

Declarative Integration of Interactive 3D Graphics into the World-Wide Web: Principles, Current Approaches, and Research Agenda

1 Abstract

With the advent of WebGL, plugin-free hardware-accelerated interactive 3D graphics has finally arrived in all major Web browsers. WebGL is an imperative solution that is tied to the functionality of rasterization APIs. Consequently, its usage requires a deeper understanding of the rasterization pipeline. In contrast to this stands a declarative approach with an abstract description of the 3D scene. We strongly believe that such approach is more suitable for the integration of 3D into HTML5 and related Web technologies, as those concepts are well-known by millions of Web developers and therefore crucial for the fast adoption of 3D on the Web. Hence, in this paper we explore the options for new declarative ways of incorporating 3D graphics directly into HTML to enable its use on any Web page. We present declarative 3D principles that guide the work of the *Declarative 3D for the Web Architecture W3C Community Group* and describe the current state of the fundamentals to this initiative. Finally, we draw an agenda for the next development stages of Declarative 3D for the Web.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality I.3.6 [Methodology and Techniques]: Standards—Languages

Keywords: Declarative 3D, HTML5, DOM Integration, Polyfill

2 1 Introduction

The Web evolved from a text-based system to the current rich and interactive medium that supports images, 2D graphics, audio and video. These types of new media have made the Web experience richer, more attractive to users, etc., than ever before, and opened up possibilities for new types of applications and usage. The major media type that is still missing is 3D: synthetic, possibly photorealistic images in 3D with animation, as smoothly integrated in the everyday Web experience as images or video. Just as the appearance of images or video could open new application possibilities, access to the 3D on a Web site would make it possible to include realistic models of 3D objects – from models of buildings to representation of the human body or the sceneries for computer games. With WebGL [Marrin 2012], a JavaScript binding for OpenGL ES 2.0, this seems feasible; however, the goal would be to achieve the same smooth inclusion of 3D content in a Web page like we experience today with images or SVG-based 2D graphics.

Although some of these goals could also be achieved by imperative means (e.g., through the usage of WebGL), developments of 3D models have a long tradition of using declarative approaches, which is also in line with some of the fundamental principles of Web development. It is therefore important to explore how the experiences accumulated in two different communities, namely the Web Development and Computer Graphics communities, can be capitalized upon to achieve the long term goal of using 3D on the Web the same way as we do with 2D graphics and video today. Moreover, while imperative graphics APIs are powerful and necessary, a *Declarative 3D* approach can provide web authors an easy way to add interactive high-level declarative 3D objects through the

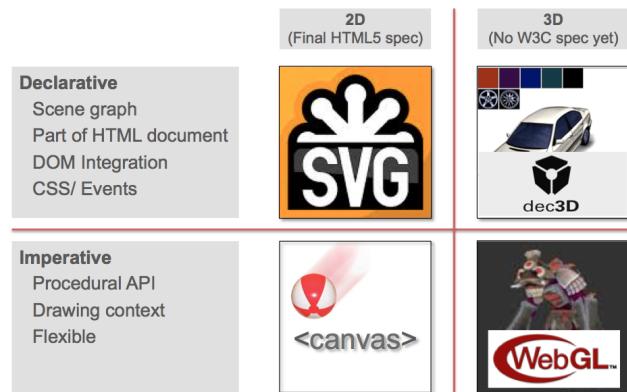


Figure 1: The position of the Declarative 3D approach in the current Web graphics technology ecosystem.

HTML Document Object Model (DOM) allowing them to easily create, modify, share, and experience interactive 3D graphics using HTML documents. Figure 1 depicts the position of the Declarative 3D approach in the current Web graphics technology ecosystem.

