

## **Liquid Book: collaborative reuse and sharing of multifaceted content**

Fabio Casati<sup>1</sup>, Maurizio Marchese<sup>1</sup>, Cristian Parra<sup>1</sup>, Luca Cernuzzi<sup>2</sup> and Ralf Gerstner<sup>3</sup>

<sup>1</sup>*University of Trento, Trento, Italy, {casati,marchese,parra}@disi.unitn.it*

<sup>2</sup>*Universidad Católica "Nuestra Señora de la Asunción", Asunción, Paraguay, lcernuzz@uca.edu.py*

<sup>3</sup>*Ralf Gerstner, Springer-Verlag GmbH, Heidelberg, Germany, ralf.gerstner@springer.com*

### **Abstract**

Liquid Books are a new way of thinking about scholarly and educational books in the Internet era. In Liquid Books, authors share material (and so, collaborate, as authors in an ordinary book or in a wiki-book) but each author is then free (within the limits of a contractual agreement that we have identified) to take any of the shared material and edit/organize it any way they want, and then to have a personalized version of the book that leverages the knowledge of the group but that does not require everybody to share the same view on the content or organization of this personalized version. Liquid books include multifaceted content, which stays up to date with the current state of the art (as it continuously evolves over time) while reducing the typical time to market interval. The key issues here are not IT-related but rather related to the setting up of a suitable contract, licensing, credit attribution, royalties and dissemination model.

In this paper, we first describe the main concepts and the conceptual model for Liquid Books together with a short presentation of related work in this area. The central part of the paper is devoted to the detailed description of the experience gained during the past year on the creation of one real instance of a Liquid Book by a group of collaborating authors. We first describe the pilot, its content, the related processes and used infrastructure; then we focus on the lessons learned and in particular on the methodological and contractual novel aspects of the Liquid Book model.

**Keyword list:** Computer Supported Cooperative Work, Social Web, Educational Publishing and Dissemination, Customizing, Personalization

### **1 Introduction**

The recent advances in the Web and in ICT have created new scenarios that radically change the knowledge production process: we have virtually unbounded storage capabilities and essentially no limits in our ability to interact with peers. This new network- and storage-mediated knowledge production process is impacting on all aspects of knowledge creation on all types of knowledge. The Web is becoming the most extensive knowledge repository that ever existed. As examples of this trend, it is sufficient to consider the success of Wikipedia, or the upcoming initiatives such as WSRI<sup>1</sup> (Web Science Research Initiative).

Surprisingly, it is easy to observe that innovative forms of educational knowledge are somehow lagging behind, and that the world of scholarly and educational publishing has been largely oblivious to the current potential of the Web 2.0. Today, education material is characterized mainly by static printed or electronic books, online slides and digital Learning Objects [2] generally managed by powerful but complex and difficult-to-use Virtual Learning Environments. In particular, current infrastructures do not encourage or support effortlessly open sharing and community evolution of academic courseware material. For a given topic, professors have at their disposal several textbooks to choose from. However, it is rarely the case any single one of these textbooks matches the professor's intended class topics and level of depth, and it is even more rare that the books are up to date, especially in a quickly evolving area

---

<sup>1</sup> <http://webscience.org/>

such as IT. In the best cases, the authors "maintain" the book more or less up to date every now and then, often with minor extensions. As a consequence, a professor wanting to teach a class that is up to date is somehow forced to use parts of several books along with supplementary material taken from the Web. Indeed, it is not surprising that some professors lately teach by using Wikipedia as a reference for the various topics, rather than textbooks.

This paper explores how state-of-the art ICT, web technologies and the lessons learned from software engineering and the social Web can be applied to provide a radical paradigm shift in the way educational knowledge and in particular textbooks and courseware is created, disseminated, evaluated, and maintained. To this end we detail in Section 2 the motivating scenario and in Section 3 we present the new concept of Liquid Book and the related conceptual model. In Section 4 we briefly revise related work and point out to the main differences of our proposal in respect to other current approaches. The central part of the paper is devoted to the detailed description of the experience done during the past year on the creation of one real instance of Liquid Book by a group of collaborating authors. We first describe the pilot, its content, the related processes and used infrastructure; then we focus on the lessons learned from such experience, in particular on the methodological and contractual novel aspects of the Liquid Book model.

## 2 Liquid Books: Motivating Scenario

When teaching, professors and students need a book which is tailored for the class, which is up to date, which includes (or points to web-based) exercises, which has a companion web site with additional readings, etc. And, at the basis, a textbook that is correct and clear. Today, this rarely happens: books are written by a small set of authors and require a lot of coordination among them. Moreover, since updating books often requires a big effort by both authors and publishers, usually new editions contain just minimal changes or more chapters at the end. From the perspective of lecturers, it is very hard to find a book tailored for their class, meaning that they have to resort to a collection of books, with the related problems (cost, inconsistent description, difficulty in exposing a coherent set of topics in a coherent way, difficulties in linking topics and classes and exercises, etc..).

In this work, we have explored and proposed the concept of Liquid Books (LB) as collaborative, evolutionary, possibly open-source and multifaceted versions of the traditional books (either printed or digital). In a nutshell, a liquid book is characterized by the following entities (in italics):

- *content* added by contributors
- content is published via *editions*, prepared by *editors* (editors are a subset of contributors).
- a *contract* defines who can add or consume content, who can create editions and personalized versions, what the frequency of editions is. A contract defines also the licensing/visibility policies for content among contributors.
- A contract also includes the specifications of a number of processes, e.g. for modifying the pool of authors (including new authors or removing previous ones), defining credit attribution mechanisms etc.
- a *publisher* publishes and distributes the editions and personalized versions of the book, typically holds the copyright or an exclusive distribution license, and possibly provides additional services, including IT services supporting LBs.

