Sophie 2.0 - a platform for reading and writing of interactive multimedia books in a networked environment

Kalin Georgiev[1]; Miloslav Sredkov[2]

1 Faculty of Mathematics and Informatics, Sofia University,
5 J. Boucher str., Sofia, Bulgaria
kalin@fmi.uni-sofia.bg
2 Astea Solutios AD, 20 Lozentz str., Sofia, Bulgaria
milo@asteasolutions.com

Abstract

Sophie is software for reading and writing networked multimedia electronic books. It has been designed to let those without professional design skills create and distribute complex documents that take full advantage of new media and the Internet. Sophie brings together all of the pieces of media-rich writing. In addition, Sophie fosters collaboration, allows instant reader feedback, and encourages interactivity. Sophie lets users create communities around projects; with Sophie, "books" become "places" where people meet. In addition to its powerful capabilities for combining various media formats and interactivity, Sophie Server, a significant part of the Sophie platform, allows authors to collaborate – working on the same content simultaneously in real time or offline, and later integrate their changes with the work of others when an Internet connection becomes available. Sophie also offers integrated reader communication capabilities allowing readers to ask questions and comment on specific sections of the book.

Keywords: e-book; e-publishing; collaboration; Sophie; rich media;

1. Introduction

The digital, networked nature of the World Wide Web provides significant opportunities for the dissemination of electronic content, opportunities not available for conventional printed content. There are several widely used technologies that today facilitate the delivery of media intensive, interactive content over the Web. These technologies include Flash, HTML, and PDF [1].

Collectively, current digital content 'carriers' offer the following features, providing the fundamental advantages that are propelling the acceptance of electronic content:

- Support for various asset types (text, images, audio, video, and others) [2]
- Support for interactive features and scripting
- Advanced layout of content organized as building blocks

The widespread availability and features of existing software tools for writing and publishing is also a major factor for many authors. Every modern authoring tool provides at least some of the following advantages:

- Ability to author, manage, and publish rich multimedia books without prior technical training (ease of use)
- Availability of facilities for the publishing and review process
- Support for author collaboration and reader feedback [3]
- Low price (or free)

There are certainly other criteria that make one authoring tool preferable over another. However, we have limited our list to the set of factors that the project team has identified as those authors take into consideration when choosing an authoring tool.

Each technology taken alone will provide many of the desired capabilities, but none of them provides all of them in a single package. For example, Flash supports various asset types and text flow (using the recently developed Text Layout Framework [4]). However, it requires technical training. It is not intended to make content *per se*, but rather animations and applications. There are many easy to use authoring tools for HTML, and it does support video assets [5]**.** However, advanced text flows around other assets are not supported, nor are some matrix transformations over media (rotation of images, for example).

We will not continue comparing existing technologies that enable authoring of electronic content since a detailed list is not the purpose of this paper. Rather, it is to make several simple points:

- There are too many authoring tools and the choice authors have to make is not trivial. Often, by choosing a particular technology, authors sacrifice the ability to make use of some of the potential capabilities of electronic content because of technology limitations
- There are too many competing concepts in the world of authoring electronic content. For example, there is a conceptual abyss between making Flash content and making HTML 5 content
- Creating quality interactivity elements in electronic content requires professional experience, tools and training

- Integration of multiple sources of content and reuse of content created with different technologies is complicated and often impossible
- Free and easy to use tools for building advanced, rich content are rare
- Publishing, author collaboration, and reader feedback facilities are often beyond the reach of technically untrained authors.

2. What is Sophie?

Sophie is a software package for writing and reading interactive books in a network environment. Its aim is to support authors of all levels in the world of electronic content by addressing the challenges discussed above. In addition, Sophie makes several new conceptual approaches possible for creating interactive content without prior technical training. Sophie also provides a platform for real time author collaboration and gathering reader feedback.

3. Static Sophie Content

Sophie content incorporates a wide variety of asset types such as text, images, sound, videos, and more. Sophie goes even further by enabling the incorporation of PDF documents, Flash, and, more exotically, HTML content. In addition to reusing existing documents and assets in other formats, Sophie books are embeddable one into the other, allowing the authoring process to be decomposed by creating smaller, reusable content components.

Text is of special importance for Sophie as Sophie is software for creating books. In addition to most of the text styling features authors have come to expect in rich text editors, Sophie provides dynamic text flow. Text wraps around other shapes in the document (images, videos, and other text areas, for instance) and is flowed into a sequence of independent rectangular shapes called *a text chain*. Manual chaining of text blocks allows authors to build various types of standard layouts like the simple book layout, or the multi-column layout with images that is the standard for newspapers. Turning the automatic chaining option on makes Sophie automatically generate new pages whenever typed or pasted text exceeds the limits of the manually created text chain.

Figure 1: Halo buttons and HUDs conceptualize available settings and tools

Supporting a wide variety of asset formats is a significant advantage; however, in most software tools, this often leads to bloating of the user interface. Many different asset types usually mean many different tools and properties, accessible through many different palettes, dialog boxes, and inspectors. Sophie addresses these challenges by proposing an innovative universal approach to the user interface for manipulating any content type. Sophie introduces halo buttons and HUDs (Head-up Displays) which contextualize the access to available operations over the specific assets.

