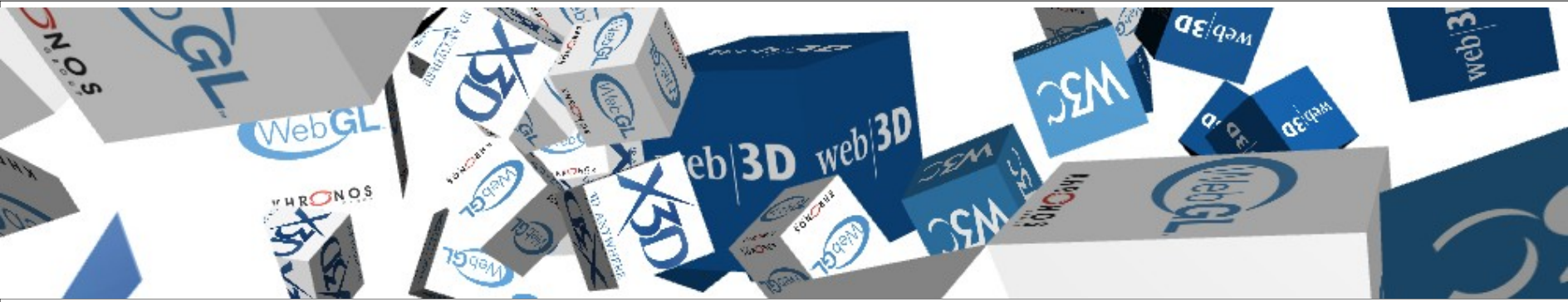


A Scalable Architecture for the HTML5/X3D Integration Model X3DOM



HTML5/X3D BOF, SIGGRAPH 2010

Johannes Behr & Yvonne Jung

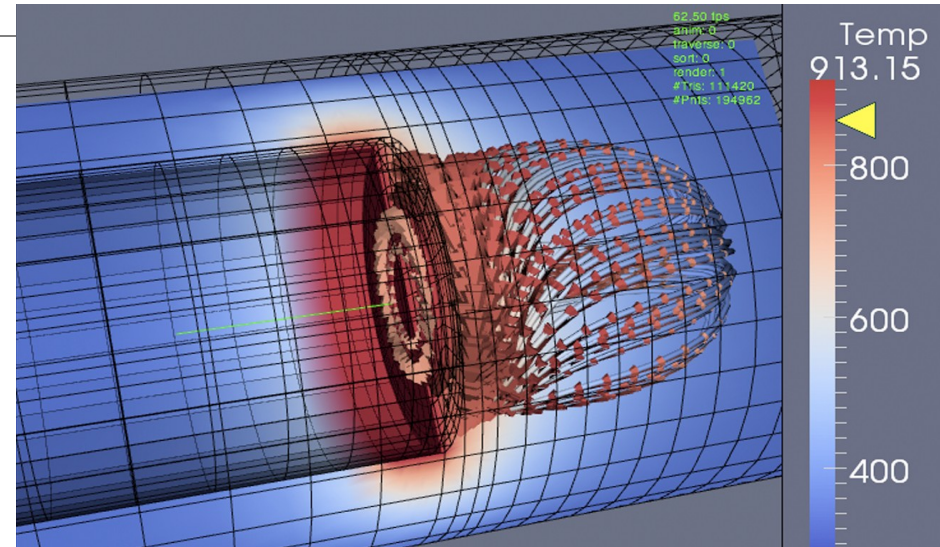
Virtual and Augmented Reality
Group, Fraunhofer IGD,



Overview



- Introduction and Motivation
- Current State of 3D on the net
- X3DOM Model
 - System Architecture
 - HTML & WebSG Profile
 - DOM Updates
 - HTML Events
 - CSS Integration
- Implementation
 - HTML & XHTML encoding





15 years of Web3D technology

Initial hardware and network **limitations are gone**

phones render millions of polygons per second

broadband connection in almost every home

X3D established and solid technology

Successfully used in various application areas

But: **few web3D-applications today**

Increasing interest in 3D web technology

Fat-client: Second-Life, GoogleEarth, Games (e.g. WOW), ...

Browser: X3D, WebGL, ...



