Future of X3D: X3D version 4 with HTML5/DOM

Don Brutzman and Roy Walmsley, X3D Working Group Cochairs

Summary. This presentation discusses working-group design and execution strategies for X3D version 4 with HTML5/DOM.

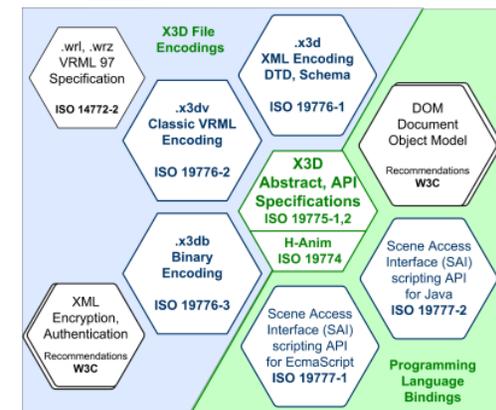## X3D v4.0 Working Hypothesis – Ultra Brief Summary

X3D v4.0 will be structured similarly to current v3.3, and will support a wide range of executions environments which will include HTML. Development is based on three strands:

a. Addition of material to cover the integration of X3D into the HTML/DOM environment
b. 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)
c. Resolution of the many Mantis issues raised against the current suite of standards.
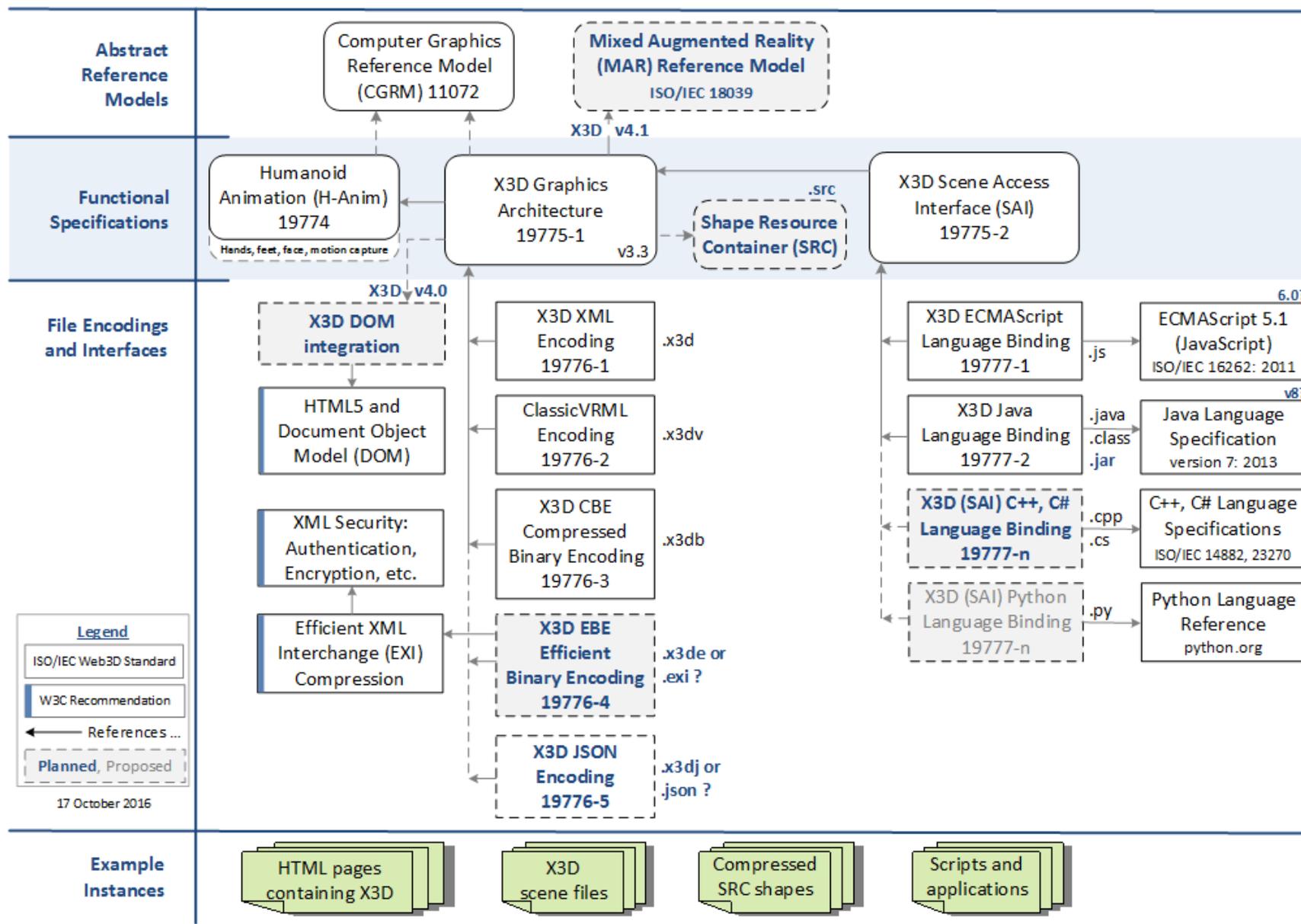
V4.0 of X3D will continue to be an ISO/IEC international standard. While it has a long development time, this also provides longevity and stability for both current and future users of X3D, whatever environment they are working in.

