The Future of X3D

X3D version 4 with HTML5/DOM

Roy Walmsley and Don Brutzman

roy.walmsley@ntlworld.com brutzman@nps.edu

7 June 2017

Web3D 2017, Brisbane Australia

Summary

What will next X3D International Standard look like?

Current plans of X3D Working Group

Your comments, insights and wishes are encouraged

What is X3D?

Extensible 3D (X3D) is a software standard for defining interactive web- and broadcast-based 3D content integrated with multimedia.

X3D is intended for use on a variety of hardware devices and in a broad range of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds.

X3D is also intended to be a universal interchange format for integrated 3D graphics and multimedia. X3D is the successor to the Virtual Reality Modeling Language (VRML), the original ISO standard for web-based 3D graphics (ISO/IEC 14772).
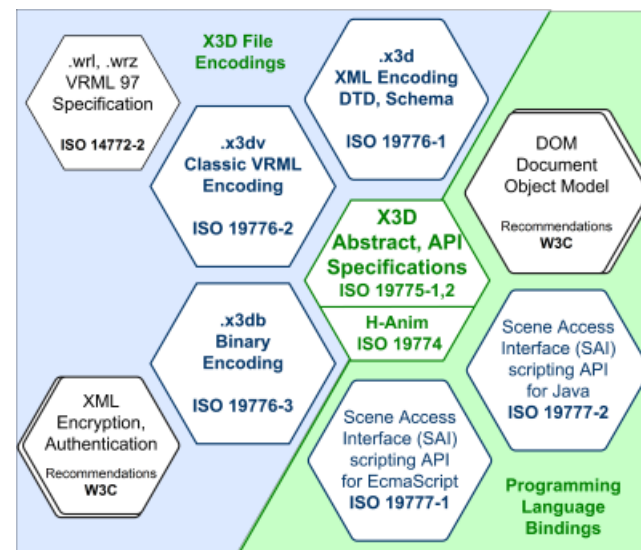
X3D improves upon VRML with new features, advanced application programmer interfaces, additional data encoding formats, stricter conformance, and a componentized architecture that allows for a modular approach to supporting the standard.
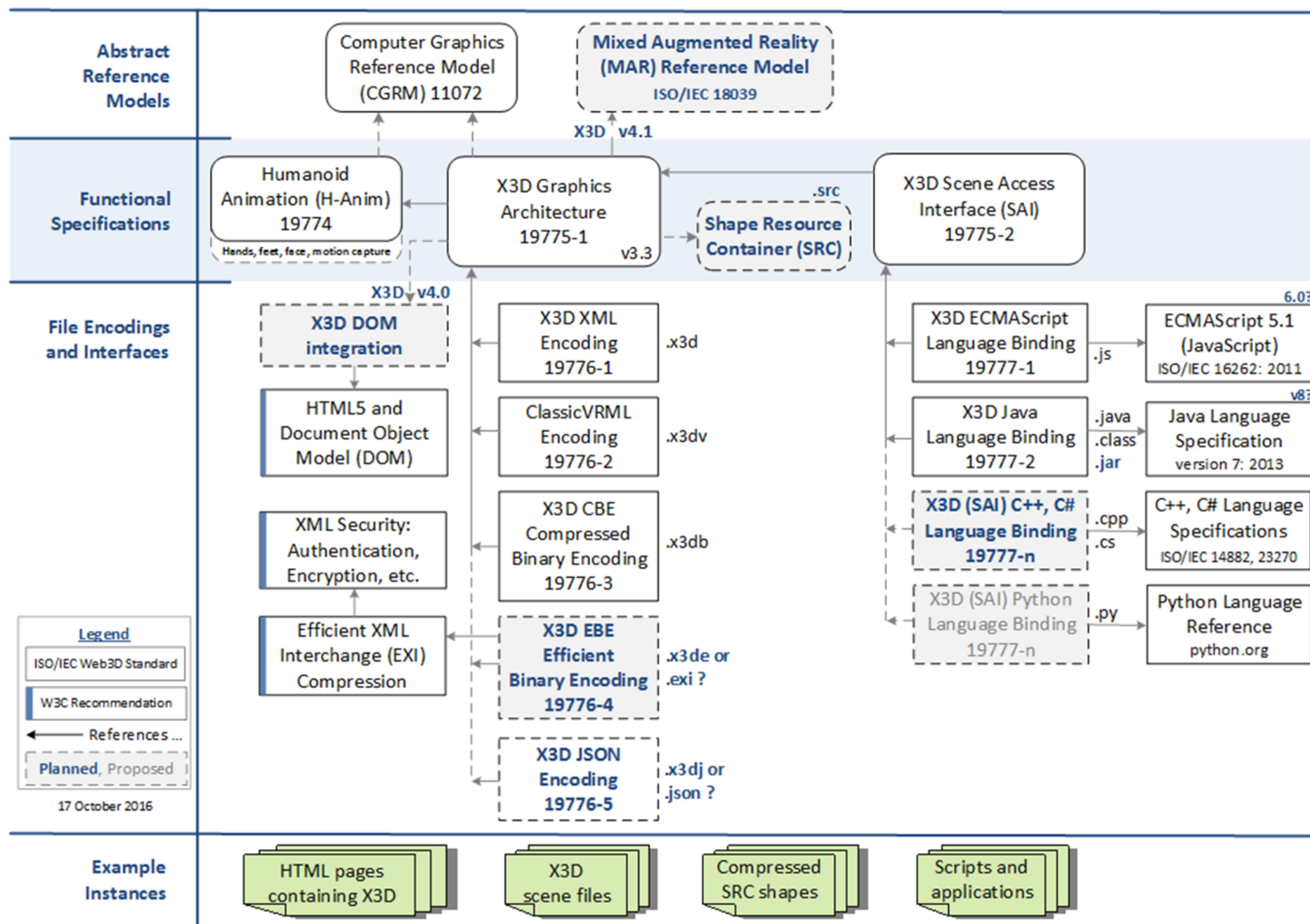
X3D V4 Working Hypothesis

