The Concept of an Unstructured Book and the Software to Publish and Read it

John Bauer Mengelberg Colegio de Postgraduados, Montecillo, Mexico

jbauer@colpos.mx

Abstract

An unstructured book is defined as a set of digital multimedia files called items, together with ordered subsets of them presented to its readers. Thus different readings of the same set of items are possible. Additionally, items may be shown in different versions and languages. The functionality of the software package SPRP used to publish and read such books is described with an overview of what it offers, in a way which should enable readers to further understand the underlying concept and its possible applications but without any details. The software allows the introduction of reader constraints which will prevent them from seeing certain items, as well as protecting the computer files from unauthorized viewing.

Keywords: e-book, publishing, unstructured, reading privileges.

Introduction

Several years ago, people started to think about the computer as a way to publish – and hence, to read – a book. Some of us regarded this as the creation of a new concept, somewhat more general than a "book" in the traditional sense of the word. This paper describes a result of such an outlook.

Let it be stated that some of the very attractive possibilities of electronic books are mentioned but not described in this paper, since they are not only well known but also have become household concepts. The most significant of them are probably the inclusion of multimedia materials, the "live" references to other published items and very specially, due to its enormous impact on the way a book is now written, the possibility of including links – be they hypertexts or some other form of using the material in non-linear ways. Of course for some readers, the possibility of enlarging letters or having a software product read the texts out aloud may be the main difference. Many of the available software products which allow authors to publish such an e-book list the advantages of doing so, but are somewhat biased in their main objective, which is to maximize the author's revenues as a result of offering his material. For example, in the description of a product called ebook88 (ebook88: ebook Resources, 2002), the list of advantages fails to include

the two features which I pointed out as the most important, but enhances environmental aspects, the cost of producing them and the ease of distribution. However, they included the possibility of making changes or adding more material to the book, which is also near the top of my list of advantages. An illustrative discussion comparing several terms, including e-books meaning the material as well as the device to display it, and e-

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

texts as plain text files instead of the bundled files produced by most e-book products, was found in the article called precisely e-book (Wikipedia, 2007). E-texts were first introduced by Michael Hart in 1971, with the purpose of making all information available to everybody (Michael Hart, 1992). This in turn implied offering the material through the most available software products, and is obviously still the goal of many informers. However, as far as the book industry, many have chosen a different strategy, consisting in the use proprietary software to read the resulting ebooks

After examining the offering of publishing tools, we concluded that the emphasis centered primarily on the economic results of the tool: one of the leading claims was that it would significantly reduce unauthorized use of the materials. As will be seen our approach to the topic is radically different, since we are not too concerned about the economic possibilities. For example, though some of our methods and possibilities could do both, our primary concern is not preventing unauthorized reading of the books but to bar individual readers from seeing something they should not see.

It will become clear that the new possibilities of the unstructured books described in this paper complement, but in no way replace, any of the existing enhancements due to the electronic medium. Neither is their introduction meant to cause a significant change in the publishing world. Its purpose is to offer a different way to communicate with readers of the material, who, incidentally, have a much bigger role in this interaction than in traditional books. Additionally, other uses of the concept are not only possible, but probably will have greater impact than the new types of books made possible by the added features.

The concept we describe was reached in three stages. Initially, I discovered that many factors may tend to bother certain readers of a book, since they often annoyed me. One of them is the way the author quotes foreign language texts, phrases or expressions: a translation replaces the literal quote, is added to the original quote or no translation is provided at all. Of course each one of these forms is totally acceptable and justifiable; however, the irritation it may produce in different readers is another topic. A feeling of inadequacy or superiority, depending on the language and the reader's degree of knowledge of the particular one being used, is quite unavoidable. Formulating a way to eliminate this possible source of dissatisfaction with the book by using electronic publishing, seemed attractive, or at least worthy of study. The idea of offering an option to the reader as to which of the three alternatives he prefers – though actually there is a fourth, consisting in the elimination of the quote altogether – via the inclusion of several versions of the corresponding paragraph, chapter or whatever division could be applicable, arose and immediately became a goal of this type of publishing. Other results of offering these different versions of the same material include the possibility of eliminating offensive material from the view of some readers, minimizing violent descriptions, suppressing certain types of technical or specialized stuff when incomprehensible to the reader and of course, extending this to any other attributes of materials which share the ability to bother or bore some readers, whilst being exactly what others enjoy or even seek. Actually the inclusion of versions and translations has a somewhat more profound motivation, namely that information reaching the reader should be intrepretable during acquisition, a primary requirement in all informing situations (Gackowski, 2005).

The second stage was produced by my desire to help an author who had been collecting material for a book for sixty years, and now faced the almost certain fact that nobody could or would publish his book. Since offering such a huge amount of materials was not possible using available techniques, I formulated the other main idea behind my new books, which is what I call subsetting: the reader is offered different readings of the materials via subsets of the published files.

Finally, the necessity to block certain readers from some subset of the book led to the inclusion of what was called *reading privileges*. Through certain features of the corresponding programs, an

author or publisher can achieve many types of protections of his materials or simply avoid a particular reader from seing certain items: you may think of a child not being able to see adult materials, or a student not having access to the solutions of the problems included in a textbook.

