

Simplifying e-book and print production

Liam Borgstrom
University of Pretoria

While XML-based single source publishing is a powerful tool for streamlining the book publishing industry, it may not be the right one for every sector of the industry. For an industry that is part cultural and part industrial, the traditional workflow allows for creativity at every stage and this craft focus still deserves a place in defining how the industry functions. With simple and familiar tools, it can be possible to integrate elements of old and new principles to create a workflow that is streamlined and advantageous without being too radical a shift in practice. This article describes the factors that are important to creating publications in both digital and physical formats.

Introduction

Where the intended output is for non-interactive print and commercially available e-reading devices, XML workflows can be made simpler. The practice of creating proprietary DTDs (Document Type Definitions) in publishing houses is a viable route, as it allows for adequate content description and powerful format conversion. However, in the production of e-books in the prominent e-reader formats ePub and Mobi, the end result of all this XMLⁱ editing is HTML/XHTML.ⁱⁱ As a result, any broad spectrum DTD made for the production of e-books is being made too broad, as its conversion stage must limit the format in order for it to function. Therefore, it stands to reason that for this type of production, the workflow should rather centre around HTML as the source format and use the class attribute or HTML5 semantic mark-up to add classification to elements.

This becomes important as the benefits of an XML-based workflow can often be hindered by the software requirements. Because a large degree of publishing today makes use of varied computer-based software the need for interoperable formats is increasing. While it is true that XML provides the best level of platform independence, it is an interim format that itself is not suitable for publication without a suitable rendering engine. Hence HTML (as a widely used and displayed language) has been chosen to represent information in e-book standards. However, this does contain its own problems, as the formatting ability of HTML is not directly translatable for print design software. While there are single-source publishing software solutions that do away with the need for visual publication design, it is not common practice across the industry. While certain publication types do certainly benefit from a regulated

repeatable layout, the organic and human feel that comes from a human- or visually-designed publication has a healthy and respected place in our visual culture, with the role of the information designer firmly entrenched in the publishing workflow. “A key thing is to focus on is the fact that a screen is not contained the same way that a printed book is, and that it is an entirely different format” von Veh in ¹. The skills needed for e-book design in HTML formats must stem from web designers with strong coding ability.

Therefore, there is a need for a bridging format that unites visual design, mark-up-based design, and authoring within the formalised publishing environment. To an extent this does exist. Adobe InDesign’s InCopy Markup Language (ICML) and InDesign Markup Language (IDML) is an XML-based language that describes text and styling information in a way that can be interpreted by the InDesign platform. Furthermore this feature is used in conjunction with the InCopy client program (or suitable interpreter) to allow for simultaneous design and authoring/editing. This format can then be converted to HTML through InDesign if the layout file is prepared properly. While this does tie publishers into that software platform, it also provides a way for information designers not versed in web markup languages to create hypertext material in a more familiar working environment.

The PressBooks publishing engine as well as the Simon Frasier publishing programme (both based on the WordPress platform) have used XSL (eXtensible Stylesheet Language) scripting to enable conversion of HTML content into ICML for the purposes of uniting the two design fields ². It is however dependent on the use of a web-based engine which has many benefits (including least of all cloud-based working), though it does also mean the learning of another system which may or may not satisfy the needs of the publishing house.

An interim format must also be malleable and free to impose custom limitations on its intended output formats from the start. This can be the problem with web-based working environments where the typesetting/styling potential is limited to what the [often] JavaScript authoring tool allows. There are changes that can be made to this and Vook (an e-book production company)ⁱⁱⁱ so far has proven its excellence in making an authoring tool that does adapt for unique styles by working with live CSS (Cascading Style Sheet) ³. But, when the engine was still available to the public it was intended for self-publishing authors and not to be used as a multi-user content management system (CMS). As such, while a fantastic engine, it could not support the collaborative benefits which are integral to the efficiency of single-source publishing platforms (which are necessary in formalised publishing institutions). Unfortunately, Vook has now moved its engine to be used completely in-house, and acts now entirely as a custom publishing enterprise.

A further requirement for the interim format is that it should be easily shared, distributed, and integrated into existing content management systems without increasing the learning curve too drastically and thus leading to users reverting to prior practices. What is needed is a file that can contain textual, visual and associated styling information and be highly ubiquitous

and if not easily interpretable between all software, at least be useable through very common software.

