Title
Philobiblion: Problems and Solutions in a Relational Data Base of Medieval Texts

Journal
Literary and Linguistic Computing, 6(2)

Author
Faulhaber, Charles B.

Publication Date
1991

Peer reviewed

Philobiblion: Problems and Solutions in a Relational Database of Medieval Texts

CHARLES B. FAULHABER
University of California, Berkeley, USA

Abstract

Philobiblion is a relational database for the cataloguing and study of the primary textual sources, manuscript and printed, for the study of medieval and early modern culture, along with the institutions and persons who created them. Here we present some of the problems we encountered in fitting the data into a relational dbms format (Advanced Revelation) as well as the solutions devised: (1) sorting manuscripts and texts in topographical order; (2) explicitly noting the contingent nature of the data as well as its source; (3) using a data type field to classify sets of associated multivalued fields; (4) establishing views into two different tables from a single field; (5) devising mechanisms to allow dissemination in printed and networked form as well as on diskette or CD-ROM disk; (6) subject access. Unsolved problems include: (1) deciding how to define a 'text'; (2) establishing compatibility with previous standards; (3) normalized searching on unnormalized text; (4) defining reciprocal but asymmetrical relationships; (5) remote data entry; (6) balancing the need for complexity and completeness against interactive usability.

Introduction

The *Bibliography of Old Spanish Texts* (*BOOST*) records the primary sources, manuscript and printed, for the study of medieval Spanish culture. Established in 1974 as part of the computer-based *Dictionary of the Old Spanish Language* at the University of Wisconsin (see Nitti 1978 and Kish 1979), it was originally intended to aid in the selection of the text corpus for the dictionary; and the first two editions (Cárdenas *et al.* 1975, 1977) focused almost exclusively on that purpose. However, it soon became evident that *BOOST* could also serve as a union catalog of medieval Spanish texts; and the third edition (Faulhaber *et al.* 1984) was undertaken by an international team with that specific aim in mind.

BOOST's db system at Wisconsin was derived from FAMULUS (cf. *FAMULUS* 1977 and McCrank and Batty 1978), with the data in a single flat file with nineteen data fields in each record. When we moved it to Berkeley in 1984 we ported it into SPIRES, an IBM mainframe system developed at Stanford, in the same format, a relatively simple process. In 1987, because of the increasing difficulty of maintaining the data in a flat file format, with its inherent redundancy, we ported it again, this time into seven related tables with a total of over 300 data elements running under Advanced Revelation, a high-end PC-based relational/hierarchical db package with some significant advantages over most of its competitors: variable-length fields, no limits on the number of individual fields or records, and a maximum size of any record or any field within a record of 64 Kb. Access to indexed fields and related records in other tables is facilitated by a variety of pop-up menus and other window-based mechanisms. The new dbms was christened *Philobiblion*, after the fourteenth-century French scholar Richard of Fournival's description of an ideal library and cataloguing system. Currently it consists of ten tables with 450 data elements. In addition to *BOOST*, it is now being used for the *Bibliography of Old Portuguese Texts* (*BOOPT*) and the *Bibliography of Old Catalan Texts* (*BOOCT*; see Concheff 1985),[1] and we intend to continue to enhance it as a general-purpose tool for similar text corpora. Since there are no Library of Congress MARC standards for catalogs of older manuscripts, we created our own format. It is the fruit principally of my experience in compiling the catalog of medieval manuscripts of the Hispanic Society of America (Faulhaber 1983 and forthcoming *b*) and of a study of the file structure of MEDIUM, a database of medieval manuscripts in European libraries developed by the Institut de Recherche d'Histoire des Texts (see Minel 1986).

The major tables are MS.ED, UNIFORM.TITLE, and ANALYTIC, the latter of which can be visualized as the intersection of the first two. Each record in ANALYTIC describes a copy of a given text in a given manuscript or edition, with the title of the text as given in that copy, the incipits and explicits of the various parts of the text in narrow paleographic transcription, and any other features of interest to that copy. In traditional bibliographical terms, ANALYTIC analyzes the contents of a given physical volume, manuscript or printed, whereas MS.ED gathers together all of the information concerning that volume (shelfmarks, codicological data, provenience, etc.) and makes it accessible to the ANALYTIC record of any text contained in it via a relational link. Similarly, all of the invariant information about a given text (authorship, standardized title, date of composition, etc.) is gathered together in UNIFORM.TITLE and linked to every record in ANALYTIC containing that text. These three major tables are supported by seven satellite tables which serve as authority files (BIOGRAPHY, LIBRARIES, INSTITUTIONS, BIBLIOGRAPHY, GEOGRAPHY, SUBJECTS, COPIES).

