A quick introduction to the future of reading

Sophie

SOPHIE

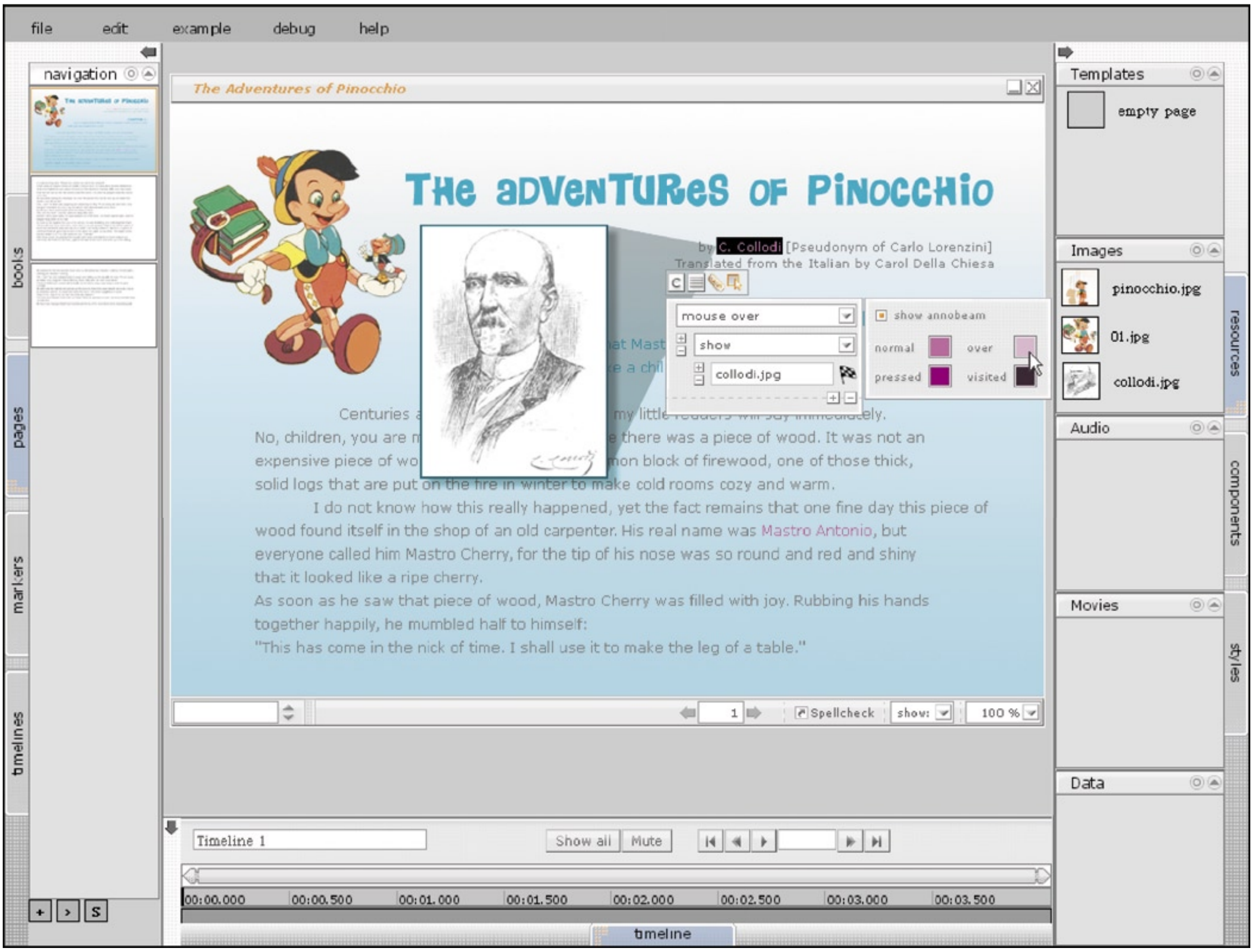
B. Eckardt, J. McIntosh, S. Riggins, T. Rowledge, M. Rueger, B. Stein
The Institute for the Future of the Book, Impara, funded by the Mellon and MacArthur Foundations and affiliated with the University of Southern California
http://sophieproject.org

NEW STRUCTURES FOR BOOKS

Sophie is an all purpose tool for dealing with media. It allows users to easily create books that can contain any sort of media on hand – text, images, sounds, videos, animations. Sophie does for media what a physical book does for text and images: with Sophie, authors can create multimedia books. You might think of it as a wrapper for anything digital, but it's more than that. Sophie differs from previous platforms for electronic reading by giving the author as much control over the form of what they're making as the content.

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

You might object that it's possible to make something vaguely like this in HTML. Sure. But you're going to have to spend a lot of time concentrating on the structure of the website you're creating: do we really need to think about information architecture when we're trying to make things? Sophie makes this seamless.



