The Use of U-books to Inform Delegates to a Conference

John Bauer Mengelberg
Colegio de Postgraduados, Montecillo, México

jbauer@colpos.mx

Abstract

Delegates to a conference usually have enough information to allow them to choose the papers they wish to attend. However, some of them might profit from several presentations or orderings of schedules and data about the authors or presenters of the papers, for example including the presentation itself. Thus, U-books – unstructured books, where items are offered as sequences presented one after another to a reader, may provide an alternative. The items of the book include files containing descriptions of the event, the delegates and the papers, and information stored in databases about the same topics and others, such as schedules. Several sequences will present the information in different orderings and levels of detail. The restricted privileges component of the software can serve a delegate to select his areas of interest, thus limiting displays to papers which he might attend. Information about previous instances of the same event, as in previous years, may be integrated. Delegates, who might be given a copy of the software, can update their own copy of the book to include comments on papers they attended, or create a sequence of the U-book with their planned schedule, which might subsequently serve as a memoir of their attendance to the conference. Two new features of the software, along with the previous ones, are described as elements of a system proposed to organize the materials or even as a planning tool, though usually conferences have their own systems for these tasks.

Keywords: e-book, unstructured books, SPRP software package, delegates to a conference, schedules of papers, several presentations of schedules.

Introduction

Attending a conference with one's peers is a special event for many of us. We leave with additional friends, experiences and often, knowledge or new ways to look at things. Though of course the main attractions are the papers presented, several other factors will determine the success of an event as perceived by a particular delegate, as attendees are called in this paper. I have thought about these other factors, and commented on them with colleagues, but especially, tried to assemble remarks and opinions of other delegates to events that I have attended. One of the conclusions, actually a pretty obvious one, is that the degree of interest and effort invested by the organizers of an event regarding DRM (delegate relations management) - to paraphrase CRM, inappropriate since the business association of this concept should not be applied - varies notably from one event to another.

To quote examples of events which practically lie on opposite sides of the scale, INSITE makes every effort to offer delegates timely, exhaustive, well presented and up to date information about papers, authors, schedules, other activities and details about almost everything, from travel and locations to recommendations of all kinds, including a printed and a digital copy of the Proceedings, available to all delegates before the start of the conference (Proceedings of INSITE, 2010). On the other end of the attention grade was a conference where I presented two papers. By the time it started, a Monday at 8 am, I had not been able to find out when I was due to present. At 7:45 that morning, after queuing up for half an hour, I was handed a 2-column page printed on both sides with the schedule which included about 220 papers and several invited conferences. The print was in a rare font, similar to Niagara solid 8, and spacing between lines was about 5 pts., which makes it perfectly legible if offered as a digital file – which it was not – since one can zoom it to double its size. The result was obvious: all sessions were empty; in mine, the chair of the session arrived after the first talk was completed: he was not informed of his duties, and it took him 10 minutes to find the room in the huge university building. After 3 years, proceedings remain unpublished, not to speak of the papers themselves.

To make a difference in an event such as the latter is too easy, so I decided to shoot for bigger game: what could be done to enhance my experience of attending even a very well organized and delegate-driven event such as INSITE? Since I did not have the technical details of their systems and the way they put them to use to offer us everything we need to enjoy the conference, I had to imagine some of the details and invent others.

Though many factors will determine the satisfaction or opinion of a delegate, I focused on paper selection: two or more papers attract my attention, but they will be presented simultaneously. My preferred approach to decision making, though it is a bit on the pessimistic side, is Savage's minimax regret (or minimax loss) principle (Blackwell & Girshick, 1964; Savage, 1951) so regrets play a huge role in the grade I assign to my participation in such an event, notwithstanding many other agreeable experiences. And regrets usually concern papers missed, though they might also reflect the opposite. I have found out that many others feel the same way.

Besides the description of the papers contained in their abstracts and access to the papers themselves, one may use additional information to make choices amongst papers. Expert advice ranks high in this aspect, where an expert is a colleague who has either read the paper thoroughly or can provide some information about the presenter. Since everybody is in a certain sense a Bayesian, being able to accumulate experiences to affect certain probability distributions also helps, so the proposed system allows the inclusion of previous events, especially other instances of the same conference. This will allow modifications of Savage's minimax regret decision rule, especially using Hannan Regret (Hannan, 1957), which exploits both the experts' advice and one's own experience in similar decision making.

I compare the choice of one of simultaneous papers to attend with picking a horse at the track. Each of us has his own strategy, so that we need the data in different presentations. For example, I use information about the odds of the participants in their previous races, besides their performances. Since this is not too common, gathering such information from a racing form involves a lot of work, so I usually am frustrated in two dimensions: besides never picking a winner – an outcome practically independent of the strategy – the failure to assemble and process the necessary data in a short time bothers me even more.

Extracting certain phrases from Gackowski (2005) that apply to the use of information for a choice between alternatives, data or information values or any combination thereof should not only be usable but also useful. It means they can be used effectively, for which they should be interpretable during presentation (Wang & Strong, 1996). Usefulness, however, is contextual, depending heavily on the situation. For designers of decision support systems, any data/information is useful that changes the outcome of decision situations under consideration. So offering diverse presentations of the information about papers seemed attractive to me, and led to the design of a system that, with little additional effort, will produce several ways for us delegates to choose the right papers.

U-books (Bauer Mengelberg, 2007) are, essentially, collections of items, which may be thought of as files containing multimedia objects, and sequences, through which ordered subsets of these items are offered to a reader. The software product that implements them, called SPRP (structured publishing with restricted privileges) besides being designed precisely to allow different orderings and levels of detail for a particular use of the materials of the conference, offer easy ways to eliminate certain papers completely from consideration, for whatever reason one might have, but mainly because they concern a field of knowledge different from our interests or for which we feel we lack the background to comprehend the paper.

Though U-books were conceived to offer different readings of the same items, it turned out that they could well serve a purpose such as the paper selection decisions. The attempts at using these books to enhance the experience of delegates to a conference or similar event resulted in design difficulties which led to the addition of certain features to U-books. The purpose of this paper, besides describing the announced tool for delegates to a conference, is to show the impact of these changes on the type of applications of U-books.

First, U-books are described, including the features that were added to the original version of the software package SPRP. Next, the use of U-books to enhance a delegate's experience at a conference is explained, as well as the database used to include information about the papers and their authors and the time slots in which they will be presented. The construction of sequences, and the ways in which the book can be used by organizers and delegates is followed by a brief analysis of the usefulness of U-books in this context and the conclusions. A somewhat more technical explanation of how queries to a foreign database are included as items of the U-book constitutes the Appendix to the paper.

U-books and the SPRP Package

Introduction

Several years ago, people started to think about computers as a means to publish and read a book. An electronic book or e-Book is a text and image-based publication in digital form produced on, published by, and readable on computers or other digital devices (Gardiner & Musto, 2010). This definition does not - explicitly - include some of the new possibilities offered by digital books. 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 way to access 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 emphasize the maximization of the author's revenues. For example, in the description of a product called ebook88 (ebook88, 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 did include the possibility of making changes or adding more material to the book, which is also near the top of my list of advantages. This allows a book to offer current information, a very desirable attribute of an informing process (Gackowski, 2005). Of course the concept does not apply to any type of book, but it is vital for processes that offer information, precisely one of the uses of U-books we are describing in this paper.

In “E-book, an article” (2007) we found an illustrative discussion comparing several terms, including e-books meaning the material as well as the device to display it, and e-texts as plain text files instead of the bundled files produced by most e-book products. E-texts were first introduced by Michael Hart in 1971, with the purpose of making all information available to everybody (Hart, 1992). This in turn implied offering the material through the most available software products, as is still the goal of many informers. However, as far as the book industry, it seems to have chosen a different strategy, consisting in the use of e-readers (or e-book devices) defined as a portable electronic device that is designed primarily for the purpose of reading digital books and periodicals. These devices have gained acceptance far beyond expectations, so that new models and brands with expanded features, as well as products for use on smart cell phones, are offered to the rapidly growing population of electronic book readers. The products include Kindle (2011) and Nook Color (2011) but there are many others. An up-to-date comparison of available devices can be found in (Comparison of e-book readers, 2011) from where the definition of these devices was also reproduced.

Some of us regarded the above mentioned possibilities – and several others - of digital publishing as the creation of a new concept, somewhat more general than a *book* in the traditional sense of the word, which would eventually replace most printed books, not just due to distribution and ecological considerations, but precisely because they add new dimensions to the information that may be transmitted to a reader. U-books are a result of precisely such an outlook: they intend to take advantage of most of the new features made possible by the fact that they will be read on an electronic device. Additionally, the concept was formulated as an informing tool, and thus, follows Cohen’s sentence definition of Informing Science (Cohen, 1999): The fields that comprise the transdiscipline of Informing Science provide their clientele with information in a form, format, and schedule that maximizes its effectiveness. The informing tool we describe in this paper is very client-oriented, so many parts were defined following Birdsall (2009) and Gill and Bhattacherjee (2007).

As stated in the original paper bout U-books by Bauer Mengelberg, they complement the existing enhancements due to the electronic medium. Their introduction is not meant to cause a significant change in the publishing world, but 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 any new types of books made possible by the added features.

No claim was 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 of 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.

The concept we describe was reached in three stages. The initial idea was to offer parts of a book (or other material) in different orderings. Of course one must refer to Julio Cortazar’s Hopscotch in this context: the book offered several readings by changing the order of its chapters (Cortazar, 1963). U-books actually offer items to be rearranged, not entire chapters.

Next, versions and translations of the same item were included, pursuing the goal that Gackowski (2005) formulated as “information reaching the reader should be interpretable during acquisition, a primary requirement in all informing situations”.

Finally, the necessity to block certain readers from some subset of the book led to the inclusion of the *restricted privileges* component in the software product. Our primary concern is, rather than preventing unauthorized reading of the books, to bar individual readers from seeing something they should not – or do not want to - see. One may think of a child not being able to see adult material, or a student not having access to the solutions of the problems included in a textbook. On the other hand, a reader may not enjoy maps depicted in a history book or explanations of some item that are superfluous for an expert on the subject. An examination of several publishing tools showed that the emphasis centered primarily on the economic results of the tool: one of the leading claims is they would significantly reduce unauthorized use of the materials. I subscribe to the theory of free and unlimited distribution of knowledge,

Thus, a new book was created and called a U-book, where the U stands for *unstructured*, indicating that the parts (called items) of the book do not necessarily have a place or meaning within the book. Thus, in a particular reading (we call them sequences) only selected items are included in a specific ordering.

The use of individual items as part of the materials shown to a particular reader also enables an author to offer translated texts or the same materials in different versions, whatever the purpose served by these in their application. For example, an explanation or description of certain technical materials may be displayed differently to an expert or a beginner.

A few years ago, software products 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 persons use the possibility of building a list of items – in this case, a playlist or photo books - to be offered in a certain order: this is exactly the same as building a sequence of a U-book. 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: people will not listen to added features or they reject them automatically.

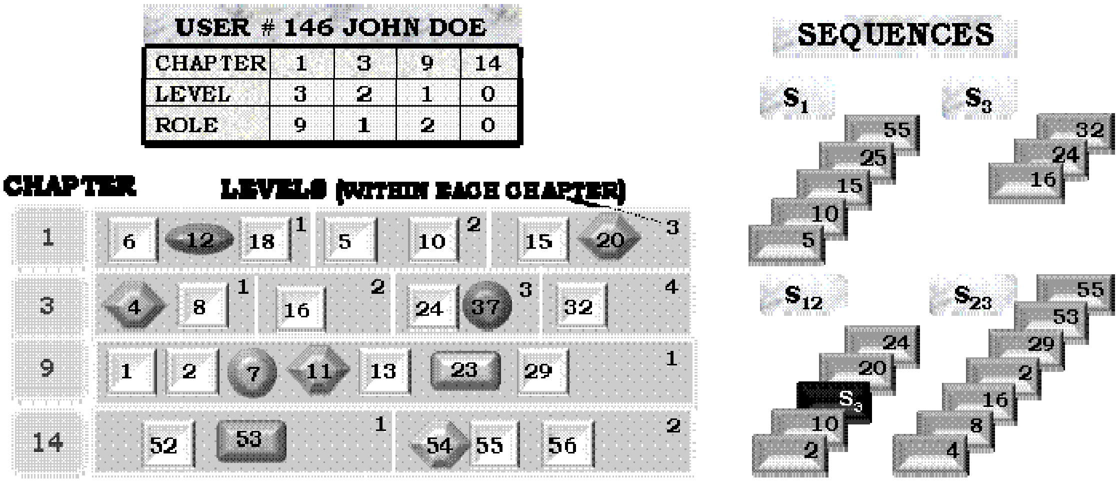
As a result of applying the concept underlying U-books to different types of texts, books or other uses, such as the preparation of class materials, presentations, shows, catalogues and many others, several features were added, mainly through changes and additions to SPRP, the software package that prepares and delivers U-books. They were included for different but converging purposes, including mainly: to increase the variety of items of a book; enhance the way to present it to its readers; and provide additional aids to those preparing sequences. Most of the new features included in SPRP were due to shortcomings that surfaced when U-books were used in different areas, admittedly most of them associated directly or indirectly with academic activities.