It is arguable that the emerging support for an imperative 3D API for the Web is useful but insufficient for broad acceptance and usage of 3D on the Web. A declarative approach that is tightly integrated with current web technologies, such as JSON or XML (for scene construction), DOM (for scene manipulation), and CSS (for styling), and that offers qualified concepts is necessary to support a fast adoption and broad use of interactive 3D graphics by the millions of existing Web developers. The provided concepts must lift the hardware-oriented imperative application programming interfaces (APIs) to an expressive and more easily usable level. Therefore not the low-level data structures of existing hardware layers must be in the center of the design but high-level elements and items like 3D objects, transformations, material descriptions, and lights. Instead of teaching Web developers 3D graphics APIs, the goal is to bring 3D graphics to the point where it is natural for Web developers to just make use of it. While this might not be possible for every possible use of a low-level API, we believe that it can cover the vast majority of use cases.

The *Declarative 3D for the Web Architecture W3C Community Group* (Dec3D) [Dec3D CG 2013] was thus founded to suggest and create methods to add high-level declarative 3D objects to the HTML DOM [W3C 2005], so users can easily create, share and experience interactive 3D graphics. The core mission of this W3C Community Group is to determine the requirements, options, and use cases for the declarative integration of interactive 3D graphics capabilities into the Web technology stack in a declarative way, which hopefully will provide a foundation for future standardization. Therefore, the group aims at presenting common use cases that define how 3D might intersect and interact with HTML5, DOM events, CSS, SVG, GeoLocation, Augmented Reality, Efficient XML Interchange (EXI) and other key working groups, whereas certain complex data types (e.g., transformation matrices) and computations are also of mutual interest. In this regard, this paper presents the current state and efforts concerning Dec3D.

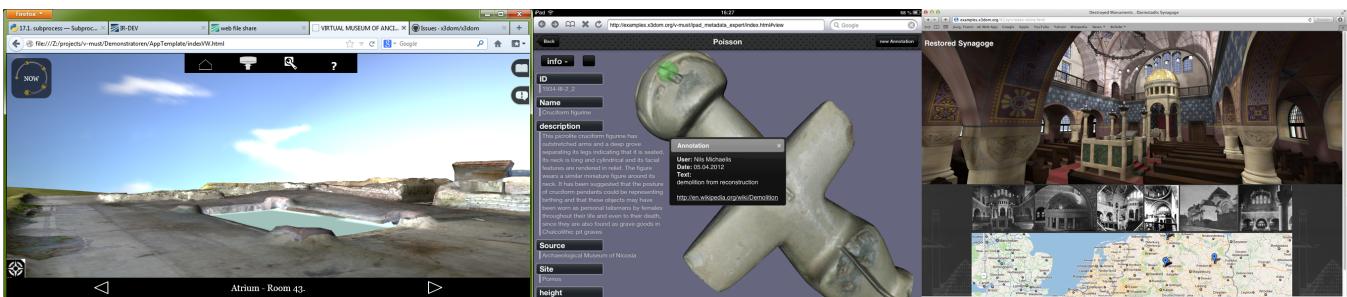


Figure 2: From left to right: walkthrough of Roman heritage site; cultural heritage object explorer with metadata; restored virtual synagogue.

2 Declarative 3D Principles

In this section we describe Declarative 3D principles, where the following key goals should guide the development of DOM-based interactive 3D graphics.

Following the Established Principles of the Web Declarative 3D is being developed to significantly lower the barrier for authoring 3D content for Web sites. It aims to lower the barrier by duplicating the key features that enabled the growth of the early Web and its further spectacular success.

Separation of structure from content - Underlying the Web from its earliest days was the separation of structure from content. The concept of a paragraph specified by the `<p>` tag was separate from the content in the paragraph. Declarative 3D is attempting to bring the same separation of structure from content to 3D graphics inside of web pages. The concepts such as definitions of 3D objects, transformations, materials, etc. should be implemented in a declarative 3D description as an extension to HTML5 using any existing or future extension mechanism.

Separation of content from style - One of the principles of the current Web is also the separation of content and style, most notably through CSS. The successful integration of SVG with HTML was made much easier due to the fact that SVG was already following this principle. The aim of the CG is to extend the use of CSS for styling 3D graphics. One example can be the use of the latest CSS3 3D Transforms to allow to manipulate not only 2D but also 3D objects in 3D space.