Choice of book formats

ePub vs Mobi

In the general trade environment, ePub and Mobi have been the dominant books formats for personal reading devices. While they are different, they share many similarities. Both are based in HTML, and rely on the publication standards set out by the IDPF OPF specification. Both are archive formats, in that they package a set of files which allow for a controlled degree of rendering. As such they are both well-suited towards an HTML-centric workflow.

Both formats can be described as the “bound web”.^{iv} That is, content defined as a web-site but encapsulated in a standalone off-line format. Here the benefits of an HTML-centric workflow are most easily employed as the output can be used directly in the publication. The production process is one of ensuring applicable containment within the format.

The 2010 ALPSP survey ⁴ revealed that 25% of the ALPSP members are producing full-text XML journals, and 15% are producing ePub. Publishing with mark-up languages has long been common practice for technical materials for the expediency and the rendering quality that can be provided using specialist languages, for example MathML, which makes it highly practical and beneficial. The new ePub 3.0 standard supports MathML directly, as well as SVG (useful for graphs and other vector based graphics) ⁵, and as such should become more prevalent for journals, textbooks and scholarly monographs in the future.

Books as mobile applications

In the creation of books as mobile applications the added purpose of production is to define the interface necessary for the intended rendering device. In such cases, the authoring environment would have to generate content in a format which the application is intended to render.

The added benefit of this form is that a combination of codes can be used. While ePub and Mobi are limited to HTML and JavaScript representation. Creating an “app book” allows for the use of software programming languages to create completely unique and interactive reading environments. However, it is a great technical adventure and expense and may not be the most viable option for large-scale, non-interactive publishing.

Books on the web

As many e-book formats are based on web-technologies, they are easily adaptable to use as a collection of web-pages (such as Safari Books Online and Ibis Reader) or to be synchronised between reading applications as is done with modern services (such as those from Amazon.com and Nook). The book on the web is therefore a natural presentation medium. It also unbinds the content from the book container and allows the material to become truly interconnected by allowing for the feeding of live data and commentary, streaming media and instant updateability.

PDF

While not ideally suited towards digital publishing, it has been the mainstay for many publication types (most notably academic journals) and does allow for media enhancement. While numerous XML-based publishing systems do exist to facilitate the creation of PDFs (often using a styling language such as FOSI or XSL-FO), the complexity comes in the combining of traditional print publishing workflows with the autonomous benefits of digital publishing workflows.

While a document's content can be authored in XML and parsed for a printable layout, that layout is fully dependent on the restrictions imposed by the styling language. This makes the type of graphic (somewhat intuitive) layout we would normally find in richly designed printed books difficult to replicate without the use of labour-intensive tagging and embedding^v within the design program. Therefore getting the best of both worlds, at the moment, relies on splitting the workflow after authoring.

Working with XML

Approaches to digital publishing production

While the publishing process has been largely digital for many years in different degrees in many of the publishing fields, the practical process of digitalising the content so that it is re-interpretable and machine-readable is only practiced within a much narrower band of publishing. This process is essentially based on XML, whereby types of content can be made identifiable by software and stylesheets which allows the formatting and archiving of that content to be automated to a large degree.

Adrian Bullock (2012:87-90) describes the XML workflow process in a five variations:

1. XML authoring
2. XML copy-editing
3. XML composition
4. XML last, or back conversion
5. Hybrid model

These variations detail the steps of integrating XML into the workflow environment at various stages in the publication's lifecycle. When integrated at the beginning of the process, there is the smoothest workflow, but also the greatest learning curve as the author will be required to prepare the XML within the boundaries of the DTD (or the sister definitional format, the XML Schema). As a result, the authoring can be hampered by the requirements of the format, especially where the schema may be too wide and seem daunting. "The challenges revolve around changing how you think about producing, editing and designing content and managing the change process" von Veh in 1.

In an in-house environment where the support can be on-hand and procedures made global, XML-authoring can be effective. All role-players can make use of the same software and plug into the same CMS. This can create a highly productive workflow system as it keeps track of revisions, and creates a non-linear platform whereby edits can be made seamlessly up to and

beyond the point of publication without creating bottle-necks. However, in a workflow in which many role-players are separated the need to adapt to other authoring platforms becomes pertinent. Therefore, a model making use of XML composition (after authoring and editing) makes sense.

XML composition brings with it many benefits, but also many challenges when considering publishing as a creative industry. With influences beginning in the formalised creation of codices, design has always been an important consideration in the production of books. And while XML workflows do allow for a highly flexible and definable level of design. It is achieved through coding, measurement and mathematical operators.