A traditional book is one instance of the model above. In a traditional book (think of authored books, not of edited books for now) we typically have one or a very small number of sequentially published editions. All contributors (the authors) are listed as authors of the book, and may get royalties based on a pre-agreed contract. Variants (except for new editions) of the book are typically subject to new contracts. In particular, an author cannot just take the content and publish its own personalized version

without contractual agreements with the other authors and with the publisher. Also, in a typical publishing process, there are rigorous quality checks. The “time to market” is overall quite high, both because it takes a long time for authors to prepare the manuscript and because of the publishing process after the initial manuscript has been delivered, that also takes several months. Overall this resembles very much a waterfall software development process, with all its benefits and limitations.

The opposite extreme of a liquid book resembles more agile and open source software development, with some twists due to the nature of book publishing and copyright management. So, at the “liquid extreme” we have a liquid book where contributors add content to the LB, possibly in a continuous fashion, and where contributors can freely decide at any point in time to prepare new personalized version by collecting and editing content put by other contributors to the same LB. With such flexibility, each personalized version could be easily tailored for a set of specific readers. At the proper phase, the associated processes (as defined in the contract) start. These processes can range from very simple rules (e.g., editors always take 20% of the royalties and the rest is shared equally with all contributors) to more complex procedures (for example, a voting mechanisms that approves royalty distributions proposed by the editor). The key here is to have these processes simple and well defined, at the beginning, otherwise the complexity is excessive and very likely this will not scale.

The main goal behind the LB concept is to enable authors to easily share and reuse their content, giving them at the same time the guarantee that their content will be used appropriately and so producing personalized versions of the book tailored for the needs of the prospective readers (students, consultants, etc.) LB can be open source, partially open, up to the case of traditional closed book. In our model we aim at covering all the possibilities, leaving to the authors the final choice of how to distribute their content.

A first - but not the only - pilot scenario we have focused on is that of educational textbooks, but all the concepts described for this scenario can be also applied to different ones, like major reference work (e.g. encyclopedias) where larger pools of authors are involved and where the need for regular updates is very strong.

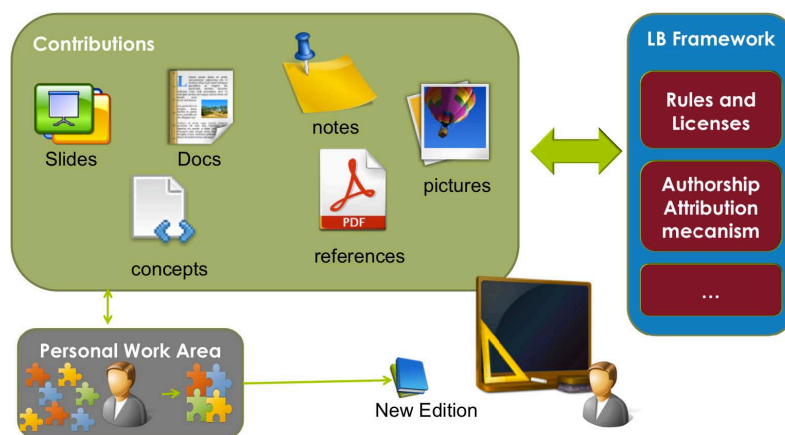


Figure 1: Liquid Book Textbook Scenario

The main concept behind a LB textbook is to let professors compose and create a customized (and as liquid, evolving) textbook for their class by putting together modular (but “evolving”) content from other colleagues. A LB textbook is an evolving and multi-author collection of materials on a given topic

that unlike traditional books/e-books, is composed by different kinds of artifacts (documents, slides, exercises, code, experiments etc.), is collaboratively written by several, possibly (but not necessarily) cooperating authors, and has various author/publisher defined copyright models (see Figure 1). The idea is to have a few professors that share their teaching materials (slides, sections on given topics, exercises), as they want to produce a book for their class, and each one of them or some of them produce a different version of the book, as they may have a different vision on how to organize the content. Then we envision that each professor at the start of the class will "create" a version of the book that is tailored for his/her class by reusing the shared material, either composing it when possible, or taking it and modifying it when needed and when allowed by the contractual framework they agreed on before to start the LB textbook. We expect modifications to existing content also because when we compose it with other material we need to make sure that the text flows properly and that terms are used consistently, but possibly also to add/edit content.

There may exist several versions of a LB textbook, per year, per professor and per class. Versions are different from editions: versions are different views on the LB content, i.e. the content can be assembled in novel ways depending on the needs of the specific audience: a book for a class of master students or PhD students, a book to improve personal knowledge of a topic, not for teaching purpose, etc. Instead, editions are "solidifications" of the LB textbook, whose content remains stable and fixed during a longer period of time (more like traditional books). Note that while editions are static collections generated from specific "internal" versions of the LB (they do not change anymore), the LB itself remains in its evolving state and may externally exist (in addition to the stabilized current edition) in several personalized versions. As a result, authors can progressively improve and maintain the book, generating new internal versions as they reach a delta modification that they consider relevant and which will then end up in a new edition, from which again new personalized versions can be derived..

This model is not that far from open source software or wiki-like projects, also here there are the concepts of collaborative efforts, sharing and reuse of materials, redistribution. However, rules and practices typical for wiki or open source projects cannot be straightforwardly applied to LB textbook scenario. The difference is that here authors have more and strong concerns about credits (intellectual property and royalties) or improper reuse of existing work. Indeed, a distinction has to be done between "functional work", (e.g., manuals, computer programs, recipes) where it is somehow easier to establish rules about reuse and modification of contents and "opinion-type work" (e.g. scientific reports) where a modification can lead to misrepresent the authors' ideas or thoughts.

