Designing for Children
- With focus on ‘Play + Learn’

Interactive Ebooks

Experiments on the OLPC XO-1 Book-reading System

Sayamindu Dasgupta, One Laptop Per Child, sayamindu@laptop.org

Abstract: This paper describes the ongoing work on the book-reading system for the One Laptop Per Child (OLPC) XO laptops. It explores the potential of electronic-books to offer an entirely new level of interactivity by incorporating not only multimedia elements, but also interactive programming “shells” and programmable “microworlds” into a hitherto static medium. While the general focus is on features and possibilities that are rarely explored by the current market forces in the ebook world, the paper also describes the various technical engineering aspects of the book-reader project, illustrating the lessons learnt and pitfalls encountered while designing and implementing the system for deployment in a varied set of use-cases and scenarios.

Key words: ebooks, epublishing, one laptop per child, XO, sugar, reading

1. Introduction

The past year has witnessed an explosion in the number of electronic book (ebook) reader devices being released and sold all over the world (Amazon Kindle, Sony Reader, Barnes & Noble Nook etc). Even more general-purpose devices have not been spared from the ebook wave, a phenomena illustrated by the fact that ebook related software is the current (as of November 2009) dominant category in the iPhone App Store. However, with almost all mainstream ebook readers (and ebooks), the dominating trend seems to be an attempt to mimic the traditional ink and paper book as closely as possible. This is not unexpected, and to many, this may seem like a repetition of a process which took place almost 500 years ago, when Johannes Gutenberg and his contemporaries tried to meticulously mimic the older handwritten book format with their new movable type technology, with painstakingly hand-drawn illuminated decorations, etc. This trend of mimicking the old, along with certain limitations of current mainstream ebook-reader hardware (lack of color screens, comparatively slow UI response times, etc) in some ways remove the focus from the new possibilities offered by the digital medium. At One Laptop Per Child and Sugarlabs[1], while working on implementing a book-reader system, we have been exploring these new possibilities in parallel with our usual engineering efforts. The initial iterations of our work have already reached more than a million children in 31 countries all over the world. This paper is a description of our efforts, as well as a discussion of the technical challenges involved.

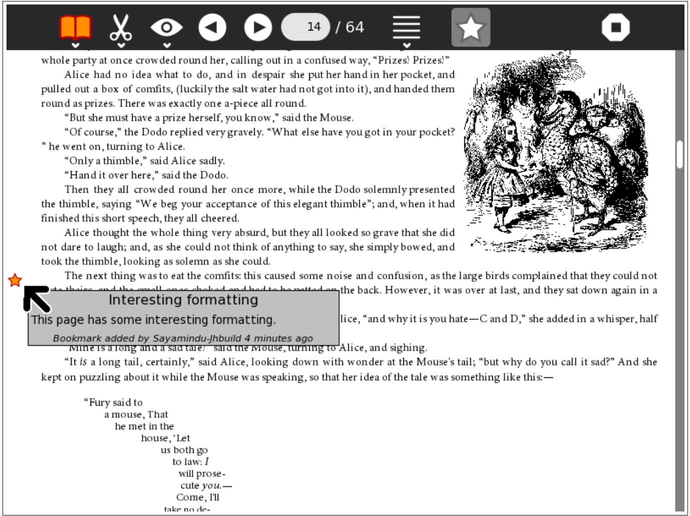
2. New Possibilities with Ebooks

Before going into an enumeration of the largely unexplored possibilities the new ebook medium may present, it may be useful to provide a short overview of the core technology involved, especially a justification of the ebook file format choice.