THE ADVENTURES OF PINOCCHIO

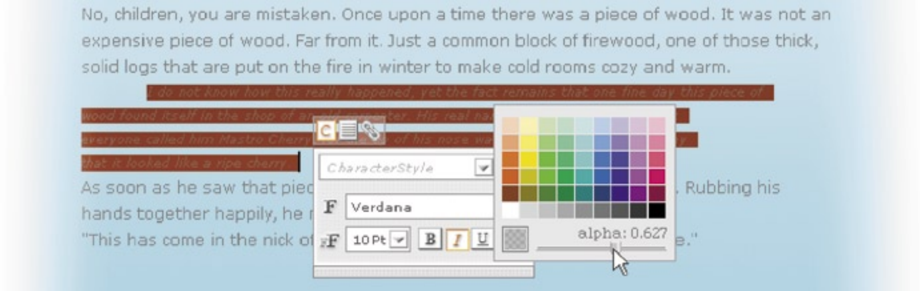
by C. Collodi [Pseudonym of Carlo Lorenzini]
Translated from the Italian by Carol Della Chiesa

at Mast
e a chil

Centuries a my little ely.
No, children, you are n there was a piece of wood. It was not an expensive piece of wo mon block of firewood, one of those thick, solid logs that are put on the fire in winter to make cold rooms cozy and warm.

I do not know how this really happened, yet the fact remains that one fine day this piece of wood found itself in the shop of an old carpenter. His real name was Mastro Antonio, but everyone called him Mastro Cherry, for the tip of his nose was so round and red and shiny that it looked like a ripe cherry.

As soon as he saw that piece of wood, Mastro Cherry was filled with joy. Rubbing his hands together happily, he mumbled half to himself:
"This has come in the nick of time. I shall use it to make the leg of a table."



No, children, you are mistaken. Once upon a time there was a piece of wood. It was not an expensive piece of wood. Far from it. Just a common block of firewood, one of those thick, solid logs that are put on the fire in winter to make cold rooms cozy and warm.

I do not know how this really happened, yet the fact remains that one fine day this piece of
wood found itself in the shop of an old carpenter. His real na
everyone called him Mastro Cherry … *of his nose wa*
that it looked like a ripe cherry

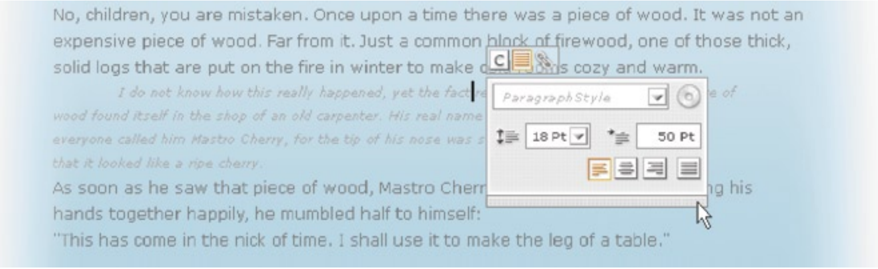
As soon as he saw that piec … Rubbing his
hands together happily, he r
"This has come in the nick of … e."

USER INTERFACE

The user interface design is based on a few simple assumptions:

the most often used functions and views should never be more than a mouse click away minimal use of modifier keys no modes, but always immediate actions combined with undo efficient, semi-automatic management of screen real estate. By following these rules we have completely avoided contextual menus and minimized the number of items in pull down menus.

When the user stops typing or selects a piece of text or a frame, halo buttons appear automatically. These buttons provide access to editors for style, trigger, linking, etc. Clicking on a halo button opens up a HUD (head up display) close to the mouse cursor.



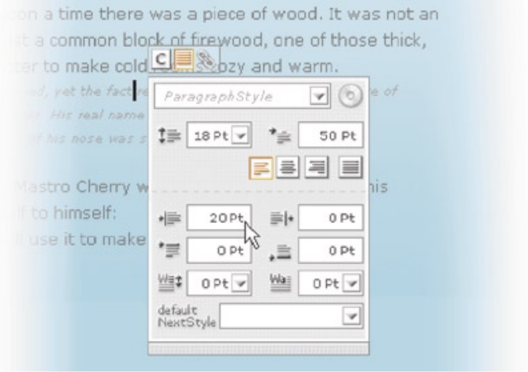
No, children, you are mistaken. Once upon a time there was a piece of wood. It was not an expensive piece of wood. Far from it. Just a common block of firewood, one of those thick, solid logs that are put on the fire in winter to make c … s cozy and warm.

I do not know how this really happened, yet the fact re … *e of*
wood found itself in the shop of an old carpenter. His real name
everyone called him Mastro Cherry, for the tip of his nose was s
that it looked like a ripe cherry.