Thus, a "new book" was created and called a U-book – where the U stands for "unstructured", with the possibility to present a material or not, change the order of such contents and enabling the author to offer translated texts or the same materials in different versions, whatever the purpose served by these in their application. It must be stated that no claim is made regarding the novelty or exclusivity of the concepts associated with the U-book. However, after a very extensive search, both in the literature and the description of software products, no similar entity was found. During the process or analyzing existing products and papers, performed *after* creating our own software, we had a number of scares. One notable such instance was the "Build a Custom Book" feature (O'Reilly's SAFARI-U, 2006). However, though the concept of creating a book – especially a textbook - by concatenating articles, papers and excerpts from existing material is quite similar to the U-book's concept of sequences, as far as the way to read them, differences definitely outweigh similarities.

Recently software such as ITUNES and IPHOTO (Apple Computer, 2006), to name just two of them, have become well known and so an ever increasing number of people use the possibility of constructing a list of items to be shown in a certain order: this is exactly the same as the building of our sequences. As a matter of fact, the existence of such tools, that appeared in the market after we had created our U-books, makes the explanation of our concept all the harder, since we are treading on known paths.

Many available products are meant to construct lists of books chosen by their users. Though this also consists of choosing from a list of objects, the way to achieve itdiffers significantly from one product to another. One of them can be found in the childrenslibrary software (International Children's Digital Library, 2006), which we think has implemented a very attractive use of the concepts of choosing and then reading e-books, based on certain descriptors of the books.

The purpose of the creation of our software and making it available to anybody who desires to use it, is to enhance reading possibilities in certain new or different ways. Some other applications, other than "books", will be suggested; others may be formulated by readers who find the concept applicable to some of their needs.

Terminology

The descriptions are based on a series of definitions of the terms. No new words were coined in the process of selecting adequate terms, but some of them were used with slightly or even completely different meanings. A few explanatory comments were included in some of them: they might add to the comprehension of why they were considered important components of our books. Actually when detail is included, it serves the purpose of dealing with a particular subject at this point so as not to mention it in the rest of the paper.

Book: a collection of items from which a reader may obtain sequences.

Book (another meaning, closely associated but not equivalent to its other one): a collection of data files which contain everything regarding that book. Note that this meaning is only slightly different from the one provided in the terminology section. When we need to think of files, we use this one (this is especially true for the author of the book.)

Item: any computer file (or what it contains, if the term is more appropriate in that sense) consisting of text, images, sound tracks, video o any other imaginable type of digital file. Digital presentations, such as enabled by Flash or Power Point, are amongst such files. Viewing an item, also referred to as seeing it: the system displays the contents of the item in the appropriate manner. For example, if it is a sound track, the reader will hear the material.

Sequence: an ordered subset of the items of the book which can be presented to a reader, who will *see* them in precisely that order.

Item of a sequence: the item but only as far as its inclusion in the sequence. The term element of the sequence was considered but later discarded, since the confusion caused by two different uses of "item" in very similar contexts disappears with the use of the package.

Reading (of the book): another term for a "sequence" which can be used when the more technical term might be discouraging or confusing.

Author (s): the person(s) responsible for creating the items.

Publisher: the person who includes the file on disk and uses the software package necessary to offer the material to the readers (via sequences).

Reader: one who uses the software to enjoy a sequence – what used to be known as reading the book.

SPRP (Structured Publishing with Restricted Privileges): the software package which enables its users to publish and read U-books.

Site: a collection of books on one computer (or a network). The term also is applicable to the collection of pages when accessing the materials via the Internet.

Edition of the book: the software allows several editions of the same book to reside simultaneously in the site, so a user can choose the edition he wishes or will use the latest one available to him. As anybody who has written a book, or even a paper, knows, as soon as it is out in print one would like to make certain corrections, additions and changes. Again, one might want to separate the corrected versions for one of many reasons, and the addition of a new edition will allow this, as well as just changing an item in the present edition. The topic of editions will not be described or discussed further, though many of the technical aspects of the package rely heavily on the concept of edition.

Subsequence: a sequence used as part of another one; this can happen recursively.

User: person who may use some or all of SPRP's functions in one site.

User of a book: somebody who may use or update data referring to that book (since there may be several books in the same site).

Some important attributes of an item are presented in a more detailed manner.

Version of an item: different contents or presentations of the same material, which could be formulated as conveying the same message in a slightly or even totally different manner. Different versions may also differ in file type (for example, in a *silent version* a text may replace a sound track.) Just to show different applications, a reference to a page on the www may be replaced by an image which conveys a similar meaning for offline readers. The term version will sometimes be used in a more general context, including what we here defined as version, but also the fact that the language may differ, or the file contains the "light version".

Language of an item: as part of a translation or simply to provide a translated text in any of the languages announced as available for the particular edition of the book. Since SPRP is offered to its users in several languages – actually it is what we call a translatable system, allowing easy translation into additional languages, it is important to avoid confusion between the language of the material of a book, and the language in which a user wishes to use the software package: they are two entirely different and separate concepts. The system may ask you in German – if you

chose to use the programs in that language – "welche Sequenz wollen Sie?" instead of "which sequence do you want?" as would be the case should somebody use the system in English, its default language. But this has no bearing on an item of a book, which may be shown in French.

