

Techniques for the Coordinated Study of Multiple Electronic Texts

Harry Plantinga
Wheaton College
Harry.Plantinga@wheaton.edu

Abstract

This paper addresses the problem of the simultaneous, coordinated study of two or more texts, such as a book with a table of contents, index, notes, translation, lexicon, or different edition. In the paper realm, these texts must be coordinated or *synchronized* in various ways, such as looking up unfamiliar words in a lexicon or paging to corresponding sections of different editions of a work. In this paper I identify meta-data that facilitates this kind of synchronized study by computer. With the relevant meta-data for books, it is possible to load them into relational database tables in such a way that all of the synchronization operations above may be expressed as database operations such as selects and joins. Finally, I propose several tools that would facilitate this kind of synchronized study of multiple texts on the world wide web.

1. Introduction

Scholars frequently study multiple texts in a coordinated fashion. A scholar studying a work in an unfamiliar language may have a lexicon at hand, looking up words as needed. One studying a well-known text may have at hand commentaries that have been written by those who previously studied the text, turning the pages of both books so that text and corresponding commentary are both visible. Other common examples of synchronization include the use of indexes and notes such as footnotes or end notes. Centuries of publishing practice have developed standard methods for handling these problems in the print, but these methods don't always translate naturally into the electronic realm, where there is no natural notion of a page.

Texts are also synchronized in other ways. Two people may put their calendars together and try to find a meeting time when both are free. A text may have a topical numbering system applied, so that when studying a particular passage, it is possible to find related passages by following the the topic number embedded in the text to the subject index.

All of these kinds of synchronization among texts can be translated to the electronic realm, so that similar sorts of coordinated study can be performed on-line, often with greater ease. However, to make this possible, certain meta-data must be represented, specifying the relationships among the texts. Then, computer tools can perform the relevant forms of synchronization.

2. Types of Synchronization

Most of the common types of synchronization that occur in the study of multiple texts can be categorized

as follows.

2.1. Dictionary

A *dictionary* is a text with keywords and their definitions (or other associated notes). The keys of the dictionary, i.e. the words defined, are arbitrary strings of characters. The use of the key in another text is sufficient to imply that the dictionary definition is relevant (or *may be* relevant -- words may have several meanings, but we don't distinguish them here). Dictionaries are not tied to a particular text; a dictionary may be used in conjunction with any other text that uses some of the words in the dictionary. Dictionaries, in this general sense of the term, include several sorts of reference material, such as encyclopedias, lexicons, glossaries, and lists of characters.

Dictionaries are often used as reference material; a word is looked up as needed. This can be automated in the electronic realm by adding a hypertext link for all of the words that appear in the dictionary. However, other kinds of synchronization are also possible. For example, a student studying a foreign-language document may wish to have all of the unfamiliar foreign words looked up and *accumulated* in a single page of definitions. Then, the foreign-language document and the relevant word definitions can be displayed in parallel columns. This eliminates the time delay inherent in looking up a word.

2.2. Index

A general *index* has topic entries and references to particular sections of a text. The topic entries are often hierarchical, consisting of two or more words or phrases, but they are not used as keys in the sense that they are expected to be located in the text verbatim. The references may point to a particular location in a text or a range of characters or words. The text referred to may be a particular edition of a text or any edition -- for example, a reference to Augustine's *Confessions*, bk. XIII, ch. iii, may be used with any translation or edition of that work. General indexes of this sort are commonly used as tables of contents, subject indexes, tables of figures, and the like.

In the normal use of an index, the student browses the index and follows references to topics of interest. This can be automated by constructing a hierarchical index of hypertext links. It is also helpful to make large indexes expandable and collapsible, so that only the top level of the hierarchy is initially visible and the user can expand and collapse lower levels of the hierarchy.

If a standard set of keys were defined and indexes were created using those keys, it would be possible to combine the indexes for many texts, making one large subject index to a library of books, for example. Finding information by topic in a whole library would then be much easier. However, a standardized set of topics may not be as apt for a particular book, and multiple indexes for different purposes may be warranted.

2.3. Parallel Edition

Multiple editions of a text are often studied side by side, with appropriate *synchronization* of the texts. A student may study a text in an original language and a translation. An editor may have several manuscripts that are being used as the basis for a critical edition. With books or manuscripts, the student turns pages as appropriate to keep the current portion of the text visible in all editions. The texts may be virtually identical in the content and order of presentation, but that will not always be the case. One text may have been expanded, reordered, edited, and the like, so that the corresponding sections of texts may not always be easy to find.