In almost three years' experience with this system we have encountered and solved a number of unexpected problems, chiefly through the ingenuity of John May, an experienced Advanced Revelation programmer and student of database design. We have also faced some design issues which remain to be solved. Herewith a sampling of each.

Correspondence: Charles B. Faulhaber, Department of Spanish and Portuguese, University of California, Berkeley, CA 94720, USA;
E-mail: cbf@~~athena~~garnet.berkeley.edu

Literary and Linguistic Computing, Vol. 6, No. 2, 1991

1. Solutions

1.1 *Sorting in Topographical Order*

In the flat file version we had already encountered the problem of sorting manuscripts in topographical order. Because of the enormous variety of shelfmark systems used in libraries around the world, and the arbitrary way in which each library establishes its canonical topographical sequence, it is not possible to use the shelfmarks themselves to sort manuscripts in topographical order. In the Escorial library just outside of Madrid, for example, the canonical order requires that shelfmarks beginning with *&* be listed *after* shelfmarks beginning with lower-case *h* and *before* shelfmarks beginning with upper-case *H*. Thus we use a *manu.seq* field in MS.ED to store arbitrary alphanumerics in order to sort the manuscripts in a given library in canonical topographical order. The real problem is that precisely because the various libraries are unsystematic, assignment of such numbers has to be done manually. A program facilitates this process by maintaining a separate index of *manu.seq* numbers and shelfmarks which can be examined at will through use of a popup. Precisely the same problem occurs with the order of texts within a given volume. Because foliation or pagination may be absent, incorrect, or repeated, ordering cannot be based simply on the contents of the *doc.loc* field in ANALYTIC, which records foliation or pagination. Instead we use a *text.seq* field for the purpose of sorting texts in the correct order within a volume, and a program which allows us to see at a glance the correlation between foliation or pagination and the *text.seq* numbers of all of the texts contained in that volume.

1.2 *Contingent Nature of the Data*

Much of the data with which we deal is contingent in nature, particularly as we go further back in time: dates are approximations or uncertain; each of a number of scholars may offer a different opinion on, for example, the authorship of a given text or the printer of a given edition. We capture the contingency of the data and the uncertainty of attributions through the device of multiple associated fields, one (or more) for the data, another for the qualifier (with a *.q* extension), a third for the source (with a *.bas* extension). For example, in dating printed editions, we must allow for a beginning date and an ending date; and in both cases they can be problematic. With *bd*, *bd.q*, *ed*, *ed.q*, and *d.bas* we can record all of the possible dates which have been proposed for a given edition, their degree of reliability, and the basis for each (e.g. the name of the scholar who proposed it). These associated multi-valued fields can be viewed as a two-dimensional matrix, with the associated fields aligned horizontally, and the sets of values in rows underneath them, with all values in a given row linked syntagmatically as part of the same association, while all values in a given column are linked paradigmatically, as members of a given data element (see Fig. 1):

bd	*bd.q*	*ed*	*ed.q*	*d.bas*
1488	[?]	1491	[?]	Vindel
1489	*c.* [?]			Haebler

Fig. 1. Associated multi-valued fields to record printing dates.

So far we have added such contingency/authority fields only for data structures containing dates and names, information inherently of interest to many scholars and therefore the object of searches as well as reports. We have not used them for the physical description of a manuscript, for example, even though scholars may disagree on its size or the number of leaves in it. In theory such disagreements can be resolved through the simple expedient of looking at the volume; in practice this is often not possible. As the amount of data has multiplied, we have frequently found that information about a given manuscript or edition may come from half a dozen different sources; and we need to be able to note those sources explicitly in order to maintain an audit trail. Thus we are beginning to see the necessity for contingency and authority fields for all data elements. To provide them systematically would triple the size of the data structure, however, with a commensurate increase in system overhead. We need a mechanism which would allow us to associate contingent and authority information with every single field, but without requiring us to set up such contingency data elements explicitly. One possible solution would be an ancillary table for such information, accessed via a hot key. Thus whenever there was disagreement, a variety of opinions, uncertainty, or simply the need to attribute information to a specific source, the hot key would bring up a pop-up window with just two fields: *qualifier* and *basis*. Information recorded there would be saved in a record linked to the specific field in which the pop-up was invoked.