Plugins:

- Installation/Security issues

- Data/Event-model is separated from DOM

- plugin specific scripting interface (e.g. SAI for X3D)

WebGL: **Imperative** API

- Based on Canvas3D (Mozilla)

- Developed with Khronos group

- Exposes the OpenGL-ES 2.0 layer to JavaScript



Introduction and Motivation

Current State of 3D on the net

X3DOM Model

System Architecture

HTML & WebSG Profile

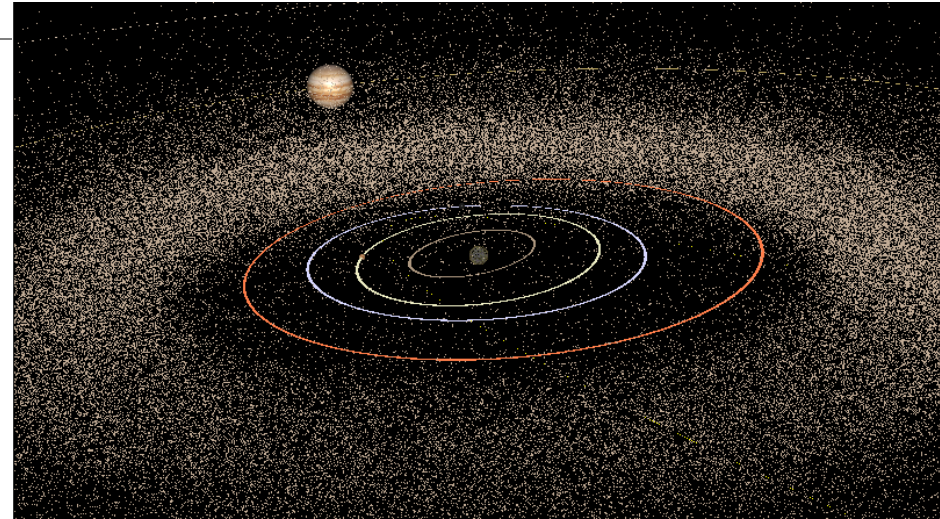
DOM Updates

HTML Events

CSS Integration

Implementation

HTML & XHTML encoding



X3DOM integration model

Declarative 3D in HTML

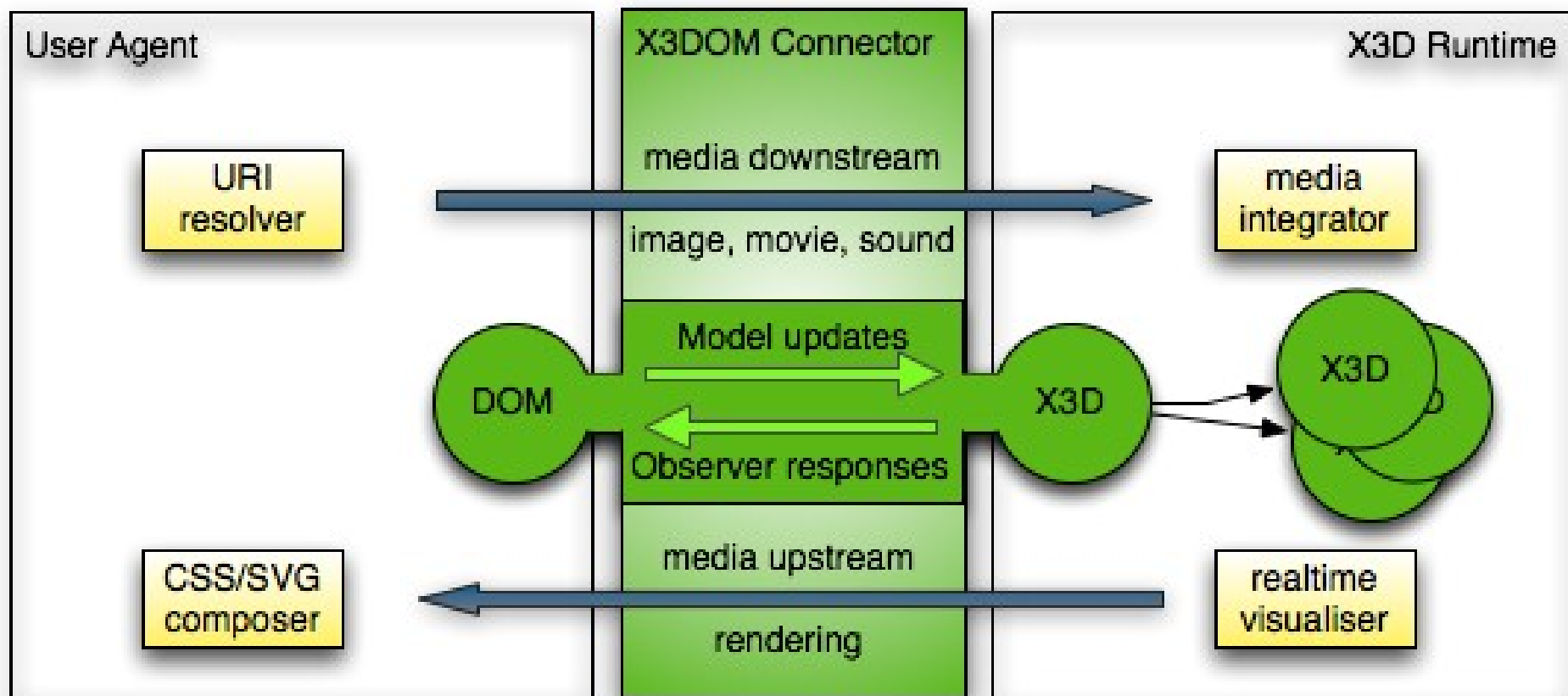


```
<!DOCTYPE html >
<html >
  <body>
    <h1>Hello X3DOM World</h1>
    <x3d xmlns='...' profile='...' >
      <scene>
        <shape>
          <box></box>
        </shape>
      </scene>
    </x3d >
  </body>
</html >
```