### **3 Liquid Books: Conceptual Model**

Figure 2 shows our current conceptual model for Liquid Books. In the model, we can see how the LB authors can start a new (either internal or personalized) version, each composed by different types of content. Both structural and representation links are used to navigate the version. Structural link defines relations within the LB as an index, while representation link leads to the actual content in a particular representation.

The LB version has also one or many associated renderings (how the content is actually presented to the final readers). The evolution of a particular version can be tracked by temporal links. Every now and then an internal version is frozen in an edition. It is possible to define who (person) contributes to each version and what role (authors, contributors, etc.) this person takes on. People can annotate (with tags, comments, opinions) both content nodes and link nodes, and the system keeps track of who tags what. We can describe the working area of a LB as a "content pool", where authors put their content to share it with others and where they can define their personalized versions on top of this content. Internal versions are liquid, as they are in a continuous evolving state, while editions are internal versions, which

have been solidified and cannot change anymore. In addition, editions form the basis of personalized versions, which can be further developed by the involved authors.

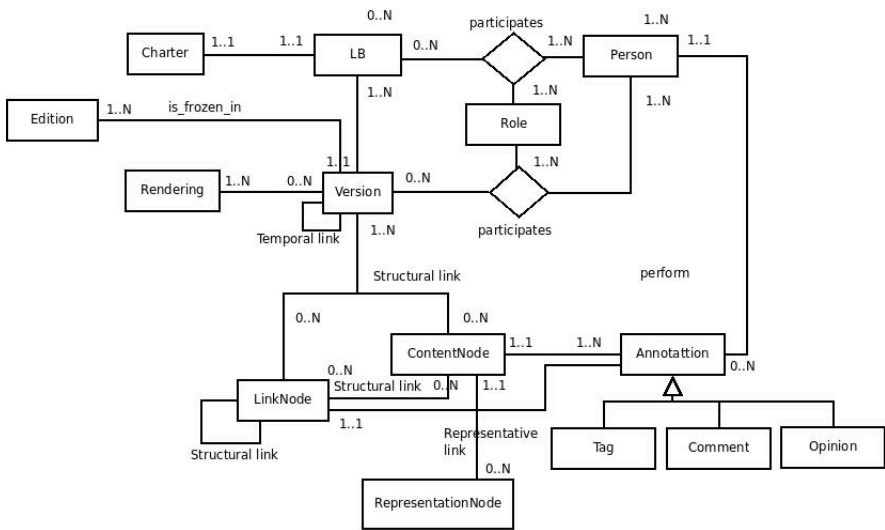


Figure 2: Liquid Book Conceptual Model

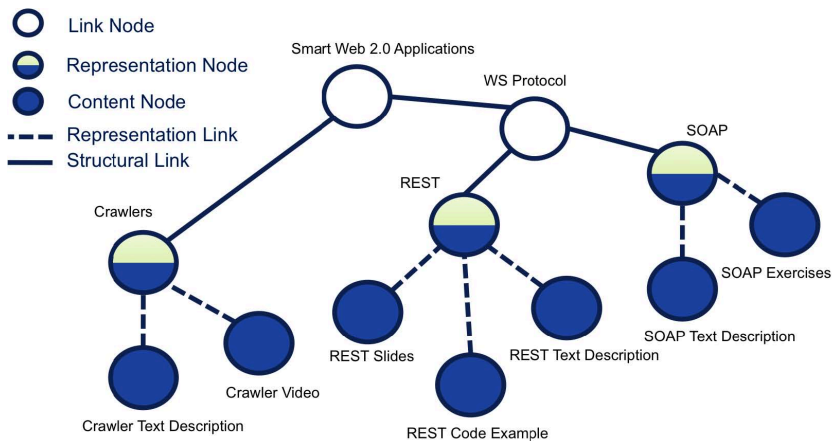


Figure 3: Example of Liquid book structure

A version is a collection of multi-faceted content. The notion of multi-faceted content comes from the idea that for any given concept, the same concept can be "represented" in different ways: slides, text, video, etc. As an example, consider three professors teaching topics related to Web Applications in different universities. They decide to write an LB called Smart Web 2.0 Applications. They can have in a particular version some content related to REST, and different representations associated with this concept: slides, code libraries, text descriptions, as well as some content on Crawler, including just some text and a video of an interesting talk. The specific LB can then be represented using a tree, where the nodes are the different topics (REST, Crawler, etc.) and the leaves the actual content (see Figure 3). Therefore, informally, we can distinguish three kinds of nodes in the tree: (i) nodes which just represent the different topics included in an LB, but do not contain actual content, like REST, SOAP, Crawler and that points to (ii) nodes where the actual content is (text, slides, code, etc.), and (iii) nodes which contain just a set of links that points to the different topics of the LB (WS protocol, etc.). More formally, looking at Figure 3, it is possible to distinguish the following three kinds of nodes in the

tree:

- a link node, which is a collection of links to other nodes, among the two in the figure we can distinguish the link node which is the tree root and identifies the particular version of the LB;
- a representation node, which is still a collection of links to different representations of the same content;
- a content node, which is the node containing the actual content. Link nodes and representation nodes are linked to each other through structural links, while content nodes are linked through representation links to content nodes.

As we mentioned before, LBs are evolving books, where versions, differently from editions, are not static collections, and are instead in constant development. To keep track of this evolution in time, we can use temporal links and series. Temporal links connect different trees of the same LB. Each tree is a version and the Series is the ordered list of tree's roots connected by temporal links, representing the evolution of the LB. A series keeps track of the evolution in the time of the LB version, where a new root node is created by the author(s) any time a significant delta of changes is reached. The structure of the LB is inspired in that of the Scientific Knowledge Object proposed by [3][4][5].