When working with highly aesthetic material however, there is still value in the “traditional” publishing workflow. With the current information design programs being more visually based, achieving the same results through XML becomes a problem, as the visual techniques which have supported the industry for so long now become difficult to implement – for they require a different skill set, aptitude and approach. Therefore it stands to reason that for publishers which require the potential of XML-authoring and copy-editing to increase throughput, in order to maximise the benefits gained by both print and web design professionals, the workflow must split at production to allow for benefits of ‘traditional’ print production. This is the “Hybrid Model” which Bullock refers to and for a publisher producing a variety of materials, it becomes the most practical.

“I don't believe there is one ‘best’ way. Again, it's down to what is the best way that suits that particular publisher. ‘XML first,’ ‘XML last’ and ‘XML in the middle’ all have their own costs, implementation requirements and benefits” ^{McNamara in 1}.

Software solutions

Searching for solutions in software is a logical way to go about such processes. And since the inception of SGML and TeX we have seen the development of such tools.

Such systems can and have been developed in-house by technical publishers and service providers such as O'Reilly media, who have been noticeably open about their adoption of multi-platform publishing through proprietary measures. In so far as the creation of mobile applications, Inkling (a developer of educational app books) has shown great promise with Habitat in creating a viable new publishing platform for books as applications, by using a cloud-hosted, XML-based system to develop books as mobile and web applications ^{7,814}. It is currently running in a closed Early Adopter program.

Furthermore, corporate and technical publishing systems such as Liquid XML and ArborText Editor are very applicable for XML-based publishing in trade, academic and educational environments, especially when used in conjunction with well detailed content management systems. Such programs edit documents with the infallible structures of XML, ensuring that every piece of content is defined in a machine-readable state, thus allowing for rigorous structuring, automatic formatting and in conjunction with a CMS, regulated,

controlled and simultaneous authoring, editing and designing of independent parts of a project with a multi-format publication being the end-goal.

Visual typesetting systems such as Adobe InDesign and Quark Express also allow for XML, HTML, and ePub export.

Other management and conversion tools are also available for doing simple conversions between format, one of the most popular being Calibre. However, these tools do not allow for the hands-on editorial creation and customisation that is needed for publishing professionals. Rather one needs to edit the source code and customise each piece of information for the different intended outputs. And this can be done either directly or indirectly through various software options.

However, in an environment which is as uncertain and turbid as that for e-books, the need for interoperability, ease of use, and familiarity may be fundamental in adoption.

A sample method

As a proof of concept, the following method was devised to demonstrate how a small change in a familiar authoring environment could be beneficial to the production of e-books in the ePub and Mobi format.

The assumption was made that Microsoft Word is the most common authoring environment in formal publishing practice. Even in those situations where other environments are used (such as open office, older versions of Word, Word Perfect and so forth), the .doc/.docx format is often a compatible format and an up to date version of Word can be used as a final editing program to standardise the manuscript. The rationale behind choosing a common platform was to lessen the necessary learning curve, and thus allow authors and editors to get along with doing their normal actions in an environment they were already comfortable with. The only request to authors and editors is that they refrain from custom formatting and use the provided styles.

One of the many benefits of advanced XML-based typesetting systems is the ability to structure the content as it is being written (both visually and semantically). This feature is controlled through the combination of rigid DTD / XML Schema enforcement working in conjunction with the styling capabilities of the software. Word processors, however, have the same functionality. Those proprietary formats make use of their own DTD equivalents to describe the data being processed. MS Word has its own way of describing text and layout in a form which suits material to be published in print or on-line.^{vi} What's more, this can also be described in XML and HTML. Therefore, as an authoring tool for single-source publishing, Word makes an ideal platform due to its interoperability, as well as the fact that it is designed for the purpose of word processing, not necessarily XML structuring.

This benefit has long been known, and in conjunction with VBScript, many macros^{vii} have been developed for the purpose of converting Word files to HTML, XML, PDF, and other publication formats. This process however does require an understanding of Visual Basic, a

programming language, and for intensive streamlining is often integrated into a specific CMS as well. However, for many titles this may not always be a necessary step.

Wiley notes that the Word DTD is not deep enough for their publications, and therefore must resort to conversion practices into a custom and deep XML (WileyML) ⁹. However, XML as a language is strong largely because it allows for anything to be defined. HTML however has the same functionality through the class attribute and HTML 5 semantic tags. And, HTML also makes the publication fit for on-line presentation, as well as for ready adaptation into ePub and Mobi.