1.3 *Subvalues in Associated Multi-valued Fields*

While the use of associated multi-valued fields allowed us to establish data structures, it did not solve the problem of subvalues within such a structure, a third dimension, as it were, in a two-dimensional matrix. For example, the imprint of an early printed book associates a place, a printer (and sometimes a publisher or patron), and a date. We have already seen how the date requires five fields to specify its various possibilities. Location requires three: a *geoid* which links the record to the GEOGRAPHY table, and the contingency/authority fields, *geo.q*, and *geo.bas*. The printer also requires three, a *bioid* field to link it to the printer's authority record in BIOGRAPHY, and the same contingency/authority fields. Frequently, however, a book was printed not by one printer but rather by several working together. Thus, instead of one bioid, we must record several, all of which must be linked to the same date and location. We have solved the problem by recording these bioids as subvalues to the printer field, a procedure explicitly allowed by Advanced Revelation. We have improved on the specific mechanism, however, by placing these subvalues in a pop-up window which appears automatically whenever the cursor moves into the printer field. The same mechanism is used for multiple authors in the *author* field of UNIFORM.TITLE.

1.4 *Data Typing of Associated Multi-valued Fields*

A fourth problem we faced was that of tracking the various ways in which people and institutions are related

to texts, to manuscripts, and to each other. In the original flat file we had concerned ourselves only with the authors and translators of the texts catalogued; so there was a field for authors and another for translators. In fact, there are a number of other fairly common ways in which an individual can be related to a given text: as its dedicatee, as the composer of the music to which it is set, as its editor or compiler, as its addressee (in the case of letters). Our first instinct was to try to capture the common relationships in specific fields, with one each for translator, editor, composer, and dedicatee, replicating the design of the flat file. Each of these fields contained the record key of a person in the BIOGRAPHY table, on the basis of which the actual name of the person was brought over symbolically into the UNIFORM . TITLE window. In terms of database design, however, this solution left much to be desired, since for any given text usually only one or two of these fields would contain data. Thus we found our data structures becoming increasingly more cumbersome; but only a small percentage of each one was actually in use in a given record (see Fig. 2).

Other Persons Related to Text:

transl	*compos*
1025 Carlos de Arago\n	
editor	*dedic*
	1031 I\n~igo Lo\pez de Mendoza
connect	*assc . nam*
BASED ON VERSION OF	2073 Angelo Decembrio

Fig. 2 Separate data fields for each relationship.

Penny Small, creator of the *Lexicon Iconologicum Mythologiae Classicae* at Rutgers (see Small 1987, 1988), suggested a technique to tighten up the data structure considerably by using a set of multiple associated fields in which one field would specify the *kind* of relationship while the others identified the person, a design strategy which we have now implemented consistently in all of our tables. In the case set forth above, the result is a set of associated name fields, where the first field in the set specifies the type of association (ADDRESSEE, TRANSLATOR, DEDICATEE, COMPILER, COMPOSER), while the others specify the individual (linked to BIOGRAPHY via the bioid), contingency, and authority (see Fig. 3).

Other Persons Related to Text:

connect	*assc . nam*	*an . q an . bas*
TRANSLATOR	1025 Carlos de Arago\n	
DEDICATEE	1031 I\n~igo Lo\pez de Mendoza	
BASED ON	2073 Angelo Decembrio[2]	
VERSION OF		

Fig. 3 Classification of associated multi-valued fields.

We have used the same strategy very effectively in BIOGRAPHY, where we were faced by a similar problem, the relationship of a given individual to others in the database. Again, our first instinct was to record basic family relationships in separate fields (e.g. *father*, *mother*), but here also it rapidly became evident that the vast majority of these fields would go unused in any given record. We instead set up a data structure which links the relationship type with the record key of the related person, the dates of the relationship, and the standard contingency/authority fields. The most common relationships (e.g. mother, father, sister, brother, patron, client, collaborator, correspondent) can then be chosen from a pop-up window, which eliminates the necessity of manual entry (and therefore the possibility of spelling errors). The other advantages of the pop-up are that it can be readily expanded should experience indicate that other types of relationships need to be added; and that it can serve as a validation mechanism to ensure that only the values listed in it will be accepted as valid. Any others will be rejected by the system (see Fig. 4).