- Structured similarly to current v3.3, and will support a wide range of execution environments which will include HTML.

- Development is based on three strands:

  - Addition of material to cover the integration of X3D into the HTML/DOM environment

  - Extension of the current node set to cover new technologies, and provide rendering improvements (NOTE: This would previously have been considered as evolutionary development for V3.4)

  - Resolution of the many comments raised against the current suite of standards.

ISO/IEC Strategy

- V4.0 of X3D will continue to be an ISO/IEC international standard. While it has a long development time, this also provides longevity and stability for both current and future users of X3D, whatever environment they are working in.

- The Web3D Consortium process has been essential for so much successful progress to occur. Our Standards Strategy is a proven process that succeeds for a simple reason: everybody working together is always smarter and better than any group working alone.

- The X3D Working Group welcomes broad consideration of these many issues. We are also ready for more to happen, and look forward to clear strategic messaging that attracts even more participants.

- Please tell us what you think, we continue listening and talking.

X3D Graphics Standards: Specification Relationships

Abstract Reference Models

Computer Graphics Reference Model (CGRM) 11072

Mixed Augmented Reality (MAR) Reference Model
ISO/IEC 18039

X3D v4.1

Functional Specifications

Humanoid Animation (H-Anim) 19774
Hands, feet, face, motion capture

X3D Graphics Architecture 19775-1
v3.3

.src
Shape Resource Container (SRC)

X3D Scene Access Interface (SAI) 19775-2

X3D v4.0

File Encodings and Interfaces

X3D DOM integration

X3D XML Encoding 19776-1
.x3d

X3D ECMAScript Language Binding 19777-1
.js

6.0?
ECMAScript 5.1 (JavaScript)
ISO/IEC 16262: 2011

HTML5 and Document Object Model (DOM)

ClassicVRML Encoding 19776-2
.x3dv

X3D Java Language Binding 19777-2
.java .class .jar

v8?
Java Language Specification
version 7: 2013

XML Security: Authentication, Encryption, etc.

X3D CBE Compressed Binary Encoding 19776-3
.x3db

X3D (SAI) C++, C# Language Binding 19777-n
.cpp .cs

C++, C# Language Specifications
ISO/IEC 14882, 23270

Efficient XML Interchange (EXI) Compression

X3D EBE Efficient Binary Encoding 19776-4
.x3de or .exi ?

X3D (SAI) Python Language Binding 19777-n
.py

Python Language Reference
python.org

Legend

ISO/IEC Web3D Standard

W3C Recommendation

→ References ...

Planned, Proposed

17 October 2016

X3D JSON Encoding 19776-5
.x3dj or .json ?