A description of U-books

Figure 1 and the explanations that follow attempt to synthesize the main concepts of a U-book, which can be described succinctly as a collection of uniquely numbered items and sequences offered to readers, and how users may be constrained to some of the items. The importance attributed to these constraints which, incidentally, can be voluntary or not, is reflected in a part of the name given to the software package (with restricted privileges).

In the sequences depicted in Figure 1, sequence S_{12} has sequence $\mathbf{S_3}$ as one of its elements, so we refer to the latter as a subsequence of the other one. Thus, S_{12} will present items 2, 10, 16, 24, 32, 20, 24 in that order. Notice that item 24 occurs twice in the same sequence: there is no reason to limit an item to appear only once in many instances. Subsequences can simplify the construction of many sequences, especially long ones such as a text-book, or even a presentation. Further details about this topic are furnished in the corresponding section below.

A user has a level in every chapter of the book: he may not see items that have a higher level. In the example depicted, user #146 may not see items 24, 37, 32 since he has level 2 in chapter 3, or any of the items of chapter 14. Thus, User #146 would see only items 2, 10, 16, 20 when he reads S_{12}. We use the terms read or see items of a sequence, though actually the reader may be listening to an audio clip or a video clip with sound.



USER # 146 JOHN DOE

CHAPTER	1	3	9	14
LEVEL	3	2	1	0
ROLE	9	1	2	0

Figure 1. The main concepts of a U-book

It is essential to point out that the term *chapter* was used with a totally different meaning than the corresponding term in a book. It was adopted because e-Books are often offered to the readers (buyers) in releases: one chapter is sold or offered free, and subsequently the interested reader will purchase the other chapters. This misnomer was caused by our inability to find an alternate designation for such groups of items. Since the term is pretty much transparent to most readers, users of U-books will soon overcome the initial confusion caused by this terminology.

Additionally, in every chapter, a user has a *role* that indicates which functions he may perform with items of that chapter. Roles are: 0=nothing; 1=only read; 2=update items; 9=assign roles of that chapter to other users. For example, User #146 is practically the "owner" of chapter 1 (he can do everything) whereas he is just a reader of chapter 3.

Next, Figure 2 shows what the term *item* indicates in U-books. The main concept is that it is a description of some type of material that may be used in a sequence, but the content itself is stored in one or more disk files, which may reside anywhere, that is, on the same computer, a local network or a remote site. Since the same material can be included in several combinations of languages and versions, an item is not necessarily a file. This is perhaps the most confusing concept of U-books, so we have decided to explain it further using an example though, since it reflects a particular case of a very broad variety of applications, it may lead the reader to a partial interpretation of a much more general concept.

Suppose that Smith, a famous person, is described. An item is included in the database, and assigned the number 24. The name of the person is used as the main identifier of the item, and a description in a free text format may be provided. At this point we will ignore reader constraints, so we will say the item is in Chapter 1, level 1. The type of item must be chosen from a catalogue of such types – which is created for the U-book. Suppose that the type "famous persons" is included, so we will use it. Of course, it could also have been biographies, since this is what the item will contain. Since the book uses 2 languages (1, the default, English and 2, Spanish), biographies of Smith are included in both languages. In English, a very good biography is available in the form of a slide presentation, especially meant for children under 12. For adults (version 1) another such presentation is found, but it is rather lengthy, so the author of the U-book decides to use one of the slides as a *light* version of the presentation. He does the same for the Spanish version. The term light version will be explained in the section devoted to a reading session.

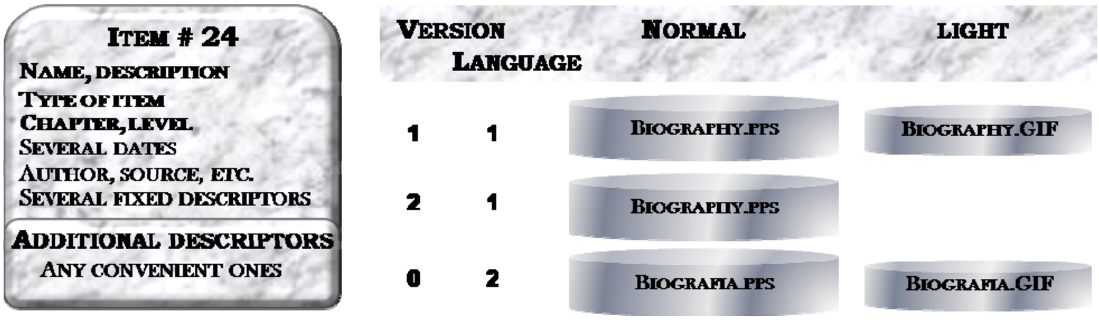


Figure 2. An item of a U-book

The point we wanted to make here is that an item not necessarily corresponds to a single content, but may have several of these associated to its use in the U-book. These different files make it possible to show different readers the same item in diverse ways, according to the selected version and/or language. Notice that files corresponding to different versions of the same item need not be of the same nature: one version may be an audio clip, whilst another version could be a text. This would enable a read-aloud file (using a print to voice product) to replace an image, or an image or text to be presented instead of an audio clip.

Each of these combinations of language and version of an item may in turn be included as two files, where we call one of them the *light* version and the other one, the normal version. For example, there might be a very heavy map (in terms of size) and present a reader with another one – the light one - with less detail: he will be shown the “normal” version if he asks for it. Of course this concept can be used in many other ways. The normal version of a paper may be the full paper, whereas the abstract could constitute the light one. Or the normal file may be a further explanation of the light one: think of a photograph as the light file, and its detailed description as the normal one.

Notice that the term version is used both as referring to an item, as well as to a particular file of such a version of the item. Experience with several users of the software indicates that this only bothers the builders of sequences the first or second time, since the difference is fairly obvious once one has used items that contain versions in both these senses. To avoid confusions, the programs used to update items make a very marked distinction between the light version of a file and the versions of an item.

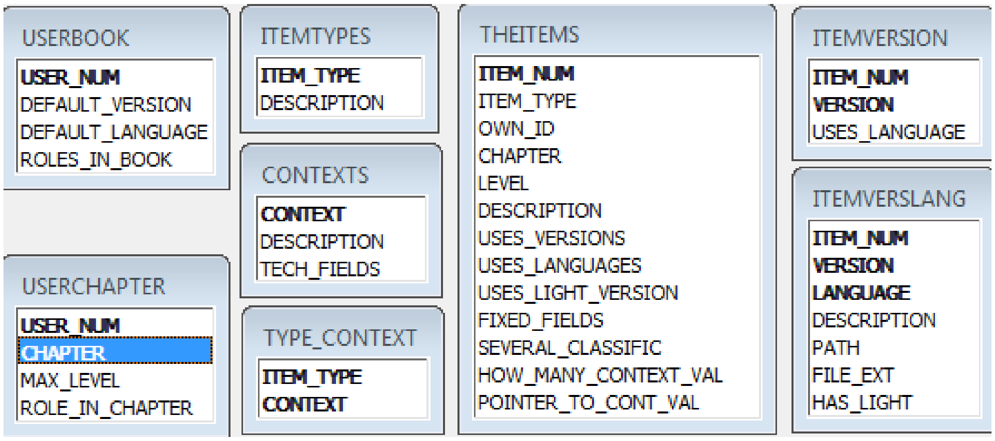
Another component of the software addresses the same objective as the versions (that it be interpretable upon acquisition), though with a slight difference: the software is available in several languages, so that a reader may choose the language in which he wishes to communicate with the computer, making the acquisition itself easier or even possible. For this purpose, the software is translatable (Bauer Mengelberg, 2001): anybody can translate the user interfaces to any language whose alphabet use 256 ASCII code without any knowledge of databases, programming languages or other technical concepts, but especially without gaining access to the source code, which could inadvertently be modified causing malfunctions.

Other desirable attributes of an informing process are its relevance and interest of the client of the information. A reader of a U-book may use chapter and level constraints to bar certain non-interesting items, of course referring to his particular interests.

The most difficult attribute to offer refers to avoiding excessive task complexity (Gill & Hicks, 2006). As is often the case, flexibility may result in added complexity: more features and similarities between them may cause the product to be quite difficult to use. Both the provider and the client of the informing process were taken into account in the design of SPRP, and constant changes reflect previous failures to achieve a certain degree of simplicity as well as improvements in this direction. The authors of a book will have to know how to use certain special components of the programs, or rely on someone else - we refer to them as editors - to perform the necessary tasks. Anybody who has published a book or prepared an e-Book will relate easily to this concept. But, alas, the reader is the main stakeholder, and it is really easy to read a sequence.

How SPRP Stores Information about Users and Items of a U-book

SPRP uses several databases. The main one of them is where items are stored. Only some of its tables and fields are depicted in Figure 3, but they reflect the main concepts of the actual data model. Several tables and many fields were omitted, and some of the fields were made up since the description of the original ones would have been lengthy and unnecessary in the context of this paper. Amongst them, all fields that only serve technical purposes or the translation of the software product, as well as everything regarding security – essentially confidentiality and protection against unauthorized viewing - were not included in the description of the model. Therefore the fields and tables described should not be compared with other descriptions of the model, not only in past papers such as the often cited one about U-books, but also others that may be published in the near future. The inclusion of the description of the data model is due to suggestions and criticisms by reviewers, some of which I tend to regard as specific petitions.



USERBOOK
USER_NUM
DEFAULT_VERSION
DEFAULT_LANGUAGE
ROLES_IN_BOOK

USERCHAPTER
USER_NUM
CHAPTER
MAX_LEVEL
ROLE_IN_CHAPTER

ITEMTYPES
ITEM_TYPE
DESCRIPTION

CONTEXTS
CONTEXT
DESCRIPTION
TECH_FIELDS

TYPE_CONTEXT
ITEM_TYPE
CONTEXT

THEITEMS
ITEM_NUM
ITEM_TYPE
OWN_ID
CHAPTER
LEVEL
DESCRIPTION
USES_VERSIONS
USES_LANGUAGES
USES_LIGHT_VERSION
FIXED_FIELDS
SEVERAL_CLASSIFIC
HOW_MANY_CONTEXT_VAL
POINTER_TO_CONT_VAL

ITEMVERSION
ITEM_NUM
VERSION
USES_LANGUAGE

ITEMVERSLANG
ITEM_NUM
VERSION
LANGUAGE
DESCRIPTION
PATH
FILE_EXT
HAS_LIGHT

Figure 3. Some of the tables and fields of the database used to store items of a U-book

In the THEITEMS table, fixed fields are the same for items of any type (though their interpretation may differ). The “several-classific” field actually consists of several fields, both numeric and alphanumeric, that can be used as descriptors of items.

However, items may need certain additional descriptors, either to locate a particular one, or to include it in a sequence. So SPRP, via a component called RISP (relate items in structured publishing) allows additional information about an item to be included as part of its description using *context-value* pairs. In RISP, the contexts used by a book are defined, and then, for a particular item, one or more values may be attached to some of them. Examples of contexts include: subject of an image, level of expertise necessary to understand an item, the source of information which provided the item, additional descriptions of the content, and - of course – any others applicable to the particular book. This allows different descriptors to be added to items which differ in nature or usage. Referring to our delegate book, a paper, an author, a schedule, a session or a social event will be described using the appropriate attributes for each of them. Some details about RISP were included in the original paper about U-books.

As many context-value pairs as desired or needed to describe an item may be added. Since these pairs are stored separately, that is, not as part of the database, the Pointer-to-cont-val field contains a pointer to the location of the array of such pairs, and the How-many-context-val indicates how many of these pairs were added to the item.

For every item, the actual files are described in the ITEMVERSLANG table: its records contain the path of the file corresponding to every combination of version and language, as well as an optional description that can be shown as a caption when the file is displayed as an element of a sequence. The "has-light-version" indicates that there are 2 files. The filename of the light one is always the same as the normal version, but the suffix ".LT" is attached preceding the extent. SPRP of course uses many such "tricks" which perhaps makes it more efficient, but definitely a lot harder to explain in detail.

SPRP offers books as part of a *site*, which has its own database: think of a publishing house that offers several books. Users are registered and assigned a site-role in this database. For every book of their choice, additional data is stored in the USERBOOK and USECHAPTER tables for that book. A user has a role in every book (for example, he may or not create chapters, include other users or build *public* sequences) and a role in every chapter of the book. The actual model is somewhat more complicated, but the fields shown as part of the tables should suffice to show how a user is constrained to certain functions of the software and use of the data itself.

Reading a Sequence

The reader of a U-book is a user, that is, he has gained access to the book. When he starts a reading session, he will choose one of a list of available sequences. If there are too many, he may *filter* this list in several ways, typically by their name or description, but also by the author of the sequence (who built it) or some other classifications which further describe them.

Once he has chosen a particular sequence, he may change certain reading options, especially the choice of a language and version: he will see the items of the sequence in the chosen version whenever they are available; otherwise items will be offered in their default versions. SPRP also *remembers* readers of the book: at the start of a session, a reader will be offered the last sequence he used, and he may even indicate to restart where he left off.