X3DOM

Revised system architecture



X3DOM

HTML/DOM Profile



Called “HTML”

Extends “Interchange” Profile

No **Script** nodes

No **Proto** types

No **PointingSensor** types

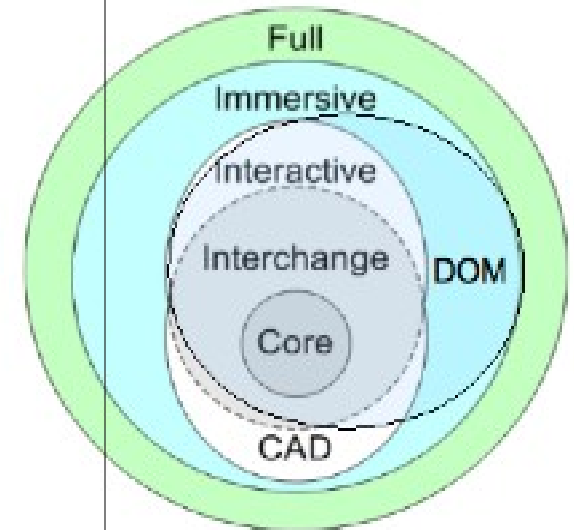
Inline from network component

Supports animation for per-frame updates

TimeSensor, Interpolator, Follower

Two New Nodes

Generic <texture > and <mesh >





<texture> - Generic texture which allows to use standard HTML5 ``, `<video>` or `<canvas>` tags as pixel provider

```
<texture>
```

```
  <img src='foobar.jpg' >
```

```
</texture>
```

<mesh> - Generic mesh-geometry which holds single-index data without any default vertex semantics: + maps directly to VBO, - no semantics for picking ...

```
<mesh primType="triangle" index="  ">
```

X3DOM

websg-Profile / HTML-*tiny* (similar to SVG-*tiny*)



Result of long discussion on the WebGL email-list
=> goal is to reduce the node-set to the absolute minimum.

Called "websg"

Minimal runtime, **no routes**

Only explicit **shader-based** appearance

<texture> uses , <video> or <canvas>

single-index **<mesh>**

always websg backend with specific interface

scene.context holds backend

scene.render() renders the scene

DOM Manipulation

Node appending/removal



HTML/X3D code:

...

```
<group id='root'></group>
```

...