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 will 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

File Encodings and Interfaces

X3D v4.0

- X3D DOM integration
- HTML5 and Document Object Model (DOM)
- XML Security: Authentication, Encryption, etc.
- Efficient XML Interchange (EXI) Compression

- X3D XML Encoding 19776-1   .x3d
- ClassicVRML Encoding 19776-2   .x3dv
- X3D CBE Compressed Binary Encoding 19776-3   .x3db
- X3D EBE Efficient Binary Encoding 19776-4   .x3de or .exi ?
- X3D JSON Encoding 19776-5   .x3dj or .json ?

- X3D ECMAScript Language Binding 19777-1   .js   →   ECMAScript 5.1 (JavaScript) ISO/IEC 16262: 2011   6.0?
- X3D Java Language Binding 19777-2   .java .class .jar   →   Java Language Specification version 7: 2013   v8?
- X3D (SAI) C++, C# Language Binding 19777-n   .cpp .cs   →   C++, C# Language Specifications ISO/IEC 14882, 23270
- 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

Example Instances

- HTML pages containing X3D
- X3D scene files
- Compressed SRC shapes
- Scripts and applications

1. **X3D v4.0 Working Hypothesis.** The X3D v4 standards suite will be structured similarly to current v3.3, and will support a wide range of execution environments such as:

    a. Desktop application (browsers)
    b. Web pages written in HTML5, without requiring a plugin
    c. Mobile devices, using device OS-specific capabilities (e.g. FreeWRL on Android)
    d. Rendering and running inside a PDF document
    e. Running as a modular component written for Windows OS and packaged as a dynamic link library (DLL) to be incorporated into Windows apps
    f. Running in native execution environments in VR hardware, commonly via a Scene Access Interface (SAI) library
    g. Model exchange and interchange, for example CAD composition or game engines
    h. Metadata-annotated models from 3D scanners or for 3D printers

Under this hypothesis for X3D v4, HTML is just another interactive display environment. The structure of the X3D standard and its prototyping, scripting, and componentized domain support are not compromised to accommodate subsetted business requirements, or developer technical preferences. 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.