The file format for ebooks which is fast becoming *the* standard (major players like Sony, Barnes & Noble, Google, etc have adopted it already) is EPUB. It is a combination of proven technologies and standards which include

- Open Publication Structure v2.0, which uses XHTML or DTBook (a specification for accessible books) and a subset of CSS for presenting the actual content
- Open Packaging Format v2.0 for describing the various content items, metadata, as well as describing navigational rules within the ebook
- OEBPS Container Format (OCF) v1.0 which specifies how all the components are organized and packaged together in a Zip file

For most EPUB files, XHTML is the format in which the actual content is stored, and we have tried to leverage the flexibility of XHTML as much as possible in our work. It must be noted however that in some of the cases, we have used HTML5 instead of XHTML, and we hope that the EPUB standard will evolve to support HTML5 in the near future.



whole party at once crowded round her, calling out in a confused way, “Prizes! Prizes!”

Alice had no idea what to do, and in despair she put her hand in her pocket, and pulled out a box of comfits, (luckily the salt water had not got into it), and handed them round as prizes. There was exactly one a-piece all round.

“But she must have a prize herself, you know,” said the Mouse.

“Of course,” the Dodo replied very gravely. “What else have you got in your pocket? ” he went on, turning to Alice.

“Only a thimble,” said Alice sadly.

“Hand it over here,” said the Dodo.

Then they all crowded round her once more, while the Dodo solemnly presented the thimble, saying “We beg your acceptance of this elegant thimble”; and, when it had finished this short speech, they all cheered.

Alice thought the whole thing very absurd, but they all looked so grave that she did not dare to laugh; and, as she could not think of anything to say, she simply bowed, and took the thimble, looking as solemn as she could.

The next thing was to eat the comfits: this caused some noise and confusion, as the large birds complained that they could not

the back. However, it was over at last, and they sat down again in a

lice, “and why it is you hate—C and D,” she added in a whisper, half

Interesting formatting

This page has some interesting formatting.

Bookmark added by Sayamindu-Jhbuild 4 minutes ago

Alice, and sighing.

“It *is* a long tail, certainly,” said Alice, looking down with wonder at the Mouse's tail; “but why do you call it sad?” And she kept on puzzling about it while the Mouse was speaking, so that her idea of the tale was something like this:—

“Fury said to
a mouse, That
he met in the
house, ‘Let
us both go
to law: *I*
will prose-
cute *you*.—
Come, I'll
take no de-

Figure.1 The OLPC/Sugar ebook reader interface

[1] Sugarlabs is a volunteer-driven non-profit organization developing the Sugar Learning Platform, the software environment that is shipped with the majority of OLPC laptops

When we mention our attempts to make EPUB *the* standard format for the OLPC/Sugarlabs book-reading system, a question that often comes up is, why not use PDF instead? While it is true that PDF is an established (and open) standard, two major concerns make EPUB more suitable for us. The first concern is about file-size – a typical PDF usually takes up more space than an equivalent EPUB book. For example, *Abroad*, a much downloaded image-rich ebook from the collection of the Internet Archive is around 11 MB in its PDF form, and 2.5 MB as an EPUB file. The second major advantage of EPUB is that it is reflowable, which means that an EPUB formatted document can easily be displayed at a comfortable zoom-level in displays of varying sizes, without the need for horizontal scrolling.

Our efforts (apart from the engineering work that goes into a generic "standard" ebook reader) have been largely concerned with making ebooks as dynamic and interactive as possible, so the reading experience can engage the reader in ways that would not have been possible with the traditional book. I give a description of some of the new possibilities with which we have experimented:

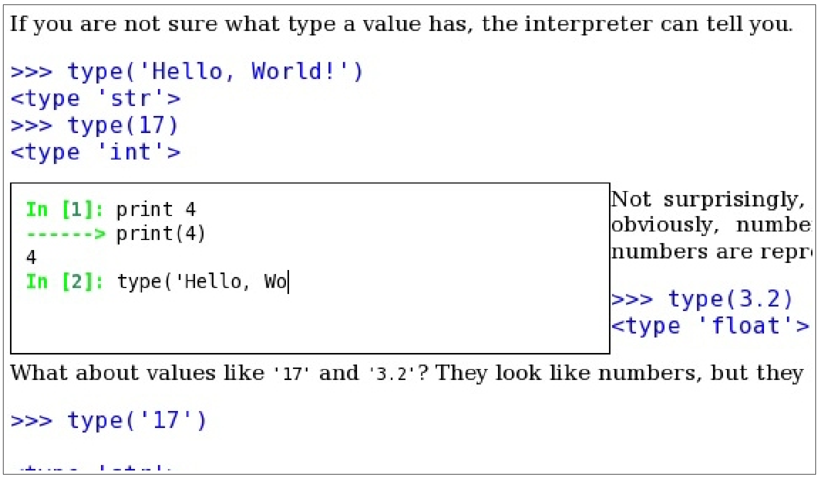
2.1 Multimedia content

Perhaps the most obvious components whose inclusion is made possible as a book becomes an ebook are video and audio. Instead of static images, a page in an ebook can contain audio-visual elements, coming alive either automatically, or manipulate-able via simple controls embedded in the page. This simple feature alone could make a world of difference in the reading experience and is the first step towards the next form of ebooks. The <*video*> and <*audio*> tags added in the HTML5 standard makes adding this kind of content into EPUB ebooks almost trivial. However, other technologies such as Adobe Flash, etc can be used for the implementation as well.

Incidentally, this particular feature/possibility in ebooks is being adopted not only by us, but also, one of the most publicized features of the latest version of Adobe Acrobat (the de-facto PDF authoring software for many) is support for audio-visual content embedding in PDF files.

2.2 Programmable shells

Most books on programming usually start off with simple constructs, and the first hurdle beginner programmers have to deal with is the process of setting up a right development environment to try out these constructs. This can be considerably difficult at times, and sometimes, it may even lead to the reader losing interest. For programming languages like Python and Scheme, which have an interactive console/shell, the initial pain of setting up a development environment can be avoided if these shells can be embedded in the book itself. Simple examples and constructs can be typed out and played around with inside the book itself, without needing to setup, configure, run and switch to an external IDE or shell.



If you are not sure what type a value has, the interpreter can tell you.

```
>>> type('Hello, World!')
<type 'str'>
>>> type(17)
<type 'int'>
```

```
In [1]: print 4
------> print(4)
4
In [2]: type('Hello, Wo
```

Not surprisingly, obviously, numbe numbers are repr

```
>>> type(3.2)
<type 'float'>
```

What about values like '17' and '3.2'? They look like numbers, but they

```
>>> type('17')
```

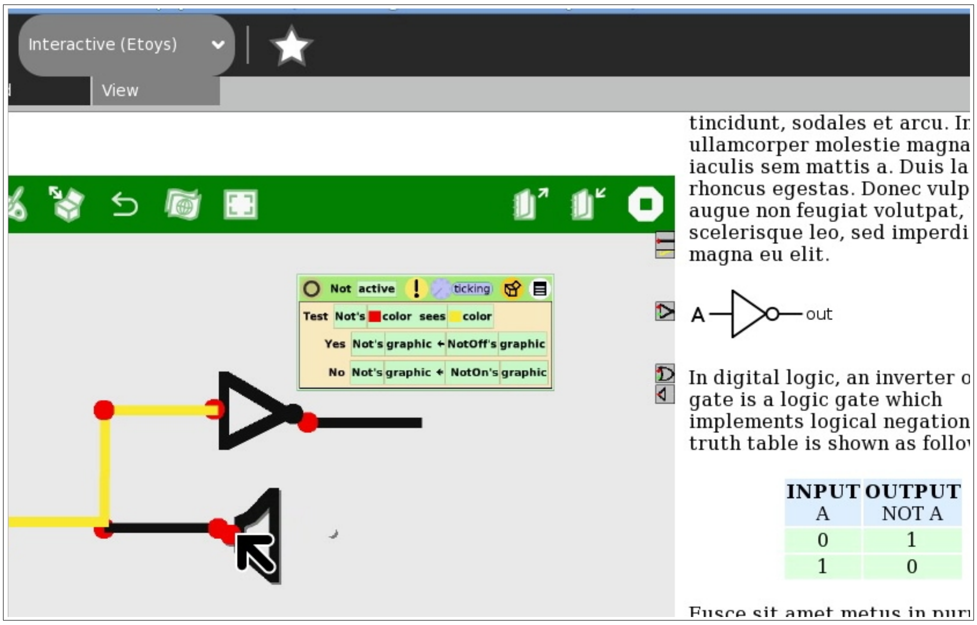
Figure.2 Interactive Python shell as an ebook element

This possibility can be taken even further, by allowing the execution of simple text I/O based scripts inside ebooks as well.