HTML-Script to add nodes:

```
root = document.getElementById('id');
```

```
trans = document.createElement('Transform');
```

```
trans.setAttribute('translation', '1 2 3');
```



DOM Manipulation

Field updates with attribute and 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

HTML Events

User Interaction through DOM Events



```
<x3d xmlns="...">
<Scene>
  <Shape>
    <Appearance>
      <Material id='mat' diffuseColor='1 0 0' />
    </Appearance>
    <Box onclick="document.getElementById('mat').diffuseColor='0 1
0'" />
  </Shape>
</Scene>
</x3d>
```

HTML Events

Events from the X3D subsystem



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

Element identification

X3D and HTML identification scope



HTML global-scope: id, name and class per document

used for script-identification (e.g. getElementById()) und CSS

X3D local-scope: DEF per name-space (= scene + every inline)

used for **ROUTEs**, re**USE** and **import/export**

- ⇒ Can be used separately: `<Box id='foo' DEF='bar' />`
- ⇒ Exclusive **id automatically** assigns internal **DEF** `<Box id='foo' />`
 - ⇒ Id can be used for **ROUTEs** and re**USE**

Accessing Inline content:

- **Bottom-Up Interface** : Import/Export declaration (pure X3D stvle



CSS: presentation semantics, look and formatting of the document

X3D: No strict separation between structure and style; no layout process

X3D **counterpart:** Transformation and Appearance (including Shader)

Appearance:

Fix assigning of existing properties: color to diffuseColor

=> very limited, only works with fix material

Dynamic use of existing properties:

diffuseColor='@color'

=> very flexible, really what we want ?



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); }
```

Implementation



Introduction and Motivation

Current State of 3D on the net

X3DOM Model

System Architecture

HTML & WebSG Profile

DOM Updates

HTML Events

CSS Integration

Implementation

HTML & XHTML encoding





```
<!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>
```



HTML Encoding (current UA limits)

without xmlns, lower-case tags, no-self closing tags

```
<!DOCTYPE html >
<html >
<body>
  <h1>HTML5 Hello World</h1>
  <x3d profile='... >
    <scene>
      <viewpoint position='...'></viewpoint>
      ...
      <route ... ></route>
    </scene>
```

XHTML Encoding (current UA limits)

with xmlns, case-sensitive tags, self-closing tags



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <body>
    <h1>HTML5 Hello World</h1>
    <X3D xmlns="..." profile='...' >
      <Scene>
        <Viewpoint position='...' / >
        <ROUTE... />
      </Scene>
    </X3D>
  </body>
</html>
```

Unified XHTML/HTML Encoding

with xmlns, case-insensitive tags, no-self closing tags

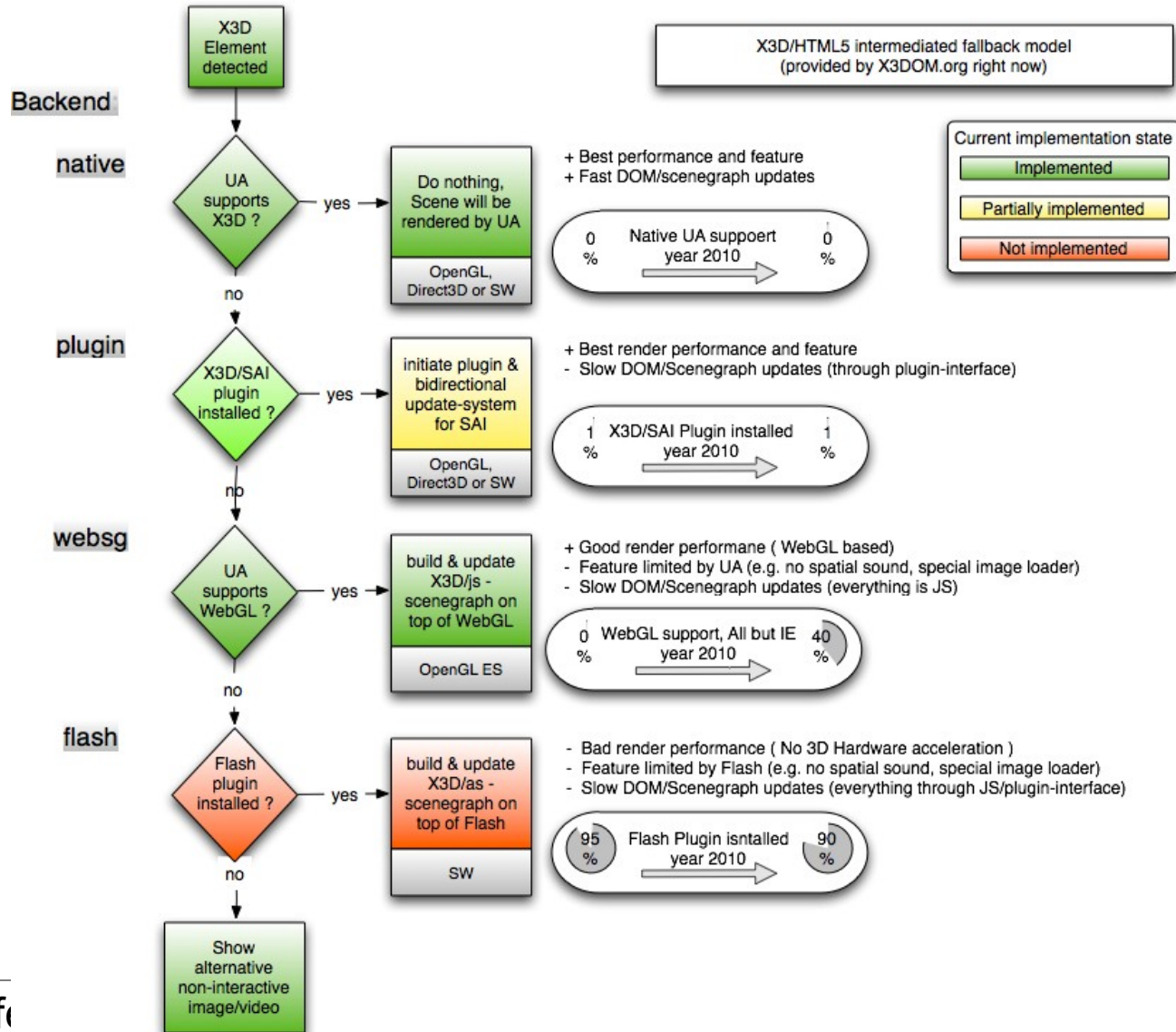


```
<!DOCTYPE html >
<html >
<body>
  <h1>HTML5 Hello World</h1>
  <x3d xmlns="..." profile='...' >
    <scene>
      <viewpoint position='...'></viewpoint>
      ...
      <route ... ></route>
    </scene>
```