A reader will be presented the elements of the sequence one after another. He may invoke a navigator, including an index of the sequence which can be built for it, to choose the next element he wishes to see. However, he will never be shown or offered an item which he may not see due to a chapter or level constraint.

Finally, a reader can comment elements of a sequence, both for his own use or – optionally – also for the use by others. It is important to note that these comments will not refer to the items themselves, so that they will not apply to the same item as an element of another sequence.

Building a Sequence

Elements of a sequence are either items or other sequences, included as subsequences, as was pointed out before. This feature is recursive, that is, a subsequence may in turn have its own subsequences. For thorough readers, I point out that SPRP detects loops that may result from this feature. The construction of a sequence consists of adding or deleting elements and reordering them whenever necessary. At any time of this process, one may *try* the sequence.

Elements of the sequence are presented as a list, and deletion and reordering are performed on it by selecting an element and indicating the desired operation. There are essentially three ways to add elements to a sequence.

- An item can be added (or inserted anywhere) individually indicating its item number.
- A list of items is prepared, using a filter or relations based on their descriptors. The filters are based on combinations of conditions formulated using the main classifications of items, such as the type of item, one or several chapters and a minimum levels of each of them, and the fixed fields that are used to indicate other attributes of an item. Relations are formulated based on the context-value pairs.
- An existing (previously defined) sequence is used to add elements to the one we are building. The sequence can be added as a subsequence, or the list of its elements used as if it were a special case of the filtered lists of items.

From the list of items, there are once again three alternatives:

- All the items of the list are added to the sequence. The user may indicate the position in which he wants to insert them.
- Elements of the list are selected and added to the sequence.
- The file corresponding to each item of the list is displayed one after another to the user, who decides if it should be added or not. If an item has several files (that is, more than one combination of version and language) the default version is displayed.

The data model used to store the sequences is not shown, since it would not provide relevant information to this paper. However, it is important to point out that an element of a sequence has other specifications, besides the 2 main attributes: what it is (an item or a sequence) and the corresponding number. Other (optional) attributes are the following, where the default is always “it’s up to the reader preferences”:

- The element should be shown in a specific version or language regardless of the choice of these by a reader;
- The associated element (explained in a separate section) is to be displayed simultaneously or alternately;
- If the element is an item that has a light version, the order in which they will be offered: Light normal (this is the default value of this option); only Light; only Normal; or Normal – Light,

Notice that understanding all these details is in no way essential for the comprehension of the main topics of this paper. Also, the same features that are used to build lists of items for their possible inclusion in a sequence allow users to locate (find) a particular item.

An example of a list of items

Suppose the items of the books refer to objects in a large museum. There may well be hundreds of thousands of them, if descriptions, documents, invoices, certificates and other materials are added to the objects themselves. To build a filtered list, a user would indicate what he knows about the item, via its descriptors, and the RISP component – added to the functionality offered by the relational database itself – will help him find the item. RISP performs logical operations on subsets of items. For example, one could ask for the list of all items of the book that satisfy a condition such as: (year = 1923 or 1924) AND (type of document = certificate) AND (country =

Italy) and (Source = Kingsley) and (Source = John), where some unnecessary brackets were used to increase the readability of the statement. Note that an item can have more than one value in the same context, as if the case in the context “source”, where John Kingsley was indicated as 2 values of the same context. RISP will build the lists of items that satisfy each of these conditions and then perform the appropriate intersections or unions of the subsets represented by each list. Obviously the use of the descriptors of the items has a prerequisite: they must have been furnished! Precisely this last observation led to the inclusion of queries to foreign databases as items of a U-book, described below, to provide an alternative for situations in which including the descriptors represents an excessive work load.

Some of the New Features of U-books

The associated element of an element of a sequence

A new component of sequences was added to the original product: it is now possible to associate to an element of as sequence another element, which can be an item or another sequence. This means that when one item is shown to a reader, he is either offered to switch to the other one, or shown both at the same time. In the latter case, the system will use a split screen, or, if available, a different monitor (or projector) to show the other item.

A couple of examples might eliminate some of the confusion caused by these descriptions. A famous painting is displayed, and the biography of the artist may be added as an associated element of the one representing the painting. Or suppose the item shown is a video clip. One may associate a text (e.g. the lyrics of a song) or even a sequence of texts or images: as the video progresses, these texts are displayed. Actually, this example was a result of a paper by Whatley in which she concluded that, besides requiring well produced video clips, ”providing text alongside the video may be useful for some students in supplying a structure to the lecture content” (Whatley & Ahmad, 2007). The thought of offering an – apparently – necessary complement to the video gave rise to the inclusion of the associated element to an element of a sequence.

A thorough reader of this paper might suspect that this is the same as presenting the normal version after the light one. There are two differences (as far as reading the sequence is concerned): the associated item may be a sequence, and it may be displayed simultaneously with the original element of the sequence.

Actually, the associated element of an item may be a different version (or language) of the same item. Remember that a reader does not see *an item*, but the contents of one of the files referenced by that item. Think of a text book of a foreign language, where the student can switch between the texts in his own and the foreign language. More about this topic will be discussed when the conference U-book is described in a later section, since it makes use of this new feature, which was added because it may facilitate the preparation of the items of a U-book in many instances, though perhaps at the cost of confusing a first time user of the product.

Queries to foreign databases as items

Queries to databases other than the U-books own database can be included as items. This feature was added for two totally different purposes, where the second was really an afterthought. As was pointed out, in order to use a descriptor of an item, it has to *be* (I prefer this term to the usual *exist*). In many instances, this could involve a lot of work or even programming some kind of interface, if the ones provided by SPRP were insufficient.

Since items often will be imported from other systems or databases, we included the possibility to use the descriptors provided in their sources, without having to include them as attributes of the items in the U-book. Thus, a user may use a query formulated to this source – which in the current version of the software must be a relational database – to obtain a list of items of the book. The query is formulated and included as an item of the U-book, so it can be used individually (just executing a query) or as an element of a sequence. A query may have parameters: in this case, the user of the query will be prompted for their values. For example, a reader wishes to obtain a list of all delegates to a conference who work in Australia. He would use the appropriate query, and when asked "for what country" he would indicate his preference.

This proved to be advantageous when U-books are used as sources of information or live presentations. The use of queries for the conference U-book will be described in the section dedicated to them, whereas the Appendix to this paper shows how queries are included as items of a book.

Self-imposed constraints when reading a sequence

In many cases, a reader will want to block certain information, that is, he does not wish to see it. Since a normal reader (meaning he has a limited role in the book) cannot change his roles or levels in chapters, the new version of the software offer readers of a sequence the possibility of reducing their maximum level in certain chapters; these changes will only affect the particular reading session, not his profile as a user of the U-book. For example, the reader decides that "now" (in this particular reading session) he does not wish to see anything of chapter 5 (whatever that means in the particular situation) even though his level in that chapter allows him to see all its items: he just changes his maximum level for chapter 5 to zero.

Elements of a sequence may be grouped to be displayed jointly

Elements of a sequence may be grouped so they will appear together, as if they were a single item: a set of texts are shown together, or a photo-gallery is assembled from several items. Observe that grouping is only possible if the files to be displayed have the same *extent*, and only for some types of these, since you – obviously -cannot group video or audio clips, or presentations or other files that require another software product to show them, such as a pdf or a Word file.

The Conference U-Book

A Wish List of a Delegate to a Conference

As announced, an information system consisting of two main components was conceived to offer delegates to a conference the information they need about papers, schedules and authors in several presentations. The database used by the organizers to plan the event is complemented by a U-book which will be updated using the information contained in that database, as well as several files prepared specially for this purpose. Both of these components will be described below.

The U-book itself, as well as a partial version of SPRP, may be distributed to the delegates who wish to use it, besides the "public" copy available on computers connected to the network where the software is installed and the data updated by the organizers. Thus, delegates can update certain elements of the U-book on their own copy to reflect their interests and preferences.

As in any system, one must decide what it should do. In this case, the objective was to define several presentations of the information that could be useful to delegates, each one designed to satisfy one or more specific information needs. This would lead to two sets of requisites: the items to be created, and some of the sequences that could be offered to the delegates, besides such as those they could build for themselves.

Some terms related to the sessions were defined *a priori* and used in the U-book. Thus, sessions are numbered consecutively in the day (relative to the start of the conference) and the room in which they take place, and their starting and finishing times are furnished. Hence, there may be several “Session 2” on day 1: they differ as to the room in which they will take place. *Previous papers* refer to those presented in previous instances of the conference, if such information were available. Finally, papers will be presented in a slot, numbered within the session.

The probable interests of delegates were formulated as a wish list, where an attempt was made to include all imaginable – or perhaps, likely – attitudes of the users of the sequences. A delegate may want to compare information about certain papers, or he or she may want some detailed information, either about a paper, or its author(s). Please note that I shall not use two-gender designations, since I always think of “he”, “his” or “him” that way, and ask my readers to do the same. The descriptions in this list roughly indicate the content of a possible sequence or query prepared for that particular use.

- Papers (grouped by field of knowledge and topic)
- Papers by session, and slot (and room)
- Papers by room, session, slot
- Papers by author
- Papers and previous papers by author
- Papers and previous papers by research area and author
- Authors (alphabetical order)
- Authors (by research area and topic)
- Authors by session
- Authors by country or affiliation
- Authors by their photograph
- Authors by previous instances of the conference.

Additionally, it should be easy to find a specific paper by any of its attributes, including the title, research area, topic or any of its authors. The same is true about authors, including some way of finding a delegate if one only knows his face.

To properly inform delegates, but also for other purposes, there are three main tasks performed by the organizers of the event: 1) Assembling all data about it in a database; 2) preparing the schedule of presentations of the papers; and 3) adequately informing delegates of schedules and last minute changes. A U-book may be useful for the third, but in order to offer it to delegates, the first two tasks have to be performed, and then the items of the book must be created using the information gathered and generated previously. The preparation of the items should involve the least amount of extra work possible, as this is an important issue when evaluating the usefulness of the additional offering to delegates.

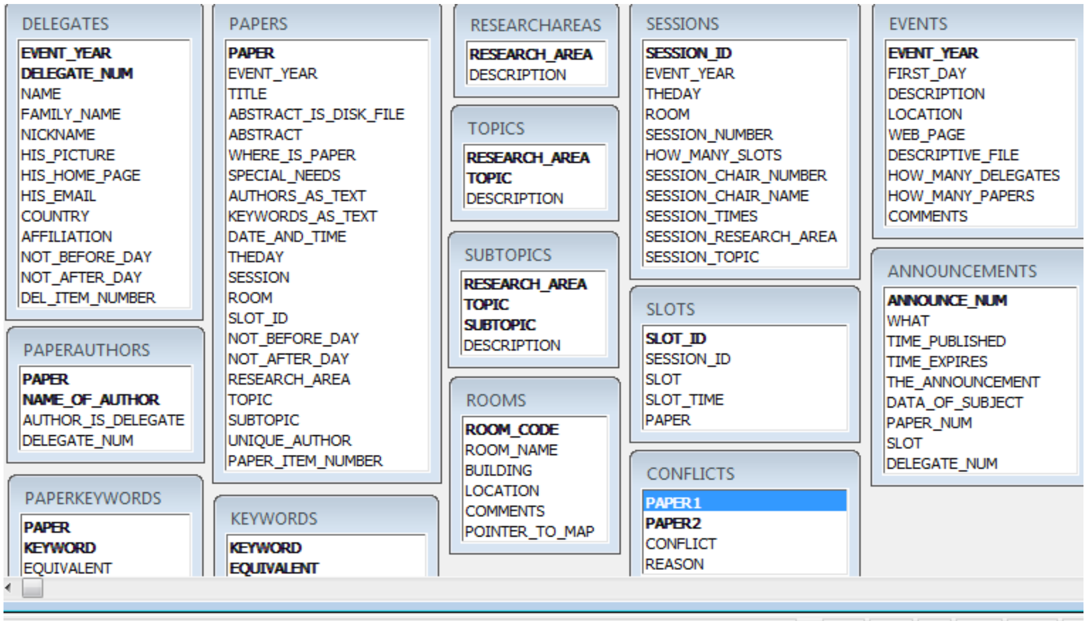
The Database with the Information about the Conference

In order to gather the necessary information about the papers to be presented during the conference, and consequently use it to schedule them for their presentation in work sessions, the use of a database seems the best alternative. Though its structure will be chosen by the organizers, either by a specific design or by the use of one of the many software products available for this type of event, one can think of a typical model, conceived precisely to contain all the information needed by delegates to choose the presentations they would like to attend, besides serving the planners of the conference. Figure 3 shows its tables and their fields. Notice that relations were not shown, and that bold faced fields constitute the principal indices of the tables. The data about registration and payments of delegates was not shown. The information that reflects constraints for the scheduling of papers was left in the model, though it is not used by the U-book and will not be commented further.

Though the names of some fields are self-explanatory, others may require a comment. Descriptions are grouped by tables, but fields that include ITEM-NUMBER in their names deserve a separate mention. As will be seen, delegates, papers, sessions and slots will become items of the conference U-book. Since the numbering rules for the future items were defined *a priori*, the eventual item numbers can be included as fields in the database.