Key supporting documents:

- X3D Version 4 Strategy,   http://www.web3d.org/x3d4
- X3D Version 4 Development, http://www.web3d.org/wiki/index.php/X3D_version_4.0_Development
    o 1 Genesis and Strategic Overview
    o 2 Legacy Issues
    o 3 Candidate Capabilities
    o 4 Backwards and Forwards Compatibility
    o 5 Architectural Considerations
    o 6 Open Questions
    o 7 Related Work
    o 8 Schedule

2. **Expected technical changes.** Potential modifications to the X3d standards suite include the following:

a. Additions for the integration (see note below) of X3D into DOM/HTML. These might be:

1. 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.
2. Possible new encodings for HTML / XHTML. However, it may be possible to simply refer to the XML encoding, with minor amendments.
3. Some changes to the abstract standard to cover the HTML environment.
4. NOTE: The formal SVG and draft XML3D specifications can provide insights into the requirements for integration of graphics into HTML.

b. Possible addition of new application oriented content profiles, for HTML and 3D printing and scanning, for example.
c. Inclusion of a formal Object Model for X3D (OM4X3D), which provides a solid foundation for consistency both within the standards suite and for implementations.
d. Corrections from resolution of technical issues raised against the existing standards.
e. Automation of Open Source implementation matching the requirements of the scene access interface (SAI) specifications.
f. Addition of new nodes as necessary to cover usage in mobiles, with VR headsets, etc.
g. Addition / extension of existing components with improvements, extensions. One example would be advanced graphics capabilities, such as shadows and bump mapping, and programmable shaders. Another is projective texture mapping (PTM).
h. Possible deprecation / removal of existing nodes, though none yet identified.
i. 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.
j. Additional encodings and language bindings, such as a JSON encoding and a C++/C# language binding, for example.
k. Comprehensive binary compression using Shape Resource Container (SRC), glTF, and EXI.
l. Formal identification of security considerations throughout the standards suite, currently missing.
m. 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).
n. Metadata mappings with broad expressive power allow X3D alignment with many external specifications and datasets. We expect that metadata will unlock new synergies for CAD, medical, heritage, printing, scanning, etc.
o. Future additions for Mixed and Augmented Reality (MAR), currently planned for V4.1.

3. **Summary**. X3D will remain a large and capable ISO/IEC international standard, incorporating new content to meet the needs of all the working groups, and to fulfil the requirements arising from collaboration with external activities.

Implementers can build applications in a modular fashion with subsets of X3D appropriate to their domain. Some are focused on working-group topics, other are hybrids such as ISO JTC-1 activity on 3D printing/scanning with SC 4/TC 184 CAD interchange.

Numerous issues have been explored to show that capabilities can coexist harmoniously. So there is no need for X3D to fork per se, but many approaches can be pursued in tandem. Cross-domain synergies on metadata continue to show the benefits of this approach.

Web3D Consortium Working Groups, http://www.web3d.org/working-groups

- CAD, 3D Printing and 3D Scanning
- Geospatial
- Heritage: Cultural Heritage and Natural History
- Humanoid Animation
- Medical
- Mixed and Augmented Reality (MAR)
- X3D Working Group (for overall coordination)

4. **Potential consequences**.

a. The standard development process takes time, but participants are guaranteed that the results work.
b. Backwards consistency is maximized, continuing to maintain longevity of content.
c. Costly mistakes are avoided. Industry initiatives are frequently terminated after one to two years of effort.
d. 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.
e. 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.
f. 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.
g. Recent dramatic progress by the Castle Game Engine and FreeWRL open source players demonstrates that X3D remains a viable and capable standard.
h. 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.

5. **Alternative strategies**.

    a. 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. Of course, if X3D is not the chosen 3D format, then we have lost out.

    b. 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 would 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.

    c. We could try to build X3D nodes on top of one of the simpler implementations, for example, XML3D. This has already been demonstrated to be feasible. However, even if browser manufacturers incorporated XML3D code, the X3D upper layer would still need the inclusion of JavaScript and CSS files, as above. This would not seem to offer any significant advantages.

Of significant note: none of these alternative strategies seem consistent with the Web3D Consortium process, Web3D working group charters, or already-declared priorities for X3D version 4.