1.5 *Pointing to Two Different Tables from a Single Field*

The 'previous owner' set of associated fields in MS . ED illustrates yet another problem, the fact that a particular function or relationship may be carried out either by persons or institutions. If the previous owner of a given volume is a person, then a link must be established to BIOGRAPHY; if a corporate body, then to INSTITU-

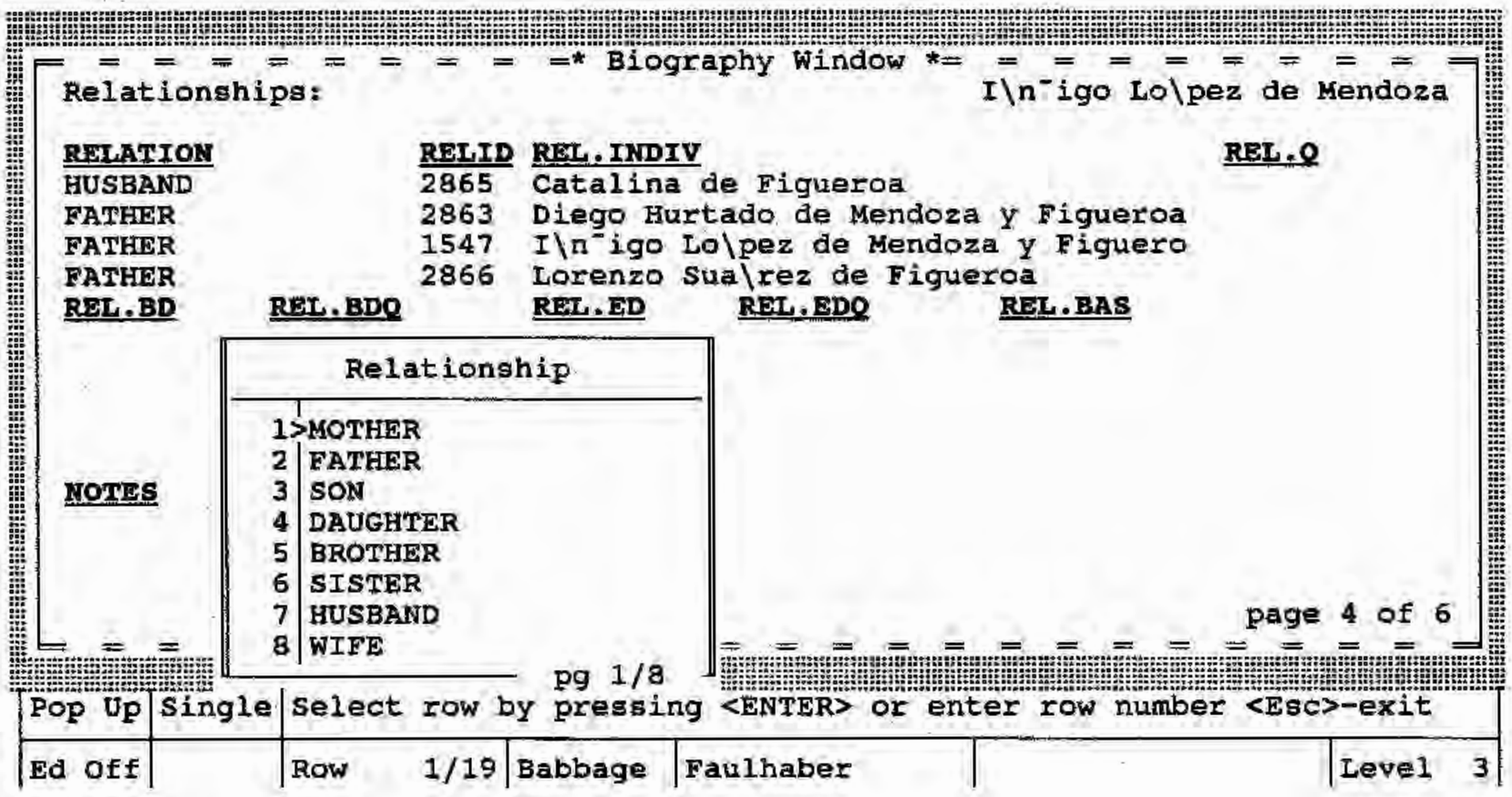

Fig. 4 Pop-up access for validated data entry.