Fallback model

matching existing backends and content-profile



Plugin-Backend

X3D-SAI Plugin for scene-management and rendering



Detection: mimetype or ActiveX-ObjectID check

Profile matching: Plugin provides profiles

Lifetime control: Plugin can be instantiated and disposed from JS-layer

DOM synchronization:

DOM serializer:

```
domString = (new XMLSerializer()).serializeToString(x3ds);
```

```
scene = browser.createX3DFromString(domString);
```

=> Works in most browsers

SAI ImportDocument:

```
scene = browser.importDocument(x3d);
```

=> x3d-dom modification: Add SAI Node pointer to every DOM

WebGL-Backend

X3D-SAI Plugin for scene-management and rendering



Detection: ask canvas for webgl-context

Profile matching: HTML and HTML-tiny

Lifetime control: is part of the JS-layer

DOM synchronization:

DOM level2 Mutation Events for DOM monitoring

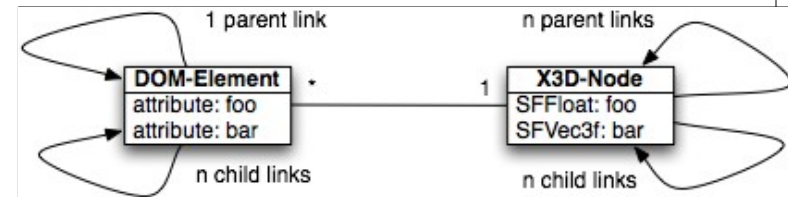
Creates JS-scenegraph with optimized data-container

One SG-Node can be linked to multiple HTML-Elements for reUSE

WebGL for rendering

Limitation

JS performance (scene traversals, culling, math, ray-intersects, ...)