Light version of an item: a different version usually related to the size of the file, as an image in a lower resolution, or a short preview of a video. Please note that the term version here, as announced previously, is misused since it does not have the specific meaning we defined for U-books, but is used in its generic context. Furthermore, this feature of including a light version can be used in different ways in other applications, since it is an additional dimension.

We separate the next two terms since they serve totally different purposes than the others. They are related to the "readers' privileges" which were considered important enough so as to have been included in the name of the software package. Their use is almost transparent to readers of the books.

Chapter: the most confusing term of SPRP. A subdivision of items for authorization purposes only. A user will obtain permission to read some of these chapters.

Level of an item: An attribute of items used to limit a user's permission to read or see that item, in terms of the maximum level of items which he may see (of a particular chapter). It is possible to use the software for books without using this term at all (should there be no need to restrict readers.) However, it is an essential and extremely useful feature of U-books, as will hopefully become clear later on in the paper.

The U-book

Let us first rephrase our definition: A U-book is a collection of items which allows different readings. A reading consists of viewing, one after another, the items of a sequence chosen by the reader amongst a set of such sequences, or which he built *ad hoc*. The relation of the reader with the book is the set of items he may see – which might or not be restricted due to a number of circumstances – and the functions of the system he may use. Additionally, a reader will have some permits associated to the site, which might be thought of as "all of the books published there".

A book will be modified to become a U-book, where the changes will be motivated by added features of the book itself. After that, an overview of the functions of the package is presented, but the emphasis is on the tree main types of tasks, namely including items, building sequences and reading a sequence. The description of any function or feature of the package will always be from the point of view of the design: what the package has to offer so we can enjoy the U-books, from both sides: the author and publisher, and the readers. One of the purposes of the inclusion of the descriptions is to convey the notion that this is not just an idea: the programs to implement the concept exist and work. We describe it this way since we could not include a demo, as the one we will show in the presentation at INSITE.

A few examples of possible applications of the concept of a U-book, naturally made possible by the corresponding software, are presented before the paper's conclusions.

The Book to U-Book Example

Suppose you have published a digital book, and that you have simply divided it into Chapters – where this means the usual division of parts of the book. In order to avoid confusion, this designation has been changed here to "parts". Without getting technical, further suppose that you used pdf files – i.e. Adobe's Reader will be used to read them. Though this is not all you can do with present software related to the e-books, it will serve our example.

You start getting all kinds of emails about your book, and amongst a lot of flattery and good wishes, several readers complain about foul language, others about pictures containing nudity in Part 3 - a Greek statue, actually –and still others, about both. Since you are using U-books, you decide not to ignore them – as many authors normally would. Therefore you now prepare a new "version" of every part which contains any one of those elements. You will prepare 3 new versions.

Besides the normal or original, call it version 1 = "Original", you will now have version 2 = "No foul language", version 3 = "No nudity depicted", and version 4 = "No foul language nor nudity". Figure 1 presents the sequences that will enable the readers to take advantage of the new versions. The image in Part 3 was now separated from the rest of the material and stored as a separate file, the item we have called 34, for which the author obtained an image of another statue which would not alter the effect on the rest of the material, but was now not unclothed. Please note that though only one such file is necessary, the U-book will use it as versions 3 and 4 of item 34. This is one of the main concepts involved, so an effort to understand and remember it will be useful in the rest of the paper.

PARTITION OF ORIGINAL ITEMS

EQUIVALENT SEQUENCES

#	DESCRIPTION	NEW ITEMS
#	PART 1	11, 12, 13, 14
#	PART 2	
#	PART 3	31, 34, 38

1	ORIGINAL	1, 2, 3
24	EQUIVALENT	11, 12, 13, 14, 2, 31, 34, 38

NEW FILES ARE CREATED WITH NO FOUL LANGUAGE FOR ITEMS 12 AND 14 WITH NO NUDITY DEPICTED FOR ITEM 34

Figure 1. New items and their use in the corresponding sequences

The rest should be obvious: you will let a reader select the version he wishes. Thus, the person who chooses a version without nudity will get to see the appropriate figure or image, and the same happens with the elimination of any offensive language.

The equivalent insertion of the items in different languages will not be described at this point, except to point out that it will allow piecewise translations of a book, and reduce the effort of publishing the translated version considerably.

As was seen here, whole parts or sections are not necessarily stored and used as a single file. Thus, to implement the changes of version, you can construct another file with the necessary changes or divide the original file into several other files, and modify only those for which some such change is called for. The sequence in which items are presented will include several of these that will replace the single file. For some of them, no versions were added, so the only version will always be displayed to any reader. Those items that do have another version will be used when the reader selects precisely that version. Exactly the same applies to the language in which the item will be presented. It should be noted that SPRP uses many types of files, not just pdf or similar ones. In the example, the image files may be stored in any format.

An Overview of the Software Package (SPRP) used for U-books

The goal of this section is to show some concepts of the software which might be of help when thinking about different applications of this type of book. Figure 2 shows a diagram of the users, meaning those who can access some of the package's functions.