An aside on the role and practices of typesetting

While single source automated publication handles typesetting in its own way, it does so in a way which is fundamentally different the “traditional method”. The missing element is the human. Typesetting is a rigid, calculable and automatable process, but with the consideration of the greater layout is also a creative venture. While creativity certainly is possible and prevalent in automated systems, it does require an engagement with the formal standards of CSS, XSLT, and also perhaps an interim interpreter.^{viii} This marks a very different process to that of more “freehand” layout, where while many styles and standards are used at the designers or client’s discretion. The layout is visually flexible and adjustable. That is to say that layout is done by the eye, and not pre-determined by numbers. As a result, while a very neat print-compliant PDF can be produced from automated systems, it takes advantage of a different kind of mind to create that styling, and that can be detrimental to the industry, as the visual design composes part of the artistic appeal of the book.

While XML is a more powerful format by definition, for electronic book reproduction HTML is the intended end-format, and still has a capability to define custom content sets. Furthermore, if there is a desire to integrate experienced print typesetters into the workflow for the production of a print-edition, using XML can be detrimental, as its coding base is incompatible for visual typesetting (i.e. through InDesign),^x as is the same with HTML. An interim format that is accepted though is .doc which is accepted by InDesign (though the latest QuarkXPress supports .docx). This makes Word a more suitable interim as it is compatible with both intended outputs (and being XML-based still counts as XML content). What this entails is that after editing, the manuscript can move into production with an XML-based source document, thus ensuring compliant information types across print and digital editions of the book.

By creating a template within Word, editors can mimic the foundations of a DTD by standardising a series of styles for the various types of expected content. The template would be a combined effort between the copy-editor and the designer as it would describe the content which the author is expected to create for this particular work as well as displays the type with the agreed upon spacing, colour, fonts, and page size. This mimics the “instant preview” feature

of XML-typesetting systems and allows the author to visualise the end product as it is being written.

Print and HTML production can already occur in advance of the manuscript with the preparation of a design sample within the layout program. The template made in Word should conform to the standards of the design template. For HTML output, a sample document can then be made from the template by exporting it as “HTML filtered”. This creates a clean HTML file which defines styled content under the class attribute and creates a CSS which will form the basis of the ePub or Mobi book. This base CSS can then be customised by the web designer for the final intended outputs.

As the styles are already defined within the Word template, once the manuscript is completed, production’s primary task is to map the typesetter’s styles to those in the manuscript, and for the CSS reference in the HTML to reflect that of the updated stylesheet.

While in some ways a more laborious process than strict single-source XML-based publishing, the positive side is the preservation of human involvement, and a division of labour according to existing skill sets and familiar tools. It is a type of method which tries to ease the production workload within the traditional publishing processes.

Making a set template is crucial as the styles are interpreted by visual-typesetting systems and linked to their own style definitions, thus negating the need to recode the text. At the same time by exporting the edited manuscript as HTML (filtered) is created

In order to organise the various chapters to make up a book, the free ePub WYSIWYG editor Sigil was used. This program created the skeletal structure for an ePub into which HTML and CSS could be imported. Within the program then (though any text-editor can be used), the CSS was edited to make the control the visual rendering and make the styling compatible with the chosen outputs. The Mobi format was created from the ePub using Kindle Previewer (which gives a detailed report during conversion) according to Amazon.com guidelines¹⁰.

The challenge of semantic mark-up

While this method does allow for valid HTML production, it is valid HTML 4.01. This may still be acceptable in the interim until more devices adopt the new standards, but with HTML 5 and CSS 3 now present, the industry should be moving towards that level of mark-up. From one point of view, it’s not too great a problem to overcome, as many of the existing HTML 4.01 tags are still valid and class attributes will still hold true. The CSS will still need to be customised by the web designer, and newer options can apply with some editorial work in the generated HTML as would need to be done in any case.^x However, HTML 5 has introduced several new semantic tags (such as [article](#), [aside](#), [figure](#) and [section](#) (see <http://html5doctor.com/downloads/h5d-sectioning-flowchart.pdf>), which not only allows for more universal interaction and control over various elements but is also required in the ePub 3.0 specifications. This particularly refers to the [nav](#) element which is used to create an ePub document’s table of contents.

Furthermore, the ePub 3.0 specification also makes use of several unique identifiers (appearing as `epub:type`). This is again another level of semantic mark-up which a Word export is not capable of producing at the moment.