#### 4 Related work

Wikibooks<sup>2</sup> (WB) are open-content and freely distributed, as they adopt a GFDL license<sup>3</sup>, which ensures that content remains copyrighted to the authors, while at the same time the copyleft licensing allows one to freely distribute and reproduce such a content. WBs are “public” in that everybody can see everything. Also, in WB anyone can contribute by adding new content, editing or deleting existing one. Everybody can participate, so a WB can have many authors. The main idea of Wikibooks follow the successful Wikipedia commons-based peer production model [1] incorporating three main concepts: (1) the content artifacts must be modular; (2) in order for a peer-production process successfully to pool a relatively large pool of contributors the modules should be predominantly fine-grained; and (3) the peer-production process must have low-cost integration -the mechanism by which the modules are integrated into a whole end product. Social norms play also an important role in sustaining some of these peer collaborations, both where there are small groups, and where there are larger groups and the platform allows for good monitoring and repair when individuals defect. At present, the Wikibooks site is listing 2,399 books in English in its repository and no specific “best-seller” in the academic domain.

LB are different from the WB model in that:

- LB are not necessarily open-content. The content is exposed through editions. This does not exclude (but does not require either) an approach where the content is also made public or freely available (based on certain licensing policies defined by the contributors or the publishers).
- The collaboration in LB is more on the contractual side: contributors may collaboratively edit a text, as done in wikis, but this may or may not happen and it is not the essence and value of LBs. The essence is in allowing reuse and modifications based on contractual agreements that define attribution of credits and royalties.
- Each piece of content in LB can exist in multiple versions, both because the original contributor can evolve it and because other contributors may take it and change it for putting it in their on edition. Around WB there is instead a social interest in having a reference point, a unique version which can be taken as reference.
- In LB, the responsibility for the content ultimately is with the LB editors of an edition. While in

---

<sup>2</sup> <http://en.wikibooks.org/>

<sup>3</sup> <http://it.wikipedia.org/wiki/GFDL>

WB the focus is on the content creation process, in LB it is on the edition creation process by reusing, modifying, and organizing content. A distinction has to be done between “functional work”, (e.g., manuals, computer programs, recipes) where it is more easy to establish rules about reuse and modification of contents and “opinion-type work” (e.g. scientific reports) where a modification can lead to misrepresent the authors ideas or thoughts [XX].

- Editions are static collections (an edition is analogous to the edition of a book today), the LB itself remains in its evolving state and contributors can keep adding or editing content - both internally to prepare new editions (e.g. to include new technological or standardization developments) and externally to create new personalized versions (e.g. to accommodate to the needs of a specific target audience). We can see this as a branch in software development.
- In LB the we still have the notions of publishers and editions with ISBN, DOI etc.

Other related initiative are the *Culture Machine Liquid Books*<sup>4</sup>, which shares the same name with our project with a model much closer to that of WikiBooks. The initial ideas of this initiative were sketched in the second chapter of [6], where the author cite also LiquidPub as a parallel and related work.

With respect to other related efforts, a first attempt to create evolving textbooks is in place by MacMillan, one of the five largest publishers of trade books and textbooks. There are several points in common between our vision of LB and MacMillan’s one of DynamicBooks<sup>5</sup>, as well as some differences. DynamicBooks will allow instructors to edit digital editions of textbooks. In this way, professors will be able to customize the book for the needs of their classes, deleting chapters, uploading different kind of materials (notes, videos, pictures, graphs). However, the big breakthrough is the fact that professors will be completely free to rewrite or delete paragraphs, equations, illustrations without consulting the original authors or publisher, they have just to log into the system.

Regarding the business aspects, MacMillan will sell online access to the students, and the modifiable e-book editions will be much cheaper than the traditional printed version; even if the publisher will also offer print-on-demand versions of the customized books for a higher price. This can be seen as a particular case of liquid book with specific and very simple predefined rules and processes. They do not envision any customizable publishing process or legal framework to constrain possible changes to books, publishing of editions, attribution of credits and royalties, and the like. On the contrary they rely completely on the instructors and on students or other instructors to help monitor changes, even if the publisher reserves the right to “remove anything that is considered offensive or plagiarism”.

With respect to the collaborative aspect, there are several examples of collaboratively created books, even if they are more similar to classical e-books, without the multi-facet aspect or the evolving one. A first example is “How to Think Like a Computer Scientist”<sup>6</sup>, series of publications by Green Tea Press, where the same core programming text has been adapted to several different programming languages (Java, Python, C++, etc.). On the same line, 97 Things Every Programmer Should Know<sup>7</sup>, another example of collaboratively written book, with hundreds of contributors, where multiple and varied perspectives are collected about things a programmer should know. Contributors have become involved through announcements, specific invitations, recommendations and word of mouth. Another example of collaborative written book is Business Model Generation<sup>8</sup>, with 470 co-authors and distributed without

---

<sup>4</sup> <http://liquidbooks.pbworks.com/>

<sup>5</sup> <http://dynamicbooks.com/>

<sup>6</sup> <http://openbookproject.net/thinkcs/>

<sup>7</sup> [http://programmer.97things.oreilly.com/wiki/index.php/97 Things Every Programmer Should Know](http://programmer.97things.oreilly.com/wiki/index.php/97%20Things%20Every%20Programmer%20Should%20Know)

<sup>8</sup> <http://www.businessmodelgeneration.com/> <sup>7</sup><http://www.fastpencil.com/>

a publisher. The last three initiatives do not have a specific platform for collaboration; they usually rely on call for contributions and then on manual integration of contributions. A first attempt to facilitate the search of contributors and the collection of contributions is the one by Fast pencil<sup>9</sup>, a platform to write, publish and sell collaboratively written e-books. The above mainly focus on collaboration on content creation, which again is not the focus of LB. A wider overview of highly collaborative environments (such as that of LB) and the particular problem of credit attribution within them are discussed in [7].