Example Instances

HTML pages containing X3D

X3D scene files

Compressed SRC shapes

Scripts and applications

Execution Environments

- X3D V4 will support a wide range of execution environments such as:
  - Desktop application (browsers)
  - Web pages written in HTML5, without requiring a plugin
  - Mobile devices, using device OS-specific capabilities (e.g. FreeWRL on Android)
  - Rendering and running inside a PDF document
  - Running as a modular component written for Windows OS and packaged as a dynamic link library (DLL) to be incorporated into Windows apps
  - Running in native execution environments in VR hardware, commonly via a Scene Access Interface (SAI) library
  - Model exchange and interchange, for example CAD composition or game engines
  - Metadata-annotated models from 3D scanners or for 3D printers
- For X3D V4, HTML is just another interactive display environment.
  - While this commitment to treat the HTML execution environment on a more-or-less equal level with the others applies only to the standards document, developers and companies will still be free to tailor their implementations for what they see as an HTML-centric world.

Modularity

- X3D is a large standard, with 233 nodes plus additional statements.
- Profiles
  - General, e.g. Interchange, Interactive, Immersive, Full
  - Domain related, e.g. CAD Interchange, Medical Interchange
- Components
  - Geospatial, CAD geometry, Rigid body physics, Particle systems, Volume rendering, and several dozen other implemented areas of interest

- Developers can choose what to support in their implementations
- Authors can similarly specify what support they want for their content

Potential modifications to X3D standards suite 1

- Additions for the integration (see note below) of X3D into DOM/HTML. These might be:
  - A new standard, currently thought of as 19775-3. This is an additional abstract standard that includes mappings of X3D nodes into the DOM, and merging of HTML methodologies such as event handling into X3D scenes.
  - Possible new encodings for HTML / XHTML. However, it may be possible to simply refer to the XML encoding, with minor amendments.
  - Some changes to the abstract standard to cover the HTML environment.
  - NOTE: The formal SVG and draft XML3D specifications can provide insights into requirements for integration of graphics into HTML.
- Possible addition of new application oriented content profiles, for HTML and 3D printing and scanning, for example.
- Inclusion of a formal Object Model for X3D (OM4X3D), which provides a solid foundation for consistency both within the standards suite and for implementations.
- Corrections from resolution of technical issues raised against the existing standards.
- Automation of Open Source implementation matching the requirements of the scene access interface (SAI) specifications.
- Addition of new nodes as necessary to cover usage in mobiles, with VR headsets, etc.
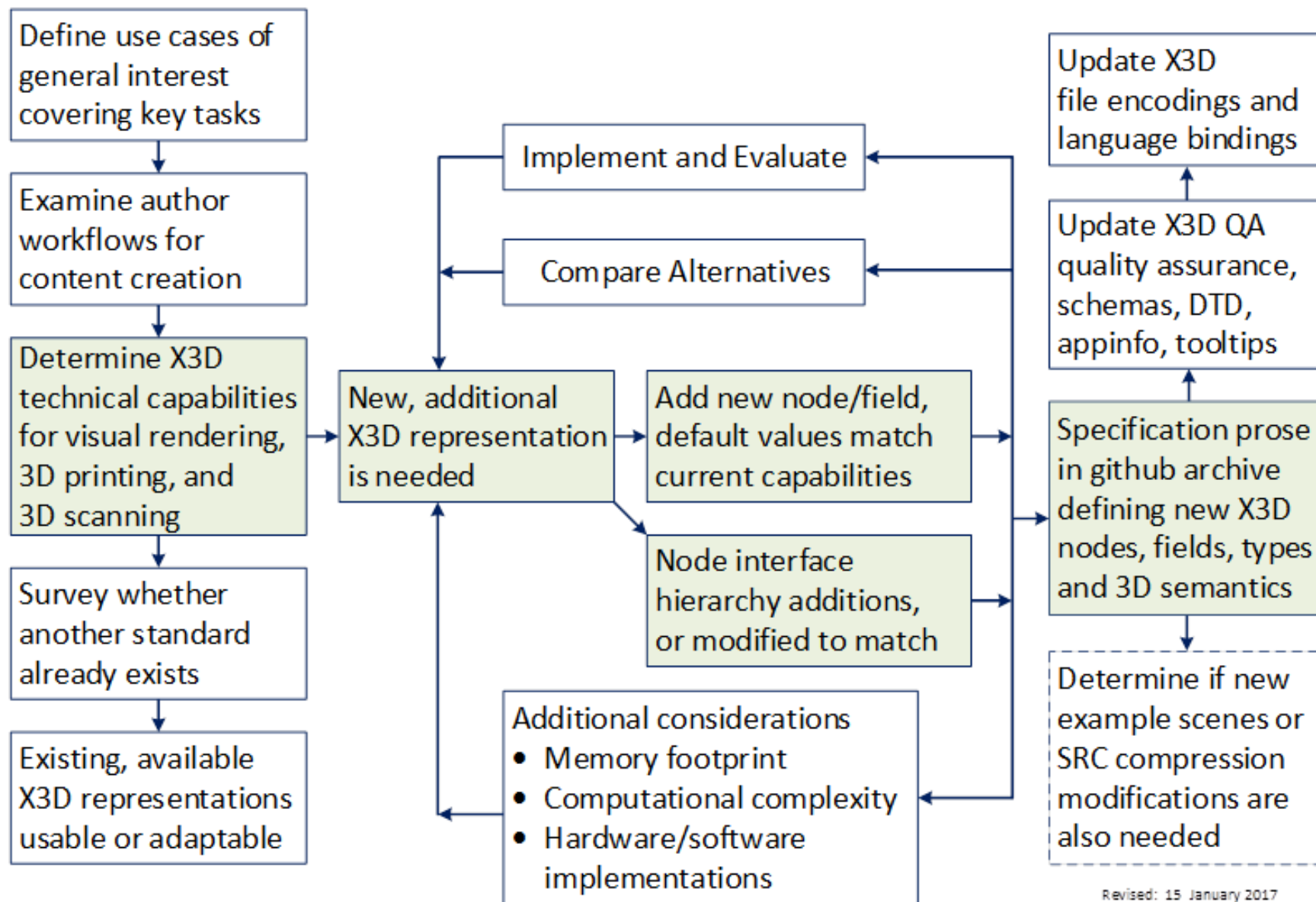
Potential modifications to X3D standards suite 2