Use of the Document Object Model - The Document Object Model (DOM) is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of Web documents [W3C 2005]. Declarative 3D should use DOM API to examine and modify elements on the 3D scene and their attributes by simply reading and setting their properties. As DOM provides access to user actions (e.g., pressing a key or clicking a mouse button), it should also be used as a main interface to interact with 3D content.

Moreover, the CG should reuse existing W3C techniques (specifically from HTML5 and SVG) as far as possible and propose extensions only where 3D specific features are necessary or where they provide significant benefits. Where new concepts are introduced their relation to and effects on existing Web standards should be analyzed, evaluated, and discussed with the respective W3C working groups.

3D Content Creation and Reuse While the creation of original 3D geometry and appearances still requires 3D specific know-how, the reuse, configuration, and manipulation of such content should be made similarly easy as for 2D Web content now. The solution should hide internal data structures and algorithms and provide users convenient ways to edit and manipulate such scenes. A key success factor for Declarative 3D on the Web will be the ability to generate new or reuse existing content. This requires that suitable exporters and converters can be built. However, as 3D on the Web is supposed mainly as a delivery mechanism, it is not necessary to include the ability to semantically represent all but the important 3D features

Platform Independence The aim of the CG is to describe 3D content in a way that does not rely on a specific render API such as OpenGL or DirectX nor on a specific rendering technique such as rasterization or ray tracing only. The technology investigated should allow for content to be portable across user agents, rendering techniques, and hardware platforms, while taking advantage of available features wherever possible. The results of rendering content under such different environments should be highly predictable.

Efficiency and Scalability Interactive 3D graphics operates in real-time, which enables new forms of interactivity on the Web but also adds significant new requirements on user agents. A key requirement for the selected technology therefore is the possibility to implement it efficiently. Since 3D scenes can sometimes become large, any solution should target scalability in the sense that 3D content should run across different platforms (from mobile devices to high-end graphics hardware) with predictable performance. Mechanisms should be in place to handle cases where the performance provided by a user agent on some platform is not sufficient, e.g. by allowing for switching to different content (e.g. lower LOD) or provide alternate methods of delivering the content (e.g. server-based rendering delivered via streaming video).

Security & Digital Rights Management Secure delivery of Web content is a general problem and not specific to 3D data. However, the economic value of 3D data might make the problem more acute. Any proposed solution should therefore be based on a general approach to secure Web content. The CG should, however, collect use cases, extract requirements and examine how far existing methods and standards can be transferred to the proposed architecture. It is already demonstrated that application of XML Encryption and Signature is needed for document fragments as well as full documents, since high-fidelity or sensitive portions of 3D models often need special protections.

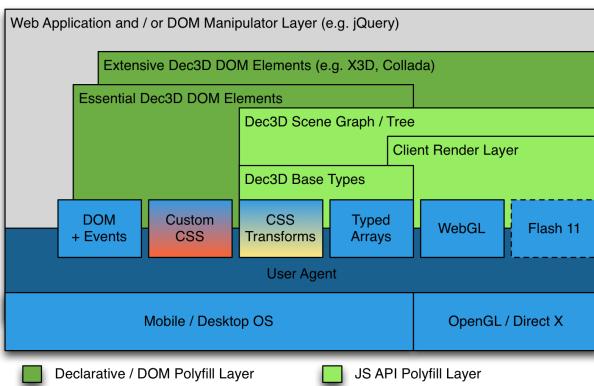


Figure 3: Declarative 3D “polyfill” runtime architecture.

Accessibility & Usability A large body of work has shown that accessibility improvements serve all users, not just people with disabilities. A curious aspect of many 3D graphics approaches is that user navigation is implemented inconsistently. Therefore, users familiar with one approach are impeded when navigating or interacting with other 3D scenes and models. Examination of relevant Web Accessibility Initiatives (WAI) principles might provide significant benefit. Conversely, use of declarative 3D graphics models might provide major benefits when describing the accessibility features and constraints of real-world objects and locations. Declarative 3D goals and potential solutions may achieve significant benefits if they are harmonizable with WAI imperatives.