TION. Therefore we must be able to place record keys from two different tables into the same field, a procedure for which Advanced Revelation makes no provision; or we must establish two separate fields in the association, one for corporate owners, one for personal owners. In all cases, one of those fields would remain blank, a trivial matter in itself but one with important consequences for searching. If we wished to search for all manuscripts of known provenience, we would need to look at the contents of both fields rather than one, a needless complexity.

The solution was to indicate the type of owner at the beginning of the association in a field which accepts either 'P' for personal or 'I' for institutional. On the basis of the contents of that field, a program automatically connects the *owner* field to BIOGRAPHY or INSTITUTIONS respectively. Thus all searches for previous owners need look at only one field; while—should it be of interest—we can still specify a search of either corporate or personal owners. The same mechanism can be used to indicate corporate or personal authorship in UNIFORM . TITLE.

1.6 *Formatting for Printed and On-line Versions*

For years to come databases like *BOOST* must exist in both print and machine-readable form; and, in the latter, in both standalone and on-line formats. The printed version is needed for the still-numerous scholars who do not use computers or who wish to have a handy reference work; on-line formats are required in order to provide ready access to updates. Thus we must be able to map *Philobiblion* into MARC (machine-readable catalog) format as well as provide phototypesetter output for printing. Solutions which take into account only a single electronic format are not viable, given the scholarly clientele which must be served.

The fine-grained nature of *Philobiblion*'s data structure is such that mapping it into a template for printing or into MARC format is a relatively straightforward process, even when the data are complex. For example, the associated multi-valued fields of *doc . loc* and *dl . type* in MS . ED allow us to mix pagination and foliation in the description of a single volume simply by associating the appropriate data types. Thus in order to indicate flyleaves in a volume otherwise paginated:

DOC . LOC	DL . TYPE
2	fl.
99	pp.
3	fl.

This becomes, in a printed report: '2 fl. + 99 pp. + 3 fl.', with the report generator supplying the connecting '+'.

A similar technique is used to record the incipit(s) and explicit(s) of a text in ANALYTIC. We can identify each section of the text (e.g. prologue, translator's prologue, preface, text, etc.), the location of the incipit of that section, the incipit, the location of the explicit, and the explicit:

IE . TYPE	I . LOC	INCIPIT	E . LOC	EXPLICIT
prol	93vb	Este segundo fue	95rb	escripta en tabla

Formatting such a set of fields for typesetting is then a straightforward exercise, e.g. '[prol. inc. f. 93vb] Este segundo fue ... [expl. 95rb] ... escripta en tabla'.

To map *Philobiblion* to the MARC format, the standard for on-line public access catalogs (OPACS) as well as for the exchange of bibliographical data, is also straightforward, despite the difference in the data structures and file format of the two systems. *Philobiblion*'s UNIFORM . TITLE table, for example, allows us to link the *author* field either to INSTITUTIONS or BIOGRAPHY, depending on whether the author is a person or a corporate body, via the use of an *author.type* field containing either 'I' or 'P'. MARC, however, requires that the names of personal authors be placed in field 100, and corporate names in field 110. A report generator would accomplish the appropriate mapping through a simple if-else declaration:

If *author.type* contains 'P', then map *author* to MARC field 100
Else map *author* to MARC field 110

1.7 *Extensions to the Database Structure*

In terms of additional tables, the only one which we know to be needed but have not yet implemented is a topical subject table (the equivalent of the MARC 650 field). A relational system is tailor-made for subject indexing, since it lends itself nicely to the construction of a thesaurus, with each main subject-heading given a separate record. Within the record itself the main heading, rejected headings, related headings, and broader headings can each be given their separate fields and indexed together. Thus, no matter what the user searches for in the index, the authorized subject heading will be returned. Indented fields are associated in a multi-valued data structure with the preceding field:

subid = Record key
head = Heading
  h . type = Heading type (e.g. topic, subtopic, location of event)
  h . bas = Authority for assignment of *head* (e.g. Library of Congress)
othr . head = Variant heading
  oh . type = Type of *othr . head* (rejected, French, Spanish)
  oh . bas = Authority for assignment of *othr.head*
broader = Broader heading
related = Related heading

2. Unsolved Problems

The answers to other problems are less obvious. Sometimes we can sketch out several possible solutions, but it is difficult to say which is correct without testing them in a production environment. The issues which still continue to trouble us are listed from most specific to most general.