DELEGATES: Note that some “authors” of papers will not have a delegate number, since they did not attend the conference. Delegates may have been assigned different numbers in previous events, and so the number of a delegate is always qualified by the year to which the information relates.

PAPERS: a unique research area, a topic and – optionally – a subtopic will be determined for each paper. Since a paper may have several keywords and authors, these are included in separate tables. Additionally, all authors and their affiliations are included as a single “Authors-as-text field”, since often their names will be shown in this manner. Similarly, all keywords of a paper are included in a “Keywords-as-text” field. The “where-is-paper” field contains the name of the path of the file containing the full paper; of course, it may be null or void if the paper is not available or is not offered on-line at this time. The abstract is included as a (memo) field of the table. Should additional information about the paper be available, for example, the full presentation or the handouts, this would not be included in this database. Instead, an item of the U-book would be created and there, in the U-book data, this item would be related to the item reflecting the paper. The slot allocated to the paper, as well as the date and time it will be presented, are added as a result of its scheduling in a session.



DELEGATES
EVENT_YEAR
DELEGATE_NUM
NAME
FAMILY_NAME
NICKNAME
HIS_PICTURE
HIS_HOME_PAGE
HIS_EMAIL
COUNTRY
AFFILIATION
NOT_BEFORE_DAY
NOT_AFTER_DAY
DEL_ITEM_NUMBER

PAPERAUTHORS
PAPER
NAME_OF_AUTHOR
AUTHOR_IS_DELEGATE
DELEGATE_NUM

PAPERKEYWORDS
PAPER
KEYWORD
EQUIVALENT

PAPERS
PAPER
EVENT_YEAR
TITLE
ABSTRACT_IS_DISK_FILE
ABSTRACT
WHERE_IS_PAPER
SPECIAL_NEEDS
AUTHORS_AS_TEXT
KEYWORDS_AS_TEXT
DATE_AND_TIME
THEDAY
SESSION
ROOM
SLOT_ID
NOT_BEFORE_DAY
NOT_AFTER_DAY
RESEARCH_AREA
TOPIC
SUBTOPIC
UNIQUE_AUTHOR
PAPER_ITEM_NUMBER

KEYWORDS
KEYWORD
EQUIVALENT

RESEARCHAREAS
RESEARCH_AREA
DESCRIPTION

TOPICS
RESEARCH_AREA
TOPIC
DESCRIPTION

SUBTOPICS
RESEARCH_AREA
TOPIC
SUBTOPIC
DESCRIPTION

ROOMS
ROOM_CODE
ROOM_NAME
BUILDING
LOCATION
COMMENTS
POINTER_TO_MAP

SESSIONS
SESSION_ID
EVENT_YEAR
THEDAY
ROOM
SESSION_NUMBER
HOW_MANY_SLOTS
SESSION_CHAIR_NUMBER
SESSION_CHAIR_NAME
SESSION_TIMES
SESSION_RESEARCH_AREA
SESSION_TOPIC

SLOTS
SLOT_ID
SESSION_ID
SLOT
SLOT_TIME
PAPER

CONFLICTS
PAPER1
PAPER2
CONFLICT
REASON

EVENTS
EVENT_YEAR
FIRST_DAY
DESCRIPTION
LOCATION
WEB_PAGE
DESCRIPTIVE_FILE
HOW_MANY_DELEGATES
HOW_MANY_PAPERS
COMMENTS

ANNOUNCEMENTS
ANNOUNCE_NUM
WHAT
TIME_PUBLISHED
TIME_EXPIRES
THE_ANNOUNCEMENT
DATA_OF_SUBJECT
PAPER_NUM
SLOT
DELEGATE_NUM

Figure 4. The tables and fields of the database

EVENTS: the current event (year) will be included here. If information of other instances will be offered, additional records are created.

SESSIONS: a unique session-id is computed as follows:

Session number = theday * 1,000,000 + session * 10,000 + room * 100 where theday 1 represents the first day sessions are presented, rooms are numbered consecutively (note that their names or designations are also recorded) and sessions are numbered within each day and room number; thus, once again, there may be several sessions "1" on day 1, one for every room in which there is such a session. For example, the second session of the first day in room 3 has session_number = 1000000 + 20.000 + 300 = 1,020,300. Notice that we use the field name THEDAY since DAY is a reserved word in SQL.

The name and delegate number of the session chair are indicated for each session. So are the starting and finishing times of the entire session and the research area, a topic and – optionally – a subtopic of the papers to be presented in the session.

SLOTS: they are also assigned a unique number (slot-id), by adding the number of the slot (in a session) to the session number. Observe that all session numbers end in "00". So 1020301 is the first slot of session 1020300 (note that we can refer to it by its session-number).

The other tables are catalogues, built according to the criteria of the organizers and/or the subject matters of the papers. The exception might be the KEYWORDS table, so it is described separately.

KEYWORDS and PAPER_KEYWORD: This table may be populated dynamically as keywords appear in accepted papers. Actually, as shown in this system, we prefer to include a standard keyword for searches, so we establish equivalence (for the purpose of this conference only) between certain keywords and a standard word used to convey that meaning. The keywords of a paper will be indicated as such, but the system will see if there is a standard term for that keyword indicating the same meaning; if not, the equivalent word will be the same as the keyword furnished by the authors. Notice that both are included as fields of the table. However, the combination of both of them must be valid, meaning there is a record in the Keywords table that has precisely those values.

If a delegate wishes to find papers that have a particular keyword, he will use the standard version. This will enable him to find papers that have the terms "education" or "teaching", if in the framework of the conference the organizers consider them as referring to the same topic or knowledge field. Actually, in many conferences papers may be admitted in several languages, so that this equivalence will enable retrieval of papers with the "same" keyword, but in different languages.

ANNOUNCEMENTS: here any relevant announcements are posted. The alternative is to place this table in a separate database, instead of as a table of this one. Since this affects neither its use nor its understanding, it was included in the conference database in this paper. It may be used by the U-book (an item will be created consisting of a query to this table) or a separate program will be made available to delegates. Several of the table's fields are optional, that is, they will not be used either because they do not apply, or because it was not deemed necessary to populate them when an announcement is posted as a record of the table. A possible use of this feature would be to include a program on the computers available to delegates and allow them to invoke a query to this database: it will display last minute announcements, and the program might offer filters should there be a considerable amount of last minute changes or other important news.

The Design of the Conference U-book

The design of the U-book consists essentially of determining what items must be included and how they will be presented as sequences. However, SPRP offers an additional feature: a user may formulate a request to find a particular item. Thus, he will be offered a list of queries defined as items (explained below) and choose the one he needs to obtain specific information. Besides, since queries are stored as items of the book, they can be included in sequences as well as invoked individually.

The elements of the design will be presented in three sections. The main items that will be created are described first. Then some of the sequences created for delegates are shown, followed by a list of queries that could be prepared for them. This is followed by a description of how a delegate can "read" sequences, use the restricted privileges features of SPRP to limit certain sequences to elements that might interest him, and build his own sequences to reflect either his choices of papers or to record those he actually attended.

The main items of the U-book

We started the design of this U-book by defining its main items. Some assumptions were made about the conference. Observe that the resulting numbers are not a limitation of the system: they were only used to show a typical numbering scheme to make things easier for the organizers, and to shorten some of the examples.

- Papers of the conference are numbered 1 to 19,999.
- Delegates are numbered in the range 1 to 50,000.
- The conference will use 4 rooms.
- A session will have a maximum of 3 papers.

Furthermore, as was mentioned before, the *day* is relative to the start of the conference, the *room* is a consecutive number assigned to every room, *session* is the session number (in that day and room), and finally the *slot* refers to the position of the paper in that session.

For every paper, there will be an item with the same number as the paper, consisting of an rtf file reflecting the title, the authors, the abstract and the keywords. Using images for these items is also possible, if these were easier to prepare than the text files, for example by scanning existing printed documents. The research area, topic and subtopics may be added. Finally, the slot in which it will (probably) be presented is also shown. Later changes of schedule will not be reflected in these items, but will be informed to the readers whenever they access the U-book. These rtf files will constitute the "light" version of the item's files. The reference to the full papers will be added as the "normal" version of the paper items. Thus, when a paper is shown as an element of a sequence in its light version, the full paper may be offered to the reader with a simple click on the "show full paper" button. If available, photographs of (all) authors of the paper will be included as items with an item number obtained by adding 20,000 to the paper item.

An item is created for every delegate to the conference: its item number if obtained by adding 100,000 to his delegate number. The items will represent a file containing the name, affiliation, a biography and a photograph of the delegate. However, the photograph may be included as the light version, and the full description as the normal one (or vice-versa). Notice that non-registered authors will not have such an item: their photographs may be seen directly in the full length paper.

Session and slot items will be prepared as follows. For every session and for every slot, there is an item, where its number was not only described above, but included as a field in the corresponding record of the data base. For sessions, the date, room and starting time, the chair and the research area will be displayed. For slots, the starting time, title of the paper, its keywords and the full author text will be shown. As will be seen in the next section, sequences may use the paper item instead of the slot item: they contain similar information, but the abstracts are included in the items reflecting the papers; besides, they offer the photographs of their authors as an option.

Items can also be created for coffee or lunch breaks and other events, so they may be included in sequences. All such items will be in a separate chapter, so a delegate can decide not to see them every time he uses the sequences that contain them.

Observe that information about the event itself, locations, dinners, outings and similar were omitted in this description: their inclusion involves essentially the same activities as the ones usually performed to divulge such information (posters, notices or announcements).

The basic sequences that will be offered to delegates

We will "build" some of the sequences by listing their elements, and for each of these, indicate what they will show: an item, a query (actually as an item) or another sequence. A name of the sequence is furnished, so that a reader can choose a particular sequence from the list offered for that purpose. In SPRP, these names can be indicated in several languages. Every sequence is also provided with a (non-translatable) description – dedicated to the builder of the sequence, but also available to a reader when its name is not sufficient to distinguish it from other sequences.

The sequences for the conference book will use the new features described before. This is not surprising, since they were partially introduced precisely for books that, rather than provide a normal reading, are used as sources of information. Thus elements of a sequence may have an associated element (another item or sequence). Furthermore, in some sequences elements will be grouped to be shown as if they were a single element. Though grouped elements may not have individual associated material, the group itself may have an associated element.

The numbers assigned to these sequences do not reflect any particular order or have any other meaning. However, we divided the sequences into 3 great groups for our own convenience, as might well occur in any application, where a numbering scheme helps organize tasks.

Group 1 to 100: Sequences with a number in this range are meant to provide information somehow connected to schedules, that is, papers to be presented in the slots of the work sessions. The basic sequence, denoted by Sequence #1, is a reproduction of the conference schedule usually available to delegates. It will have a group of elements for every session: its first element is the session itself, that is, the item created for the session, containing the date, time and room, the field of knowledge or research and the session chair. The other elements of the group are the slots of that session, that is, the items created for slots corresponding to the papers to be presented. This is the "usual" schedule, but shown one session at a time, ordered by day, session number and room number. If sessions consist of many papers, and the information does not fit on one screen, the builder of the sequence will divide them into 2 or more groups. Every group has an associated element: it is a sequence formed by the items corresponding to the papers, not, as in the group itself, the slots. If a reader is not interested in the papers, he will simply skip the associated element, that is, he will not make use of the command button that allows him to see the details of the papers to be presented in the session. The papers cannot be shown simultaneously, as on a split screen, since grouping of items will usually use the full size of the monitor.

In Table1, the first elements of Sequence #1 are shown as an example of how a sequence is built. Sessions are indicated as theday/session number/room number, and for the slots (papers) the slot number is added to the other numbers. The paper numbers are only shown as a reference to see how sequences 101 and 102 would be defined.

Table 1: Sequence #1 – the usual presentation of the scheduled papers for a session

Position	Group	Type of Element	Associated elements			Description	Paper #
In Seq.			What?	Number?	Show		(informative)
1	1	ITEM	Seq	101	ALT	Session 1/1/1/0	
2	1	ITEM				Slot 1/1/1/1	245
3	1	ITEM				Slot 1/1/1/2	612
4	1	ITEM				Slot 1/1/1/3	183
5	2	ITEM	Seq	102	ALT	Session 1/1/2/0	
6	2	ITEM				Slot 1/1/2/1	754
7	2	ITEM				Slot 1/1/2/2	92
8	2	ITEM				Slot 1/1/2/3	200

Sequence #101 is built with the 3 items corresponding to the papers that will be presented in the slots shown as part of the session. Remember that the item number assigned to a paper was the number of the paper in the conference database. Thus, sequence 101 would have elements: Item (245), Item (612), Item (183) and of course sequence 102 would be similar contain the 3 papers to be presented in session 1/1/2.

The reader will be presented with all 4 items of group 1 on the same screen. There will be an adequately labeled command button (SPRP offers a simple way to provide the label) with which he can invoke the details about the papers of the slot, so that the description of the papers included in Sequence #101 will appear, one after another, on his monitor.

Sequence #2 is almost the same as sequence #1, but sessions are ordered by room, day and session number. This is equivalent to the schedule which could be posted outside the corresponding room. Observe that this sequence is not illustrated by a table.