Leveraging Web Development Infrastructure The Web has become the new application development platform for our time. Countless rich internet applications are being created at a dizzying pace. The “apps” world for mobile applications has also become a huge phenomenon. The single most important technology, among many, that has emerged to enable this explosion of creative applications is JavaScript. JavaScript has evolved from a toy language into a robust and richly nuanced tool that is the basis for much of the recent explosion in Web applications. Declarative 3D, by living within the structure of Web applications is poised to leverage the huge collection of tools and infrastructure that already exists like e.g. the JavaScript library jQuery [Resig 2012]. The ability to take existing Web debuggers, editors, and viewers and use these tools to create, edit and debug 3D content is a tremendous benefit and just begins to scratch the surface of development tools, created for Web application development, but which we can use for 3D content development.

3 Declarative 3D Frameworks

As we have already mentioned, our Community Group has been formed to determine the requirements, options, and use cases for the declarative integration of interactive 3D graphics capabilities into the Web technology stack which will provide a foundation for future standardization. While the standardization is our goal, we still need platforms allowing for the experimentation and for the assessment of our design decisions. We also need to reach out to Web developers who could provide us with valuable feedback as early as possible.

With X3DOM [Behr et al. 2009] and XML3D [Sons et al. 2010] two experimental declarative 3D Web publishing frameworks are available. In the following we briefly introduce these open-source

frameworks, which support the ongoing discussion in the computer graphics and Web communities how an integration of HTML(5) and Declarative 3D content could look like. This discussion is followed by the presentation of the *polyfill* approach for the further X3DOM and XML3D development activities.

3.1 X3DOM and XML3D

Fraunhofer IGD’s X3DOM [Behr et al. 2009; Behr et al. 2012b] is a JavaScript-based open-source framework for declarative 3D graphics in HTML5 that aims at extending the HTML DOM tree with declarative 3D objects while employing modern Web technologies like CSS3, Ajax, DOM scripting, as well as WebGL and – as fallback – Adobe’s Flash 11 with Stage 3D for GPU-accelerated rendering. The proposed 3D elements are mostly based upon the open ISO standard X3D [Web3D Consortium 2008], though X3DOM introduces a special HTML profile that basically extends the X3D Interchange profile. Additionally, instead of implementing the somehow odd X3D pointing device sensor component, X3DOM simply uses, and appropriately extends, the HTML UI/Mouse events such that 3D pick events are likewise supported.

Furthermore, to overcome various problems that come along when embedding 3D mesh data directly into the DOM (which typically consists of several megabytes of vertex data), the developers are working on efficient 3D transmission formats that separate the node structure from the raw vertex data [Behr et al. 2012b]. Based on this work, this is now also a major topic at the Khronos Group [Trevett 2012], where it is similarly argued that raw 3D data, just like image, video or audio data, needs to be externalized from the HTML document. DFKI’s XML3D [Sons et al. 2010] is a similar approach. In contrast to X3DOM, this format proposes a new tag set that is not based on X3D anymore. In recent versions of XML3D, mesh data can also be externalized from the HTML document by storing it in separate JSON files.

3.2 Polyfill Approach

Recently, the CG proposed to measure a level of integration for 3D graphics in terms of W3C technologies (DOM, CSS etc) [Dec3D CG 2012]. The proposal aims to explain existing and possible integration levels and finally to identify an integration level the CG is aiming for.

It quickly became clear that a higher level of integration requires additional APIs in user agents. After discussions with browser vendors (namely Firefox and Chrome), who made it clear that integration of the desired extensions natively into their frameworks is not currently their priority, we decided to consider a polyfill approach for the further development activities of the Community Group.

A *polyfill* is downloadable piece of code, which provides facilities that are not built-in to a Web browser [Sharp 2010]. For example, many features of HTML5 are not supported by versions of Internet Explorer older than version 8 or 9, but can be used by web pages if those pages install a polyfill. Polyfills can also be used to add entirely new functionality to browsers.

Polyfill based approaches allow the Declarative 3D Community Group to derive hard requirements for related and utilized W3C standards and UA APIs immediately. This leads to a much more evolving concepts and solutions in contrast to an overall declarative 3D specification.