## **5 Pilot: A Liquid Book on Service Oriented Business Process Management**

Starting mid-2010, a group of authors have initiated a real experimentation of the LB concept. The group consists both of lecturers involved in the teaching of courses related to service design and business process management. Namely, the group includes:

- Fabio Casati, (LiquidPub): teaching "Business Process Management" at the University of Trento;
- Maurizio Marchese, (LiquidPub): teaching "Service Oriented Architectures and Application" at the University of Trento;
- Alejandro Mussi, Xiaola: teaching "Business Process Management" at the University of Trento;
- Boualem Benatallah : teaching "Service Oriented Architectures" at the University of New South Wales;
- Helen Paik: teaching "Service Oriented Architectures" at the University of New South Wales;
- Marlon Dumas, (LiquidPub): teaching "Enterprise Software Architecture and Business Process Management" at the University of Tartu
- Marcello La Rosa: teaching "Business Process Modeling" and "Enterprise Software Architecture" at the Queensland University of Technology
- Ralf Gerstner, (LiquidPub) representing Springer as the publisher of this work.

Since all authors are involved in lecturing on Service Oriented Architectures and Business Process Management, the decision has been to work collaboratively towards the creation of a novel text-book supporting teaching in these topics. The title of the first edition of the Liquid Book has been tentatively defined as "Service Oriented Business Process Management". At the same time, a first version of the Liquid Book, focusing on the business process management aspects, is also being shaped under the tentative title "Principles of Business Process Management".

The initial content of this work is directly derived from the teaching experience of the authors, starting from the current material and experiences they have been using independently so far. The aims of this proposed prototype of Liquid Book is to provide a complete and personalized set of teaching materials (including textbooks, presentation slides, exercises, laboratory material, assessments and single and group projects) that can be adapted by each author (or group of authors) to their specific course. The topics covered aim to provide the basis for a number of fundamental courses in the general area of service science, design and engineering and addressing ongoing convergence between Business Process Management and Service-Oriented Computing.

Following the concept of the Liquid Book model, the starting point therefore has been the common decision to share the current individual teaching material (at present the size of the repository of combined lectures' material is about 2GB). Thereafter, the authors have identified a number of macro-processes involved in this editing experience. The first process is our Liquid Book pilot life-cycle (see Figure 6). It contains a number of phases, namely:

---

<sup>9</sup> <http://www.fastpencil.com/>



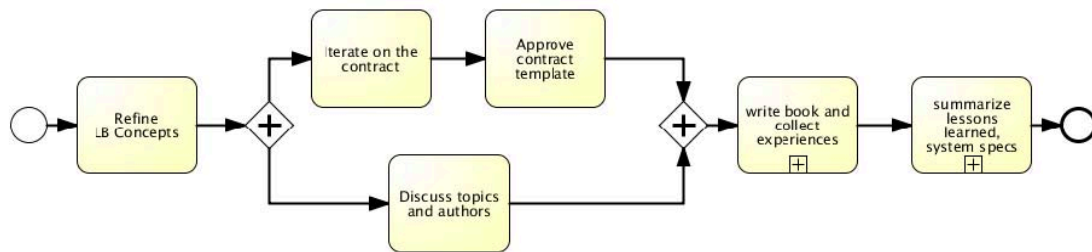


Figure 6: LB general process

- a first phase devoted to the refining and sharing among the pool of authors of the Liquid Book concepts and conceptual model. This phase has been running from end of August 2010 until early October 2010.
- a second phase with two parallel threads of activities: in the first thread the authors have been discussing the specific topics, type of materials, IT infrastructure, tools et al. until a sustainable agreement has been found; on the second thread a discussion with the publisher has initiated on the specificity and peculiarity of the contract type that could support such kind of work. Also in this case, discussion has been iterated until convergence. This second phase has lasted from October until early February 2011.
- the third and fourth phases are currently running and involve the actual creation of the first edition of the work as well as the collection of the involved actors (authors, publisher, students..) experiences. We have however experimented with the writing and reusing of content and in credit attribution within draft editions of a liquid book, available on our shared repository, and students of the course in Trento are already using the liquid book as main teaching material and reference book for the course. The book grows weekly as new lectures are done and as content from co-authors becomes available.

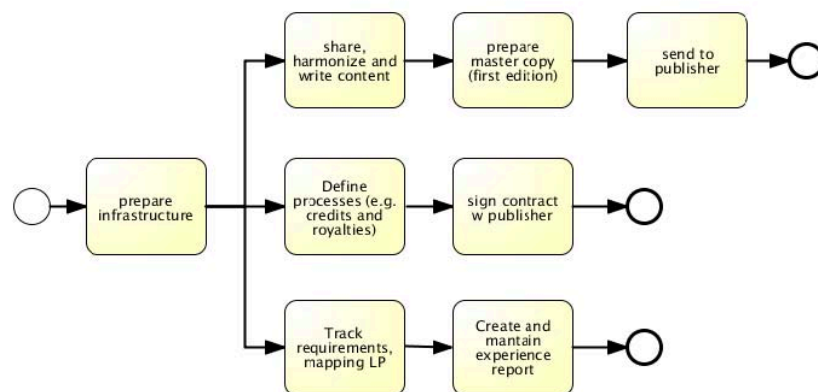


Figure 7: LB Contractual process

Within this generic process life-cycle (valid also for more traditional authoring processes), we have specialized two more specific processes: one centered on the infrastructural and contractual aspects and the other emphasizing the elicitation from the pilot experience of the IT requirements specification and in general a more detailed understanding and mapping to the generic IT Liquid-Pub infrastructure (in

terms of services needed, API required etc.). Along with the model, defining the contract is really at the heart of the work. Having a contract model means that now we have a way to promote and reuse the liquid book concept beyond this pilot. Over time and as people (and publishers) gain practical experience in writing liquid books, the contract model is also supported by methodologies and best practices.