Therefore, with the development of increasingly semantic mark-up, the role of XML-authoring becomes more prominent as these custom elements would need to be accurately converted and attributed.

As a result, with newer mark-up, the method described above begins to fall short, as it would require a greater level of mark-up editing, and/or the use of macros or other software. For the method to work with HTML 5, Microsoft would have to continue from their previous efforts. Already in Word 2010, the styles strong and emphasis are match to `strong` and `em` tags. Therefore, if the same efforts were made for article, figure ... the software could become more compliant.

Resistance to change

Iacob and Dekhytar ¹¹ address the need for “document-centric XML documents”. That is, XML documents that are authored more around the primary content than around the structure, and describe such authoring as more “human-mode”. The xTagger (a custom XML authoring and editing tool) system was designed for humanities scholars not trained in technical/digital authoring environments to allow for XML-based but free-form writing.

The concept is based around the idea that writing according to a template can hamper the authoring process. And some software has made an attempt to address this by making the structure almost seamless and invisible. Arbortext XML documents achieve this well as the stylesheets are linked to the documents and enforce a rigid writing structure without the author needing to change levels in the hierarchy manually (except when starting a new section). The same is true for the above-described method in which the format of the document was pre-set so that the author be able not only to see what the final outcome may look like, but be able to write unimpeded by the structural requirements.

Madans ¹² explains how the Hachette Book Group had difficulties in making the XML tagging less complex for editors. In the findings it is suggested that “Editorial tagging of files requires better tools. XML editors are too complicated [and editors need] ease of use of Word”.

Pearce ¹³ notes that the learning curve with learning to use XML editors is high but is significantly flattened if done through Word. He poses the idea of working within Word and then exporting WordML and using XSLT to adapt the content for other formats. But the interim formats that DTP programs accept are Word files (.doc/.docx) and when creating e-books, the ideal output is HTML. As such, the ideal method maybe a simpler one, as the mind-set of / aptitude for mark-up languages is not for everyone.

Planning for the future

With the development of ePub 3.0, and the increased adoption of HTML 5 and CSS 3, the method detailed above loses its applicability as it is only suited towards ePub 2.0 publications. Naturally, with the introduction of a new standard, it is up to the software manufacturers to develop tools that support it. The argument presented above is that the creation of new tools is a positive influence, however, adoption is never guaranteed. Hence a well-made market needs to exist prior to use.

Apple has already made a great advancement in this respect with the development of their iBooks author, which uses Apple's famously user-friendly interface to create e-books to the new standard. However, the modifications imposed on the standard means that it is not an objective, store- / device-neutral tool. But it has a market within the Apple fan-base. For other markets (fan-bases being an applicable term within a broader device-segmented market), other tools would need to be used. The audience that needs to be targeted here is not the web-development professional but rather the writing author, the first user and reader who must create the content with the final electronic format in mind.

As such, what is needed is a further development within popular software to support the new semantics that are defining the developing semantic web 3.0. Within the shift to using a greater template based writing system, existing word processors will need to enable those developing the templates to create to assign new semantic mark-up based on the style settings.

Typesetting systems may be ahead in this regards in that Adobe InDesign CS6 now allows for the mapping of style options to some specific tags. However in order to make the process fully compliant for ePub purposes and extension into .ibooks and KF8 and Mobi formats, the need for flexible and custom tag mapping is a necessary development.

However, it is this researcher's belief that the encoding is better done within the authorship stage, and that the author understand the inherent structure they should be building into their works.

The process is not new and practised in forms of publishing where stream-lining is necessary. Whether using an XML-based system or not, publishers are increasingly making use of template-based systems where the design can be influential in the authoring process. In highly structured publications this inclusion method may be the most ideal. The trick though will be to make it easy.

The important question

If all of these new production processes rely on adoption, the necessary research topic for discussion is how well do authors adapt to changing methods within familiar environments versus in new environments (where 'environments' is used synonymously with software). This would need to be the topic of further research to ascertain whether this system of augmenting the familiar environment is more suitable than creating a new authoring environment.

ⁱ eXtensible Mark-up Language

ⁱⁱ Hyper Text Markup Language / eXtensible Hyper Text Markup Language

ⁱⁱⁱ Vook initially began by making “video books” in a new format which combined video and textual interfaces in a double page spread. They have since redeveloped themselves and now create e-books in the standard formats for the major stores on behalf of individual and corporate clients.