To sum up, the X3DOM and XML3D experimental declarative 3D Web publishing frameworks were designed to explore different options for adding interactive 3D graphics to HTML. We want to

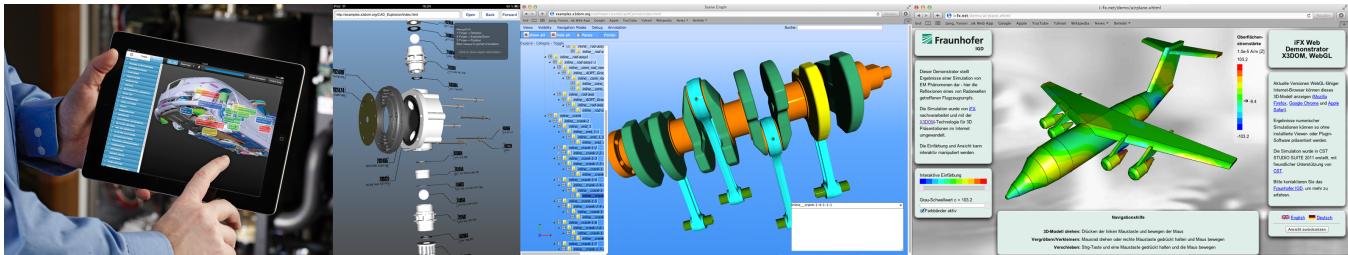


Figure 4: From left to right: web-based design review application with annotation markers; interactive explosion of CAD model on iPad; lightweight web-based viewer for 3D CAD data; visualization and interactive exploration of simulation data by using extended mouse events.

279 stress that the whole integration model is still evolving and it is
280 open for discussion.

281 4 Declarative 3D Agenda

282 In this section we aim to shape the agenda and identify upcoming
283 research issues for the next development stage of the Declarative
284 3D for the Web.

285 During the 1st International Workshop on Declarative 3D for the
286 Web Architecture (Dec3D2012) [Behr et al. 2012a] as well as during
287 more informal group meetings at Web3D in 2011, 2012 and
288 SIGGRAPH in 2011, 2012, the members and supporters of the
289 Declarative 3D W3C Community Group agreed that what is needed
290 to make the effort successful and for the W3C to adopt/develop a
291 declarative 3D standard include the following topics.

292 **Encourage Participation** All relevant stakeholders, e.g. developers,
293 designers, researchers, 3D artists, industry professionals,
294 representatives of standards organizations, accessibility experts,
295 and user-agent implementers, are encouraged to participate in the
296 Declarative 3D Community Group. Participants must be willing to
297 actively develop and donate materials towards the group's deliverables,
298 as well as attend the group's teleconferences and face-to-face
299 meetings.

300 **Clear Definition of Use Cases & Requirements** The Community
301 Group needs to agree on a collection of use cases, where embedding
302 3D data in HTML using declarative approach provides
303 significant benefit. Each use case should explore how publishers
304 and consumers benefit from Dec3D. From these use cases, the
305 Community Group needs to derive and prioritize the different required
306 dimensions for the Declarative 3D technical specification
307 (see [Jankowski 2012; Le Feuvre 2012]).

308 **Clear Technical Specification** The next step of the CG should
309 be the creation of the clear, detailed and extensible technical specification
310 of the implementation concepts and features necessary to cover the majority of useful requirements (but not necessarily all of them). Measurable properties should be defined to quantitatively or qualitatively evaluate the quality of the solution, document the pros and cons of each solution based on these measurements, demonstrate that, based on the above analysis, there is a good chance of success in creating a W3C standard for Declarative 3D for the Web.

317 **Outreach & Exemplar Applications** The CG needs to continue
318 its outreach activities through the high quality demonstrations of
319 the Declarative 3D philosophy using the experimental X3DOM and
320 XML3D frameworks. In this regard, the CG should identify several
321 applications, each requiring and demonstrating different capabili-

322 ties of Declarative 3D. For example, one application could require
323 huge 3D datasets which are impractical for inclusion directly into
324 the DOM; another could require real-time control of an external
325 system (inter process communication); another could integrate with
326 a complex data base; another an illustration of complex data (infor-
327 mation visualization) without a physical analog and another a more
328 traditional scientific visualization. Key to selection of these applica-
329 tions is the use and demonstration of a Declarative 3D requirement.