One crucial issue that had to be kept in mind while implementing this particular feature was security, since it essentially allowed for arbitrary code execution in the reader's device/computer. For us, the security system originally designed for the OLPC laptops, called Bitfrost can effectively deal with at least the majority of the posed security risks. Bitfrost's software implementation (called Rainbow) works by isolating each software process from one another (by allocating each process a unique sandbox environment and and a unique user id), so even if malicious code gets executed somehow, the damage done would be restricted to only the sandbox environment, and hence, would be minimal.

2.3 Microworlds

In his 1980 book *Mindstorms: Children, Computers, and Powerful Ideas*, Seymour Papert introduced the concept of microworlds – simulated environments with idealized conditions where children can programmatically conduct experiments. Example of a microworld include a computer simulated environment where Newton's laws of motion hold perfectly, and children can try out various experiments without having to deal with real-world factors such as friction, atmospheric wind-resistance, etc. For this particular micro-world, Papert suggested modifying the familiar Logo turtle to become a “dynaturtle”, with additional properties such as velocity and acceleration. These dynaturtles could then be controlled by children to get an intuitive and more relevant understanding of the Newtonian laws of motion.



```
Not active ! ticking
Test Not's color sees color
Yes Not's graphic ← NotOff's graphic
No Not's graphic ← NotOn's graphic
```

tincidunt, sodales et arcu. Ir ullamcorper molestie magna iaculis sem mattis a. Duis la rhoncus egestas. Donec vulp augue non feugiat volutpat, scelerisque leo, sed imperdi magna eu elit.

In digital logic, an inverter o gate is a logic gate which implements logical negation truth table is shown as follo

INPUT	OUTPUT
A	NOT A
0	1
1	0

Fusce sit amet metus in puri

Figure.3 A digital logic microworld inside an ebook

With the technology available to us today, such microworlds can be easily implemented with software such as Etoys, etc. We have succeeded in incorporating a full-blown Etoys environment into an ebook, thus making real the possibility of having entire microworlds inside books – as a result of which, without needing to go out of the ebook reading process, a child would be able to explore and play in various areas of science, construct and invent their own theories, and slowly evolve them through concrete and personal experiences and experiments.

Apart from Etoys, we have also enabled TurtleArt, a microworld for geometry and mathematics, which draws its inspiration from the Logo programming language, to be incorporated as an element inside an ebook.

2.4 Exercises

A very common request from teachers in various OLPC deployments is for better (technical) ways to grade students, to take short tests, to conduct quizzes, etc. For this, we are currently working on a feature that would allow self-contained exercises to be included in ebooks. This is largely similar to contemporary “online” tests, but instead of a central server, the entire logic behind the test system, along with the correct answers, marking rules, etc is included in the ebook itself. The Offline Storage API (or DOM storage), introduced with the HTML5 standard makes it possible go even one step further and make the entire process persistent, so that the examinees can come back to their exams later in order to reflect on and review their performance.

From a question pool included in the ebook, a randomized set of multiple-choice-type questions is generated in a specific page of the ebook as an exercise, and on submission, the answers are checked automatically. Finally the generated questions, the results and the provided answers are stored for future reference via the Offline Storage API.

A logical extension of this feature is to extend it to support submission of exercise results to a centralized School Server ("XS", in OLPC terminology), so that aggregated reports, etc can be automatically generated.

