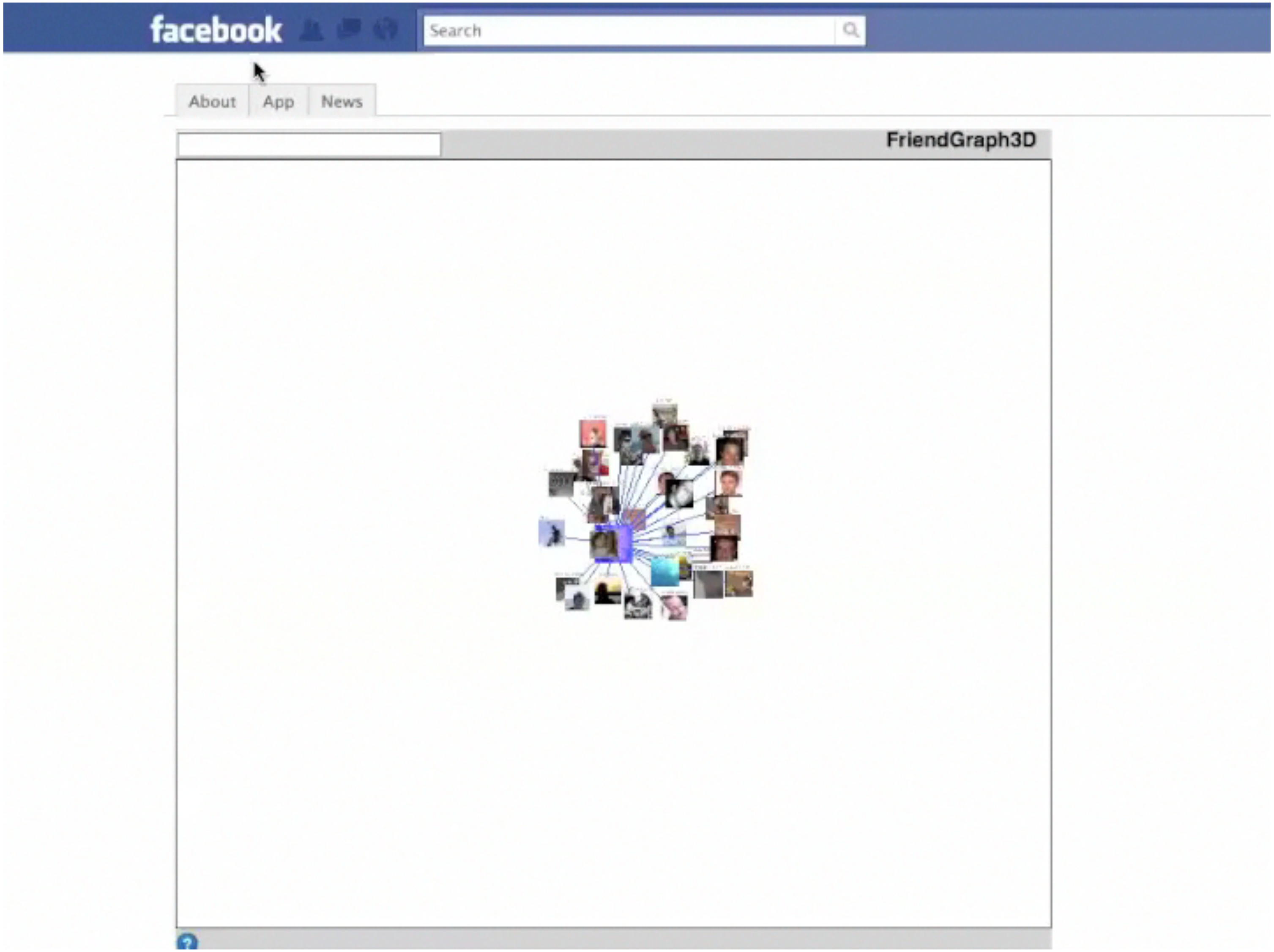
Show-cases and Third-party application

Developed during the last 12 month, (TPAC 2009 – 2010)



