AN AUTHORING TOOL FOR DEVELOPING HIGHLY INTERACTIVE DIGITAL BOOKS
The IntBooks Platform

Eugénio Rocha, Ana Breda, Mário Soares and Telmo Parreira
Departamento de Matemática, Universidade de Aveiro, Portugal

Keywords: Authoring tool, Educational software, Interactive digital books.

Abstract: We present a document platform, denominated "IntBooks", that seamlessly aggregates web technologies into digital books, giving the reader an interactive, intelligent and rich environment, and authors a "simple" and collaborative way to build and share them. It addresses some of the common technical issues as: how should different media (e.g. mathematics formulae) be delivered and displayed in a large set of devices; how users interaction should be made and statistics collected; and how authors may reuse and mesh up content from different web repositories. A particular IntBook planned for the fifth and sixth grades of Basic Portuguese School is also discussed, emphasizing its specificities and additional features.

1 INTRODUCTION

Any multimedia educational software is a structured and varied blend presentation of multiple media, such as text, formulae, graphical images, animation, audio sound and full-motion video, by a computer-mediated software or a single interactive (web) application. Multimedia software may use some or all of these modes of communication, however, it is more than a collection of multiple media and hyperlinks (Chang, 2000). As a complex interaction of stimuli (McKerlie and Preece, 1993), multimedia software for education aims to put users in contact with a specific set of knowledge, in an organized and reactive way. In fact, it can be a powerful tool in the hands of the performance technologist and multimedia designers to be used by educators. The effects of multimedia-based instruction versus lecture-based instruction on teaching has been investigated in several works in the literature, e.g. in (McKethan et al., 2001). The last decades have witnessed a steady, irreversible trend toward the globalization of e-learning tools, and education through the use of software-intensive web technology in particular. But the production of multimedia artifacts demands guidance for both engineers developing software and designers generating content. This is a clear obstacle for non-technical technology aware authors (or educators) that want to produce multimedia educational software to be used in their teaching activities.

Here, we present a document platform, denominated "IntBooks", that seamlessly aggregates web technologies (as many as possible) into digital books (web applications), giving the reader an interactive, intelligent and rich environment, and authors a "simple" and collaborative way to build and share them. The developed platform strives not only for content deployment but also as a set of tools that enable authoring and re-usage of contents extracted from repositories. In an IntBook production, the author can use any mixture of formats, be it simple text, HTML, LaTeX, or any other which he is more comfortable with. It can also embed java-applets (e.g. Geogebra constructions or others), in fact, any current web technology (e.g. flash objects) and some application specific formats. All the source is written in a XML-structured file (the IBK format) and the set of tools takes care of the rest, transforming it accordingly and as necessary for a consistent web presentation in the client browser targeted for a specific device (a PC with a small or wide screen, a PDA, etc.). Figure 1 presents an example of a IntBook by Pedro Miguel Duarte of the University of Lisbon, generated from an IBK file that contains LaTeXand Java applets.

To achieve such high goals, there are a multiplicity of important issues and technical decisions that

Rocha E., Breda A., Soares M. and Parreira T. (2009).
AN AUTHORING TOOL FOR DEVELOPING HIGHLY INTERACTIVE DIGITAL BOOKS - The IntBooks Platform.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 302-307
DOI: 10.5220/0001981003020307

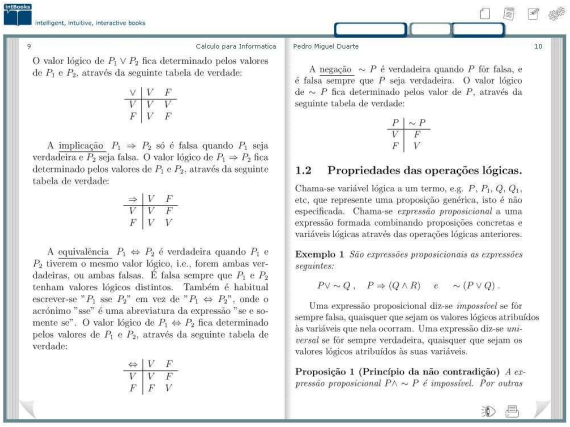

Figure 1: A snapshot of an IntBook, with a double-page layout, seen in the web browser of a widescreen PC.