The main steps of the infrastructural and contractual process are summarized in Figure 7. They include:

- a first activity concerned the selection and implementation of the infrastructure selected in the previous process. It is important to underline that in this pilot, our focus has been on the methodological and contractual part and not on IT development. Therefore, we have used available state-of-the-art infrastructure and tools supporting collaborative authoring work. In particular, we have decided to use:
  - Dropbox as the repository for our material. Dropbox<sup>10</sup> is a Web-based file hosting service operated by Dropbox, Inc. that uses cloud computing to enable users to store and share files and folders with others across the Internet using file synchronization. The Dropbox client enables users to drop any file into a designated folder that is then synced with Dropbox's Internet service and to any other of the user's computers and devices running the Dropbox client. All files stored online by Dropbox are encrypted and kept securely on Amazon's Simple Storage Service (S3) in data centers located along the east coast of the United States. Dropbox's focus is on synchronization and sharing: it supports revision history, so files deleted from the Dropbox folder may be recovered from any of the synced computers. Dropbox's version control also helps users know the history of a file they may be currently working on, enabling more than one person to edit and re-post files without complications of losing its previous form. The version history is limited to 30 days for normal users while a paid option for unlimited version history called "Pack-Rat" is available. The selection of this cloud infrastructure has been made mainly for the interesting combination of simplicity of use and offered functionalities.
  - Latex as the main document preparation system. LaTeX<sup>11</sup> is a well-known document preparation system for high-quality typesetting for medium-to-large technical or scientific documents.
  - Microsoft Office suite<sup>12</sup> for draft text authoring and more specifically for the authoring of the slide presentations.
  - Signavio BPM Process Editor<sup>13</sup> as the drawing suite for the textbook diagrams. Signavio Process Editor is a software tool for graphical business process modeling and analysis and we have been using it in its Software as a Service on-line version.
  - Apache Subversion<sup>14</sup> open source version control system for the Java code of the exercises, assignments and student's projects.
- a number of parallel activities, namely:
  - authoring a first complete and self-contained master edition and deliver it to the publisher;
  - arrive at an agreement among authors and publisher on the credit attribution and royalties sharing within the Liquid Book model in order to sign the contract between authors and publisher;
  - track all process requirements experienced during the pilot and maintain an experience re-

---

<sup>10</sup> <http://www.dropbox.com/>

<sup>11</sup> <http://www.latex-project.org/>

<sup>12</sup> <http://office.microsoft.com/>

<sup>13</sup> <http://www.signavio.com/>

<sup>14</sup> <http://subversion.apache.org/>

port

Finally, in the pilot we have also analyzed and discussed IT issues related with the overall Liquid Book model and infrastructure . In particular we have proposed the following IT-oriented plan and related process (see Figure 8):

- during the pilot experimentation, the participants will refine the detailed model mapping between the LB conceptual model and its realization. This mapping will be used as the basis for the definition of a first set of API for the LB infrastructure;
- at the end of the delivery of the first edition a detailed IT requirement specifications document will be compiled;
- on the basis of the two previous tasks, we will set-up and launch an Open Source development project for the development of an liquid book IT platform

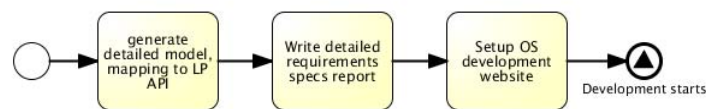


Figure 8: LB IT-related process

## 6 Lesson Learned

In this section, we briefly summarize the main Liquid Book specific issues we have been able to learn during the pilot experience. In particular, we will focus on: (1) Contractual issues (2) Methodological Issues (3) Infrastructural Requirements and IT Issues.

### 6.1 Contractual Issues

When designing a contract template for liquid books, it became clear that several issues had to be taken into account, which are not relevant for traditional books, but are related to the special features with respect to creation, personalization, maintenance, and distribution of liquid books. The main "non-standard" issues that we have encountered have been:

- How to fit regular updates in the traditional set-up of ISBNs and editions
- How to fit personalized/derived versions in the traditional set-up of ISBNs and editions
- How to handle the addition of new authors and the retirement of old authors
- How to handle the change of content by people who are not the original authors of this content
- How to make sure that authors produce updated versions on a regular basis
- How to distribute royalties for personalized versions
- How to handle royalties for retired authors
- How to handle various distribution strategies and platforms

Apart from these issues, some others, which are also relevant for traditional books, become nonetheless more important and more complicated for liquid books, e.g.

- How to protect the author's work from undesired/illegal re-use
- How to make sure there's a viable economic exploitation possible
- How to make sure that possible new editions fit in the life-cycle of the target purpose and audience

As it had turned out very soon, there is more than one answer to each question, and the answer depends on the purpose of the publication (e.g. major reference work versus textbook), the priorities of the authors (not all authors are interested in all features of a liquid book), or the intended business models (which are still in an experimentation phase themselves). Thus, as a (preliminary) result, there's not the "one-size-fits-all" contract template, but rather a "toolbox of contractual stipulations" from which the necessary and appropriate items can be taken for an individual book project (and most probably this toolbox is still far from being complete).

In the following we present "generic" approaches how we tried to resolve each of the above issues and leave the actual phrasing to the rights and licenses departments of an interested publisher. We will now go through the above mentioned single issues and present possible approaches how to resolve them.

