# HTML.NEXT PANEL DISCUSSION

Standardized Innovation via the W3C

Author:
Patrick Dengler, Microsoft, Member of W3C SVG Working Group and SVG-CSS Task Force

Contributors:
Sylvain Galineau, Microsoft, Member of W3C CSS and Web Fonts Working Groups
Michael Champion, Microsoft, W3C AC Representative
Travis Leithead, Microsoft, Member of W3C HTML and WebApps Working Groups
Adrian Bateman, Microsoft, Member of W3C HTML and WebApps Working Groups

W3C Working Groups are moving HTML 5, SVG 1.1 and portions of WebApps and CSS 3 specifications into Last Call. This is great progress.

Looking back, it is my opinion that these specifications have been disadvantaged by:

- Silos in the Working Group structure which inhibits alignment
- Incongruent technologies that lead to integration problems
- Incompatible timelines which contribute to low interoperability
- A lack of consistent guidelines that blur ownership boundaries

I am not sure what innovations will surface in HTML6, but I do know that changes need to be made in order to move forward and allow future innovation to thrive.

HTML has such momentum, and is expanding quickly. Consumers struggle to make sense of the differences between SVG and HTML, which CSS specification should apply, or even more importantly, how something is supposed to work due to incompatibilities. For example:

- Do animations and transitions apply to SVG?
- Why did ‘border-radius’ require vendor prefixes for seven years?
- Do SVG Filters apply to HTML?
- Why are CSS Transforms, as they apply to SVG, being examined in hindsight?
- Should SMIL animations still work inline in HTML as they do in XHTML?
- Why are there so many interoperability problems?
- What delayed the SVG 1.1 2nd Edition for so long?

Current technology requires a period of **alignment.** We as members of W3C Working Groups need to make changes in our methods to accelerate this alignment. We need to:

1. Establish a single set of guiding principles across working groups and apply them to aligning current technologies.
2. Modularize specifications and share the responsibility of developing these specifications across Working Group members, and remove silos.
3. Temporarily focus Working Groups exclusively on normalizing a prioritized set of functionality in order to deliver predictable consistency to developers.

In order to facilitate this process it will benefit all to establish *and agree on* cross-working group guiding principles up front. To rationalize these technologies more quickly, there is a need to put some stakes in the ground about the general approach.

The intention of the stabilization portion of the process is to provide a more general and consistent target for the end of alignment process. As an example, for SVG, the focus should only be on the SVG Integration Specification, the SVG DOM, and general animation. For the most part, that is the SVG Working Group's current priorities.

Other Working Groups focused on Web technologies would use this same approach to establish this virtual single-version of the Web. Aligned and stabilized, vendors can then rally around these specifications to provide an interoperable, stable platform upon which Web developers can depend. For Web Developers, testing in different browsers should be a sanity check, not a significant amount of the development process. Both the CSS and SVG Working Groups have already identified this disconnect and it is the highest priority for their 2011 calendar year. This should expand across the Working Groups that have common, complementary and dependent functionality. There should be a specific end date for this process, preferably within the 2011 calendar year.

Simply stated, to give more functionality to end users and developers, there is first a need to enable a fast track to aligning features currently in demand. This, in turn, will expedite future innovation.

## ESTABLISH CROSS GROUP GUIDING PRINCIPLES

In order to be successful during a stabilization period and beyond, some cross-group guiding principles should be established. While guiding principles exist today, they are different across working groups, do not contain architectural direction and, most importantly, are not always agreed upon before moving forward.

The following are some principles required.

### 1. MAINTAIN STABILITY AND CONTINUE FUTURE INNOVATION

If developers cannot depend up on a stable cross-browser platform, they must create such dramatically different code paths that can almost double the development cost, often leading them away from standards and towards singular third-party solutions. If developers knew which features were stable and interoperable up front, they could choose to wait on new features, or develop separate content as they do today. But most importantly, it becomes their choice instead of an unpredictable, unplanned future burden and cost.

> **Proposal**: As part of an alignment milestone, an interoperability process should be introduced that deepens the testing quality of the specification; major browser vendors should participate in this process across Working Groups; and lastly, date-driven feature sets need to be agreed upon up front across Working Groups.

Improving interoperability was Microsoft's intent when submitting the CSS 2.1 tests. However, there were unintended consequences of slowing the Working Groups. Developing tests before or in lock step with new assertions written for each feature will increase interoperability during the process.

Lastly, by establishing a target date for specifications to stabilize, seams between functional areas and architectures will dramatically decrease in number.

### Future Proof – Innovation

While it is the primary job of the Working Groups to deliver specifications, vendors must be free to innovate and introduce new functionality. Methods already exist to allow for the introduction of new features without destabilizing the Web. However, because there is no process by which to stabilize portions of specifications, innovation is stifled today. This is why, for example, vendor prefixes were required on border-radius for so long. There was no process by which to stabilize a portion of that specification. Some vendors may be hesitant to implement too many vendor prefixes for fear of creating instant legacy.