Application, Conclusion and Future Work

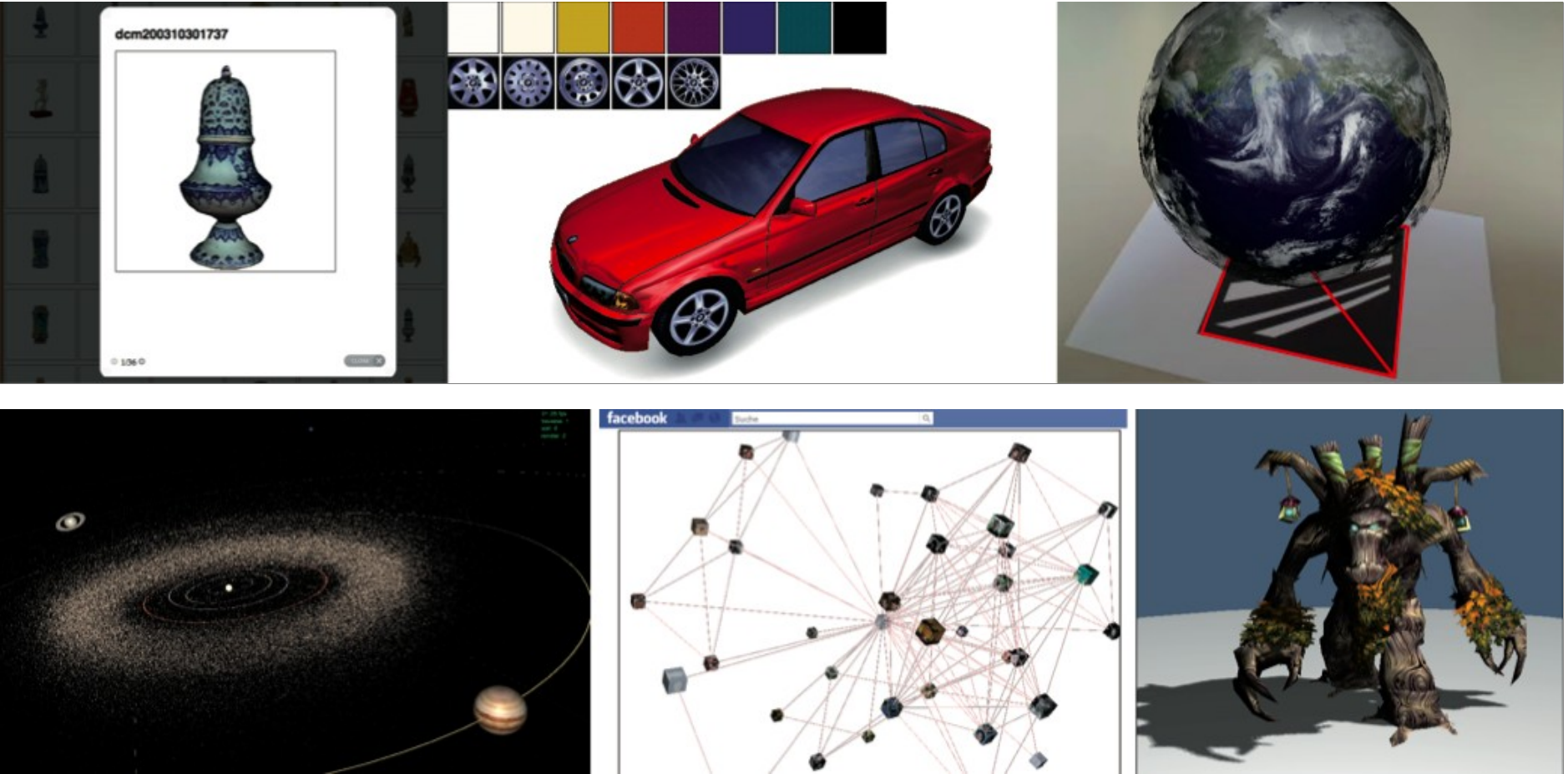


Introduction and Motivation
Current State of 3D on the net
X3DOM Model
 System Architecture
 HTML & WebSG Profile
 DOM Updates
 HTML Events
 CSS Integration
Implementation
 HTML & XHTML encoding