must be considered. For example, the platform should produce digital books that render appealing mathematical expressions and formulae, e.g. to be suitable for the production of supporting material for courses at higher levels of education. Among the multiple types of content, mathematics presents the biggest challenge, since there is no consensus in the community about which format is simultaneously better integrated and produce the best visual results (e.g. PDF, MathML, LaTeX converted to an image as in Wikipedia), see (Borwein et al., 2008). Generally and briefly, we must clarify the following crucial points: (a) how mathematics (and other media) should be delivered and displayed in a large set of different devices; (b) how users interaction should be made and statistics collected; and (c) how authors may reuse and mesh up content from different web repositories. In this work, we convey our point of view, answering the above questions through the design model and development of the IntBooks platform. In Section 2, we focus in the display problem, presenting our "unified" model. Section 3 gives a sketch idea of how the platform deploys an IntBook (web application) from the IBK file (source code). The user interaction is discussed in Section 4 and the aggregation concept in Section 5. Section 6 makes a brief explanation of the platform management and access control. Finally, in Section 7 we present a particular digital book, covering the new curricula in mathematics for the 5th and 6th grades of the Portuguese Education System, which is a clear example of the potential and features of our authoring tool.

2 CONTENT MESH UP DISPLAY

In our everyday use of Internet and web experience, we interact with a multitude of technologies capable of displaying different types of documents and enabling different types of interaction. The most common are HTML and Javascript, but also Flash, SVG, Java, etc.. Browsers enable the presentation of content through external plugins or mechanisms already embed in their source code. Authoring complex web content requires more than one tool and various steps. Often one has to take *a priori* in account the final layout, the target device screen dimensions and capabilities, and the software to be used, before its real production. IntBooks reduces significantly the effort to create online interactive books, feeding content from various formats and sources. These pushes for the need of an unified and coherent representation of content, not only to avoid dependencies but also to be able to transform between formats (sources) in a consistent way. A file format specification was designed, in order to comprise such needs and to be extensible to open formats (see subsection 2.1). The user should also be able to produce rich content, regardless of end form of presentation, in the software more familiar to him. IntBooks takes this into account, and makes an effort to support as many formats as possible, and put your content where it needs to be, and how it needs to be. Using a back-end set of tools, it transforms the input into the suitable format for web usage.

Figure 2 shows a IntBook's page which was generated from three different source formats.

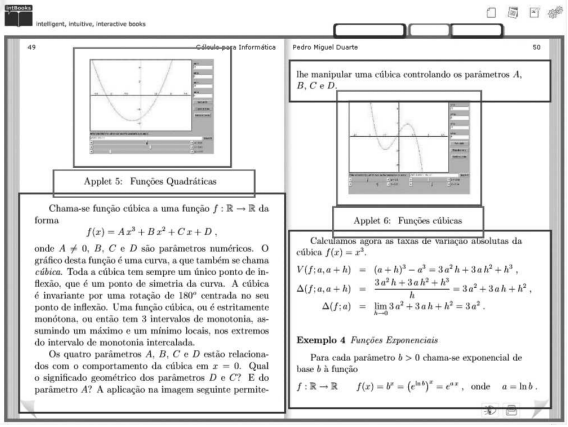

Figure 2: A snapshot of a calculus IntBook, where different source fragments are marked with boxes: LaTeX(the bottom boxes), Java applets (the interactive graphs), and HTML (the figure captions).

2.1 Structure and Language

The proprietary source package is a ZIP file containing a XML based file, so-called IBK file, and external binary files (e.g. images, videos, SWF files, etc.). The IBK file main purpose is to be a consistent container and structured aggregator of the different source formats (called IBK fragments, embed in specific XML tags), which additionally has the file referencing for the external binary content. It is a file structure intended for multi-format document amalgamation, editable in any simple text editor. In what follows, there is an example of the top level syntax of an IBK file:

```
<?xml version="1.0" encoding="utf-8"?>
<intbook-system>
  <book id="3" title="TexMat" authors="Geometrix" year="2008">
   <header>
     (information / parameters setting commands)
   </header>
   <body>
     (fragment / block commands)
   </body>
  </book>
</intbook-system>
```

The IBK file is then parsed and processed by conversion tools and filters, which apply to each fragment, in order to generate a visually consistent and interactive book (see Section 3).

2.2 Different Layouts and Devices

To enable content delivery to different devices some restrictions would need to be enforced and considered. Examples of potential targets are, the nowadays common, mobile devices, which have specific web capabilities (varying from manufacture and model), e.g. screen resolutions and web extensions availability are in general very limited. One key point is that authors (content producers) should not be obliged to produce different sets of content to different devices. The platform should (ideally) mold itself accordingly. However, content size and formation can be a real obstacle. IntBooks make use a set of algorithms that enable a reasonable and automatic splitting of content into blocks that best fit in the page size templates, pre-defined in the IBK file. This is possible since the server stores the original IBK source file, and not a compiled version, so it enables content re-edition and output generation on-demand, i.e. content may be recompiled with different parametrization and templates to better fit the graphical demands of the supporting layout and the different target devices. Figure 3 is a snapshot of the same IntBook of Figure 1, but with a different layout (template).