Sequence #3 is another slight variation of the previous ones: here papers are displayed ordered by day, session, slot number and room, so that a delegate can compare simultaneously presented papers. The first part of this sequence is shown in Table 2 using the same notation as above. We have assumed there are 4 conference rooms. Observe that the “sessions” do not refer to a room in this sequence (this is indicated by a value of 0 in the room-number position). In order to offer the description of the sessions, 3 additional items describe the starting and ending times of the 3 slots of the first session of the first day, but omitting the room number and the chair of the session.

Here the elements of Sequence 135, for example, will be the items corresponding to papers to be presented in the first slot of the first session of the first day, ordered by room number. Notice that the papers to be presented in rooms 3 and 4 were not shown in Table 1 (which showed the first sequence) for the sake of brevity. Also observe that sequences 1, 2 and 3 use the session and slot items, not the papers themselves, whereas sequences 101, 102, 135 and 147 use the items corresponding to the papers.

Table 2: Sequence #3 – Schedules to see simultaneous papers

Position	Group	Type of Element	Associated elements			Description	Paper #
In Seq.			What?	Number?	Show		(informative)
1	1	ITEM	Seq	135	ALT	Session 1/1/0/1	
2	1	ITEM				Slot 1/1/1/1	245
3	1	ITEM				Slot 1/1/2/1	754
4	1	ITEM				Slot 1/1/3/1	1001
5	1	ITEM				Slot 1/1/4/1	1002
6	2	ITEM	Seq	147	ALT	Session 1/1/0/2	
7	2	ITEM				Slot 1/1/1/2	612
8	2	ITEM				Slot 1/1/2/2	92
9	2	ITEM				Slot 1/1/3/2	980
10	2	ITEM				Slot 1/1/4/2	456

Sequence #4, #5 and #6 are almost the same as sequences #1, #2 and #3 respectively, except that its elements are not grouped. The item corresponding to the session is followed by the items corresponding to the papers presented.

Table 3: Sequence 4 which shows papers one after another

Position	Group	Type of Element	Associated elements			Description	Paper #
In Seq.			What?	Number?	Show		(informative)
1		ITEM				Session 1/1/1/0	
2		ITEM	ITEM	NN1	ALT	Item 245	Paper 245
3		ITEM				Item 612	Paper 612
4		ITEM	SEQ	MM3	ALT	Item 183	Paper 183
5		ITEM				Session 1/1/2/0	
6		ITEM				Item 754	Paper 754
7		ITEM				Item 92	Paper 92
8		ITEM				Item 200	Paper 200

Every paper will have a way to display its authors' biographies and photographs, and offer a link to the full paper. The way this is done is to display the light version of the paper (title, abstract, keywords and authors) and allow the "normal" one to be invoked via the corresponding button (this would be the full paper). The associated item might be used to display the photographs of the authors. Table 3 depicts the first elements of sequence #4. Item NN1 would refer to the photograph of the (only) author of Paper 245. Seq. MM3 refers to a sequence whose items are the photographs of the 3 delegates who are authors of paper 183. Remember that only photographs of authors who are also delegates will be available.

The next range of sequence numbers, 101 to 200, was reserved for sequences that provide information regarding papers and authors, regardless of their schedules. For example, sequence #109 could be: papers of a certain research area, or even a given topic. Individual sequences could be prepared for every research area, and then another sequence could offer, one after another, those sequences as subsequences. As will be pointed out in a separate section below, since we recommend that chapters are defined for every research area, a delegate can indicate that some of the papers are not to be shown to him, by excluding himself from the corresponding chapters. Furthermore, one may provide the sequence with an index, which would allow the reader to find the papers of a particular research area. In every sequence, papers would be displayed as described in sequence #4, that is, one paper after another with the possibility to access the full paper (if available) and/or the picture of the authors.

Yet another sequence could contain all papers presented by authors from a particular country or university. A sequence I would particularly appreciate is one that will show the pictures of the delegates and, as associated items, the paper they will present (or a sequence of their papers, if they will present more than one). This "photo" sequence would profit from an index by country.

Finally, the range of sequence numbers 201 to 300 was reserved for sequences that include (or consist exclusively of) queries as items. Some examples were included in the section about these queries. Once again, technical details are omitted at this point. Suffice it to say that the tasks involved in the preparation and update of the data (contained in the foreign databases) as well as the specific SQL statements to obtain the information and other descriptions of the query, will usually be performed by technical staff, though a reader may be allowed to do so as well.

Building sequences for the delegate U-book

The steps to build a sequence were outlined in a previous section, so only some very specific comments will be made here. Specifically, to include an item as en element of a sequence, the easiest way is to indicate its item number. Since often the number is not known by the user, the procedure would be to obtain a filtered list of "candidate" items and select the desired ones from this list. The way to define the filters, however, requires some explanation.

When adding sessions, for example, the easiest way is to define a filter based on the item type "session". Personally, I would start the definition of Sequence #1 in this manner, adding that I wish to order them by session number. Then I would add all the session-items to the new sequence. Next, ask for item type "slot" (by changing the filter) and ask the list to be ordered by slot-id. Now add the slots corresponding to every session immediately after the session item. Once this is done for all sessions, group the elements of every session: each group includes the session and its slots. Now there are two alternatives: either store this sequence (save it though it is not totally defined), and proceed to define the other necessary sequences; or build these sequences beforehand. In either case, the final construction of Sequence #1 will consist in adding the sequences corresponding to each session as an associated element of the group for that session.

To build the individual sequences, invoke a list with a filter by item type "paper" and ordered by the slot in which they will be presented. Observe that there is a prerequisite to be able to do this. As part of the creation of the items for the papers, the data from the database has to be transferred or indicated individually during the process. SPRP offers several ways to "import" data from other sources, though some of them will require certain skills to adapt them to a particular data model from which the information is being obtained. More about this will be discussed in the "evaluation of the proposed system" section below.

For every sequence, a name and (if possible) a description should be indicated. Both of these serve the same purpose: to allow a user to choose the correct one from a list of sequences, though the description is meant for the builder of the sequence. We can at this point ignore the translation of the name to furnish them in other languages, but it would be an added activity should these translated names be a requirement or an advantage for a certain event. Observe that the system automatically records who built the sequence: it uses the user's U-book number for this purpose.

Finally, the sequence can be provided with an index, consisting of two levels of grouping: SPRP calls them parts and divisions. The list of elements of the sequence is used for this purpose, and with a very simple operation the break points are indicated. If names are to be assigned to these parts and divisions, SPRP offers an awkward way to do this. The names have to be introduced in specific catalogues (one for divisions, another for parts) and then the parts of the sequence are baptized choosing a "valid" name. This is necessary to enable readers in other languages to be offered the names of the parts in their preferred language.

Once a sequence is built, it can be used as a source to define other sequences. A copy is made and assigned a different sequence number, and its name is changed. Then its elements can be changed in any way, that is, by deleting and reordering them. Of course, other elements can be added. This provides a very efficient way of building sequences #2 and #3 starting with sequence #1, though care must be taken since the associated sequences will be different.

As for sequence #4, which is similar to sequence #1, except that there is no grouping and the slots are replaced by the items referring to the papers, a trick is useful. Starting with the elements of Sequence #1, first add the associated sequence (of each group) as an element, immediately after the session itself, and remove it as an associated element of the group. Observe that you could add each of the papers individually with exactly the same effect. Next, eliminate the items corresponding to the slots of that group. After you have done this for all groups (sessions), remove the groupings themselves.

How sequences are presented to a reader

A reader logs on to his U-book, using his user number (actually, he will also be assigned a log-on or nickname, as we call them, to sign on.) Usually he will have to provide his password (this requirement can be overridden, for example, for a delegate who is using his copy or the book). He will be offered a list of available sequences for the book, as was explained before. After choosing a sequence, he may be prompted for certain options: in the case of the conference book, no version or languages were included, but the reader may want to change some other options. He may with to specify how to "end an element" and show the next one: the default is an OK button, but he may indicate different actions to produce the same effect. For example, any key pressed could cause the next element to appear. There is a "previous element" button, but one may specify that a given key serves this purpose (one would typically assign the "page up" and "page down" keys to show the next or previous element, or the usual left and right arrow keys. None of these keys is reserved for other actions, so they are all available.

One may also invoke the navigator described as part of the showing of a sequence. This feature offers the usual options (first, previous, next, last element) but the index of the sequence may also be called upon. Even if there is no index, the next item to be displayed can be chosen from a list of the elements of the sequence. This feature is also useful to provide a glimpse of the contents of a sequence.

Whenever an item has an associated element or is shown in a light version, the appropriate buttons will appear on the screen and the user can use them or ignore them. Associated elements in the delegate book would appear as "alternate" elements, though somebody might conceive of a use of simultaneous displays using a split screen: the item will appear on the left side of the screen, whereas the associated item will occupy the other side. When the associated element is a sequence, the "next" item will refer to the next element of that (the associated) sequence: the left part will remain unchanged.

How a reader may constrain what he is shown

An important observation is that an item is never shown to a reader that does not have the necessary level in the chapter containing the item. Since for a book such as the conference book delegates may reduce their levels in the chapters of the book with a very simple operation, they can use this feature to block papers in any sequence. Additionally, they might be allowed to change the chapter and level of items in their own copy of the book.

As an example of how to use this feature, some possible chapters are defined in Table 5, where some *suggested* (that is, what I would do) use of the levels was also indicated. The current version of SPRP allows a maximum of 64 chapters to be defined.

Table 4: Suggested chapters for the delegate book

Chapter	Level	Name or usage
1 – 29		Research area
	1	Papers that do not require specific expertise
	2	Only for experts on the subject
30	**1**	Author data
31		Sessions and slots
	1	Days I can attend
	2	Days I cannot attend
32	**1**	"recycle bin": items moved to this chapter will not be seen
40	**1**	Queries regarding papers
41	**1**	Queries regarding authors of papers and delegates
42	**1**	Other queries
60	**1**	Papers I (the owner of the copy) attended
61		Papers I might attend
	1	Papers I "must" attend
	2	Papers I am very interested (but not a must)
	3	Papers in which I am mildly interested
	9	Papers that I will not attend (perhaps I wanted to)
62		Announcements

Remember that a delegate can only change items in his copy of the database, so the changes will not affect other users of the U-book.

Initially, all sessions are included with level 1 in chapter 31. A delegate can move sessions to level 2 if he will not be able (or want) to attend the session. Of course one could do this for rooms instead of days, but not both (an item has a unique level in a unique chapter).

Similarly, a delegate could "move" (change their corresponding items' chapters and levels) all papers that interest him to chapter 61. Personally, I would use the 3 levels indicated for the chapter. By changing *my* level in this chapter to 2, for example, I would no longer be shown papers in which I am not at least very interested. Note that by using this grouping of papers I am loosing the possibility to use the research area as the corresponding criterion.

Though the current version of SPRP does not offer this feature, two additions will be made in the near future. The first concerns the index of a sequence: a reader may indicate he does not want the items of a part or one of its divisions, and for the remainder of the sequence, no elements of that division will be shown (this operation will not offer an undo, so if he has regrets, he will have to invoke the sequence from the beginning.) The other planned addition is to allow the builder of the sequence to include certain inclusion-exclusion criteria: the readers will be offered certain options. For example, a reader may indicate that he will not want to see anything that takes place in room # 3. Once again, these features are not available in the current versions of SPRP, where a reader may achieve similar results by changing the sequence (probably in a copy of the sequence made *ex profeso*).

It might be stated that precisely the many options - and the way the elements are displayed accordingly - are one of the benefits of using U-books, instead of other ways to present a series of objects to a reader. Though almost everything can be done preparing an image, a pdf file, or a presentation, with the appropriate links and navigators, usually this involves much more work - and possibly, knowledge – on the part of the person preparing the materials. For ecample, it is quite difficult to include the features that will cause certain elements to disappear from the material shown to a particular reader.

Changes of a sequence

A reader of a sequence may change a sequence as long as he or she is the author of the sequence. Additionally, as mentioned repeatedly, he may ask for an identical copy (of his own) of any sequence and then change it in any way he needs. Actually, changes implemented in the current version only allow deletion of elements or associated items of elements during a reading session: addition and reordering of items require the use of the normal program to build sequences.

Queries as items of the U-book

One of the new features of SPRP is that it handles several types of queries formulated against a foreign database, that is, not the U-book's database itself. Since this is supposed to be a non-technical paper, some details regarding queries were included as an Appendix. They are not necessary to understand the use of the queries, and some readers might not wish to determine which descriptions to skip.

Dynamic queries are performed when an element of a sequence is an item that is actually a query or whenever a reader wishes to *see* an individual item simply by asking for it. The item will consist of an SQL statement, and several other specifications, especially the parameters included in the statement. Other data concerns how the results will be shown (as a grid, a list or others) and the column headers needed to understand the output. A trigger may be defined: as a result of the selection of a row of the output, the item (of the U-book) corresponding to that record is displayed. Future versions of the software will offer other triggers as well.

Queries may have parameters. A user of the query will be prompted for values of the parameters included in the query. For this purpose, the parameters and appropriate prompts are included as part of the definition of a query-type item. The user will be presented with a simple form in which he will indicate the desired values. For some of these, the value is chosen from a list presented by the program; the list is built using a rule defined as a component of the query.