2.1 *Porting from a Flat File dbms into a Relational dbms*

The porting process, not so much a theoretical problem as a practical one, was the first major difficulty we faced. We solved it by brute force rather than elegance. Although John May spent almost three months writing a program to distribute information from the original nineteen fields in the flat file to the more than 300 then

in *Philobiblion*, it was impossible to accomplish the task algorithmically. Even after massaging the data intensively, there still remained a great many inconsistencies and ambiguities, particularly in such unstructured fields as *notes*, used as a catch-all for material which did not fit elsewhere. In fact, it took almost two and a half years of slogging to check systematically all of the data from the *c.*4000 records in the flat file and to make corrections (e.g. spelling errors in personal names created duplicate or even multiple records in the BIOGRAPHY file, which had to be eliminated). (For more detailed comments on the porting process, see Faulhaber forthcoming *a*).

2.2 *Definition of a 'Text'*

While defining the concept of 'text' is a problem of some moment for post-modern critics, we are concerned with it for entirely pragmatic reasons. We are still not entirely satisfied with the way we define a text within the UNIFORM.TITLE table. In theory a text is the smallest independent unit within a printed edition or a manuscript. Thus a manuscript may contain only one text or many, each of which is duly linked via ANALYTIC to its appropriate record in UNIFORM.TITLE. But how do we treat works which contain other works, such as poetic anthologies? Such a work is a unique collection, with its own peculiar physiognomy and, occasionally, artistic identity. Should there be an entry in UNIFORM.TITLE for the *Carmina burana* anthology as a whole or for Petrarch's *Canzoniere*? If the *Canzioniere* is extant in a number of manuscripts and editions, each containing substantially the same set of lyric poetry, the same cannot be said for the *Carmina burana* or indeed formost medieval and Renaissance poetic collections (*chansonniers, cancioneros, cançoners, cancioneiros, liederbücher*, etc). Each of these is indeed an anthology with its own particular physiognomy, but at the same time, for manuscript collections, there exists a one-for-one correspondence between that particular collection and the physical volume which contains it. Thus, if we record these collections as bibliographical types in UNIFORM.TITLE, by definition each type corresponds to a single token—a situation quite different for the individual text type, which may have many tokens. But if we do not record such anthologies in UNIFORM.TITLE, where do we record them? In fact, we have listed these collections along with independent texts in UNIFORM.TITLE, despite the theoretical objection as well as some practical problems. We have found, for example, that the existence of such collections skews the data by inflating the number of single-manuscript works. If we leave them in the database, as I suspect we shall, we must find a way to identify them as a class, with a field for text type where type = collection (cf. Sperberg-McQueen and Burnard 1990; 175).

We must also find a more satisfactory way of indicating the relationship between collections and the texts which they contain. Currently we indicate this relationship at the physical level, simply because all texts in a given manuscript share that manuscript's *manid*, or record key. This works satisfactorily for unique manuscript collections but falls down precisely in the case of more generic ones, like Petrarch's *Canzoniere*. A possible solution would utilize the *relation* and *r.texid* fields in UNIFORM.TITLE, which record the relationship between a given text and other texts in the database. Thus, for every text contained in a particular collection, we would specify, under *relation*, 'contained in', and under *r.texid* the record key of that collection. The only real drawback is that the presence of individual texts within the larger collection must be recorded twice, once for the physical volume itself, in ANALYTIC, and once for the collection type, in UNIFORM.TITLE.

2.3 *Compatibility with Previous Standards*

Since what we are attempting to provide is essentially a bibliographical tool, it must be compatible with previous efforts to provide standard bibliographical descriptions. Of these the most important are the *Anglo-American Cataloguing Rules* (2nd edn.) and the *ISBD(A): International Standard Bibliographic Description for Older Monographic Publications (Antiquarian)*, which set forth precise rules for the cataloguing of rare books. Unfortunately, neither of these norms was written with an eye toward electronic databases; and they are not even handled by the MARC format very gracefully. For *Philobiblion* the basic difficulty lies in the requirement to record the precise form which publication data takes in the edition itself, a capability which we do not yet have, except on an *ad hoc* basis. *Philobiblion* is geared toward recording the canonical forms of personal, corporate, and place names in order to facilitate searches, reports, and data entry. Thus, to record the imprint of a printed book, we simply establish a relationship with the record in GEOGRAPHY containing the name of the place where the edition was published and a relationship with the record in BIOGRAPHY containing the name of the printer. We need to find a way to record the fact that a book printed in Lyons gives the place of publication as Lugdunum. In fact, because book piracy was rampant, frequently the imprint given in a book is totally false. Thus of four early editions of the early sixteenth-century Spanish classic, *Celestina*, all purporting to have been published in Seville in 1502, three were indeed published there, but in 1511, 1513, and 1518, while the fourth was published in Rome in 1520. We need to be able to record both the true and the false imprint and indicate that the former has been determined editorially.