Figure 2. The model of the site, its users and books

For the purpose of the system, a U-book is a collection of computer files which contain different messages which, in some order delivered to a reader by means of *sequences*, purport to conform what would usually be referred to as an e-book. These files contain what are called the versions of the items of the book. Figure 3 shows the model for a book with its items and sequences. The diagram is complicated due to the fact that it attempts to show the relation between the files and the pointers to them – the items of the book as far as our package is concerned.



Figure 3. The conceptual model used for the items of the book

The six main groups of functions of the package are enumerated here with a brief comment. The following sections further describe some of them.

Introducing a book (title, authors, options and editions). This creates the files and prepares the the system so it can accept all the data regarding that book.

Authorizing users to use the functions of the package in regard to this book, and include constraints on the items they may see or use. Constraints are formulated in terms of chapters and levels of items.

Adding items of the book, with their descriptions and different versions.

Creating or modifying a sequence, including a brief description of RISP (Relate Items in Structured Publishing), to allow further descriptions to items, as well as relations amongst them and about items which are still in preparation.

Reading a sequence.

Changing items and users en masse. Special functions allow changes to be made in many items or several users; for example when changes in constraints are needed, such as in a course where following the exam, solutions are made available to the students. Similarly, we wish to block some items since the materials are not available (for example, a room of a museum is closed.)

The very simple book described in the example above will be used whenever it is deemed useful.

Authors' and Publishers' Activities

The publisher – who may of course be the author himself - will introduce the book as mentioned above. It might be pointed out that none of the activities require any knowledge of file management or some other specific computer concepts, much less information about the files actually created on disk.

Amongst the options used in regard to one particular book (all are TRUE of FALSE options) we might point out: uses versions of items, will use items in different languages, uses editions, uses chapters and uses levels of items. Some of the options or values selected for the book will be final, whilst others may be changed at some stages of the use of the book. Should the book not use one of the optional features, such as the version, the user of the programs will never be prompted to choose a version or indicate something about them. This is true for all such options.

He will then determine the chapters, in the U-book sense of this word, which is a subset of items. This is a partition of the set of items, in other words every item will be assigned to one of the chapters. Usually this will be done to restrict the user's access to the items in some way, but other uses can be formulated in some applications.

Similarly, he will decide which levels he will use for each chapter. Levels of items are represented in SPRP via a number. In our example, a single Chapter is assumed. There are 2 levels: 1 = text items only, 2 = images.

The author – there may me more than one – must write the book. He will create a set of files, some of them of different file types. For example, in addition to any rtf files or pdf's, he may use text files for plain text, meaning he does not want to include colors, underlining, changes of fonts nor other types of materials. As an explanation of the probable use of this feature, it might facilitate the use of "read aloud" features. He may include images in his files or store them separately; this also applies to sound tracks and video clips. Last but not least, he may use as many Power Point or Flash files as he wishes: they will be presented whenever they are included in a sequence. Naturally the list of file-types presented here is neither exhaustive nor final, since the

package will be dynamic in this sense, that is, SPRP will be able to handle new file types without recompiling the application.

The author may introduce several versions of the same material, or translate any of the files to another language. Of course he will be able to edit his materials any time, as long as he does it in their final files, meaning the ones to be used by readers, or he will have to inform the publisher who in turn will substitute the old files. Please note that none of these activities requires nor uses SPRP.

Thus the author, in a directory he called "MY.BOOK.FUTURE.UBOOK" has placed files which he called part1.piece1.pgf, part1.piece2.pgf, part1.piece3.pgf part1.piece4.pgf, part2.pgf, part3.piece1.pgf, part3.sculpt.jpg, part3.piece8.pgf, where names are only for his own benefit – to remember what he put there. Of course he could write it down – as we will see, not only in SPRP's database, but also with RISP. As an afterthought, he prepared another 3 files: part2.nofoul.pgf, part4.nofoul.pgf and part3.dressedsculpt.jpg. It is important to remember that he may change or add files anytime he wishes: however, they might not reach his readers if somebody doesn't do something about it.

Adding items of the book, with their descriptions and different versions

To update the items of a book, items may be added to, or selected from, a list of items and changed. Perhaps it is worthwhile to point out that an item is actually a set of files, since it may have such files for different versions, languages and what is called the light version of the corresponding files. Thus, an item is described by what it contains and the options which are active for that item. Figure 4 shows the main data items which are used to describe and store the items, including the oft mentioned options.

Some fields are not worth mentioning here. The field Risp-id will be explained below.

The list of items for our little example will not be presented since it is deemed obvious. One of them could be ITEM 31. Description: first part of PART 3 of the book. Item type: text. Chapter 1. Level 1. Uses versions: yes. Uses languages: no. An abstract or further description is optional.

On the contrary, item 34 will be: image of a Greek sculpture. Chapter 1. Level 2. uses versions: yes. Uses languages: no.

The point is that the main information about every item is stored here – actually, in the database tables roughly depicted in Figure 4. Of course the most important datum is the file, but since there might be more than one, the path is stored in the item-version entities (actually this is not true, as you see in the diagram, but you may think of it this way in this example, since no languages will be used.) We like to think of these data as pointers to the files in which the contents reside. Naturally all this technical information is not necessary to use the programs, though in order to use some of their functions, a thorough understanding of the underlying concepts is necessary.

