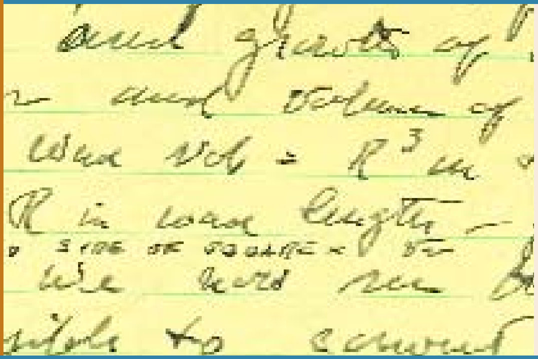
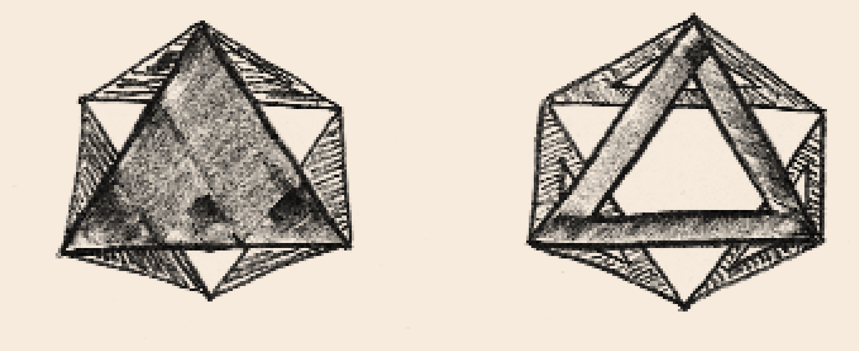
ROBERT STEIN AND DANIEL VISEL
UNIVERSITY OF SOUTHERN CALIFORNIA





Sophie and the Future of Reading and Writing

In the 1440s, Johannes Gutenberg invented a printing press that enabled the mass production of books. In the ensuing five and half centuries, books—printed pages bound together—have served as the central mechanism for storing and transmitting knowledge. But their fundamental nature as static, finished products has changed little since Gutenberg's time. Today, however, the traditional roles of authors and their readers may indeed shift as electronic, screen-based technologies allowing the creation of interactive, networked books are developed and refined. Robert Stein and Daniel Visel of the Institute for the Future of the Book, affiliated with the Annenberg Center for Communication at the University of Southern California, describe the Sophie platform, an all-purpose, user-friendly tool that gives authors control of the form of their work by allowing them to use multimedia to convey their content—without having to rely on expert programmers for support. Stein and Visel believe that Sophie has the potential to blur the lines between reading and writing in ways that printed books simply cannot.

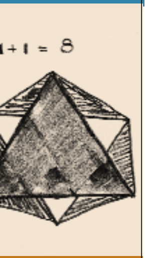
NOTEBOOK

The fundamental nature of books as static, finished products has changed little in more than 500 years.

Sophie will blur the lines between reading and writing by expanding the capabilities of printed books. It is an all-purpose tool that will allow users to easily create books with any sort of media on hand.

It is a mistake to think that a book is merely a container for text; a well-designed book presents text in a useful way. Similarly, Sophie is more than just a wrapper for media; it will allow users to structure information intelligently.

While Sophie can be used in many settings, it is aimed squarely at the world of education.



What's Wrong With Electronic Books Today

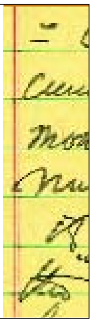
Although conveyed via a bewildering plethora of formats, just two basic models of electronic books exist today: the PDF-based model and browser-based books. While both are useful in certain cases, each has significant inherent drawbacks that inscribe themselves indelibly on work based on these formats. Primarily, both betray a paucity of imagination in that they treat electronic books essentially as texts—yet a text is only part of what a book can be.

PDFs are what many people think of when they think about electronic books. The appeal of PDFs is that every printer or computer that understands how to decode a PDF document will print or display the document in exactly the same way. This has made producing books as PDFs the path of least resistance. While it is possible to put forms, buttons, and links into PDFs, as well as audio and video, only advanced users can do so. Creating a PDF remains very similar to printing a document: the printout is static—a snapshot of one instant in a document's life.

The second common model of electronic books is based on a markup language such as HTML or XML—in effect, a specialized use of the technology underlying most of the Web. Content created in this way is more fluid than a PDF: in many browsers, for example, you can change the font and its size, or change how many words there are in a line of text. No two readers necessarily see the same thing. Programmers can transform documents made in XML so that they can be used in other ways. The design philosophy behind the markup languages has a major conceptual flaw, however, in that it assumes that all documents can be turned into an outline—markup languages work by taking all the text in a document as a single string of letters and chopping it into differently formatted pieces. This is very natural to computers, but it is not natural to people, and while it does work with many documents, it is a constrained approach that limits what can be done with a book. For example, HTML text-based books that contain slightly unordinary content, such as math books with formulas in them, tend to fall apart on the Web.