4. Dynamic Features for Sophie Content

Content displayed on a computer not only allows incorporation of predefined dynamic behavior (such as effects, transitions, and time-based behavior), but also enables interactivity. Sophie's timeline feature allows authors to synchronize dynamically changing properties of page elements in a Sophie book with time. Effects such as "start the video at the n-th second, while at the same time making the page title turn white" are achieved by simply clicking the desired moment on the timeline and setting the attributes of the elements

to be changed. Further, the time line allows synchronization of audio, video, images, and transcripts through the same intuitive interface.

Sophie provides a set of triggers that fire when a certain event occurs within page elements, when an element is clicked, for example. Authors can associate triggers with an action from a predefined set, such as hide or show an asset, play or stop a video. Advanced links can be built by defining dynamic behavior of content elements to be triggered by user input. Possible actions include changing the current page to any other page, which makes possible the implementation of non linear reading.

Sophie's scripting system lets authors write JavaScript code that accesses the Book's DOM model to implement complex interactivity or batch operations.

5. Publishing with Sophie

No matter how impressive the content itself is, publishing and distribution of interactive, rich media content is commonly problematic to authors. Electronic content is, for example, often published on web sites in PDF format.

Sophie 2.0 consists of three building blocks - the Sophie Author, the Sophie Reader and the Sophie Server. Sophie Author seamlessly connects to Sophie Server, allowing authors to publish their books on the Server with a single click. Books become accessible to readers through the Sophie Server's web interface. Readers read books using Sophie Reader. Sophie Reader is capable of running in a web page (as an applet), which allows Readers to read without installing any software on their computers and authors to integrate Sophie books into web pages. Additionally, Sophie Server supports the full history of book editions.

Sophie introduces the concept of 'book extras', additions to the main book content. An example of a book extra is the use of Sophie annotations that allow authors and readers to create additional notes to book sections, paragraphs, or individual assets. The novel aspect of Sophie's book extras is that they can be distributed separately from the book. In this way there may be several differing sets of book annotations made available to different target groups of readers.

6. Collaboration and Feedback

Sophie Server provides real time collaboration features for authors connected to the server. If multiple authors are connected to Sophie Server and working on the same book, changes made by any author are reflected to the local copies of all other authors. Authors will also be able to work offline, without a connection to Sophie Server, and their work will be automatically integrated with the work of others when a connection becomes available.

Through the use of a Sophie Feature called "comment frames", authors are capable of requesting live feedback from readers. When reading a book, readers are able to input their comments or suggestions in the comment frames making their input available to other readers and the author in real time. Comment frames can be associated with specific book segments, which allow focusing on gathered feedback on specific parts of the book.

7. Sophie as a Platform

In addition to being open source, Sophie 2.0 is being developed with extensibility in mind. Java is chosen for implementing most of Sophie 2.0 because of its high popularity and rich availability of tools. Sophie 2.0 runs on Java SE 5 and higher. Modularization support is built on top of OSGi. Each of the products (Author, Reader, and Server) is implemented as subsets of the modules of the platform. Extension points for additional content types, additional user interface elements, and others are defined to allow extra features to be added without the need of modifying the existing source code. Third party module developers can take advantage of the underlying implementations for resources, text layout, graphic scenes, collaboration and other fundamental Sophie libraries. The file formats are open, documented and XML based. The source code is documented and follows strict conventions.

8. Applications and target users

Sophie 2.0 has been developed by academics and focused at academia. Sophie is being evaluated by professors creating content to be used for their classes and other academic activities. However, the broad set of capabilities Sophie provides opens it to various other target groups, such as self publishing authors, traditional publishers, artists, designers, and others. Sophie is under

continuous development, and new features and improvements are being introduced that enhance Sophie's publishing capabilities as well as its content authoring features.

Acknowledgments

Sophie 2.0 is a project developed by the University of Southern California's School of Multimedia Literacy and Astea Solutions AD of Sofia, Bulgaria, under a grant from The Andrew W. Mellon Foundation. Sophie 2.0 is distributed under the Educational Community License (ECL). The project is a continuation of the work on Sophie 1.0, initiated in 2006 by Robert Stein, founder of the Institute for the Future of the Book. Software development has been done in close collaboration with the Laboratory of Interactive Multimedia at the Faculty of Mathematics and Informatics of Sofia University.

This report is supported by the SmartBook project, subsidized by the Bulgarian National Science Fund, under Grant D002-111/15.12.2008 and the EC FP7 SISTER Project, Grant Agreement Number 205030.

Notes and References

[1] WIKIPEDIA. *Comparison of e-book formats*. Available at http://en.wikipedia.org/wiki/Comparison_of_e-book_formats

[2] WIKIPEDIA. *Multimedia*. Available at. http://en.wikipedia.org/wiki/Multimedia

[3] NEW MEDIA CONSORTINUM AND THE EDUCAUSE LEARNING INTIATIVE. *The 2010 Horizon Report*. Available at http://wp.nmc.org/horizon2010/

[4] ADOBE LABS. *The Text Layout Framework*. Available at http://labs.adobe.com/technologies/textlayout/

[5] W3C. *HTML 5 reference*. Available at http://dev.w3.org/html5/html-author/