- Addition / extension of existing components with improvements, extensions. Examples include advanced graphics capabilities, such as shadows and bump mapping through extensions to programmable shaders; physically based materials; projective texture mapping (PTM).

- Possible deprecation / removal of existing nodes.

- Addition of updates for X3D to match the developments currently underway with the H-Anim standards. These developments include high-fidelity hands and feet representations plus motion capture (mocap) capabilities.

- Additional encodings and language bindings, such as a JSON encoding and a C++/C# language binding, for example.

- Comprehensive binary compression using Shape Resource Container (SRC), glTF, and EXI.

- Formal identification of security considerations throughout the standards suite, currently missing.

- Automated generation of validation tools for the various encodings and open source APIs, for example, JSON schema validation and X3D Java scene access interface library (X3DJSAIL).

- Metadata mappings with broad expressive power to allow X3D alignment with many external specifications and datasets. We expect that metadata will unlock new synergies for CAD, medical, heritage, printing, scanning, etc.

- Future additions for Mixed and Augmented Reality (MAR), currently planned for V4.1.

Usage Environments

- Metadata
- 3D Printing and scanning
- PDF documents
- Semantic Web
- 3D data visualization in literature
- HTML
- Immersive VR
- AR / MAR

Specification design process for new capabilities

Define use cases of general interest covering key tasks

Examine author workflows for content creation

Determine X3D technical capabilities for visual rendering, 3D printing, and 3D scanning

Survey whether another standard already exists

Existing, available X3D representations usable or adaptable

Implement and Evaluate

Compare Alternatives

New, additional X3D representation is needed

Add new node/field, default values match current capabilities

Node interface hierarchy additions, or modified to match

Additional considerations
- Memory footprint
- Computational complexity
- Hardware/software implementations

Update X3D file encodings and language bindings

Update X3D QA quality assurance, schemas, DTD, appinfo, tooltips

Specification prose in github archive defining new X3D nodes, fields, types and 3D semantics

Determine if new example scenes or SRC compression modifications are also needed

Revised: 15 January 2017

Consequences

- The standard development process takes time, but participants are guaranteed that results work.

- Backwards consistency is maximized, continuing to maintain longevity of content.