The study of multiple parallel editions in the electronic realm is possible without too much added effort, for example by scrolling multiple windows. But more assistance can be provided if corresponding passages in parallel editions are known. In that case, whenever the location in one window is changed, it would be possible to bring up the corresponding location in the other windows automatically (*synchronized scrolling*). This would be especially valuable when material has been reordered in the

different editions. Another option would be to display the corresponding sections of the different editions in parallel columns, with corresponding passages lined up horizontally.

2.4. Annotation

In many instances, one text will contain notes or explanations on sections of another text. Footnotes and end notes can be considered as separate documents commenting on a text; in fact, in electronic texts, such notes are often stored in a separate document or a separate section of a document. Other examples include a text which is an interpretation of another text or a reply to another text.

In print usage, notes or commentary might appear as footnotes, end notes, parallel columns, or a separate text that must be manually synchronized. In the electronic realm, they may be handled by adding hypertext links to a separate notes document, by displaying the notes in parallel columns, or by the synchronized scrolling of windows.

2.5. Keyword

Another sort of synchronization occurs when texts are tied together by arbitrary keys. For example, a professor may give an exam with ten questions. Students may write their answers in any order. The professor may wish to synchronize a student's answers with the exam according to a *keyword* scheme: a keyword is assigned to each question and answer -- in this case, the question number. Then, the student's answers are matched with the questions by keyword.

When the texts are in electronic form, the exam and the responses may be synchronized automatically and displayed in parallel columns or synchronized windows. The professor could, for example, synchronize all of the responses, so that she can grade all responses to question one first, then question two, and so on.

In addition, other sorts of synchronization become possible. It would be possible to build an index of student responses and add links from the exam to that index. One could then follow links from the question to each of several answers. This kind of synchronization is not often found in the print realm, because it involves so much looking up in indexes, but it can be very valuable for use with electronic texts.

Another use of this sort of synchronization would be a topical numbering of a text, which enables a student to get from a passage of the text to a subject index to a related passage elsewhere in the text or in another text. Or, a document with many points for discussion can be linked to various written responses. These documents can be presented on a computer by hypertext links, parallel columns, or synchronized windows.

Keyword synchronization is similar to a bidirectional index, in which the links from an index into the document have been inverted, so that it is also possible to get from a section of the document back to locations where that segment occurs in the index. However, in the case of keyword synchronization, none of the document are really indexes; each has content along with keyword synchronization information.

2.6. Total-order

A final sort of synchronization occurs when two documents have sections numbered from more-or-less continuous range of possibilities. For example, two people might want to synchronize their calendars to find common free time. This sort of synchronization is similar to keyword synchronization except that for a key in one text there may be no corresponding key in the other text. Instead, the keys are linearly ordered.

In order to synchronize two texts linearly, sections of the texts must be identified with keys on which a total order is defined. The sections may or may not be in order within a text. The normal use of such texts would involve sorting the section by the key and displaying the result in order, in parallel columns or

synchronized windows, horizontally aligned by key value.

3. Meta-data needed for synchronization

After analyzing these typical cases of the coordinated use of multiple texts, it is possible to identify four major types of information that may be represented in a document to make possible all of these types of coordinated use and others. The types of information are document addressing standards, synchronization points, index entries, and dictionary entries.

3.1. Document addressing

It is necessary to be able to refer to a particular part of a document, either a point or a range. An index entry, for example, may refer to three particular sentences in the text. In order to be able to coordinate the use of different editions of the same book, the addressing scheme should be able to refer to a class of texts as well as a particular text.

The scheme used by some SGML DTDs¹ such as TEI² and HTML assign unique ID's to elements of the text. References may then be made to a particular element of a particular text. This scheme is not really adequate for the above purposes; the addressing scheme must be able to represent a reference to a certain section from any one of several different editions or versions of a text. It would be possible to apply some semantic interpretation to the IDs chosen in this scheme, but this would seem to be an abuse of the semantics of the language.

Instead, it is probably necessary to generate a standard addressing scheme for individual texts. For example, a particular edition of Augustine's *Confessions* could be divided up into books, chapters, and sections. Other editions could be divided up so that corresponding sections have corresponding addresses. Several texts will be said to be *synchronization-compatible* when they share a common addressing scheme.