Another drawback of digital work is simply that it does not seem to last as long as print-based work. Some print books have lasted half a millennium; even the most poorly printed book should still be readable in half a century. The same cannot be said of digital files. Media change: the 5.25-inch floppy disks used to save files 25 years ago are entirely useless now, and it's increasingly difficult even to find a drive that can read 3.5-inch disks. Formats change too: presentations designed to run on the operating systems of 10 years ago might not work today. Even in the world of the Internet, untethered from the confines of hardware, links break and code goes bad.

While reading a book seems like a solitary activity on the surface, underneath it has always been a quietly social experience, a communication between the author and the reader across time and space. Sophie will make these metaphors concrete.



Sophie

Sophie will blur the lines between reading and writing by expanding the capabilities of print books. It is an all-purpose tool that will allow users to easily create books[1] that can contain any sort of media on hand—text, images, sound, video, animations, and so on. Sophie does for media what a physical book does for text and images: one could think of it as a wrapper for anything digital, but it is more than that. Sophie allows authors to make complex multimedia books without specialized training.

Our aim in creating Sophie is to have books present a *reading environment* that allows scholars to share and develop knowledge in a dynamic space. It is true that millions of people read on the Web every day. But no one would claim that the Web is ideal as a reading environment (to say nothing of it as a writing environment). At this point, it merely serves both purposes well enough to get by. While the various technologies available on the Web afford new opportunities and forms for writing—consider the blog—no good environment exists for easily creating unified works that take advantage of the multimedia possibilities of modern computers. Today, if an author wants to mix media forms to convey ideas on the Web, he or she likely will need to hire a programmer.

For example, if I'm reading an electronic book in Sophie and find something that reminds me of a Web page or a video clip or a podcast, I can attach a link to it. I can pass this annotated book on to a friend; she could read the book, link to my annotations, attach her own, and pass it on. While reading a book seems like a solitary activity on the surface, underneath it has always been a quietly social experience, a communication between the author and the reader across time and space. Sophie will make these metaphors concrete.

While Sophie can be used in many settings, it is aimed squarely at the world of education. A platform is needed to teach composition—not just in writing but also in terms of how to use multimedia effectively, an increasingly necessary skill in today's world. Programs currently available simply are not up to the job. More and more students are taught to make presentations with PowerPoint, a limiting program that, as Edward Tufte has pointed out, encourages gimmicky special effects at the expense of coherent thinking. Students often spend more time working out flashy transitions between slides than writing the report the slides are intended to accompany. No one gains anything from that.

Sophie will make it possible to integrate the writing of a report with the slide show that previously would have accompanied it. Sophie treats all media equally: if adding a slide show would be helpful to a primarily written report, the student can add the slide show to the page it is intended to illustrate without having to switch from Word to PowerPoint.

It is a mistake to think that a book is merely a container for text; a well-designed book presents text in a useful way. Similarly, Sophie is more than just a wrapper for media; it will allow users to structure information intelligently. Sophie differs from previous platforms for electronic reading and writing by giving authors as much control over the *form* of what they are creating as they have over its content.

Much of what Sophie will enable users to create is possible with existing technologies. The problem, however, is that working with existing technologies can be enormously difficult—so much so that in many cases a staff of programmers or designers is needed to do the job. As a result, media production tends to be the province of corporations, which generally have the money and the staff to pursue interesting media projects. Software has traditionally played to this dichotomy between authors and designers, with few authors mastering the skills needed to design. Sophie's aim is to democratize the world of multimedia by making it possible for individuals and small nonprofits to express themselves via compelling multimedia books.

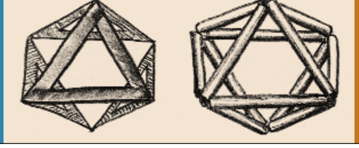
Finally, Sophie addresses the problem of obsolescence because it is based on Squeak, a platform-independent environment that uses the Smalltalk programming language and is not tied to any particular device. Sophie books will open in exactly the same way across any platform that runs Squeak—be that a Mac, a PC, a Linux box, or some yet-to-be-invented E-book reader. If an entirely different platform or operating system is released in 50 years, all Sophie books will still run in exactly the same way, provided that a Squeak virtual machine can be built for it—a far simpler task than completely redoing books to fit new platforms.