We have divided the queries suggested for the conference U-book into 2 types, according to the expected results. The first type include those that allow papers to be found using several search criteria based on any combinations of the following: keywords, research area, topic, subtopic (if they were indicated for the papers), author names, their affiliation or country, and the session and slot in which they will be presented. Selection of a paper included in the result will display the item corresponding to that paper. The other type obtains information about delegates and authors.

A unique query that offers all of these combinations of search criteria would involve an extremely complicated SQL statement. Thus, several queries are created, each with the proper combination of parameters: their results will be lists of papers that satisfy the search criteria provided as parameters as well as by the query itself.

To find information about papers (either those of this year or presented in previous events) some typical queries are the following, where only the parameters are specified, since they describe the use for which they were defined. For all of these queries, the factors are used as conjunctions (this AND the other.) However, some of the parameters may accept more than one value: for these, they are used as disjunctions (OR). Once again, details about these topics were included in the

Appendix. An obvious comment: these queries may return a list of papers (not just the one you are looking for). The more you specify as search criteria, the shorter the resulting list.

- Search based on authors: parameters include author name, delegate number, country, affiliation, another paper number by the same author. Example: find a paper presented by an author, but you don't know his exact name.
- Search based on data of the paper itself: research area, topic, subtopic, part of the title, or even day, session, whatever.
- Search based on mixed data: research area, country of author, and whatever one might add.
- Some other limited queries may be offered to make life easier for the users. They are the same as the above but with fewer parameters.

Queries to obtain information regarding authors of papers or other delegates will produce a list of authors that satisfy them, and the selection of one of them will display his photograph and the papers he will present this year. Typical search parameters include the name (or part of it), affiliation and country, or the name or number of a paper he is presenting or has presented in a past event.

The system will include a feature that presents a list of these queries: a user will select one, indicate the values of its parameters and obtain the results. Response times to these queries depend on the location of the database: if it resides on the same computer, or a local network, responses will be instantaneous for a conference or similar event, since the number of papers and delegates will not be overwhelming. If the connection to the database is via internet, the response times will depend on several factors, but will never be very long.

Evaluation of the Proposed System for Delegates

The Evaluation Method Applied

This section describes part of the methodology used for this project. After conceiving the U-book based delegate system as an informing tool, considerable effort was spent in determining if it would be of any use at all, to clients as well as to providers of the information. During this period, the author wavered repeatedly between rejection and approval. Perhaps the section about "some drawbacks or likely objections" reflects this stage of the project. Anyway, the system was approved, probably due to the bias caused by my constant search for applications of U-books for all kinds of situations. However, I decided to treat it as a system to be developed for a client.

Whatever the nature of an information system or software package, once its design is completed it must be evaluated, even though previous analyses led to the conclusion that the system makes sense and that it will work (the two essential attributes of a system). The latter of course depends on several factors, including the quality of the implementation and a successful installation process. Unfortunately there is no set of golden rules, let alone a template, to determine if a system is "worthwhile" or the "correct" one for a given situation. The judge must formulate certain criteria and grade the system in every one of them. And, to make things worse, this task is usually performed by someone who has some kind of stake in the system; the worst scenario is of course the present one: I am supposed to evaluate it objectively, even though I suggested and designed it myself.

After thinking about this role, three alternatives were contemplated. First, ask somebody else to valuate it. Since nobody agreed to do so, it was discarded due to infeasibility. The next approach was just to let it go: forget the evaluation, perhaps nobody would notice. Unfortunately, I would!

So a utility function was conceived to compare benefits with all types of costs. The rest of this section reports the conclusions of this analysis.

Cost-Benefit Analysis of the Proposed System

Costs were divided into 4 groups:

- The additional work for the organizers of the event
- Extra hardware and software components
- Activities necessary to offer the system to delegates
- Some likely objections.

The benefits, not surprisingly, have only two dimensions, since they constituted the reason for the system's conception: delegate satisfaction, which hopefully would result in increased attendance in future events; and organizer satisfaction via a "mission accomplished" feeling. To get this point out of the way, every attempt at quantifying these benefits failed even before serious consideration. So the decision rule was changed from a cost-benefit analysis to a "are the costs excessive" decision rule. The "verdict" section below contains the conclusions of this analysis. I did not provide any references that support this type of decision-making: all the blame is mine, though of course, I am not the only one to have adopted such a method.

The additional work for the organizers of the event

The major task of the organizers would be to prepare the items of the book. As described, items will include those referring to papers, delegates (especially authors of papers), schedules, and optionally, information about past instances of the same conference. Additional information could be about buildings, rooms, other events besides papers or conferences, and perhaps some information about other persons, including the organizers of the event and guests.

The tasks to be performed by the organizers due to the preparation of the U-books include those listed below, where optional or "not so easy" tasks were separated from the others. Following every task, an (over) estimate of the work-load, expressed in the percentage added to a similar activity necessary for the planning and organization of the event is indicated. A value of zero means the task will not take more than a few minutes. The pronoun they refers to the organizers of the event.

- A site is installed: the programs of the software package (SPRP) and the databases and files used by the package are copied. (0)
- Options of the site are indicated: what features will be activated. Default values make this an easy task. (0)
- Organizers are included as such (USERS). They may create books. In our case, they would create the conference book. (0)
- They will populate their own database with the descriptions of papers, delegates, dates, days, rooms, sessions, slots and other data. This activity is not a result of the use of the U-book, though some additional data items and catalogues will be included.
- Define how chapters and their levels will be used to enable delegates to use their constraints in their favor. (2%)
- Include the items. Some will be indicated individually, whereas others will be imported from the event database. Items of the book were described previously. (4 - 10%). This point will be commented below.

- They will build the sequences and prepare the queries. (5%)
- They may use the system for session planning (if they will not do so, they would not provide several data items mentioned in the description of the database.
- They will add a special item: LAST MINUTE NOTICES and MESSAGES. The easiest way to do this is using a slide presentation or an RTF file. The item will "point" to the corresponding file, and would be included in all sequences (as its first element). An alternative way to do this is to include notices as individual items, and build a sequence of these items. This would be added as a subsequence to all sequences: any changes to the notices will instantly be reflected in any sequence that shows them. (0).

The amount of work due to the creation of items of the conference book is a function of two factors: which descriptors of items will be added as context-value pairs, and can somebody write a program to use the SPRP' planned import features to prepare the data, which serves both the creation of items as well as the addition of descriptors to existing items. Such a program requires knowledge of the sources of information: the databases used by the planners, as well as the locations and filenames of other files, such as portraits, full papers and others. This might not always be the case if a conference-preparation software package is used. Additionally, the programmer must understand the way to prepare the data for use by SPRP to update its files. To provide an example, suppose keywords of papers are to be added as context-value pairs. The program will generate a file which, for every paper, will specify its number, the context corresponding to keywords, and each keyword separated by a comma. The other part should be easy for anybody with some experience in the matter: prepare an SQL statement and, for each paper, generate the appropriate record. On the contrary, if the decision is to do this by hand, including the keywords as context-value pairs might turn out to be a lot of work.

The "not so easy to do" components (some are optional)

- Build an "index" for some of the sequences: 1%
- Group the "several items at a time" elements of sequences: this is not difficult, but involves learning how to use another function. They must be selected or indicated individually, and then grouped: 1%
- Construct the images (or other objects) to be shown as items of papers, authors, sessions, slots, etc.: 3%
- Update extra data - not present in their own files: Not estimable.

Extra hardware and software components

Since the use of this system does not imply extra computers or communications, this component was ignored. However, in specific conferences, more computers may be installed for the delegates to offer the system, but they probably would be used anyway for other purposes. As far as software is concerned, the proposed system is priceless: unfortunately, here this only means it does not yet have a price, and probably never will. Any conference that is sufficiently concerned about their delegates to offer this kind of service deserves the product free of charge. Perhaps there could be some minor expenditure associated with learning how to use it, or even some made to order programs it the staff of the conference needs help in these matters.

Activities necessary to offer the system to delegates

The system is installed on a computer, preferably a LAN server. If this were not possible, it is installed on several computers so that delegates can use them to be and stay informed. The goal is that the system should be available to delegates anytime. The files could also reside in a remote host if that were convenient. However, the programs will never reside on the host: they are executed on a computer on which the software is installed. SPRP in its current version only runs under (almost any recent version of) Windows.

A copy of the system may be distributed to every delegate. SPRP contains a module that prepares a version for readers only, and provides them with a somewhat modified database for the book, which will include some information that resides in other databases in the full version. For many computers, the installation would require a brief setup, since some DLLs may have to be added. The language preference is included (initially, there will be only 2 languages, English and Spanish). Every delegate will obtain a copy in the language he prefers or selects for that purpose. The way delegates can use their own copy was described previously.

In these delegate copies of the system, though they would have the SPRP program and the databases on their computers computer, if these were connected in any remote way to the computer hosting the item content files, they would not need to be copied to every computer.

Some likely objections

This section was quoted before, since the questions and answers played a huge role in the decision to go ahead or abandon the system.

- It's a lot of work: not all has to be offered; the system may be installed in stages (say a bit more every year);
- We are quite happy with what we have now: most people are, but professionals always should try to improve in any possible way;
- Delegates do OK with what we offer now: some actually think they could use additional tools, especially in large but concentrated events, meaning papers are about the same fields of knowledge and even topics, and many papers are presented simultaneously;
- Guys can search our materials (that is why they are given a CD!): think about why you gave them the CD, instead of just the printed schedules;
- Question: could it replace some of the work we are doing now? Probably not, except question and answer activities;
- Might there be a reduction somewhere regarding the previous point? Probably not, except in subsequent events;
- Duplicating activities is no fun: Right.
- INSITE does not need this tool. Though the system was partially illustrated with INSITE concepts, it is supposed to be a SW package. As was pointed out, many conferences and conventions do not offer the same help as INSITE. Some are quite pathetic in this aspect. Since the added value obviously is "absolute", if there was a great deal of delegate satisfaction before, the proportional increment due to the system will be less.

The Verdict

Is this system useful or superfluous, or a bit of both? Conclusions will be presented in bullet format. After all, they are not really conclusions; they are furnished to elicit further observations. A well known observation is pertinent: no system (or software package) is ever *finished*. The conference U-book is in its first version, so many additions might enhance its usefulness. Of course this is one of the reasons it was offered as a presentation in a conference, rather than just a paper in a journal: criticisms and suggestions will be valued.

- The system is not *necessary*: neither is an electric antenna in your car, or a very nice looking door-knob;
- Does it contribute "non superfluous" elements? What does it add to the event?
  - DRM (delegate relations management): delegates might enjoy it more, or even make fun of it, and - hopefully - attend the next one in greater proportions;
  - Delegates might even comment on the "special treatment" with colleagues or friends.
- How could it help (some or all) delegates?
  - It will save (some) delegates quite a bit of work (decision making). One of the advantages is that they will not have to switch “sources”: for example, look up an author in one “site” or page, and view a full paper in another one;
  - It might reduce regrets due to “wrong” decisions. This factor was assigned the largest weight in the decision to proceed with the system;
  - It may be used to create a memoir of the event, for example, create a sequence of all the presentations he attended and his comments about them;
  - Include a memoir of the paper(s) he presented, preferably recording who attended, questions and answers and other comments.
- The integration of data of several events (say, years) provides an added dimension to the data.
- Some potential advantages for presenters of papers
  - They will be able to post their presentations or hand-outs;
  - They can use it as a discussion and opinion forum;
  - They may add petitions or requests for collaboration or participation in a project, participants in a session next year about the topic, or similar information for delegates interested in the presentation.

Conclusions

A way to prepare material for delegates to a conference in several different presentations, based on the use of U-books in conjunction with the database used to plan and schedule the presentations sessions, was designed as a specific system, though it is an application of the software package for any U-book. Delegates can obtain a copy of the databases, files and software so they can build their own sequences and record experiences regarding papers or presenters. The system allows the integration of several instances of the conference, which may be useful for decisions regarding paper selection or serve other purposes.

The use of a database designed and populated by the organizers as part of the U-book itself reduces the amount of work usually involved in defining the items of the book, aided by the different ways to build a sequence. Though some queries may involve an advanced knowledge of SQL, most of the other activities require few specific knowledge.

The system described would make the choice of papers by delegates easier, faster and perhaps even more successful, especially for persons who tend to forget many details about presentations they have attended, or comments of others regarding papers they did not hear for any reason.

Though it was not mentioned in the paper, a language attribute may be added: a delegate may wish to know something about the way a paper will be presented.

The design of this system proved that the use of U-books for situations of this type was not an easy task, so several changes were made to the software package SPRP. These modifications enhance the power of the package, allowing additional applications of U-books. Current and further research includes the addition of options of several types, but especially some technical aspects, especially regarding queries based both on the foreign database and the U-book data. Since one of the main areas of application of these books is in teaching, the author expects some of these additions to furnish more tools to teachers that use U-books to prepare and present their classes.

The system designed for conferences achieves three main purposes: it illustrates the use of U-books to enhance the informing service to delegates, it shows the new features of the books and how they serve the informing process; and it might somewhat reduce the effort to prepare materials for the delegates.