Particular addressing schemes may be represented as a relational table³ in a database, containing the number of levels of hierarchy in the scheme, the allowable values for each level, and some characteristics of the scheme such as whether it is a total order. A text can then be associated with the addressing scheme used. The use of the addressing scheme in the text can be validated, and the texts that share that scheme will be known to be synchronization-compatible.

These document addressing schemes will be thought of as partitioning the large document into several smaller documents. Sub-addressing within the smaller documents will still be possible, for example, by word or character numbering. However, the larger addressing scheme is needed to handle documents which are not identical and which may change over time. The document address is hierarchical and may also be used to generate an outline of the document.

The scheme will be thought of as partitioning the document into several sections, each of which is stored in a row in a relational database table. Each row will have an attribute for each level of the addressing hierarchy and another attribute for the text of the document of that section (or an external reference to the text). It is assumed that the text within each unit is self-sufficient and does not depend on markup codes from earlier sections.

3.2. Synchronization Points

¹ SGML, the Standard Generalized Markup Language, is used to define markup languages such as HTML and TEI. These languages are defined with DTDs (Document Type Definitions). See <http://www.sgmlopen.org/>.

² The TEI DTD, defined by the Text Encoding Initiative, is a rich markup language designed for literary analysis. See <http://etext.virginia.edu/TEI.html>.

³ A relational table is a collection of *rows* with the same set of *fields* and related information. For example, a table may be defined for addresses. The fields might be name, address, and phone, and each row in the table would contain the address information for one person.

Synchronization points specify points of texts that correspond in various ways. These correspondences are more general than the kinds specifiable in TEI or HTML. In TEI, for example, it is possible to specify that a particular section of a particular document is related to a particular section of another particular document in some way, for example, as an analysis, a translation, and the like. For the more general synchronization problem, it is necessary to be able to specify standard synchronization methods such as *dates* and synchronize any two texts with date synchronization points. Also, IDs in TEI and HTML must be unique, but it may be desirable to have multiple instances of the same synchronization key in the more general synchronization problem. For example, in a calendar, there may be several entries for a certain date.

The synchronization points are identified with keys at particular locations in a text. The key universes have several relevant characteristics. Keys may be arbitrary keywords or keywords chosen from a named, standard set. These keys are not ordered; they are merely identifiers. Keys may also be chosen from ordered sets such as lexical values, numeric values, dates, or standardized textual references such as "chapter and verse." These key spaces are totally ordered, in the sense that for any two keys, we can decide which is larger. Also, it is possible to compare a key *K* from document *A* with another document *B* not containing *K* by finding its appropriate location in the key space order.

There are also several important properties of the *use* of a key type in a document. A document may have at most one instance of each key, or several. The keys may appear in order in the document, or nearly in order, or not in order. All of the keys in a particular key space may be represented, or only some. As we will see, the properties of the document determine the kinds of coordinated use that may be made of a document. Synchronized scrolling doesn't work well for documents in which keys are not unique, for example.

The synchronization points are thought of as partitioning the text. There may be several different sorts of synchronization points in a text, each forming a separate partition. The keys for synchronization points are also thought of as keys of tables in a relational database. A text is stored as a table with two attributes, the synchronization key and the contents of the text section.

3.3. Index and Dictionary Entries

To handle dictionary and index entries, the relevant semantic information must be represented. An index entry consists of a hierarchy of keys and a list of references. The references are points or ranges from a particular reference scheme. Thus, the index could be used to refer to different texts. Dictionary entries consist of a word and a definition. These must be appropriately marked for use as such.

4. Synchronization and Databases

In order to translate these types of synchronization of multiple texts into the terminology and practice of databases, we must first represent the electronic text as relational tables. The *content table* for a text will have a row for each section in the partition defined by the document addressing scheme. Each row will have fields for the document address: one field for flat addressing schemes, and for hierarchical schemes, one per level of the hierarchy. One additional field is required, representing the addressed section of the text. This field may contain the text itself, possibly stored as a Binary Large Object, or a reference to the text such as a URL or URN.

A synchronization scheme is represented as a relational table naming the scheme, specifying the allowable values, and giving their standard order (if relevant). Synchronization points from a particular scheme can be represented in the database as a separate *synchronization table*. This table contains the synchronization values, the beginning and ending document sections (which are keys from the primary table for the book), and possibly word or character counts within the sections. For example, a set of date-based synchronization points would be represented as a table with a date field and two foreign key fields for the primary table for the book. Two books with date synchronization points could then be

synchronized by joining⁴ the two date-synchronization tables and the two content tables.