### Backward Compatibility

It seems inevitable that because at least one of these specifications, SVG, has not changed significantly for over a decade, this process of stabilization will create backward incompatibles. The desire and goal should be that breakages be minimal, or optimally, none.

**High Level Use Case / Scenario**: As Web developers begin to adopt new or converging technologies, there should be as few as possible inconsistencies ported forward. This will lower the barrier to understanding, development, debugging, and adoption.

> ***Proposal*** : Do not sacrifice well established Web development processes for backward compatibility if there is a belief that the tolerance for adopting a different practice by developers is very low; features that should absorb breakages are those that have much less presence on the Internet as well as the intranet.

Always be mindful of backward compatibility, but not lead with the statement that it is an absolute. This should be paired with a single design model that eliminates incompatibilities in the future.

## 2. Modularize Specifications and Focus Efforts

It is not easy to communicate progress across concerned Working Groups. This was simpler when there were fewer specifications and more participants. However, that is no longer the case with the complexity of specifications relative to the number of participants. The recommendation below may appear to be contradictory; I assure you it is not.

**SVG** is a single monolithic specification. There have already been discussions on the need to modularize the specification for SVG 2.0 in order to address different use cases.

SVG as a technology will have two primary use cases: as a document/image format, and as vector graphics for the Web. The SVG Working Group has driven a lot of the features that are now being adopted by CSS (transitions, animations, gradients, filters, and transforms). Some or all of the members in SVG Working Group are recognizing that ownership of these modules for Web scenarios is being transitioned to the CSS Working Group.

The following are recommendations on what should be modularized for an alignment milestone.

SVG Integration: The purpose of SVG Integration is to define “SVG language into other markup languages”. This is extremely important for integration into HTML. SVG’s primary use case going forward is within an HTML document. This work is good, but still needs to be scoped.

SVG DOM: The SVG DOM is considered to be cumbersome and is not aligned with general DOM programming concepts. This specification attempts to address several issues. A more detailed proposal that scopes the problem set to programmability when working with HTML and SVG is being studied.

SVG Animation: Synchronized Multimedia Integration Language (SMIL), the animation technology for SVG, should be preserved for historic scenarios, whereas CSS should apply when the root document is HTML.

The **HTML** specification would also benefit from an alignment milestone. Modularity in the next specification enables rationalization more quickly and avoids fractured efforts that have to be repaired later. Progress was made in areas such as <canvas> but there are more opportunities. For example, rationalizing audio and video between SVG and HTML should be another, concerted Task Force and done in a separate module.

Both **CSS** and **WebApps** have already modularized their specifications. They enjoy the ability to push forward on any area as needed. The problem here, and it is a running theme, is lack of focus. With so much area to cover, specifications that are in higher demand must get the attention required to stabilize.

The above may seem to create more silos. On the contrary, by modularizing efforts, participants or area experts can make better use of their limited time, and there would be much less of a chance for overlapping and colliding functionality. There is a need to identify the overlapping functionality and form additional Task Forces for area experts to rationalize technologies by using the same guiding principles. The results of these Task Force efforts need to avoid interrupting current specification processes and avoid stifling innovation.

## 3. Drive Specifications with Scenarios, Use cases and Tests

More often than not, functionality in W3C specifications is not supported by scenarios or use cases. Scenarios and use cases are the best way to spot check the design along the process and avoid prolonged exploration of edge cases. By incorporating these into the specification writing process, the working groups will be more productive and more efficient. Simply stated, there is evidence of overdesign.

**Proposal**: Each major feature requires documenting scenarios and accompanying use cases.

These examples are very high, lightweight scenarios that demonstrate my position on this subject. Complex UML diagrams are not expected as these will slow up the process. Narrative forms are sufficient and tend to provide the means by which to check feature development against real world use.

Lastly, as indicated before, to provide more up-front interoperability, a test should be written for every new assertion. This would significantly decrease the amount of churn among implementations, as well as reduce the cost of writing or reviewing tests in hindsight. Tests should be submitted with each new feature going forward.

## Conclusion

Enough W3C Web Specifications and individual modules are already in or near Last Call. We need to identify the set of modules or functional areas that will benefit from an alignment and stabilization milestone without stifling innovation.

In order to move this process forward quickly, work should be scoped and driven by shared design principles such as driving resolution through scenarios and supporting assertions up front with test cases.

Modularizing and prioritizing specifications for a particular date across working groups, continuing to form functional task forces, distributing the editing ownership, and allowing individual sections of specifications to stabilize will reduce overall costs and increase innovation.