- Costly mistakes are avoided. Industry initiatives are frequently terminated after one to two years of effort.

- Consistency and interoperability between all encodings is maintained, permitting, for example, the inclusion by reference of X3D scenes into HTML web pages. Therefore, progress in any one typically benefits other areas, as well.

- The browser manufacturers are unlikely to want to implement a large suite of nodes. Subset profiles, however, can directly meet such needs. Experience has shown that application capabilities tend to steadily grow when multiple implementations are available.

- Two active open source JavaScript implementations are available that allow us to test all aspects of X3D/HTML integration. These both require JavaScript and CSS files to be included – that is just two lines to be added to any HTML web page. Their uptake needs to be encouraged.

- Recent dramatic progress by the Castle Game Engine and FreeWRL open source players demonstrates that X3D remains a viable and capable standard.

- We anticipate and are preparing for increased participation by API designers who want to implement the X3D SAI in other programming languages, such as C++ or Python.

Alternative strategies

- To vastly reduce the number of nodes, breaking backwards compatibility. X3D would then look more like A-Frame or XML3D. A small number of nodes. Web designers would need to add a lot of JavaScript to extend the functionality that currently exists in the broad X3D.

- To "fork" X3D, so that the broad X3D is maintained for the classic environment as an ISO/IEC standard, and a reduced set is developed for the HTML environment. This latter might not be an ISO/IEC standard. Perhaps it should be a W3C standard. It would likely mean that X3D files developed for the classic environment could not be loaded into the HTML environment. Over time, the divergence would likely become wider.

- We could try to build X3D nodes on top of one of the simpler implementations, for example, A-Frame or XML3D. This has already been demonstrated to be feasible. What impact would this have, though, on the DOM structures in the HTML environment?
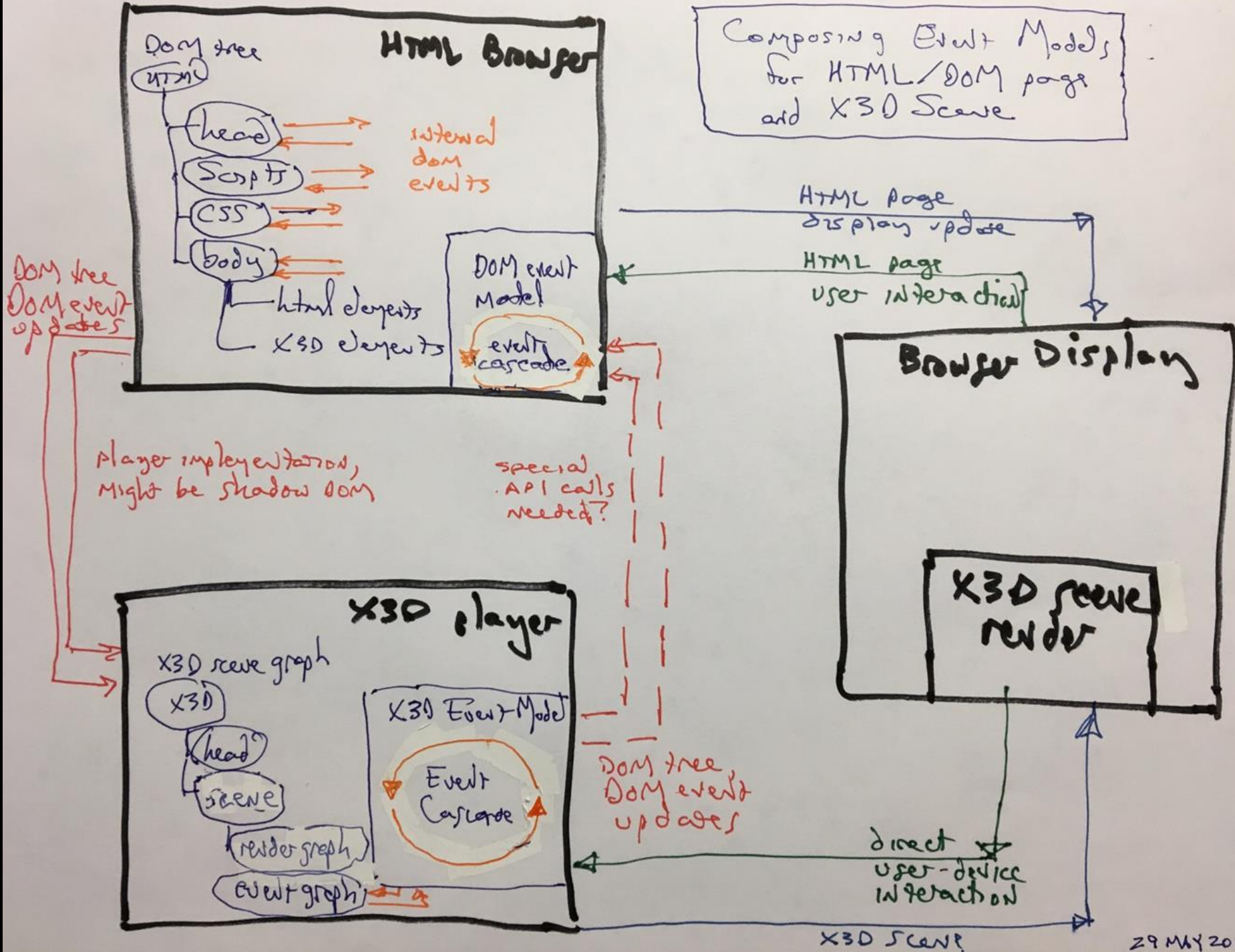
X3D/HTML5/DOM integration considerations