References

Apple Computer Inc, Iphoto and Itunes. (2006), Software product description. Retrieved February 21, 2011 from http://www.apple.com/ilife/iphoto/

Bauer Mengelberg, J, (2001). A set of components to include security, on-line translation and other features in information systems. (2001). *AMCIS 2001 Proceedings.* Paper 247.

Bauer Mengelberg, J. (2007). The concept of an unstructured book and the software to publish and read it. *Issues in Informing Science and Information Technology, 2007*, 791-810

Birdsall, W.F. (2009). The role of the client in informing science: To be informed and to inform. I*nforming Science: the International Journal of an Emerging Transdiscipline*, *12,* 147-157.

Blackwell, D. and Girshick, M.A. *(1964). The theory of games and statistical decisions*. New York. John Wiley & Sons, London 1964

Cohen, E. (1999). Reconceptualizing information systems as a field of the transdiscipline informing science: From ugly duckling to swan. *Journal of Computing and Information Technology*, *7(3),* 213-219.

Comparison of e-book readers. (2011). Wikipedia.Retrieved from http://en.wikipedia.org/wiki/Comparison_of_e-book_readers March 8, 2011.

Cortázar, J. (1963). *Hopscotch*. Julio Cortazar's Hopscotch. Retrieved August 18, 2010 from http://www3.iath.virginia.edu/elab/hfl0117.html

E-book, an article. (2007) . Wikipedia. Retrieved from http://en.wikipedia.org/wiki/E-book March 2, 2009

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 http://inform.nu/Articles/Vol8/v8p101-122Gack.pdf

Gardiner, E. and Musto, R.G. (2010). The electronic book." In Suarez, M. F. and Woudhuysen, H.R.. The Oxford companion to the book. Oxford: Oxford University Press, 2010.

Gill, T. G. & Hicks, R. C. (2006). Task complexity and informing science: A synthesis. *Informing Science Journal*, *Vol 8*, 1-30. Retrieved February 20, 2010 from http://www.inform.nu/Articles/Vol9/v9p001-030Gill46.pdf

Gill, T.G. & Bhattacherjee, A. (2007). The informing sciences at a crossroads: The role of the client. *Informing Science Journal*, *Vol. 10*. 17-39. Retrieved February 20, 2010 from (http://inform.nu/Articles/Vol10/ISJv10p017-039Gill317.pdf

Hannan, J. (1957). Approximation to Bayes risk in repeated play. Contributions to the theory of games, III, *Ann. Math. Study Number 39*, Princeton University Press, 97-139.

Hart, M. (1992). *History and philosophy of project Gutenberg*. Retrieved March 3, 2007 from ttp://promo.net/pg/history.html

Kindle 3 Wi-Fi 3G. Amazon. (2011). Retrieved from http://www.amazon.com/s/ref=nb_sb_noss?url=search-alias%3Ddigital-text&field-keywords=kindle+wifi+3g&x=0&y=0 March 7, 2011.

Nookcolor: Barnes & Noble.(2011). Retrieved from http://www.barnesandnoble.com/nookcolor/index.asp March 7, 2011.

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

Proceedings of INSITE. 2010. *Proceedings of the Insite 2010 Informing Science and IT Education Conference*. Informing Science Press, Santa Rosa, Ca.

Savage, L.J. (1951). The theory of statistical decision. *Amer. Stat. Assoc*. 46 (253), 55-67.

Whatley, J., & Ahmad, A. (2007). Using video to record summary lectures to aid students' revision. *Proceedings 2007 Informing Science and Information Technology Conference.* InSITE, Ljubljana, Slovenia – 22-25 June 2007. Available at http://ijklo.org/Volume3/IJKLOv3p185-196Whatley367.pdf

Appendix

How queries are included as items of U-Books

Introduction

The new version of SPRP allows queries to databases that are not part of the U-book itself. The queries are saved as items of the book, so they can be invoked either directly- as an item - or as an element of a sequence.

The preparation of such a query involves several steps, which will be described below with the main purpose of explaining how they are performed in SPRP, so few technical aspects are provided:

- Declare the database
- Specify the query (with or without parameters)
- Indicate how the results will be displayed
- Specify actions triggered by certain events
- Test the query
- Where the definitions of the queries are stored.

In the descriptions that follow, the terminology of Visual Basic was used. Thus the results of queries are called *recordsets;* in other programming languages, they would be referred to as datasets or cursors. They are collections of records with the fields specified by the query. These fields are numbered (starting with 0) according to their appearance in the SQL statement.

The screenshots provided for some of the forms do not exactly correspond to the real ones. They were adapted for this paper. Essentially, some font sizes were augmented, and elements that appear only when they are necessary (superimposed or added) were separated from the form in which they appear. Finally, certain advanced options were not included at all to simplify the explanations.

We will use a single query to illustrate its elements and how they are defined. The choice of a query for this purpose took into account that the SQL statement should be as simple as possible, so that even not very experienced readers in these topics could the example, but at the same time it should illustrate almost all features offered to users,. Additionally, it should show how the inclusion of a query as an item of a U-book is specified in steps.

The query used as an example in the rest of these descriptions is: the user (of the query) wants to know all the papers of a particular research area, but he may want to constrain them as follows:

- They may be of a particular topic or of any topic
- He may limit the papers to those offered on certain days.

For this purpose, the query will be formulated against the PAPERS table of the Conference database. The fields of interest are: Paper (the number of the paper), the title of the paper, the paper_item_number and the authors_as_text field, that is formed by all the authors of the paper. The query will also use fields to formulate the conditions that will determine the inclusion of certain papers in the result of the query: theday (the paper will be presented); research-area (the code of the area) and the topic of the paper. Since the query will return the item-number of the paper as an item of the conference U-book, the user may ask to see the item corresponding to the paper.

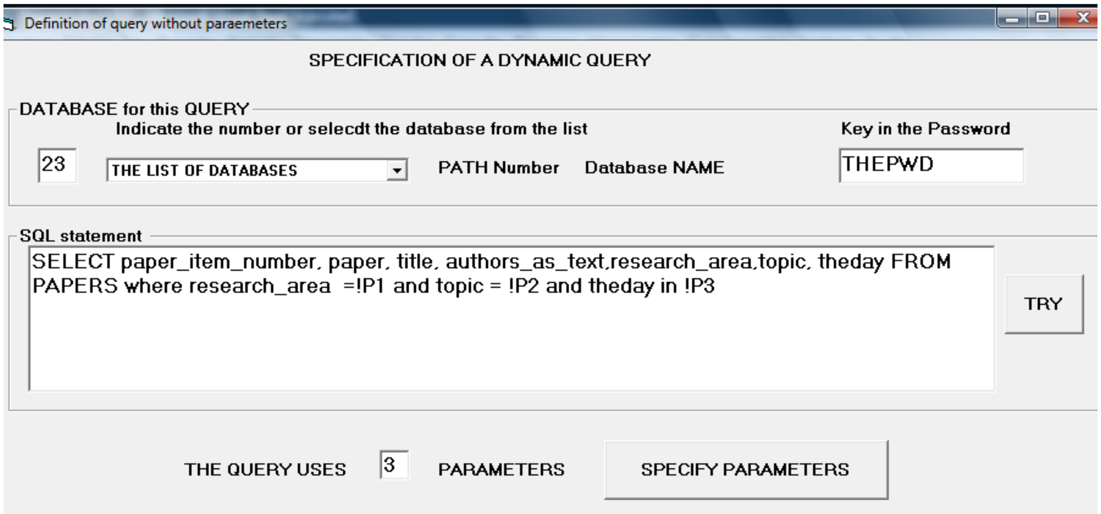
Declare the database

Suppose there is a foreign database (that is, different from the one used to store items and other data of the U-books themselves.) In order to include a query as an item, first the database as such must be included in a table (of the U-.book) called THEBASES. This is done providing 3 elements:

- the PATH (number) and name
- RDBMS used (technology) which must be one of the managers included in your version of SPRP: most common database providers are included
  - A password corresponding to the security features defined for the database.

The database assigned a number and recorded in the THEBASES table of the book's database. Queries against that base will simply specify its number.

Specify the query



SPECIFICATION OF A DYNAMIC QUERY

DATABASE for this QUERY

Indicate the number or selecdt the database from the list

23 THE LIST OF DATABASES PATH Number Database NAME

Key in the Password THEPWD

SQL statement

```
SELECT paper_item_number, paper, title, authors_as_text,research_area,topic, theday FROM
PAPERS where research_area  =!P1 and topic = !P2 and theday in !P3
```

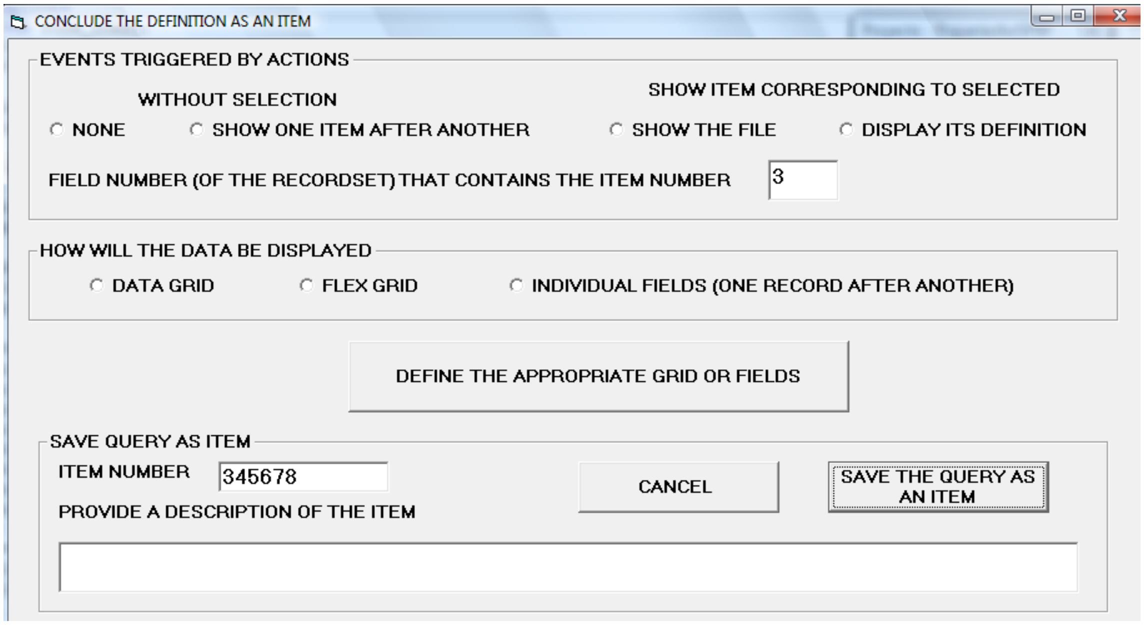
THE QUERY USES 3 PARAMETERS

Figure 5. The form used to define the first elements of a query as an item

After one of the databases is chosen, and the necessary password provided, a query may be formulated and included as an item of the U-book. A query may have parameters; before the query is sent to the database engine, they are substituted by their values specified by the query's user. Parameters are indicated as part of the statement preceded by an exclamation mark "!" and numbered consecutively. More about these parameters will be explained in a separate section below.

Figure 5 shows the forms used to define a query. After selecting a database, its name will be displayed in the appropriate fields to avoid mistakes. Next, the SQL statement is indicated, and the number of parameters included in it is specified. This can be done before or after entering the statement.

Figure 6 shows how the query is saved as an item of the U-book. Additionally, other elements of the definition are indicated on this same form. Notice that the sequence of tasks actually performed may be different, especially when the query involves parameters, but it will also be interrupted to specify details about the way the results will be displayed.



CONCLUDE THE DEFINITION AS AN ITEM

EVENTS TRIGGERED BY ACTIONS

WITHOUT SELECTION

○ NONE ○ SHOW ONE ITEM AFTER ANOTHER

SHOW ITEM CORRESPONDING TO SELECTED

○ SHOW THE FILE ○ DISPLAY ITS DEFINITION

FIELD NUMBER (OF THE RECORDSET) THAT CONTAINS THE ITEM NUMBER 3

HOW WILL THE DATA BE DISPLAYED

○ DATA GRID ○ FLEX GRID ○ INDIVIDUAL FIELDS (ONE RECORD AFTER ANOTHER)

DEFINE THE APPROPRIATE GRID OR FIELDS

SAVE QUERY AS ITEM

ITEM NUMBER 345678

PROVIDE A DESCRIPTION OF THE ITEM

CANCEL

SAVE THE QUERY AS AN ITEM

Figure 6. The form used to complete the definition of the item

As can be seen, a triggered event may be invoked for the results of a query. The only action implemented in this fashion in the current version of the software is to show items of the U-book associated with one of the records of the recordset resulting from the query. For this purpose, the corresponding item number must be a field of the recordset. The number of that field (remember that the fields are numbered) is entered, and the system can display either the definition of the item (with all its attributes and descriptors) or the file corresponding to the default version and language of the item. For example, if the item number corresponds to a paper, the image (or RTF file) corresponding to the paper can be shown, or the full definition of the item may also be displayed. Observe that one can ask the system to display the item corresponding to every record of the recordset. This will happen automatically, that is, no previous action by the user is necessary.