It is important to note that the book is not published in the usual sense of the word, just its items. A reader would have no way of seeing these items, without the sequences which will be introduced subsequently. Actually, the files may even be protected against any viewing except through the software package itself – either scrambled or deformed in other manners to prevent unauthorized use.





HAS LIGHT VERSION, READ-ALOUD

Figure 4. Diagram of an item

RISP (Relate Items in Structured Publishing)

It is easy to predict that the difficulty in maintaining control of the different items will be proportional to the number of such files, or to the length of the period in which their authors create or obtain them. Having several versions of items of course will make it even harder. Though a clever numbering and naming schema may help authors remember and use their items, it is quite probable that they would welcome some way of recording certain information about the files and where they put them, meaning which item corresponds to that file. Enter RISP, which might be succinctly described as follows.

RISP is a way to manage a set of texts and fields that describe items or other objects which may be of use to the author, publisher or whoever wants to construct a sequence, where the texts and some of the fields offer the potential to be "marked" or indexed, in such a way as to enable the use of the object described by the text in any application. We refer to these indices as contextual keywords. The main purpose of "risping" items is to produce easy to formulate lists of items for their inclusion in a sequence. Very powerful operations between these keywords are supported by the package.

An example will have to do the job of explaining this concept, since otherwise the entire system would have to be described. In a museum or a large art gallery, they have decided to use a Ubook to produce all kinds of brochures, guides and even (digital) booklets. Every particular art work is introduced as an item, that is, there is an image of it in the computer. The title of the painting and the painter are included in the item's description. In Risp, we include the period, style, technique, nationality of the painter, school, a description of the subject, the people who are depicted, the source of the object (donated by somebody, belonging to a collection, bought in 1954) or whatever data seems useful. This information can then be used in gueries which will produce lists of the items of the book (in our case, the art collection described as items) which satisfy the conditions or selection criteria. Country: France, Period: 17th century, topic: "landscape" and "sunset" but "no river" might be such an operation. Note that the keywords are used in a certain context or meaning, and that several such words can be provided for the same meaning. RISP uses its own numbering for objects, which might or might not coincide with the item's number – thus the Risp-id field in Figure 4, which creates an inverse pointer. The types of keywords, our contexts, are defined for a particular book. So if we were describing historical documents, we would probably have persons, region of the map, exact date, battle, countries involved and so on. The contextual keywords would allow us to separate Paris, the Homeric personage, from all things which happened in Paris.

Often, when describing a subsetting technique such as the one provided by RISP, people will say that the problem is that, in order to use these descriptors, they have to "be there". Of course this is an accurate statement. However, the computer can use any data it possesses, either explicitly or implicitly, as long as we indicate how to use the latter. In RISP, there is a component – not quite finished and implemented – which enables the insertion of keywords based on available information, which in some cases may be supported by human intervention. Rather than thinking of a completely automatic processes, which if often possible, we like to consider this as a way to significantly reduce the effort of indicating the appropriate values for future searches. One could have the computer show every art work, or those of a previously defined subset, and indicate if there is or not a sunset depicted, or if one can see a river. We refer to such operations as a "mass marking" of texts.

Creating a sequence (or changing one)

This is perhaps the main module of the package, since it puts into practice the concept of the Ubook. Figure 5 attempts to schematize sequences; some of the terms will be described below.

Note that the elements of a sequence are either items, specified by their item number, or sequences, which become subsequences of the one being built. The inclusion of a subsequence is different from adding the items individually, especially since a modification of its elements will affect all sequences which contain it as a subsequence.

Several fields are included in the databases and other files for technical use only, for example to save time or other computer resources when offering the books to a reader. Options of sequences include the possibility to use certain features available to them or not, notably if the sequence may or may not have subsequences or if it will have an index or not.

There are several ways to edit the items of a sequence, since this is the essence of SPRP – it reflects the possibilities offered to the readers. Perhaps one might say that if you can think of a way to edit a sequence, you will find exactly that way of doing it amongst the functions offered by SPRP. However, it may take a while to find it. A group of persons have worked hard, and continue to do so, to make it easier to find exactly the desired function without having to study the package thoroughly, as happens in many software products. Some of the main edit schemes are described succinctly here.



Figure 5. Schema of a sequence

The most obvious way to do this is to start with a sequence – an ordered list of the sequence's items. It might be empty, if the sequence is new. There are functions to add items to the existing list, to move items up or down – remember that it is an ordered subset of items - and to delete them from the sequence.

Adding items to the sequence can be done "an item at a time" or creating a set of items to add in a single operation. The straightforward procedure would be to indicate the item number. However, there are many other ways to do it.

Add items from a (filtered) list of them. Filters include the possibility to limit these lists by asking their description to contain certain word or words, to be of a certain item-type or via other fields. This results in a list of items of the database. For any item, one may display all its attributes and furthermore, ask to view it (read it, see it, listen to it.) Items can then be added to the sequence by choosing them from the list.

The alternate way to add items is to create a set of them – which in this context is called a *list* of items. The operation consists of two steps, namely

- constructing a list of items which is an ordered subset of the items
- then collectively adding these items to the sequence.

Before showing how the list can be constructed, the alternatives for the actual operation of adding the items to the sequence is described, since there are three different ways to do this. It is worth-while observing that the list is available to the program's user interactively. This allows it to be edited – items may be removed from the list.