^{iv} ¹⁵ define this type of a creation as a document, which “is self-contained in that it should be possible to communicate it as a single unit (e.g. in an email).”

^v Referred to an anchoring in InDesign

^{vi} Largely thanks to the MS Office suite now running on XML-based formats.

^{vii} Pre-programed tasks that can customise the workspace and automate certain tasks.

^{viii} Arbortext’s ‘.style’ for instance

^{ix} That is to say in its stand-a-lone form. Visual typesetting systems can create template documents which are fed by XML, however it is best suited simple list based structures and repeatable elements of known length, as otherwise the process for synchronising the XML data source can become overly cumbersome, time consuming, and taunting to human error. It is worth noting that this ‘Data Merge’ function (as it is called in InDesign) is designed to work with comma- and tab-delimited documents, and when imported into the typesetting system create XML structures. However, it is unsuitable for large volumes of text due to the labour requirements.

^x Such as updating !DOCTYPE tag and perhaps adding additional per chapter metadata.

References

1. Webb, J. The agile upside of XML - O’Reilly Radar. *O’Reilly Radar* (2011).at <<http://radar.oreilly.com/2011/10/xml-publisher-workflow-ebook-design.html>>
2. Maxwell, J. W. & Fraser, K. Traversing The Book of Mpub: an Agile, Web-first Publishing Model. *Journal of Electronic Publishing* **13**, (2010).
3. Vook Vook Guide. *Manual* (2012).at <<http://guide.vook.com/>>
4. Cox, L. Scholarly book publishing practice: the ALPSP survey findings. *Learned Publishing* **23**, 347–356 (2010).
5. IDPF EPUB 3 Overview. (2012).at <<http://idpf.org/epub/30/spec/epub30-overview.html>>
6. Bullock, A. *Book Production*. (Routledge: Oxon, 2012).
7. Empson, R. An Answer To Apple: Inkling Creates First Industrial-Grade Publishing Platform For Interactive eBooks | TechCrunch. *Tech Crunch* (2012).at <<http://techcrunch.com/2012/02/14/an-answer-to-apple-inkling-creates-first-industrial-publishing-platform-for-interactive-ebooks/>>
8. Greenfield, J. Inkling Unveils E-Book Creation Software for “Professional” Use | Digital Book World. *Digital Book World* *Digital Book World* (2012).at <<http://www.digitalbookworld.com/2012/inkling-unveils-e-book-creation-software-for-professional-use/>>
9. Grazioli, F. How Wiley Uses Word to Invite Authors, Engage Editors, Improve Prod... (2009).at <<http://www.slideshare.net/toc/how-wiley-uses-word-to-invite-authors-engage-editors-improve-production-and-put-xml-at-the-source-of-its-content-presentation>>
10. Amazon.com *Amazon Kindle Publishing Guidelines*. 63 (Amazon.com: 2012).at <<http://kindlegen.s3.amazonaws.com/AmazonKindlePublishingGuidelines.pdf>>

11. Iacob, I. & Dekhtyar, A. xTagger: a new approach to authoring document-centric XML. ... *Libraries*, 2005. *JCDL'05. Proceedings of ...* 44–45 (2005).at <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4118513>
12. Madans, P. The Evolving Role of Authors and Editors. 13 (2009).at <<http://www.slideshare.net/toc/the-evolving-role-of-authors-and-editors-presentation-920817>>
13. Pearce, J. XML publishing workflows. 25 (2009).at <<http://www.slideshare.net/newgen/xml-publishing-workflows>>
14. Cromwell, R. Putting Pages to Pasture. *Inkling Blog* (2010).at <<https://www.inkling.com/blog/putting-pages-to-pasture/>>
15. Battle, S. *et al.* DIY eBooks: collaborative publishing made easy. *Proc. SPIE 7540, Imaging and Printing in a Web 2.0 World; and Multimedia Content Access: Algorithms and Systems IV, 75400B (February 10, 2010) 75400B–75400B–9* (2010).doi:10.1117/12.839077

Contact information

Physical address:

Dept. of Information Science
Information Technology Building 6-40
Hatfield Campus
University of Pretoria
Corner of Lynnwood Road and Roper Street
Hatfield
0083

Postal address:

Dept. of Information Science
University of Pretoria
Private Bag X20
Hatfield
0028
South Africa

Tel +27 (0)12 420-2246

Fax +27 (0)12 362-5181

E-mail: liam.borgstrom@up.ac.za

Web: <http://web.up.ac.za/default.asp?ipkCategoryID=12481>