3. Other relevant features of the OLPC ebook reading system

While the features and possibilities mentioned above are the ones that evoke the most interest and is the focal point of this paper, to ensure the best possible reading experience for our young users, there are a number of other "standard" features that need to be taken care of as well. These features address real world challenges, which include issues like availability of power, network connectivity levels, language support, etc. I will describe in short, some of the most pressing issues that our deployments face, and how our ebook-reader software stack tries to work around them.

3.1 Power

OLPC's deployments are almost exclusively in developing nations, where power is scarce, and in some situations, close to non-existent. There are a number of ways in which the OLPC XO laptops try to conserve power. The display of the laptops is dual-mode, and can switch between backlit mode (somewhat similar to standard laptop displays) and a low power-consuming reflective mode (suitable for reading in the sunlight). A secondary DCON (DISplay CONtroller) unit is present between the primary display controller and the display. This secondary DCON's function is to retain the image on the screen even when the CPU and the primary display controller is switched off to save power (what is commonly known as suspend mode for standard laptops). This allows for most of the laptop components to remain off while in ebook mode, powering up only when a page transition is requested by the user. It must be mentioned here that due to software limitations, this feature has not been enabled by default in the first generation of XO laptops (XO-1), but will almost certainly be present in the XO-1.5, due to be released in early 2010.

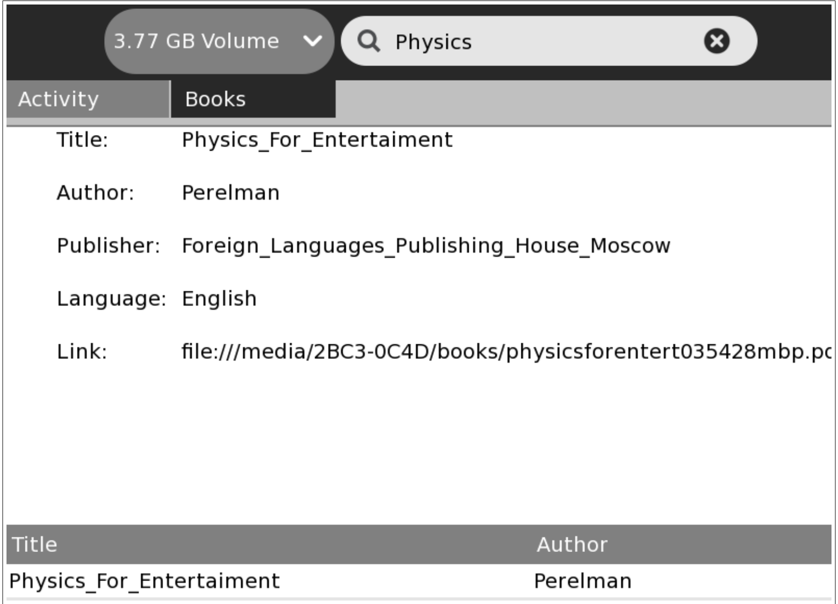
Even with this kind of aggressive power saving features disabled, the power consumption of an XO-1 in a typical usage scenario is around 4-7 Watts, which is still much less than any comparable laptop available currently.

3.2 Network connectivity

While there are country-wide OLPC deployments such as Uruguay which provide Internet access to the children using XOs, proper connectivity is still a distant dream for many regions. So when the Internet Archive announced in November 2009 that it is making available more than 1.6 million books in the EPUB format so that they can be readily accessed by the software that runs on the OLPC laptops, the most common question was, how will the books be made available to deployments without any proper Internet connectivity? Indeed, this is a valid and important question.

The Internet Archive uses a XML based (draft) standard called OPDS (Open Publication Distribution System) to publish its catalog online. While the book-acquisition component of the book-reading stack used by OLPC can access these catalogs over the Internet and let users search through them, it has a local device "sneakernet" mode as well. In this mode it can automatically detect and access storage devices such as USB pen-drives attached to the laptops which have OPDS catalogs present in them. Thus, at a centralized location with Internet access, a few thousand books can be chosen and downloaded, copied to a USB pen-drive with an auto-generated catalog and finally copies of the drive can be sent by conventional means to schools without connectivity. Thus, the equivalent of an entire library's collection can be easily distributed at a significantly lower cost.