One can add the items of the list to the sequence, in the position indicated for that purpose. The entire list can be used, or items included one after the other, after confirmation by the user. This operation results in the individual insertion of every item of the list constructed previously. Alternatively, the list may be added as a subsequence – actually a new sequence, which is then added as a subsequence to the sequence.

The list can be prepared from the descriptions contained in the database table, which means that attributes of the items are used to create a subset of these which will constitute the final list to be

added. Filtering and editing in other ways, plus ordering the resulting elements are some of the operations included in the functionality of this module.

The other way to prepare a list is to use RISP – after all, this is the purpose of including RISP-descriptions, as was briefly described above.

Alternative ways of building sequences include copying an existing one and then editing it. This will be very frequent when a reader wishes to build a sequence for his own use. He will eliminate items from the sequence and perhaps add others.

How the U-book is read

Please note that the images presented do not constitute exact screenshots of the package, since these were too large for this paper: they are prototypes where not only their sizes, but also the number of options and details, were reduced for the purpose of illustrating different concepts at this point.

A reader will login and will be offered a list of available sequences, such as shown in Figure 6. The description of the sequence which is selected appears below, in order to enable him to confirm that the choice he made was the one he wanted –but also to avoid having to scroll to see the complete description.



Figure 6. Choice of a sequence in a reading session

Should he choose to modify his options for the reading session, using the selected sequence, he would be presented with a way to do this which we depict in Figure 7. At this point the version should be clear. The language refers to the items which are available in the selected language; any item not published in the selected language will be presented in the default language of the book. Of course the list of languages will only present those in which there are some items available.

Other options simply indicate details referring to certain possibilities offered by U-books to its readers. Knowing the item number, for example, might be useful to some readers whilst others

simply would not know what to do with them. The *complementary texts* consist of any explanation, description or note the author of the book wishes to associate with an item as part of its presentation to a reader. Our favored example is the description of a picture. This complementary text may differ from one version to another (whereas the picture remains constant.) The version for an art expert would indicate dates, perhaps current owner and size. For the general public, it would include the artist, technique, setting and description of the style. Of course, it is easier to assimilate this in reference to the language: the item would be described in German or in English.

Should a book not use versions, as many will not, the corresponding options, indications and selections disappear from all screens. The same is true for other attributes of a book.



Figure 7. Change active options for this reading session

Finally the reading session begins – in most cases, login will result precisely in this step, since the computer *remembers* users and their last sessions, and will offer to restart from the point of last interruption. The system will display, in the appropriate manner, each item of the sequence, as diagramed in Figure 8, where the images suggest that there might be several "pages" per item.

Upon conclusion of the viewing of the current item, normal operation will show the next item of the sequence, but there is a navigation bar which can be invoked to change this action. Items can be grouped in *parts*, and these in turn subdivided into *divisions*, which constitute the elements of the index of the sequence. In a book, these would roughly correspond to chapters and sections. In other collections of items the terms might differ. For a particular application, its publisher may baptize these terms according to the nature of his items. For example, he might divide our museums guide into "rooms" – the parts of the index. The general picture is shown in Figure 9.





Figure 9. Navigation options for the sequence

Some comments about U-books

When asked about the difference U-books will make to the author of a book who would normally have used one of the many software tools available to publish it as an e-book, an answer seems to reflect the entire process quite adequately. Most current software products will "bundle" several files, of different types in most cases, into one single file and the product will deliver it to the reader, though there are many exceptions to this way of doing things.

U-books are not published that way. The publisher produces a list of items which, in computer speak, may be referred to as pointers to the corresponding files. Then he, or others, will construct sequences of these items, which then constitute the basis for the delivery to a reader. There are some disadvantages in this way of doing things, but the advantages far outweigh them.

The author still produces "files", though he will now be able to include versions, translations, light versions or even several different files as substitutes of others. As I often have to point out to listeners of explanations of U-books, the software will not write the books for you. The news is that now somebody will have to record and partly describe his files in a computerized list, typically assigning a number and a description to each of them. Perhaps a slightly different way to look at the possibilities offered by U-books is the fact that often an author has a large amount of materials, which due to their excessive number cannot possibly be included in his book. Now he has a way to offer different "views" of the product of his hard work and research without affecting the readability of the book itself. Thus, one could even say that the main use of U-books is not producing books at all, but different applications of the concept to other types of communication.

At some point in this paper, a reader is bound to ask the question: how do I read a U-book? In other words, what can or should I do to read it? The U-book was created to be read on-line, which nowadays has acquired an "Internet" connotation, but which to us signifies "to read it on a computer with the files being accessible somewhere." The easiest way to understand how to read it is to assume you have all the items (files) and programs (software) on your own computer, since this is equivalent to any actual situation you will encounter. The difference lies precisely in the software and to some extent, in your reading pleasure which might be affected by the time intervals necessary to present the material to you.

The package will be available to readers on their own computers - or local networks - for Ubooks which are to be distributed that way. On the contrary, some of them will probably be offered in a different way, typically viewing the files on a particular site.