Three additional forms allow the user to specify the way the results will be shown. Every one of the options uses a different form, where the sizes of columns and their captions are specified. The details of the forms and the way to specify the objects to be displayed were not included in the paper, since they are not essential to understanding how queries are defined and used.

For example, to show results in a flex-grid, on the appropriate form:

- The program will display a LIST of all fields of your recordset.
- There is an (initially) empty list of selected fields.
- You will select a field at a time and include it as a selected field.
- You may reorder these fields, which will become columns of the grid.

Next, the program will show the resulting grid (with an empty row representing the first record) and the column widths can be adjusted. The font size can also be changed for the grid. The person defining the query is encouraged to try it “live”, that is, see how the results will be presented to a user of the query.

Parameters of a query

Parameters must be numbered consecutively in the SQL statement where they appear. The number must be preceded by the 2 characters “!P” to indicate the position of the parameter. In our example, the statement would be: SELECT paper_item_number, paper, title, authors_as_text, research_area, topic, theday FROM PAPERS where research_area = !P1 AND topic = !P2 AND theday IN !P3.

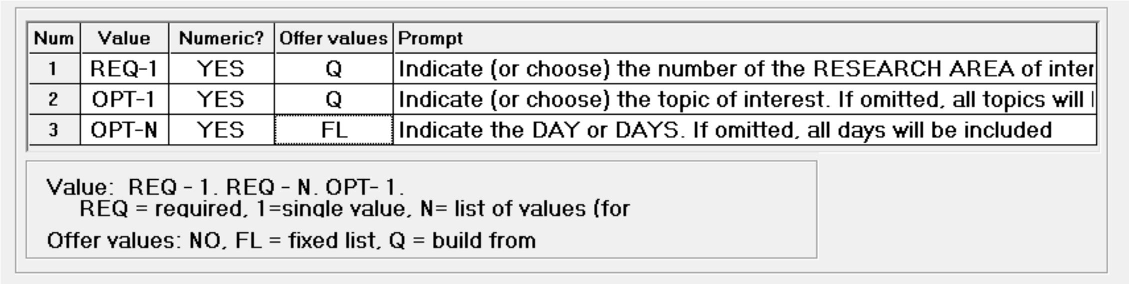
When the query is invoked, the reader will furnish the desired values of each of the parameters. Notice that the program will *fix* the statement: it will eliminate AND, OR and even the WHERE clauses when they are not used: this can happen when no values for optional parameters are furnished.

A parameter may be

- Numeric or not
- Optional or required: this means that the query may be executed without a certain constraint
- Correspond to a single value or several values. In the latter case, they will be interpreted as: the condition is satisfied if any one of the values is present. This is implemented using an “IN” clause, and providing the list of values specified. Observation: if a LIKE clause is included, it will only allow single value parameters. The user will have to supply the exact value, that is, include symbols to indicate exactly what he wishes to be the selection criterion. Therefore, the prompt corresponding to such parameters should be very precise and complete.

The user may also be offered a list of valid values for a parameter. This can be done providing the values in what is called a fixed list, or by the specification of a rule to populate the list of values offered to the user as possible choices for the parameter. If such a list is offered, the user of the query will choose one of its values. If the parameter admits several values, he will be able to select more than one from the list.

Though in figure 7 the form that is used to describe the parameters is depicted as a separate form, it will be superimposed on the form shown in Figure XX1. This form is automatically displayed between the first two shown above when there are parameters.

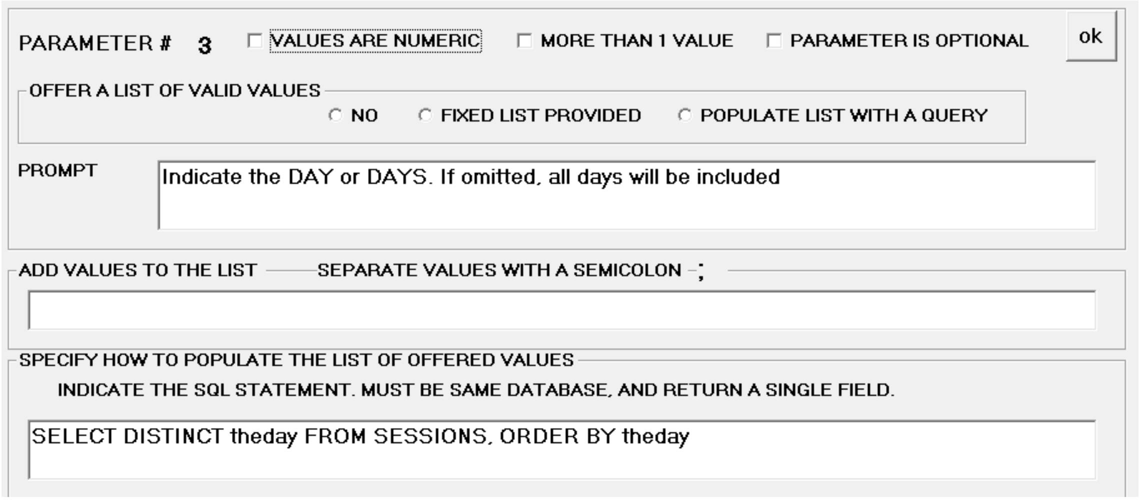


Num	Value	Numeric?	Offer values	Prompt
1	REQ-1	YES	Q	Indicate (or choose) the number of the RESEARCH AREA of inter
2	OPT-1	YES	Q	Indicate (or choose) the topic of interest. If omitted, all topics will l
3	OPT-N	YES	FL	Indicate the DAY or DAYS. If omitted, all days will be included

Value: REQ - 1. REQ - N. OPT- 1.
REQ = required, 1=single value, N= list of values (for
Offer values: NO, FL = fixed list, Q = build from

Figure 7. Form that shows the specifications of the parameters of the query

A click on a row of this grid will display the form shown in Figure 8, which will appear superimposed on the previous one – to avoid the flash effect produced by a change of form. The "add values to the list" and "specify how to populate the list" components will only appear if the corresponding option is chosen in "Offer a list of valid values".



PARAMETER # 3 □ VALUES ARE NUMERIC □ MORE THAN 1 VALUE □ PARAMETER IS OPTIONAL

OFFER A LIST OF VALID VALUES
○ NO ○ FIXED LIST PROVIDED ○ POPULATE LIST WITH A QUERY

PROMPT: Indicate the DAY or DAYS. If omitted, all days will be included

ADD VALUES TO THE LIST SEPARATE VALUES WITH A SEMICOLON ;

SPECIFY HOW TO POPULATE THE LIST OF OFFERED VALUES
INDICATE THE SQL STATEMENT. MUST BE SAME DATABASE, AND RETURN A SINGLE FIELD.

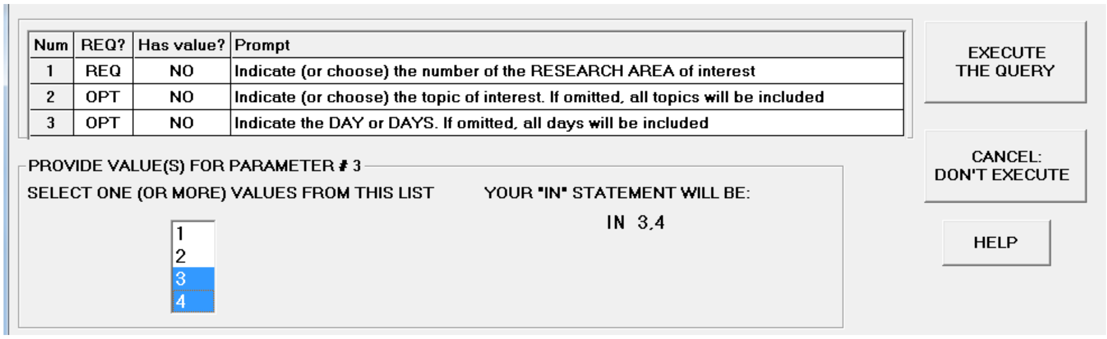
```
SELECT DISTINCT theday FROM SESSIONS, ORDER BY theday
```

Figure 8. Form to change the specifications of a parameter of the query

Use a query

As was mentioned before, a query may be invoked by asking for a particular query (by item number) or select one from a list of available queries, or may be invoked as an element of a sequence. The effect is exactly the same in all cases.

If the query does not involve parameters, the reader will obtain the results, of course in the way specified as part of the definition of the query. If there are parameters, a form is displayed such as the one depicted in figure 9. Parameters used by the query are shown: the user identifies them by their prompts. He is informed if the parameter is optional or not. Additionally, there is an indicator called HAS VALUE? If it indicates "no" this means that the user has not furnished any values for that parameter. The EXECUTE THE QUERY button will not be enabled if one of the required parameters has not been assigned any value.



Num	REQ?	Has value?	Prompt
1	REQ	NO	Indicate (or choose) the number of the RESEARCH AREA of interest
2	OPT	NO	Indicate (or choose) the topic of interest. If omitted, all topics will be included
3	OPT	NO	Indicate the DAY or DAYS. If omitted, all days will be included

PROVIDE VALUE(S) FOR PARAMETER # 3

SELECT ONE (OR MORE) VALUES FROM THIS LIST

1
2
3
4

YOUR "IN" STATEMENT WILL BE:

IN 3,4

Figure 9. Form to provide the values of the parameters of the query

The user clicks on the row corresponding to one of the parameters, and the objects depicted on the bottom part of the form are offered. These objects differ according to the nature of the parameters: for each one, the appropriate fields will appear. Here, only the parameter 3 (selection of the day) is shown, after the values of 3 and 4 were selected from the list of valid options.

There is an exception to this sequence of events: if the query uses only one parameter, the list is not offered: the fields to indicate the values of the unique parameter are offered directly.

The system, once the query has been confirmed, substitutes the parameters in the SQL statement. During this process, as was mentioned before, certain useless parts may be eliminated. This may happen when there are optional parameters. Finally, a command is issued to the database engine, which in turn returns the desired recordset.

As soon as the program receives the recordset, it will display the results in the manner defined for the query. Only one of the options will be described here. Suppose the results are shown in a flexgrid (this is an array where the columns will be the fields of the recordset (usually all of them). Each record will occupy a row of the array, except the first one, which will contain the column names.

If the “show the item” action was included for the query, the user may click on a row of the grid and thus see the corresponding item in either of the ways described before. Note that this way one can obtain information about papers (in the example used in this section) that satisfy the conditions we imposed. Then the individual papers can be accessed by a click on the row.

Where the definitions of the query are stored

Some of the information will be recorded as part of the item using the THEITEMS table’s fields. Since there are many components of the definition of a query that cannot be stored in these fields, other tables were added ad hoc to the database. As we have seen, the following data has to be available to execute the query when invoked:

- Which database (name) and some technical information necessary to connect and use it;
- The SQL statement to be executed;
- If the statement contains parameters, their attributes and the appropriate prompts that will inform the user what he needs to specify;
- The way the results are presented to the reader;
- Triggered events resulting from actions by the user.

Thus, there is a table we called QUERY_AS_ITEM, some of whose fields are the following:

- ITEM_NUM: to relate it to the item

- Database_num: which of the databases is involved;
- SQL_STATEMENT: the statement which may or not include parameters;
- How_many_parameters does the query contain;
- What_field_contains_item_number: the field number in the recordset produced by the query in which the item_number to be shown (if appropriate);
- How_are_results_shown: using a data-grid (one record per line), a flex grid (built as desired), one record at a time or not at all, where this means that, for every record in the resulting recordset, the system will automatically invoke the item indicated as part of the answer and "show" it.

If the query contains parameters, records are added to another table called QUERY_PROMPTS. Its main fields are the following (the principal index is denoted by bold face letters):

Item_num: to relate it to the record in the QUERY_AS_ITEM table.

Param_number the consecutive number;

Is_optional: yes or no field. No means the user must provide a value for the parameter.

Is_numeric: yes indicates that only numeric values will be accepted.

Single_value: yes or no field. No indicates that several values may be indicated; they will be combined with an "OR" operator, but actually usually the SQL statement will present an "IN" clause for that parameter.

Prompt: a text to inform users of the query what information they have to provide.

To define the way the results will be presented, certain values have to be provided so that the output is legible and, if possible, even attractive. For this, a table called QUERY-SIZES-AND-POSITIONS (a misnomer) was included in the data model. The details of this table are not furnished here, especially because their description is lengthy and probably confusing, since the same fields are used in different manners depending on the objects used to display the results, and furthermore, some readers will not be interested at all, whilst others will easily be able to imagine the type of data and the appropriate fields.

Biography

Juan R. Bauer Mengelberg (aka.John). After obtaining a degree in Mathematics at the Universidad de Buenos Aires, Argentina (1963), he got a PhD in Statistics and Operations Research at the University of Wisconsin, at Madison (1970) 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 also offers graduate degrees tin-Statistics and Applied Computing, 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 software packages, and his current research includes informing processes - especially teaching and electronic publishing – as well as hybrid databases and data-warehouses.