For a text that is an index to another class of texts, entries consist of a hierarchical set of topics and a reference to a point or range in the text. An index could be represented with an *index table* consisting of a compound primary key corresponding to the subjects and a pair of foreign keys for the content table, specifying the referred text.

Dictionary tables contain the term being defined and the definition. They differ from the other tables in that the term is not naturally the key of another table. In order to use an index with the machinery of a database system, an inverted word occurrence index for the book is constructed and represented as an additional *concordance* table in the database. This table has three fields: a word and the section and location for an appearance in the text. The program that creates the inverted index could omit words that are too common for a concordance.

All of the above synchronization tasks can be handled in a natural fashion when the electronic texts are stored as tables in a relational database as described above and when a few additional tools are available. The tools are used to generate indexes and links from the information in the tables and to combine information into parallel-column documents or synchronized windows.

4.1. Dictionary Synchronization

Dictionary synchronization is possible when there are two texts, one a dictionary and one which contains words found in the dictionary. There will be a table D for the dictionary and C for the concordance. Using the symbol \otimes for an inner join, the table

$$D \otimes C$$

contains a row for every occurrence of a dictionary word in the book. Each row contains a definition and the location of a word in the original text. This information may be used to add hypertext links to the text. Alternatively, the definitions could be accumulated in a single document and displayed in parallel columns.

4.2. Index Synchronization

An *index* table contains a hierarchy of subject entries and a foreign key for the content table. The index information could be presented as a hierarchical, collapsible index document with appropriate formatting of the information by the index tool. Additionally, if the index were joined with the content table, a few words of the text referred to could be inserted into the index to assist in finding the desired section of the text.

4.3. Parallel Edition Synchronization

The content tables for two texts, C_1 and C_2 , that are translations or parallel editions of each other are represented as tables using the same primary addressing scheme. Thus, they can be joined. The resulting table,

$$R = C_1 \otimes C_2$$

has two fields, containing the contents of the two texts, appropriately synchronized. The results can be

⁴ A relational *join* operation makes one large table out of two existing tables that share a key field. Rows from the two tables that have the same key value are joined to form one large row. In an *inner* join, rows from each table are discarded if there is no row from the other table with the same key value. In a *left- or right-outer join*, rows from the left or right table are included in the joined table, even if there is no row in the other table with the same key value.

displayed in columns or synchronously scrolling windows.

4.4. Annotation Synchronization

When one text contains annotations on another text, the situation is similar to the previous situation of parallel editions except that the semantics of the second text are different. The situation may be handled in the same manner, though, and displayed in columns or synchronous windows.

4.5. Keyword

When two texts are synchronized by keyword, four tables are involved. Both texts have content tables, C_1 and C_2 , and synchronization tables, S_1 and S_2 . The joined table

$$S_i \otimes C_i$$

represents the content keyed by the synchronization scheme, so the texts can be synchronized with the following operations:

$$(S_1 \otimes C_1) \otimes (S_2 \otimes C_2)$$

The resulting table contains two fields, one for the content of each book. The results of this operation may be displayed in columns or synchronized windows.

If the joins are inner, only the sections common to both books will be included; if the joins are left- or right-outer, sections missing in one of the books may be included in the result. If keywords are not unique in a text, corresponding sections from the other text will be repeated.

4.6. Total Order

In this case the synchronization points are values from a continuous type such as lexical, numeric, or date values. The tables should be sorted by the keys. The sorted tables can be displayed in columns or synchronized windows.

4.7. Other Capabilities

Finally, note that complex searches may be performed with standard SQL. The presence of the concordance table makes it possible to quickly search for sections containing any Boolean combination of words, and the other tables make more sophisticated searches possible. For example, in a text with dates as synchronization points, one could add additional search criteria concerning the date of the entry. A search interface could be constructed that would form user-specified search criteria into SQL Select statements.

5. A Proposed Implementation for the World Wide Web

In this section I propose a system for building these database tables from suitably-marked electronic texts and performing the above-mentioned types of synchronization for the world wide web. Since the synchronization markup described above is meta-markup, it is removed by a program that reads the text into a database and forms the described tables. The meta-markup therefore does not need to be a part of the underlying markup language -- in this case, HTML.