However, hopefully many of the readers of this paper will view it as a new possibility of publishing papers or books. In this paper, one must assume that the software is powerful enough to support any way we may want to do this. Again, please use the "your own computer" model as a way to conceptualize the activities involved, which might be synthesized as: including your items and constructing the sequences you wish to present, remembering that somebody else may or may not add other sequences, where this decision is the publishers' privilege.

Some examples of U-Books

Consider a normal textbook, including problems, answers, opinions as to different topics and references to other materials. Of course everything is divided in such a way as to provide the opportunity to include or exclude certain topics or details, and to introduce versions, translations or even "light" versions of some of the items. One could imagine that the publishers of the textbook have provided students and teachers, in addition to the possibility to formulate their own, with several standard readings, such as:

Initial browse - enables the reader to have an overview of the materials to be covered in a course.

Detailed study of a topic: sequences which include everything about the topic. Here there might be 2 versions: the problems without answers, and another one where answers are included.

Final exam cramming version: only what one would normally have underlined or highlighted.

In some or all of these sequences, references may be provided in the traditional book manner (quoted, listed) or the reference may be included as an item (of course only if it resides on some disk, somewhere we can access it.)

Our research and development group is fond of using U-books for the documentation of information systems, where their main contribution is through RISP. Actually we are currently using them to document SPRP and other systems, creating many files containing descriptions, decisions, our data models, screen shots, help files, elements of the user manuals, source code and many other item types. Defining appropriate sequences, we can offer descriptions of anything somebody might need.

The use of subsequences was particularly helpful in this type of use. The typical cut and paste operations we all perform in such instances can now be replaced by a sequence. I recently used the SPRP programs for a presentation scheduled for the next morning, where I was informed that several attendants had no knowledge of the Spanish language. Furthermore, the technical level of the presentation was not clear, and some applications were to be mentioned or not according to the presence or absence of some VIP's. I had a number of Power Point slides in several files and after selecting the ones I could use, I decided to store them individually, meaning one slide per file. Thus, the presentation would be shown via one of several sequences I quickly prepared. I produced a parallel set of files in English, so as to offer my presentation simultaneously in both languages using 2 projectors. The epilogue was unhappy: only one projector was available. And, though I chose the appropriate sequence once I saw the audience, nobody even noticed the way the presentation was prepared.

We are currently working on the formulation of an interactive restaurant menu which makes use of all of U-books concepts, including versions, translations, light versions of the item and constraints. For example if somebody informs the system he never eats red meat, he will never be offered such an "item" - we put them in a chapter where he has no rights, or at a level higher than the non-meat eaters user previously introduced for that book. This example makes extensive use of the dictionary (any available digital dictionary the author or publisher may select) which we may associate with a book in SPRP, since translation and explanation of unknown terms is one of the purposes of the system. It is worth pointing out that one of the as yet unmentioned advantages of e-books is precisely the possibility to offer immediate access to a dictionary at any point. In Ubooks this still is possible, but will be build into the files themselves, much as in any other e.book. The dictionary we mention above is an additional feature, provided by the book and not by its items.

Finally, two products which were planned as subsystems of SPRP lost a lot of their urgency due to the appearance of very sophisticated play-lists and similar programs, as we mentioned when we introduced the topic. However, RISP still makes our products much more powerful, especially for very large collections of photos or music. Additionally, the subsequences are invaluable – though some of their functionality can be obtained through the other available software as well.

A word about the author of a U-book's knowledge of computers

One of the most frequent comments formulated when presented with the notion of U-books – and for that matter, for e-books as well - is made by future authors: "I don't *understand* computers. Now, I have this software product and I just feed it my file and it produces the book." Initially, my comment is that if he has "a file" he probably started with several others, especially if he included some multimedia material in his book. Then he had to choose a product, though a friend or expert usually is available to help him do this without understanding it or comparing it to others. However, he now knows how to use a computer in certain ways – even if only through another computer user.

The knowledge of computers necessary to be able to use SPRP to write a book is the same. He will have to understand the new concepts involved, but they refer to the contents of the book itself rather than the way the computer will handle them. And creating sequences requires very little or no computer knowledge, except some patience to understand the ways SPRP will let him do this. There are several ways to construct sequences, but some are really very simple to understand and use: you just add items to the sequence and perhaps move them around or eliminate them if a trial reading does not satisfy you. Perhaps RISP requires a bit more understanding, since it implies

using a special program, though it is authentically user friendly – meaning almost anybody can use it, enjoy it and, especially, not forget it every time he uses it.

All of the above led us to formulate the concept of "publisher". The functions of what is usually called an author in the e-book literature are now divided into two separate groups. The author will create the files. The publisher will prepare them as items and thus enable somebody (authors, publishers or even readers) to create the sequences. Nobody will be required to perform all these tasks, nor prevented from doing so provided he has the necessary authorization from whoever is in charge.

Additional features

A frequent comment about e-books is that their readers cannot make notes about them the usual way – scribbling, underlining or even book-marking. Of course many available e-book technologies offer certain ways to provide the continuity of some of these habits. SPRP offers readers of a book the possibility to record comments about the material. The user can indicate he wishes to make a comment, which he will associate with the item. A global user can also be included, so that all readers may introduce their comments or share them with others. Both of these functions require special permits which may be constrained to certain readers of a book.