330 **W3C Working Group Proposal** Finally, the Community Group
331 should deliver a report documenting its progress, any conclusions
332 it arrived at with respect to standardization of *Declarative 3D for*
333 *the Web* and, if reaching a positive conclusion, recommending a
334 standardization approach as a basis for a future W3C working group
335 on the same topic.

336 5 Conclusions

337 While WebGL 3D imperative graphics API in the Web context is
338 getting more and more traction, we are still missing an easy way
339 to add interactive high-level declarative 3D objects to the HTML to
340 allow anyone to easily create, share, and experience interactive 3D
341 graphics with possibly wide ranging effects similar to those caused
342 by the broad availability of video on the Web.

343 The goal of the Declarative 3D CG is to evaluate the options for a
344 successful standardization of a declarative approach to interactive
345 3D graphics as part of HTML documents. The goal is to collect
346 suitable use cases, derive requirements from them, and then find
347 the essential set of features and concepts that enables broad uptake
348 by authors and users of interactive 3D on the Web. We are abso-
349 lutely aware that our goal is ambitious and it will take some time to
350 implement these features. Therefore, we call for more participation
351 from the 3D Web community that we believe is crucial to achieve
352 our common and ultimate goal: 3D for everyone and everywhere.

353 References

- 354 BEHR, J., ESCHLER, P., JUNG, Y., AND ZÖLLNER, M. 2009.
355 X3dom: a dom-based html5/x3d integration model. In
356 *Web3D'09*, ACM.
- 357 BEHR, J., BRUTZMAN, D. P., HERMAN, I., JANKOWSKI, J., AND
358 SONS, K., Eds. 2012. Proceedings of the 1st International Work-
359 shop on Declarative 3D for the Web Architecture (Dec3D2012 at
360 WWW2012), Lyon, France, April 17, 2012, vol. 869 of *CEUR*
361 *Workshop Proceedings*, CEUR-WS.org.
- 362 BEHR, J., JUNG, Y., FRANKE, T., AND STURM, T. 2012. Using
363 images and explicit binary container for efficient and incremental
364 delivery of declarative 3d scenes on the web. In *Proc. Web3D*,
365 ACM, New York, NY, USA, 17–25.

- 366 JANKOWSKI, J. 2012. Writing effective use cases for the declarative 3d for the web architecture. In *Dec3D'12*, CEUR-WS.
- 367
- 368 LE FEUVRE, J. 2012. Towards declarative 3d in web architecture.
369 In *Dec3D'12*, CEUR-WS.
- 370 MARRIN, C., 2012. WebGL specification. <https://www.khronos.org/registry/webgl/specs/latest/>.
- 371
- 372 DEC3D CG, 2012. Levels of integration of 3d graphics on the web. http://www.w3.org/community/declarative3d/wiki/Level_of_Integration/.
- 373
- 374
- 375 DEC3D CG, 2013. Declarative 3d for the web architecture community group. <http://www.w3.org/community/declarative3d/>.
- 376
- 377
- 378 WEB3D CONSORTIUM, 2008. X3d. <http://www.web3d.org/x3d/specifications/>.
- 379
- 380 RESIG, J., 2012. jQuery. <http://jquery.com/>.
- 381 SHARP, R., 2010. What is a polyfill? <http://remysharp.com/2010/10/08/what-is-a-polyfill/>.
- 382
- 383 SONS, K., KLEIN, F., RUBINSTEIN, D., BYELOZYOROV, S., AND SLUSALLEK, P. 2010. Xml3d: interactive 3d graphics for the web. In *Web3D '10*, ACM.
- 384
- 385
- 386 TREVETT, N. 2012. 3d transmission format. In *Seventh AR Standards Community Meeting Talks*.
- 387
- 388 W3C, 2005. Document object model (dom). <http://www.w3.org/DOM/>.
- 389