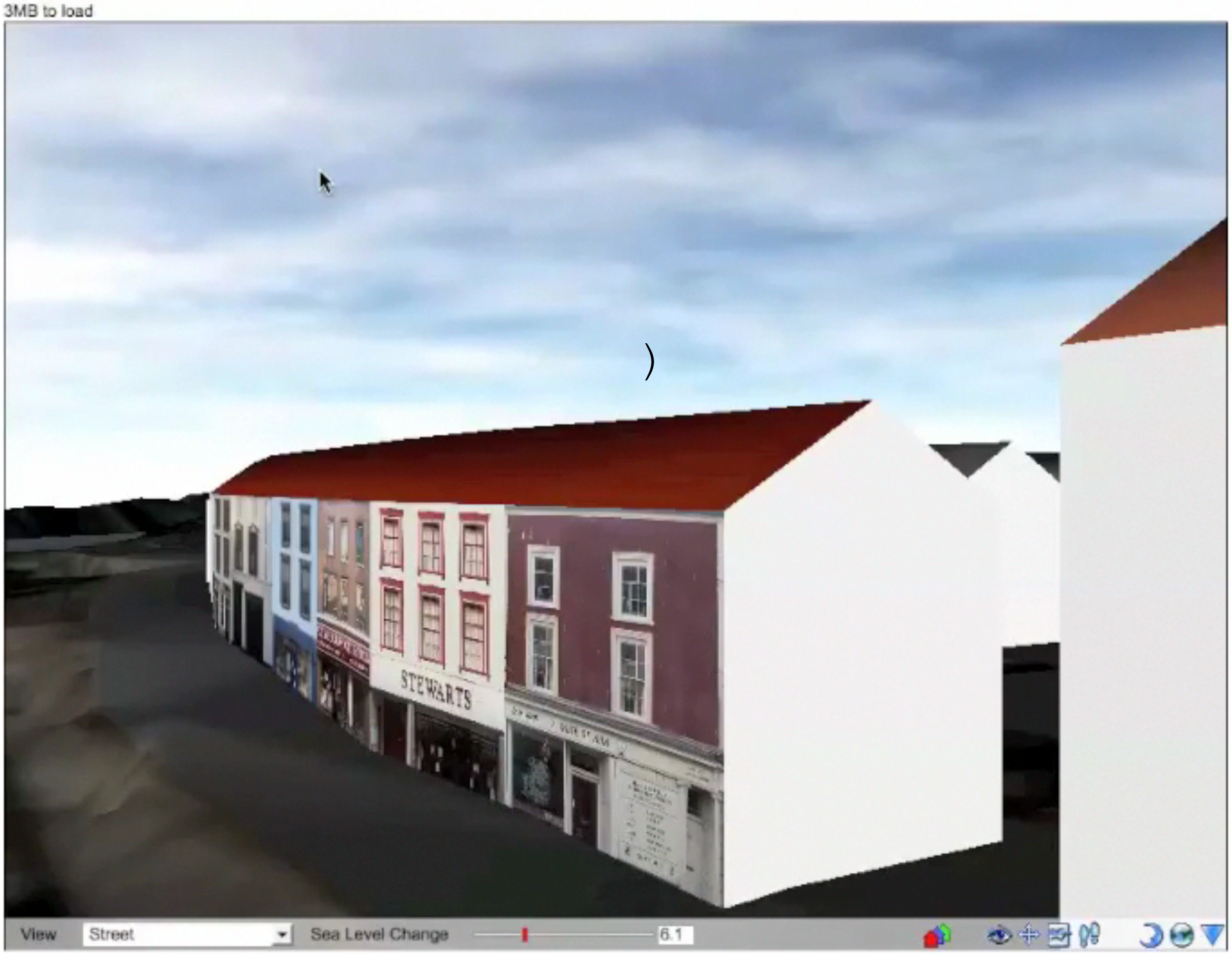
Application show-case

Facebook Friendgraph in 3D



Application show-case

Thematic data on climate change in cities



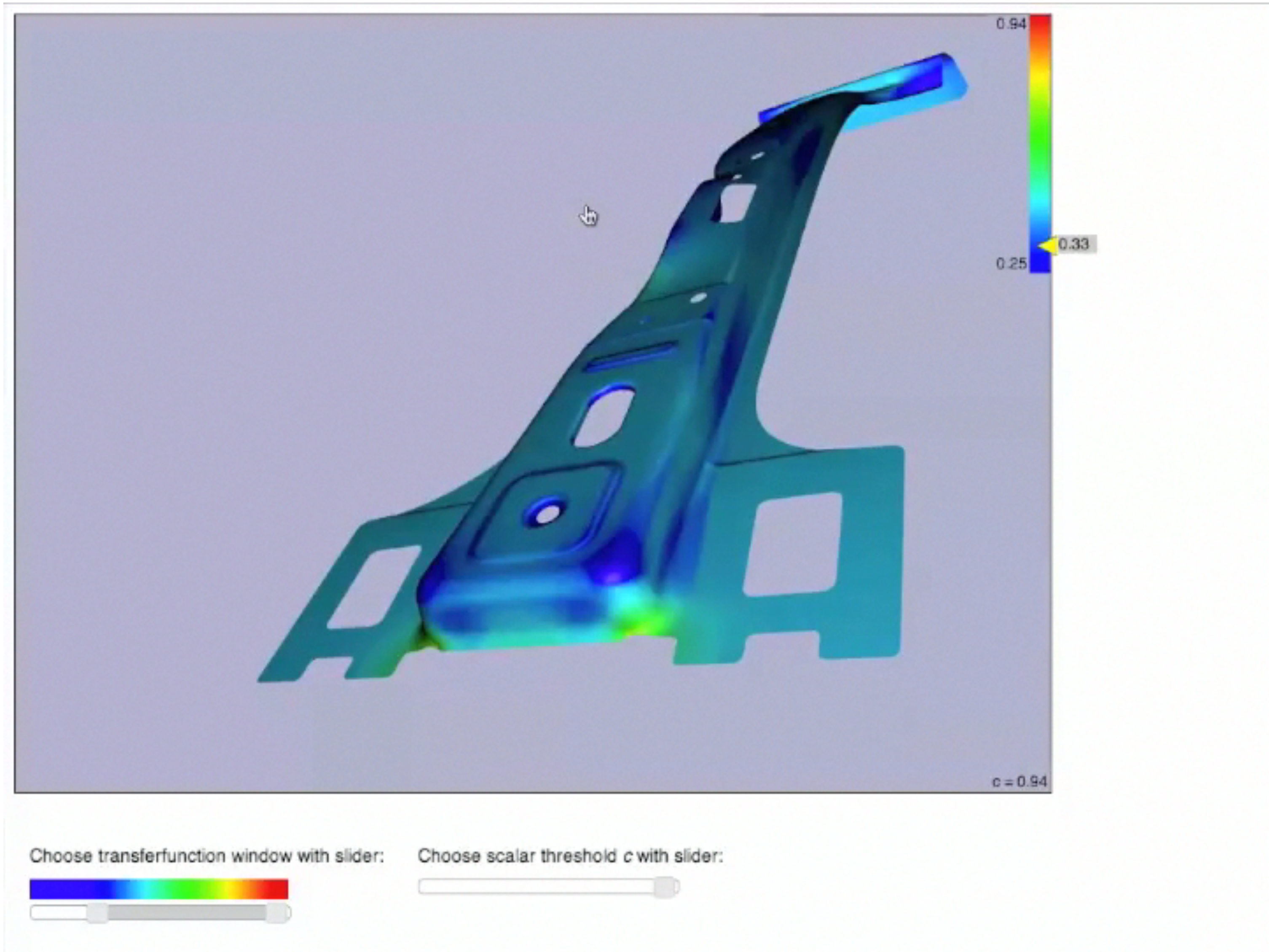
3MB to load

View: Street

Sea Level Change: 6.1

Application show-case

CAE: Data visualization for Volkswagen, IFX



Choose transferfunction window with slider:

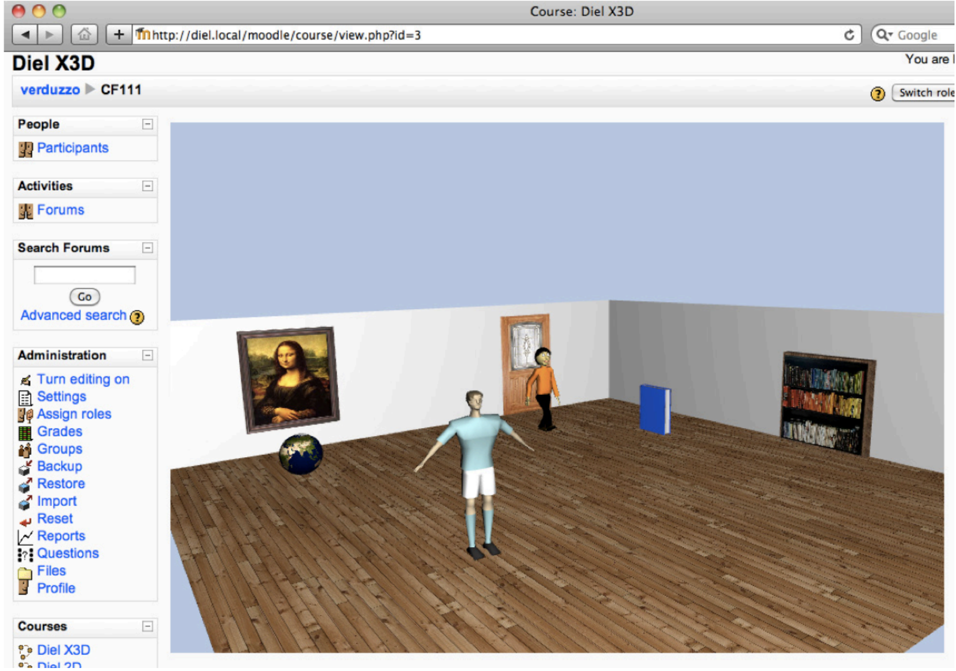
Choose scalar threshold *c* with slider:

Conclusions – X3DOM

X3DOM (pronounced X-Freedom) is an experimental open source framework and runtime to support the **ongoing discussion** in the **Web3D and W3C communities** how an integration of HTML5 and **declarative 3D** content could look like.

Targeted Application Area:

- **Declarative content** design
- Builds on an **HTML5** layer
- Application concepts map well to generic **scenegraph**



Thanks!
Questions?

System:

www.x3dom.org