Much of the effort in publishing resides in constraining some functions of future readers. Of course the leading one is the possibility of reading, copying or lending the book or some of its parts to others without the consent of the author. The word piracy keeps creeping up in this context. Again, the consent usually is formulated as some kind of previous agreement or contract, or simply a payment for obtaining it. U-books of course have to take this into account, though they mainly address free publishing. Copyrights and author fees are not a topic of this paper; perhaps my opinion about these issues can be resumed by the fact that no *satisfactory* answer was given to a - cheeky - question formulated to a speaker at a "Copyrights in the digital world" event: Why shouldn't a physician ask a patient from whose foot he removed a thorn, to pay a penny for every step he takes the rest of his life? The italics indicate that this term has various meanings, namely, referring to who needs to be satisfied.

Nonetheless. SPRP contains several ways to hide information contained in files in such a way that they can only be read or viewed through SPRP's own routines. One may also want to prevent the copying of a file to a different computer and very often, the possibility of somebody altering a files contents. This is done at certain levels of confidentiality which are indicated for every item, resulting in different ways we either deform or scramble the data. Incidentally all algorithms used for these purposes were specially designed for this package at the appropriate protection level: the cost of violations should be far greater than the expected benefits. And when we included these features, we thought about some of the different applications, not about the u-book as a generalization of an e-book.

Conclusions

This paper did not delve into the question of e-publishing versus traditional publishing, nor its advantages and the probable future of both of these procedures. Neither does it side with anybody in the quite acute topic of the death of the "real" books. Some of the differences between the U-book and books the way we know them, including the many added features made possible by e-publishing or at least, by using digitalized files instead of hardcopy versions, were pointed out, as well as the way to implement these features.

When presented with the descriptions of my unstructured books, one must admit that they are useful, and probably agree that there existed a void which could be filled by these new books. I hope to have induced some of my readers to define their own applications of the technology. In

entertainment and education, such applications seem obvious enough. In literature, however, somebody will have to come up with some new ideas, or at least adapt some of the existing ways to think of a book's contents and the manner the author communicates with his readers, to fully exploit the added possibilities. As far back as 1963, Julio Cortázar (1963) in his famous "Rayuela" (translated as Hopscotch) wrote a book which admitted different readings by changing the order in which one read its chapters. Nowadays, apparently some authors feel inclined to include several endings of their novels or even let the readers make up their own. We shall see if some other features will appear in the future to enhance, or at least diversify, the pleasure or reading.

To state this in a different way: applications based on subsets and ordering of some of the items are easy to find. Other features of our U-books, such as the possibility to present items in different versions, which leads its possibilities in my preference, may suggest alternate uses of the package. I am working on a fairy tale to provide a good example of this; it was not included in this paper but will hopefully become a part of the demo soon available via internet. Imagine reading a story to a toddler, and then let him "read" it in a different version – typically a lot more graphical, especially if he can't read. Or selecting a shorter version of the tale to tell at bedside, when the spouse is waiting to go to a dinner party!

References

- Apple Computer Inc, Iphoto and Itunes. (2006), *Software product description*. Retrieved February 21, 2007 from <u>http://www.apple.com/ilife/iphoto/</u>
- Cortázar, J. (1963). Hopscotch. *Julio Cortazar's Hopscotch*. Retrieved August 18, 2007 from http://www3.iath.virginia.edu/elab/hfl0117.html
- ebook88: ebook Resources. (2002). *Software product description*. Retrieved November 3, 2006 from http://www.ebook88.com/articles.html#3.%20What%20software%20do%20I%20need
- Gackowski, Z (2005). Informing systems in business environments: A purpose focused view. *Informing Science Journal*, 8, 101-122. Available at <u>http://inform.nu/Articles/Vol8/v8p101-122Gack.pdf</u>
- Hart, M. (1992). *History and philosophy of project Gutenberg*. Retrieved March 3, 2007 from <u>http://promo.net/pg/history.html</u>
- International Children's Digital Library (2006). *Software product description*. Retrieved November 16, 2006 from <u>http://www.childrenslibrary.org/index.shtml</u>
- O'Reilly's Safari Online (2006). *Product description*. Retrieved September 7, 2006 from http://safari.oreilly.com/. http://www3.iath.virginia.edu/elab/hfl0117.html
- Wikipedia (2007). E-book, an article. Retrieved from http://en.wikipedia.org/wiki/E-book March 2, 2007

Biography



John Bauer Mengelberg, after obtaining a degree in Mathematics at the Universidad de Buenos Aires, Argentina, got a PhD in Statistics and Operations Research at the University of Wisconsin, at Madison, where he also taught courses in the area of Stochastic Programming. He has since worked in Mexico, where besides teaching at the Colegio de Postgraduados, a school primarily involved in the field of Agronomy but which has Statistics and of Applied Computing departments, he has held several positions, always connected with the field of Information Systems, in which he has also been a consultant all his professional life. He is primarily concerned with the subject of "systems

that work", a concept he has extended to signify that they work even in abnormal circumstances. He has created and implemented many computer packages, and is currently working on several

software products regarding publishing papers or books in the electronic media, and what he calls subsetting in very large data collections. He has often complained he has to work by himself, and his main interest in attending conferences seems to consist in finding ways to change this.