6. HTML5/DOM integration.

The question is often asked, what we mean by HTML/DOM integration? Given years of experience with plugins, an earlier VRML97 External Authoring Interface (EAI), and multiple JavaScript players Cobweb and X3DOM, there is no shortage of both similarities and differences. Lots of work keeps sorting out and solving issues (i.e. the essence of an X3D Specification). Here are a few items:

1. HTML global event handlers are supported.
2. X3D specific event handlers are supported.
3. CSS is supported (there have been no significant discussions on this yet, though)
4. The use of DEF for X3D usage and ID for HTML usage are both supported.
5. When X3D is embedded into an HTML web page (as opposed to referenced as an external file) all the nodes in the scene appear in the DOM just like regular HTML elements. They can, therefore, be accessed using typical HTML approaches.
6. In principle, all nodes in the standard, including scripts and prototypes, can be supported.
7. Not a goal: Specifying precisely how an implementation must be constructed.

One short description of current views on how the integration is achieved is that there can be 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.
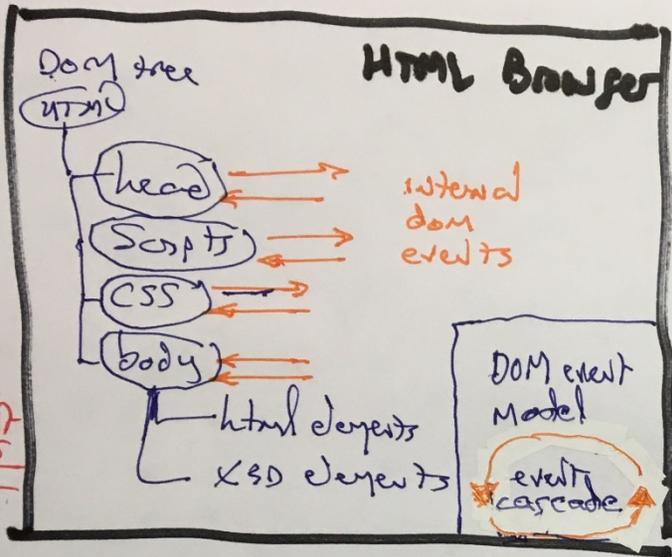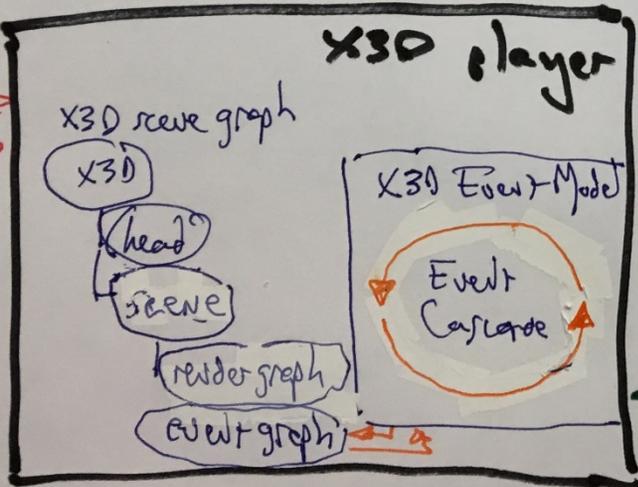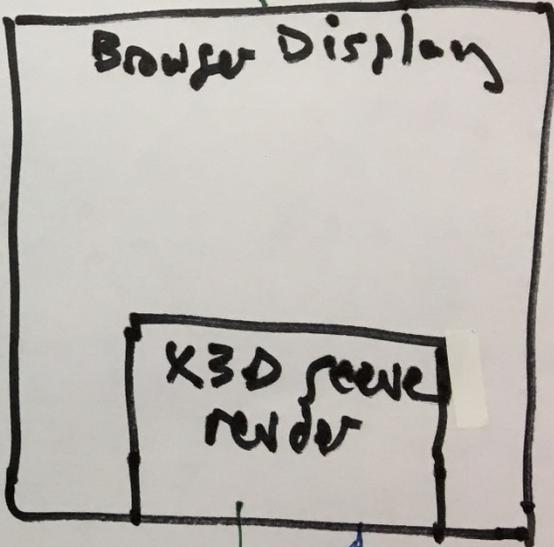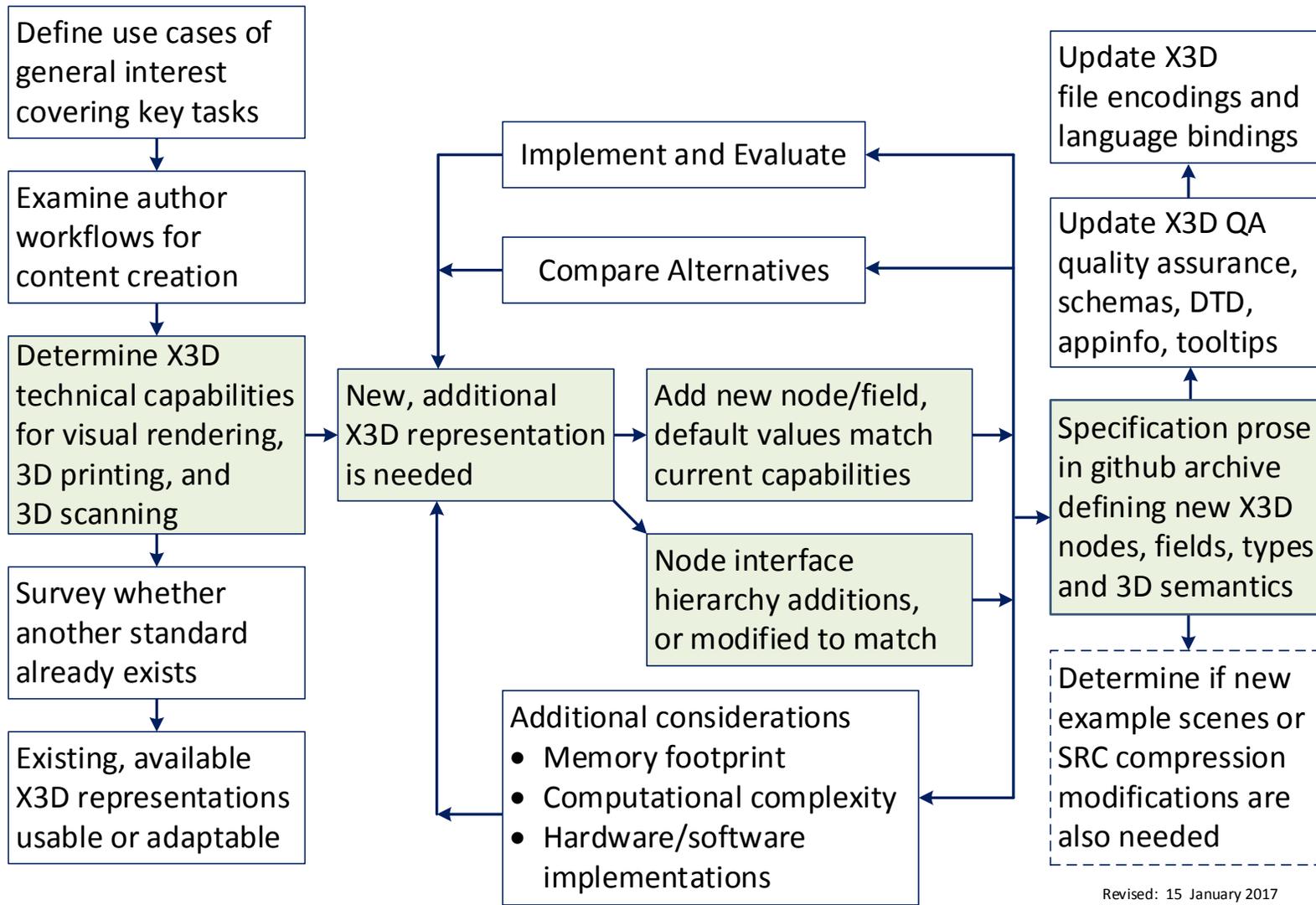
7. **Progress and Process**. As ever, somebody has to do the work! We scrupulously follow the [Web3D Standards Adoption Process](#) and pay careful attention to [Web3D Consortium Intellectual Property Rights (IPR) Policy](#).

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:

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

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