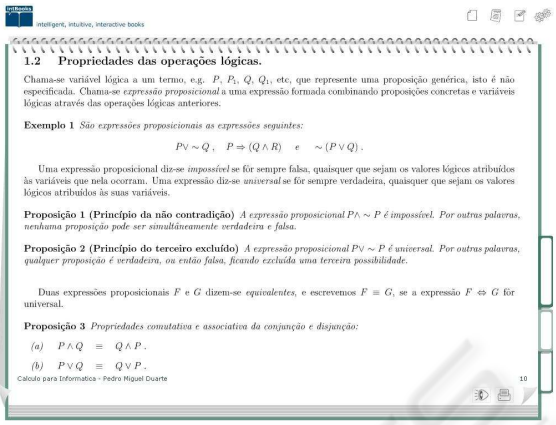

Figure 3: An IntBook with a one-page layout.

3 DEPLOYMENT

The processing and deployment of the source package (an IBK file + a set of external binary files) is very technical and deserves to be explained in detail in a future publication. In short, the processing is supported by the simple idea of a pipeline of transformations, where each IBK file containing some fragments formats are converted into a new IBK file with fragments with different formats. For example, we convert all the LaTeX to Flash vector scalable objects, which give the best visual and zooming performance.

After parsing and processing the IBK file, the output is intended for a multi-user server based environment. An installation step is needed in the server environment. Although this could be a simple package extraction process, and the content exclusively a client based application, the server-side also enables interaction and a user-based session.

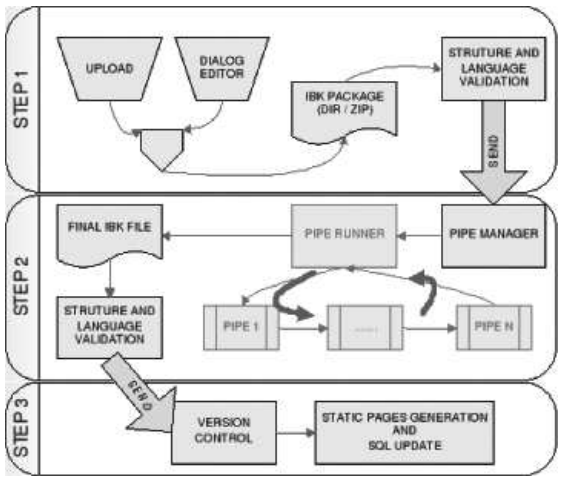

Figure 4: The IntBook processing scheme.

Although access to an IntBook may be restricted by an optional authentication layer, the installed book can be readably available for generic usage. Inputting the book URL into the browser starts the user interaction. The IntBooks web-application is responsible to deliver all the content and multimedia assets to the user browser, also providing the turn page functionality to the book. The generated content that initially could be supplied in the form of a LaTeX document, is now readable and has been transformed, for example, into Adobe Flash format. The server is also responsible to choose the most suitable output format, according to the browser capabilities.

For more sophisticated IntBooks, the server delivers a web-service for variables storage where the client application can read and optionally write. This functionality is configured as part of the IBK file. The IntBook's example TexMat (see Section 7) makes intensive use of this functionality, making it a highly responsive environment to user interaction (e.g. allowing randomly generated exercises).

4 RICH USER INTERACTION

The basic user interaction for every IntBook layout is the page turning functionality. But this can be a little limiting to the content that can be delivered. To warp this, a read/write link is available for the client-side delivered content, enabling conditionality linked to the smallest user interaction. This way during user interaction with the book, let's say a particular page as a multiple choice survey, and the next question would only appear after the previous is correct, using this read/write link, the client application could check if the answer was correct and record every answer the user would give.

Implementing a similar functionality with page refreshing would drastically increase band-width requirements, create a not so smooth user interaction, and would probably break page navigation. This is a relevant characteristic to maximize the user experience and intended for quizzes, pools, surveys, exercises, tests, etc, and it's achieved through storable variables on the server. It enables the application to "remember" the user, and the user to review all of his past answers.

These server variables are strong-typed and declared in the IBK file. During installation on the server the variables are allocated for a per user instantiation and changes to these variables are logged for usage by reporting tools.