Application and Example

www.x3dom.org



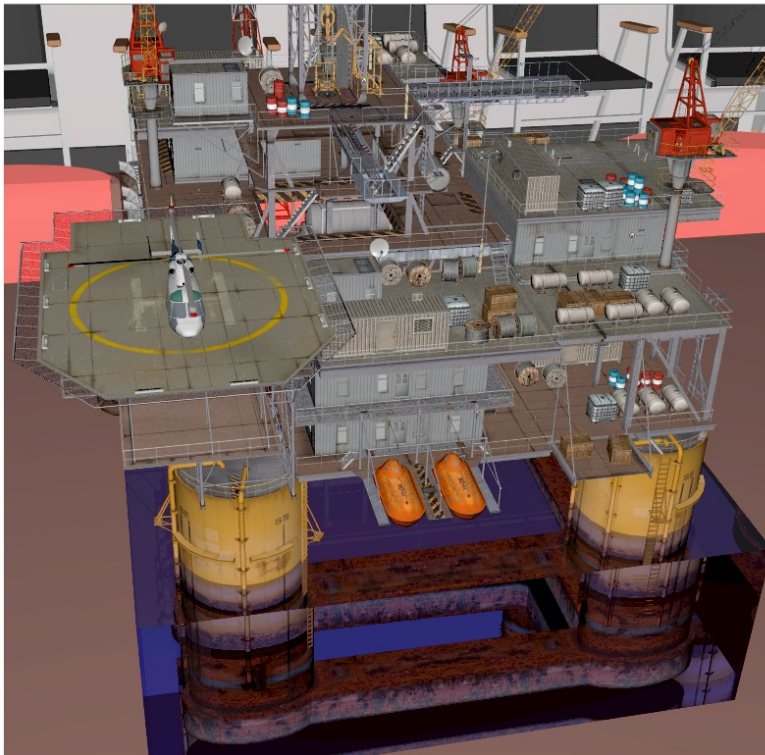
Application booth (#1024) demos



IE8 Plugin (OpenGL and Raytracing) and native iOS app with x3dom layer

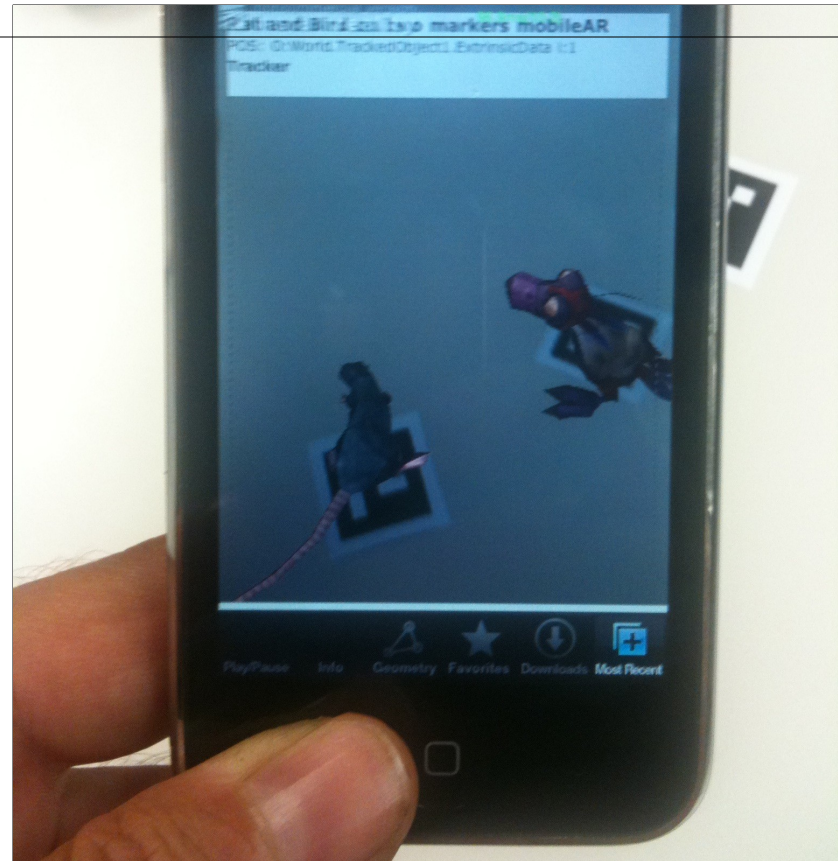
instant 3D - Plugin-based raytracing with x3D

More about x3dom: x3dom.org



Show Normal Rendering

Crane Rotation





Conclusion:

Updated integration Model

CSS integration

Unified encoding

x3dom.org implementation

Fallback-Model to match existing
backend and profile

SAI-browser support

WebGL-Scenegraph support

Future Work:

X3DOM



Thanks!

Questions?

System:

www.x3dom.org

X3D/HTML5 interest group

http://www.web3d.org/x3d/wiki/index.php/X3D_and_HTML5