- Support for HTML global event handlers.
- Support for X3D specific event handlers.
- Support for CSS (though we have only had limited discussions on this topic)
- Support for the use of DEF for X3D and id for HTML.
- When X3D is embedded into an HTML web page (as opposed to referenced as an external file) all the nodes in the scene should appear in the DOM just like regular HTML elements. They can, therefore, be accessed using typical HTML approaches.
- In principle, all nodes in the standard, including scripts and prototypes, can be supported.
- Not a goal: Specifying precisely how an implementation must be constructed.

HTML integrated implementation design

- Two parallel sets of nodes – one in the DOM, and an X3D scene graph – that will maintain bidirectional synchronization. This is effectively what both X3DOM and Cobweb do. While there are still issues to resolve, we don't see any "show stoppers." The HTML5/DOM specifications provide guidelines on how such content can be achieved.

- The following diagram shows an initial approach that looks to correlate and compose the existing event models of HTML/DOM browsers and X3D players. It is a starting point. Much further cross-comparison (of multiple specifications and implementations) is likely to produce excellent changes and improvements.

HTML Browser

DOM tree
- HTML
- head
- Script
- CSS
- body
  - html elements
  - X3D elements

internal dom events

DOM event Model

event cascade

DOM tree DOM event updates

player implementation, might be shadow dom

special API calls needed?

Composing Event Models for HTML/DOM page and X3D Scene

HTML Page Display update

HTML page user interaction

Browser Display

X3D scene render

X3D player

X3D scene graph
- X3D
- head
- scene
- render graph
- event graph

X3D Event Model

Event Cascade

DOM tree DOM event updates

direct user-device interaction

X3D Scene

29 MAY 2017

Process and Progress…

- As ever, somebody has to do the work! We scrupulously follow the [Web3D Standards Adoption Process](#) and pay careful attention to [Intellectual Property Rights (IPR)](#).

- Weekly teleconferences and daily mailing list dialog has been continuing for several years to support X3D version 4 development. Historical note: HTML itself took 14 years to progress from version 4 to version 5, and those development efforts are continuing.

- Here is how you can engage:
  - *Everyone.* Adapt X3D models and build mixed HTML/DOM content using Cobweb and X3DOM.
  - *Everyone.* Listen to conversations on x3d-public@web3D.org mailing list, also comment when you like.
  - *Everyone.* Help with test examples, especially for new nodes or existing components with limited coverage.
  - *Everyone.* Help with implementations of new nodes and capabilities.
  - *Members.* Help with Mantis issue tracking
  - *Members.* Help with github specification writing.
  - *Everyone.* Help review specification drafts.
  - *Everyone.* Enjoy! ☺

Your Ideas are Welcome

- We are listening. Reactions are especially important.
  - What do you want?
  - What don't you like?
  - Do you see things differently?
  - Do you agree?

- Let us (and everyone) know. Thanks for considering the possibilities!
  - Public mailing list x3d-public@web3D.org
  - Roy Walmsley roy.walmsley@ntlworld.com
  - Don Brutzman brutzman@nps.edu