Moreover, at the school or at the community level, the book-reading software itself allows for horizontal sharing of any ebook being read, and hence, any child with a book she finds interesting can "invite" over her friends, and the book will automatically get transferred to the friends' laptops as they respond to the "invite".



Title: Physics_For_Entertaiment

Author: Perelman

Publisher: Foreign_Languages_Publishing_House_Moscow

Language: English

Link: file:///media/2BC3-0C4D/books/physicsforentert035428mbp.pd

Title	Author
Physics_For_Entertaiment	Perelman

Figure.4 The ebook acquisition system displaying a search result from a USB drive

3.2 Local language/script support

While the ebook reader software user interface itself is translated into more than 40 languages, ensuring that books in non-European languages get rendered properly has been a slightly more tricky issue. The common problem behind this difficulty is usually fonts (it is impossible to include fonts for all the scripts in the world in our Operating System builds), but in rare cases, we have also come across issues where a script could not be rendered correctly (a majority of non latin scripts have complicated rendering rules, requiring support at the user interface toolkit level). As far as fonts are concerned, the problem can be worked around by utilizing the *@font-face* CSS rule in EPUB, by which one can bundle fonts inside a EPUB ebook. Also, here is an use-case where the PDF file format really shines as it has excellent support for embedded fonts and is largely font rendering system independent.

4. Conclusions

Digital technology has provided us with the tools and means to redefine not only the processes of publishing and distributing books, but also the entire concept of reading. It is important that we utilize this opportunity to the fullest, and perhaps children could be the greatest beneficiary of this change. As with traditional books, the largest amount of innovation and “deviation from the norm” has happened with childrens' books, and hopefully, in the new age of ebooks, this trend will continue.

Acknowledgment

Nearly 40 years ago, in his famous paper *A Personal Computer for Children of All Ages,* Alan Kay proposed the idea of the “Dynabook”, a device through which children could not only have access to digital media and content, but also the power and ability to manipulate and control the content creatively. Our ebook-reader efforts draws inspiration largely from his work, as well as work done by the Seymour Papert and others.

The work we do at One Laptop Per Child and Sugarlabs is largely a community effort, as with other Free and Open Source projects. The contributions of both the OLPC and Sugarlabs community has been invaluable, in terms of feedback, ideas as well as code. My heartfelt gratitude goes to the ever-enthusiastic members of the community for their constant support, help and encouragement.

References

Malik, O. (2009) The Next Hot e-Reader: The iPhone [Online webpage]. Available at <http://gigaom.com/2009/11/01/iphone-e-book-reader/> [Accessed 10 November 2009]

International Digital Publishing Forum. (2007) Open Publication Structure (OPS) 2.0 v1.0 [Online webpage]. Available at <http://www.idpf.org/2007/ops/OPS_2.0_final_spec.html> [Accessed 10 November 2009]

International Digital Publishing Forum. (2007) Open Packaging Format (OPF) 2.0 v1.0 [Online webpage]. Available at <http://www.idpf.org/2007/opf/OPF_2.0_final_spec.html> [Accessed 10 November 2009]

International Digital Publishing Forum. (2006) OEBPS Container Format (OCF) 1.0 [Online webpage]. Available at <http://www.idpf.org/ocf/ocf1.0/download/ocf10.htm> [Accessed 10 November 2009]

Krstić, I. (2007) System security on the One Laptop per Child's XO laptop: The Bitfrost security platform [Online webpage]. Available at <http://dev.laptop.org/git/security/plain/bitfrost.txt> [Accessed 10 November 2009]

Papert, S. (1993) Mindstorms: Children, Computers, and Powerful Ideas, 2nd Ed., Basic Books, New York.

Kay, A. (1972) A Personal Computer for Children of All Ages. In Proceedings of the ACM National Conference, Boston.