- We make a basic distinction between "Editions" and "Personalized Versions". Editions correspond to traditional editions and also to the concept "Edition" in the Liquid Book conceptual model in Figure 2. The whole version tree starts with a root Edition (i.e. its root is the first edition of the book). The current Edition is updated on a regular basis (see below), and each new Edition then corresponds to the new edition of a traditional book (and thus becomes subject to copyright, and gets an individual ISBN). Authors of an Edition explicitly allow usage of their material in Personalized Versions. Because of these dependencies, the creation of possible Personalized Versions needs to be handled already in the contract of the (first) Edition.
- In addition, we allow for Personalized Versions, which can be created based on the current Edition. A Personalized Version starts a new sub-tree, and is a new book in itself (i.e. it is again subject to copyright and gets an ISBN). It has to be fixed, who is allowed to create a Personalized Version, and how much new/changed material is necessary to distinguish it from the corresponding Edition from which it is derived.
- A process is fixed, how new authors can be added to the original author team (of the root Edition), and how authors can retire. It is stated, what happens with the material of retired authors, i.e. can it still be used in new Editions or Personalized Versions. Retired authors still keep their author rights, of course, but a stipulation with respect to copyright is needed.
- Copyright issues are clarified for reuse, not only by the original authors, but also by new authors entering the team. It is also clarified, what kind of reuse is allowed, how reuse is controlled, and what influence the original authors have (especially what the rights remain with retired authors).
- As a liquid book is intended as a living document, a regular update schedule is defined and fixed (i.e. how often does a new Edition need to be created by the authors), how much new content is required, how many iteration steps are required. In addition to traditional books, the stipulations also needed to consider the typical life-cycle of the users, e.g. it makes no sense for a textbook to have more than one Edition per year.
- A process is defined, how the authors distribute royalties also for Personalized Versions (especially those in which they may not have participated). The distribution process must not be overly complicated, as it needs to be flexible, transparent, and easily manageable for both the authors and the publisher.
- In addition to the process defining the retirement of authors and the rights these may keep, there is also a stipulation how possible royalty payments for them have to be treated..
- As the distribution of liquid books may be much more flexible and the life-cycle of a Edition much shorter than for traditional books (and as this needs to be combined with the possibility of a viable business model), appropriate licensing models have to be developed. Thus the book contract needs to take into account both commercial and non-commercial use, distribution through different kinds of media and formats, and derivative rights. On the other hand, usage of the liquid book material in the classroom must not prohibited or complicated.

Our current experience from discussions with several author teams is, that the final contracts resulting from this toolbox may actually look quite different -depending on the product type, the intended usage of the book, the priorities and the size of the author team, and the complexity of the whole project. It may even become necessary to create additional contracts for the Personalized Versions, especially if they can be created by people who are not authors of the corresponding Editions. At this point in time, we still simply lack sufficient experience from projects -both on the contractual and the (later) business level. What we currently have, however, is a good starting point, and we will experiment with the current framework, updating it whenever new results and requirements come up.

## 6.2 Methodological Issues

During the initial phases of the pilot, we have explored the following main methodological issues:

- Define resource entity type categories interesting to be included in the repository. The following entity types and instances are the ones used so far in the pilot:
  - *Resources*. For example, lectures notes, slides, exercises (text and solutions), assignments (text and solutions), student projects (requirement's documentation and code), lecture's videos etc.
  - *Links*. Mainly used to connect exercises/assignments/projects with lessons or sets of lessons and vice versa
  - *Tags*. Two main types: (1) author/user-defined to classify and support search; and (2) author-defined to limit access like confidential tag in order to limit access to a specific author only (for instance for still draft material or for material that contains private information like student's name etc.)
- Authorships: a major challenge in the overall collaborative authoring experience has been to be able to properly define and assess authorships. This issue has a number of distinctive dimensions:
  - ways for easily marking within one author's content own contributions and references to other persons contribution (outside the pool of participating authors). This kind of references must be inherited properly by all derivative version of the same content;
  - tracking and monitoring individual authors initial contributions. At present this is partly supported by the selected repository infrastructure: Dropbox in fact keeps track of the authorship of each addition to the repository in the versioning history, but it does not provide support to compute (neither automatically nor semi-supervised) the overall amount of the individual contributions.
  - tracking changes from minor modification to relevant additions within the same file and from different authors. This dimension is similar to the previous point but focused on the modifications rather than the initial contributions. Also here current tools (Dropbox but also Subversion, Google Docs among others) annotate the authorship and in some cases also store the changes -but do not support metrics to compute the amount of the modifications.
- Collaborative and asynchronous editing styles: We have experienced that in such highly collaboratively editing style it is crucial when modifying content written by other co-authors, to systematically add notes to explain the reason for the modifications. In many cases, this practice led to excessive amounts of notes. Selective hiding and visualization of change tracking comments is a desirable feature in such collaborative editing environments.
- Common agreement on a shared standard style and layout for: (1) graphical artifacts, (2) exercises, (3) assignments and (4) projects. So far, we have arrived at an agreement for graphical artifacts,

while for the other categories the understanding is that we will try to converge on a number of common criteria in order to support on one hand the need for standard style and layout for the Master Versions and on the other hand the personalized view for the Derivative Versions.