5.1. Primary Addressing

Markup for the primary addressing scheme is borrowed from TEI syntax. Tags are inserted of the form

```
<div1 type="xxx" value="xxx" name="xxx">
  <div2 type="xxx" value="xxx" name="xxx">
    [up to div8]
  </div2>
</div1>
```

The type attribute is optional, specifying the type of division, such as book, chapter, or section. Only the first type attribute for a level of div tags is used. The value attribute is only necessary when the sections of the book appear out of order. By default, the div sections will be numbered sequentially. The name attributed is used for constructing an index or table of contents. The closing </divn> tags are optional.

5.2. Synchronization Points

Synchronization points are represented with tags of the form

```
<sync type="xxx" value="xxx">
```

Here the type is one of a standard list of types, such as "date", "numeric" or "keyword". These types should have corresponding tables in the database specifying legal values. There are no end tags since synchronization points are locations.

5.3. Index Entries

Index entries are represented with hierarchical keys and a standard reference, using the following syntax:

```
<index type="xxx" key1="xxx" key2="xxx" ... key8="xxx"
  to="toRef">
```

It is not possible to specify more than eight (or some other constant number of) levels of hierarchy in an index, but that should be sufficient. The type attribute is used to signify that the index entry is of a particular type, such as subject index, table of figures, etc. If the index entry refers to a section of text, the endpoint of the range is specified with the "to" attributed. This technique is used because index entries may overlap; it would not be possible to match a closing </index> tag with the corresponding opening <index> tag.

5.4. Dictionary Entries

Dictionary entries are represented with HTML-like syntax. The type attribute specifies the type of dictionary entry, such as lexicon or encyclopedia.

```
<dl type="xxx">
  <dt>term
  <dd>definition
  ...
/</dl>
```

5.5. Additional Tools

Several additional tools are required to perform these synchronization tasks in conjunction with the database for display on the web.

- A *reference linker*, which converts cross references within the database to HTML links.

- A *columnizer*, which combines two HTML documents into parallel columns in one document, with columns synchronized according to synchronization points. The columns might be formed as HTML tables, with new rows starting at each synchronization point.
- A *synchronized scrolling* tool, which displays two texts in separate windows. When one window is scrolled, the other automatically jumps to the corresponding point in the other document. It might be possible to implement this tools with future versions of JavaScript.
- An *index generation tool*, which generates an alphabetical index from an index, dictionary or synchronization table. For large indexes, the generated index may be collapsible and expandable, using JavaScript.

6. Synchronization examples

One example of how this system might be used to address common synchronization problems is the table of contents. A table of contents can be constructed automatically from the primary addressing information or a synchronization table for a text, using the index generation tool. That tool can generate JavaScript code as well, so that the table of contents will be collapsible. The table of contents can be added to the top of a document or displayed in a separate window.

As another example, footnotes can be handled as a separate document with footnote-type synchronization points. The footnotes for a document can be displayed in a small, synchronized window below the main window, so that any scrolling to the main window will cause the relevant footnotes to appear in the bottom window.

To handle parallel editions of a text or a text with commentary, both texts should use the same primary addressing or synchronization scheme. After the relevant database tables are joined, the result can be displayed in columns on the page using HTML tables, or in different frames or windows, coordinated so that scrolling in one window causes the other to scroll to the corresponding location.

One problem that I have previously found difficult to deal with in putting books on the web is handling the subject index in the back of many books. The information there is often important for locating topical information in a book and hard to duplicate with keyword searches, but since page numbers in the index have no reference in the HTML document, the subject index is useless. This problem can be handled by inserting page break synchronization points in the electronic text at points where pages breaks occur in the paper version. A page number in the index may then be looked up in the synchronization table to find the appropriate reference in the main text. Thus, clicking on an index entry would jump to the page from the print edition.

7. Conclusions

This paper describes a framework for representing and handling meta-data useful for multi-document synchronization. The meta-data is imported into a relational database, and database operations (together with a couple of utilities) are used to handle common synchronization and presentation problems. This approach has the benefit that most of the "dirty work" of searching, sorting, and joining is performed by the database. This is especially beneficial since database operations are well understood and high-quality database software is relatively inexpensive. A test-bed implementation of this system is planned for the Christian Classics Ethereal Library,⁵ a small experimental library at Wheaton College. I anticipate that this approach will make possible a much richer interaction between documents, including sophisticated searching, synchronization of related texts, and global library indexes. These capabilities represent a potentially significant advantage of an electronic library over traditional books.

⁵ <http://ccel.wheaton.edu>