Multimedia and Interactivity

A physical book has pages. We take it for granted that pages follow pages in a sequential order, in the same way that spoken words follow each other. Certainly Sophie can be used to make books that way—it's the default setting. You might think of a book as having a spine going through it, which pages hang off. Sophie allows us to do away with the spine—or to have many spines, because different models of book construction can be intertwined. A book constructed more as a canvas on which to create could present information in several different ways and not convey judgment about which way is more important. In a print book that is mostly text, photos or graphics tend to function simply as illustrations of the text. In a print book that is mostly photos, the text sections function as captions and as such they are secondary information. Sophie doesn't force such implicit ordering; a Sophie book can have a more fluid relationship with pages.

Sophie is media-agnostic: it treats all media the same. Sophie could be used to make a book based around a long piece of text (like a traditional novel) or based around a series of photographs (similar to a slide show), adding narration or a soundtrack to play with the rhythm. Or one could produce a video-based book, or a book based around a single photograph, annotated with audio to tell a story. Or one could mix any of these (and more) forms together to create something entirely new.

Imagine, for example, a book based around a video. You are a drama teacher and your class has just rehearsed a play, which you videotaped. You put the videotape into Sophie, and, using the script, you add subtitles while the actors are saying their lines. Or you could record a commentary track about what's going on in the play; the readers/viewers of the book could switch back and forth between the recorded sound and your soundtrack—or overlay them if they wish. You could turn blocking notes into cues that show over the video. Then you videotape another rehearsal. You could add the new video to your Sophie book; you could play the two videos side by side



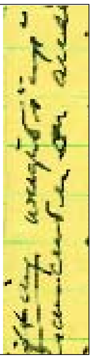
if you thought that might be useful. If a scene in one is four minutes long and a scene in the other is five minutes long, you could pause the first for a minute at the end of the scene so that the two videos don't get out of synch.

Or imagine a copy of *Madame Bovary* in French. When you move your mouse over a sentence, it might be spoken aloud in English. The same book could be set up so that it reads itself aloud, turning pages as it goes so that a student learning French can follow along.

Sophie and the Network

A book is a frozen, finished product. Multimedia books created in Sophie could be the same—except that they will live in a networked environment. It will be possible, for example, to link Sophie books to other Sophie books. All can be connected using a Web browser. In this case, the Sophie application will work as a plug-in to the browser, just as Flash does today. Sophie books, which will reside on a remote dedicated Sophie server, will open inside your browser, but changes that you make—the same sorts of changes you could make to a book on your computer—can be made on the book on the Sophie server if you choose to do so, and thus can be seen by other people accessing that book.

Imagine a copy of *Madame Bovary* in French. When you move your mouse over a sentence, it might be spoken aloud in English. The same book could be set up so that it reads itself aloud, turning pages as it goes so that a student learning French can follow along.



For example, a teacher might post a book for her class to collectively edit; they could all log in to it and make changes. Or she could assign a paper to be written in Sophie. The teacher and all the students could easily read all the papers and attach their own comments to them, which the original author could view and learn from. Because Sophie will function in a networked environment, the resources used inside Sophie books—streaming video, RSS feeds, Google maps, information from databases, and so on—will be remotely accessible as well.

Conclusion

Most users needn't think about Sophie's underlying architecture, but that architecture reflects Sophie's philosophy in important ways. Sophie is being designed so as to be fully extensible, which will ensure that Sophie books will be readable far into the future. Sophie will be written in Smalltalk, a fully object-oriented programming language, which means it will be easy to attach new functionality to Sophie via plug-ins. Sophie will be released with its own API, or application programming interface, which allows one Web application to talk to another. That is what will enable Sophie books to incorporate Internet services, for example, Google maps or content from a blog—or a blog could contain live, updated information from a Sophie book.

The Sophie server will provide a repository for all Sophie books that exist on a given network and will allow users to search Sophie books already created, as well as publish new books on it. The repository will also serve as a rich source of reusable content. Finally, Sophie will be open-source software: anyone will be free to take Sophie's source code and reassemble it as he or she chooses to suit the purpose at hand.

Clearly, Sophie is not just for creating and reading books. Rather, it presents an environment that vastly expands the boundaries of books by taking full advantage of their place in the ecosystem of information. These fully networked, organic books of the future, profoundly different from the books we have known for centuries, will in turn help expand the boundaries of human knowledge.

ROBERT STEIN is director of the Institute for the Future of the Book. He can be reached at caxton@earthlink.net.

DANIEL VISEL is a researcher at the Institute for the Future of the Book. He can be reached at dan@futureofthebook.org.

[1] For now, we use the word "book" broadly, even metaphorically, to talk about what has come before and what might come next.