### *6.3 Infrastructural Requirements and IT issues*

From both the contractual and methodological issues, a number of specific requirements and IT issues have emerged during the pilot, although none of them is really critical or a show-stopper. The main ones are:

- Easy to use collaborative authoring environment. In Dropbox (but also in similar environments) we have found adequate support for dumping content into a place and share it (as easy as copying a folder). What it is still missing are simple and easy-to-use ways: (1) to automate common tasks (such as tag a number of resources all at once); (2) to support lightweight life-cycle processes (such as the the revision process of common documents); (3) more advanced support for locking artifacts and support the merging of modifications created by different authors in the same artifacts.
- Automated or semi-automated monitoring of authorships. As mentioned in the previous section, the Liquid Book model needs more accurate and quantitative support for the tracking and assessment of the individual contributions to the overall work. Apart from the monitoring tools, also a number of novel metrics measuring amount but also impact and usage of the additions/modification could be useful.
- Enhanced search. Very soon the amount of content present in the repository has reached significant size (at present 2GB). Rapidly locating parts of content that cover same topics has become relevant. Moreover new ways of visualizing changes made (akin to "revision history visualization" features offered by version control systems, e.g. Subversion) are important.
- Support for social computing hybrid approaches to collaborative content creation such as support for crowd-sourcing for specific processes. For instance support the spawning of a "social" process that allows a lecturer to teach a lesson and then have someone (crowd-sourced? A student? A paid assistant?) that takes the recording of his class, writes it down, strips all aspects that are related to being in class (e.g., comments on lesson schedule, etc.), cleans and redraws copyrighted figures, and sends the material to the lecturer for further editing/checking/cleaning.
- Distribution of content in various electronic forms The Liquid Book model is not constrained to the printed or pdf current usage models. Web-based platform as well as the emerging tablet platforms seems to be natural candidates for other usage models. This is typically a responsibility of the publisher, however, much like newspaper on tablets are different, so our (Liquid) book will need to adapt to the new platforms and usage paradigms.
- Support for different media Related to the above point, new and specific requirements are added due the multi-facets nature of the Liquid Book content. The Liquid Book does not contain only text and figures, but also slides, videos, software artifacts etc. hence the need to provide a mix of book-ish structure with links to embedded windows showing videos, slides, etc so that most of the elements in the version at hand (either Master or Derivatives) can become actually active entities.

Finally, we mention that one key aspect experienced is that liquid books bring together the power and richness of having multiple authors and multiple perspectives (the one of all authors) with the speed and flexibility of single-author books. Writing a book with many authors is great but is also a daunting task for many reasons, and it often takes more time (and stress) than writing a book individually. As such, it is often a discouraging proposition. In Liquid Books, authors share material (and so, collaborate, as authors in an ordinary book or in a Wikibook) but each author is then free (within the boundaries of a contractual agreement that we have identified) to take any of the shared material and edit/organize in any way they want, and to then have an own edition of book that leverages the

knowledge of the group but that does not require everybody to share the same view on the content or organization of the edition, and we have experienced the benefits of this reuse and freedom already in the pilot so far.

Being liquid and evolving, a liquid book does not have to be perfect the first time around, it grows with time. This aspect and the one mentioned above together lower the barrier for good prospective authors to get their knowledge out in the public and as such it is expected to benefit authors, students, publishers, and the community at large.

## 7 Conclusions

To conclude, we summarize here the main Liquid Book contributions to overall LiquidPub objectives. With the proposed model we expect:

1. to lower the overall "time to market", since it is easier to add new content and reshape it relatively frequently. If we consider textbooks, for example (our main scenario as discussed above) this adaptation is important as lecturers frequently learn from each year delivery of a course and start revising (modifying present content and/or adding new content) their teaching material in the summer so that they can be used for classes in the fall or spring terms.
2. to lower the barrier for new authors to get to the first edition: quality or completeness will improve over time. Keeping this barrier low is important to get people who have interesting things to say but that are scared by the effort of writing a book to actually start writing one.

In the end the benefit that we envision with the Liquid Book model is:

- to increase the early circulation of innovative ideas, i.e. have more people involved in the process of writing books, including in particular busy but competent people who would not normally put the very considerable effort it takes to come up with a first edition of a traditional book;
- to ensure that the book can evolve over time in an agile fashion (also with limited effort required from edition to edition) and in this way have up to date editions;
- to ensure that it is possible and hopefully easier for contributors to prepare editions that are targeted to a specific group of readers;
- to facilitate collaboration and reuse, as editors can take, edit, and aggregate content from multiple contributors;
- to deliver innovative services and products for publishers, in order to add value to their traditional businesses.

## References

- [1] Benkler, Y., and Nissenbaum, H. Commons-based Peer Production and Virtue\*. *Journal of Political Philosophy* 14, 4 (2006), 394–419.
- [2] Ewing, Bronwyn F. Digital learning objects. *Teacher*. (April, 2009), pp. 30-32.
- [3] Chenu-Abente, Ronald and Giunchiglia, Fausto and Cernuzzi, Luca. From Software to Artifacts: Supporting the Current Scientific Knowledge Needs. Technical Report DISI-10-070 (2010). Dipartimento de Ingegneria e Scienza dell'Informazione, University of Trento.
- [4] Giunchiglia, Fausto and Chenu-Abente, Ronald and Xu, Hao and Kharkevich, Uladzimir (2010). A Metadata-Enabled Scientific Discourse Platform. Technical Report DISI-10-069 (2010). Dipartimento de Ingegneria e Scienza dell'Informazione, University of Trento.
- [5] Ronald Chenu-Abente, Juan Jose Jara, Fausto Giunchiglia, Fabio Casati, Maurizio Marchese. Supporting Scientific Inclusion: Novel Approaches for the Representation and Assessment of Scientific Knowledge Objects (SKO). In proceedings of CLEI (2010).

[6] Gary Hall. Fluid Notes on Liquid Books (2009). Chapter 2 of "Putting Knowledge to Work and Letting Information Play: The Center for Digital Discourse and Culture", edited by Timothy W. Luke and Jeremy W. Hunsinger.

[7] Birnholtz, J. P. What does it mean to be an author? The intersection of credit, contribution, and collaboration in science. *Journal of the American Society for Information Science and Technology* (2006), 57(13), 1758–1770.