5 AGGREGATION AND RE-USABILITY

In this topic we simply use the classical idea of "*reuse, not repeat*". IntBooks can also be viewed as a collaborative system, where a particular book can be built from fragments of other books or from other entire books. Different authors may work, for example, in different books, in different formats and merge everything in a uniform book. Obviously, the author dependencies are kept during all the process. Since the IBK format is a format for fragmented content, entire chapters can be imported from book to book. Because disciplines have similar chapters, concepts are interconnected. Since some courses are created from the merge of two intensive book courses and not all information is relevant, re-utilization is a key feature. Avoids content repetition and guarantees that an explained concept has only one meaning across the various books. The mechanism that manages fragments also incorporates bridges to some web repositories (e.g. for extracting LaTeX source from PlanetMath).

This linking between books is dynamic, so when a chapter or concept is improved with examples, the other books using that source are also improved. As future work we plan to improve the search engine that allow authors to find and link published contents of other IntBooks into their own books.

6 USER MANAGEMENT

As in most e-learning platforms, there is a trend to mimic the organizational hierarchy of an institution. In a learning environment there are the obvious student and teacher roles, and as in any user-based platform there are the typical administrator and manager. The IntBooks platform is no exception, and it requires a centralized, delivering and control system. The created authentication, authorization system, was molded as an institution library, makes only sense, because our object of development was centered on the book. An institution may have various libraries (sets of books) and books may be categorized. A teacher assigns a book to his teaching class (set of users), and all the students (users) can start using it. This might appear to be a complex system, but it's also very scalable. It can be as simple as an administrator uploading the books, and creating user accounts, and it's ready for usage. Using teachers role, enables decentralized management of users and permissions. This role is also meant for accessing the students data, and statistics collection. The administration system

enables creation of groups of institutions and sub-institutions (departments or pedagogical groups), allowing for the propagation of roles and books across all sub-institutions. A user might have different roles in different sub-institutions, for example: be an administrator of physics department and only a teacher in mathematics department, and a student of a art book. A sub-institution can have its exclusive users that are not users of the mother-institution. The administration interface model makes the IntBooks platform a vertical solution, enabling its deployment on large organizations. Figure 5 shows part of the administration interface developed as an Adobe Air application.

Figure 5: Snapshot of the Administration interface.

7 TEXMAT: A MATHEMATICAL INTBOOK EXAMPLE

TexMat is a highly interactive digital book covering the new curricula in mathematics for the 5th and 6th grades of the Portuguese Education System. Its design includes several features and capabilities as multilingual, modularity, the ability of being easily extendable by teachers (e.g., adding Geogebra constructions), model generated exercises, open questions, and centralized gathering of students statistics and assessments. But, technically, TexMat is a IntBook. A particular IntBook, sharing the same structure, but with additional features, specially designed for this book, since the target students need some particularities, mainly in the design and interaction level. In fact, TexMat was the application that originated IntBooks. Instead of developing a specific system, we have expanded the idea for a more general and abstract system that can build and control every book designed to be an IntBook, where TexMat can be included.

7.1 Content Display

In TexMat, almost every page is composed using Macromedia Flash. Because we are interested in keeping the users answers to the questions and in changing the appearance of several objects (visibility of text, images, animations, etc.) according to the users accesses and interactions with the book, it was necessary to define a communication protocol (using XML) to exchange information between the Flash objects and the server database. The IBK file, presented above, must now have more information about the dynamic variables and questions of any flash object.

Here is an portion of the TexMat IBK file, that represents a flash page:

```
<page>
    <flash filters="user">
      <vars_layout fName="GF19" src="flash" width="712" height="476"/>
      <vars_state>
        <var id="23" name="cErrado" property="imagem" value=""/>
        <var id="24" name="c1" property="imagem" value="caixa1a.swf"/>
        <var id="26" name="ver" property="imagem" value="ver.swf"/>
        <var id="27" name="text3" property="_visible" value="0"/>
        <var id="30" name="inp1" property="type" value="input"/>
      </vars_state>
      <questions>
        <question id="418" resposta="" tentativas="0"/>
        <question id="419" resposta="" tentativas="0"/>
      </questions>
    </flash>
</page>
```

Because of these dynamic variables, the book appearance changes in every usage, according to each user's performance.

7.2 User Interaction

As any IntBook, TexMat has two well-defined modes: student mode and teacher mode. However, in this particular case both modes have some extras, in addition to visit or edit the book.

7.2.1 Student Mode

Each user (student) must be associated with (at least) one teacher, and this defines the user's navigation in this particular book. While the student is interacting with the book, it goes changing (invisible text or images become visible, answers are given, images changed) and these changes are registered in a database. They become visible in the next user visit and in the teacher's view.