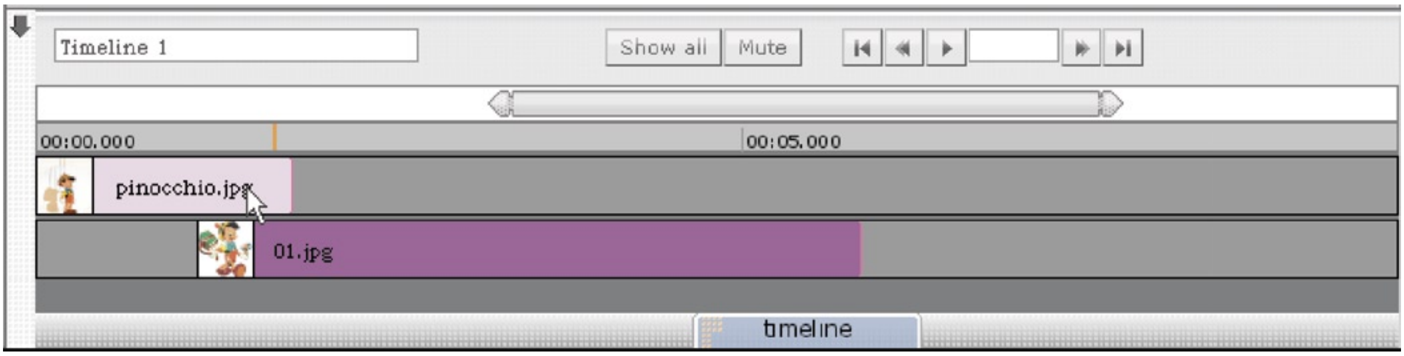
As soon as he saw that piece of wood, Mastro Cherr … ng his
hands together happily, he mumbled half to himself:
"This has come in the nick of time. I shall use it to make the leg of a table."

The HUDs can have multiple levels, presenting the most often used functions on the top one.

None of these editors has the usual moded OK/Cancel, but changes are applied immediately. Everything can be undone, the user basically works in a live preview environment. Views that need to remain visible are kept in tabbed flaps which can be easily collapsed and expanded to provide efficient management of the screen real estate.



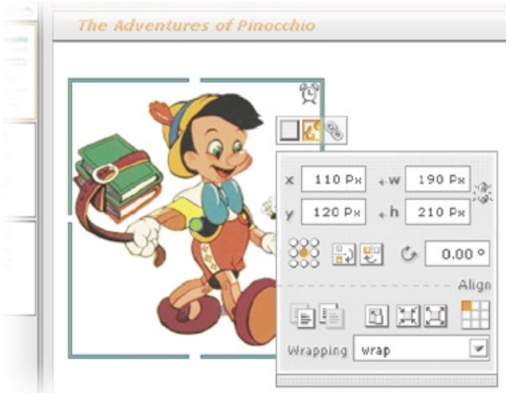
on a time there was a piece of wood. It was not an
t a common block of firewood, one of those thick,
er to make cold … ozy and warm.



ARCHITECTURE

Sophie is built around a relatively small architectural core (Sophie Common Core Architecture). It guarantees that all documents created by various means and in different environments are compatible and portable between these environments. The application (UI) is built based on the Sophie Application Architecture. Every (UI) function (even basic things like editing text) comes into Sophie as part of a so-called extension. While preserving compatibility on the document level, different combinations of extensions can provide very different applications built around the architectural core. We hope to avoid feature creep by allowing a flexible composition of extensions suiting a user's needs. The UI is built based on XUL and CSS plus the necessary support code in extensions. Existing Sophie applications can not only be themed by different CSS files, but also redesigned to a certain extent by supplying different XUL files.

The main linear content – the flowing text and embedded illustrations etc. - of Sophie books is kept in a tree structure. Each item, or node, in the tree represents either a structural component or a raw content component. Using a tree avoids the pitfalls of tag based representations (such as HTML) with the risk of dangling open or close tags, mixed up nesting etc.after editing. It also a very efficient way to represent and manipulate large amounts of content.



x 110 Px
w 190 Px
y 120 Px
h 210 Px
0.00 °
Wrapping wrap

Templates and frames define the layout and structure of book.
A hierarchy of compositors, page, frame, text, composites the content from the content trees into the template and frames layout of a page.
Display pages finally are the pre-stage for actually rendering a page. They can be viewed as device independent display lists, supporting optimized display updates.
The Resource Manager is responsible for storing, copying, updating, or altering the book data. By consistent use of URIs instead of the usual mix of filenames and URLs, the XML based pluggable storage can be local via mulitple files in a folder, or a zip based file, or remote on the internet, or interfaced to a database.
Timelines allow an author to synchronize events to time, media and pages. The author is able to arrange elements, pictures, sounds, video and slideshows along a timeline and then trigger the timeline from any marker in the system. Individual media elements will show and hide as they come into and leave scope along the timeline. Events can also be generated from user interactions. Triggers like mouse up/down can be attached to frames or marked areas of text so clicking on a certain word can open another window, start a timeline or movie, etc.

OVERVIEW INTERNALS

- cTwip - all sizes in Sophie are based on cTwips
= 1/127 twip (twip = 1/20 point = 1/1440 inch), = (1/7200) mm (integer conversion to metric and imperial)
- Rome, Freetype - Rendering based on Rome (Squeak Cairo) and Freetype fonts
- XUL, CSS - UI elements are defined in XUL files and skinned with CSS
- Storage, versioning - XML storage, everything is versioned
- system clipboard - Access to platform clipboard formats allowing to copy/paste RTF, images etc
- Quicktime - Integration of Quicktime supporting a wide range of media
- file locations - Full integration into the platform file system, use of tmp, user and application directories
- portable packaging - The same directory structure can be used on all supported plattforms. Appears as a bundle (single file) on Mac OS