

Views on a paradigm

Ben Klemens

11 May 2009

[Part two of six]

A major contribution of the IT era has been to allow multiple views of the same work. Products are becoming less like a single book, which is a fixed, indivisible object, and more like a song, which is merely a brief view on something for which hundreds of other views exist.

Your favorite pop song was probably recorded onto 24 tracks, and then loaded onto a sound mixer such as Audacity, a program built around facilitating multiple views of the music. There's a visual representation of the sound and of course the noise the thing makes. The data is multilayered, and the user can view/hear the final work with some layers processed, muted, inverted, et cetera.

Users of the GIMP or Photoshop are familiar with the same process: there are all these layers, and each can be viewed differently, separated, filtered. At some point, you set a fixed view and publish it as the final image, but with both the visual and audio systems, the base version holds much more information than the final view.

Databases have what is literally called a view, but even without them, users are encouraged to think in terms of the root data existing in the database and what's on the screen being a slice of it. The root data needs to be taken care of, but mangle the view all you want; it's disposable. HTML is the markup language used by web pages; it is plain text but then viewed via a cute renderer like Internet Explorer or Firefox. Your file browser will let you look at the pile of bits on your hard drive sorted by name, type, or size, or a number of other slices.

Almost all information processing in all media takes the views-of-base-data form. But there are three hard-and-fast exceptions to the paradigm: spreadsheets, word processors, and presentation software. There is a picture of the page on the screen, and that's the document. There are few ways to view the work differently when you're working on it than when the final output will be printed or displayed on somebody else's screen. With due creativity, you can find the outline view or other marginal shifts [by which I mean you can choose whether the page margins appear on the screen or not], but for the most part, these systems work by vehemently insisting that there be only one view. All of the document's information is present in all versions.

And that is one more reason why Word is a terrible program: it constrains the user in the classical paradigm of "one work, one view". Can I distribute drastically different views of the same work to different people? They would just be two different works, that you'll have to maintain separately. When I present to the world congress, is there

a clean version that I can use? Nah, just hit <F9> to blow up your working copy to full-screen size.

This is clearly what many people wanted, and most are happy with it. Multiple views are overkill for simple documents. The two-bit philosophy questions of which is the true version of the work evaporate; the conceptual structure of a root object which is viewed in different ways flattens out; our documents are just like they were in the 70s, but backlit. But being stuck in the seventies means that there is a clear and evident ceiling in efficacy, because the ideal view for working on a project matches the output view in only the most simple and lucky of circumstances.

The view issue dovetails with the semantic markup issue from last time, because semantic markup by definition means working with markup and then producing beautiful output from that. For example, a bibliography editor or database gives a straight-information view of the bibliography, and then the system produces a different view of the data for the APA, Chicago, or Harvard stylebook. But if we have only a one work = one view paradigm, we can't work on content in one view and formatting in a separate view.

HTML gives us some hope for the future here, because the actual work is done in a markup language and the output has myriad possible views—and people have no problem with the concept. One person can read the work on his big-screen browser, one can read it on his telephone, and one can hear it on her text-to-speech reader, and all agree that they've read the same document.

Commenting A compounding failing of Word is that it won't let me insert the wealth of expletives that it inspires in me. With the one work = one view paradigm, there is nowhere for me to leave personal comments to myself that won't go into the public version. I often leave notes about sources, the full excerpt of a passage I edited down, or other side-notes about note-worthy bits. Without properly enforced private and public views, I'd need to either keep a side document or just throw out this info.

Computer programmers have a trick known as 'commenting out'. Because the computer ignores anything marked as a comment, a coder can mark functioning lines of code as a comment to see how the program would run if that line were eliminated. It's a sort of purgatory for code that should maybe be deleted, but the judgement hasn't yet been handed down. Similarly, I write more than enough prose that is maybe a bit too verbose to put in the final work. I am reluctant to delete it, because I spent ten minutes composing that paragraph, but commenting it out is painless and reversible.

In Word, I am unable to leave personal notes to to guide myself, and I am unable to comment out sections that should probably be deleted. That is, Word gives me fewer tools to write with so that it can enforce its intuitive paradigm.