During the navigation, an interactive agent (developed in Macromedia Flash) guides the student through the book. It reacts to the user's actions according to his/her answers and controls the progress in the book, as we can see in Figure 6.

Also during the navigation process, and when a specific concept is explained in the book, it is added to the user's notebook, where he/she can add a particular note about the concept. The notebook is individual and can be acceded in any moment the user wants. If it is called from inside the lesson with a concept, goes directly to the respective definition of the associated concept. Otherwise, the notebook application opens

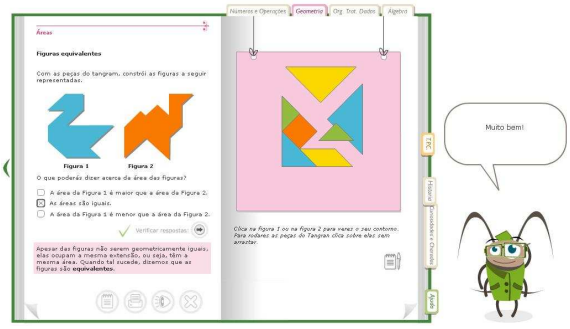

Figure 6: The TexMat agent reaction.

in search mode, permitting the user to search every concept in his/her own notebook.

7.2.2 Teacher Mode

As an editor, the teacher has full control over his/her own book, and may share it (or parts of it) with other teachers. Starting from the default TexMat book, a teacher can delete or add pages, in every location. He/She can construct a sub-book of Geometry, for example, deleting the remainder of the book. But, more interesting than deleting is adding. With a simple interface, adding pages to a book is a simple action that does not require high knowledge of web tools, and this operation can be previewed in real-time and be approved or not. Every type of document that can be viewed in the web can be added to the book, e.g. text, images, HTML, pdf, java applets, etc.. As a teacher, he/she has information about all his/her students' performances, receives the open question responses and may interact with the students directly from the book, sending particular messages or giving homework for a particular student or for all the class.

7.3 TexMat in Classroom

Not pretending to be a substitute for the classical manuals or competing with the math classes, TexMat can be used as a complement in the classroom. Specially the animations and videos can be very useful for explaining a concept, and the large amount of exercises can be used to practise. Also, the teacher can give homework directly from the book, during the class. In this case, the students are informed of the fact in the next time they open the book.

Another interesting feature of TexMat is that it can be used to plan particular lessons in the classroom. Since a teacher can add pages to his/her own book, he/she can prepare lessons from TexMat, using the included lessons or using only the platform with his/her own contents.

8 CONCLUSIONS

In this work, we have briefly discussed an authoring tool platform for an integrated and consistent delivered of multimedia content (containing different technologies) suitable to be presented in different layouts and devices, reducing the authors (content producers) effort in learning a specific set of technologies. This platform, by the set of features and structure model, is clearly different from standard projects as Wikipedia (or PlanetMath), iPaper or Google docs. In some sense, IntBooks could be viewed as a learning management systems and the generated books as learning objects. Its interface emphasizes the historical learning source of knowledge, *the book*. Integration of LMS and SCORM standards is an optional work in process. IntBooks has the simplistic model of creating the smallest dependencies needed to achieve the maximum features possible. Basing development in the standards would hurt development, and restrict ideology implementation. In this way, and using a KIS (keep it simple) architecture, standard compliance is a feasible future. Maintaining the sources, would be just a matter of adjusting the filters, to produce a compliant SCO. The IBK format, is intended for compilation, where as other formats deal with the compiled media form of the documents, emphasizing in document fragments, and enabling content reorganization, after installation, where as SCO's are the indivisible learning unit of general LMS. In IntBooks, content can be reorder to the simplest paragraph.

We hope that in, the future, IntBooks turn to be an favored and useful tool for authors to express their way of organizing and presenting educational content in a highly interactive and appealing form.

REFERENCES

Borwein, J., Rocha, E., and Rodrigues, J. (2008). *(eds). Communicating Mathematics in the Digital Era*. A. K. Peters, Ltd.

Chang, S. (2000). *Multimedia Software Engineering*. Springer Verlag, Berlin.

McKerlie, D. and Preece, J. (1993). The hype and the media: Issues concerned with designing hypermedia. *Journal of Microcomputer Applications*, 16, 33–47.

McKethan, R., Everhart, B., and Sanders, R. (2001). The effects of multimedia software instruction and lecture-based instruction on learning and teaching cues of manipulative skills on preservice physical education teachers. *Physical Educator*, 58, 2–13.