2.4 *Normalized Searching on Unnormalized Text*

Exactly similar, and far more difficult to achieve, is the necessity of maintaining, in so far as possible, the uniqueness of our textual data—orthography, punctuation, abbreviations—what Penny Small calls 'preserving the mess'. It seems self-evident that we should do this, since what appears to be irrelevant to one generation of scholars may be highly important for the next. If we carry this to extremes it implies the necessity of some sort of facsimile of the manuscripts; and we hope to marry *BOOST* and its congeners with digitized facsimiles on a optical disk which would also contain machine-readable transcriptions of the texts themselves (see note 1). Our immediate problem, however, is more limited. We can transcribe titles and incipits exactly as

they are found in the sources, but if we do so it becomes impossible to find anything. The medieval Spanish representations of modern Spanish *vivi* 'I lived', for example, can be spelled in at least fourteen different ways (*vivi, vivj, vjvj, uiui, ujuj, uivj, viuj, vjui, ujvi, bivi, bivj, bjvj, biui, bjuj*, etc.) because of the interchangeability of *b*, *u*, and *v* on the one hand and *i* and *j* on the other. Precisely to avoid this problem, John Nitti, the original designer of the Wisconsin db, insisted that every work listed in it be given a standardized title using modern Spanish orthography, the title currently found in the UNIFORM . TITLE table, as well as a specific title in medieval orthography. Searches and sorting could then be done on the basis of modern spelling.

In order to carry out the same kinds of operations on raw medieval data we need a mechanism which will allow us to find what we are looking for through 'fuzzy' searches, or what Mark Olsen calls 'inexact pattern matching' (1988). This is not a unique problem, and a variety of software and hardware solutions have been proposed for it. The approaches Olsen mentions are all based to one degree or another on phonetic and phonemic representations and are therefore highly language specific; and if we were only concerned with Old Spanish a satisfactory solution would not be difficult to find. For example, Francisco Marcos-Marín (forthcoming 1991) has provided a set of algorithms in *UNITE*, his textual criticism package, which essentially filter out orthographic variation (e.g. *ff- ss-*, and *rr-*, which stand in free variation with *f-*, *s-*, and *r-*, in syllable–initial position, are simply eliminated in favour of the latter).

Another possibility would be to use sophisticated pattern matching techniques. For example, in the case of *vivi* above it would be possible to find all of the forms listed by seeking for all permutations of b/u/v and i/j/y or by devising complex pattern matches (buv/ijy/buv/ijy). Nevertheless, the possibilities are so various, especially when dialect forms must be taken into account, that algorithmic solutions do not appear to me to be possible, even within the bounds of a single language, and even less so if we wish the program to serve as a general tool for a broad variety of languages.

We believe that ultimately we will require some sort of morphological parser or lemmatizer in order to associate all of the words found in medieval orthography with their appropriate headword or lemma, in modern spelling (e.g. OSp *dizes* as a variant of mod. *dices*). Such a lemmatizer would allow the user, interactively, to link each word form to its head word. In relational db terms, each headword and its related forms would then constitute a single record in a thesaurus table; and all searches of medieval text (in, for example, ANALYTIC), would be filtered through that table.

2.5 *Bidirectional Relationships*

By definition, all relations in a relational db are reciprocal. This is one of the most powerful features of such a system, since it enables us to keep track of relationships in both directions, both the one-to-many relationship (e.g. from the text type record in UNIFORM . TITLE to the related text token records in ANALYTIC) as well as the many-to-one relationship (e.g. from the text token records in ANALYTIC to the related text type record in UNIFORM . TITLE). From any record in ANALYTIC we can find the corresponding UNIFORM . TITLE record. (While this bidirectionality is not automatically available in Advanced Revelation, it can be achieved through manual establishment of indices.)

If, however, we wish to record the specific nature of the relationship, we have a problem. I have already mentioned the use of a multi-valued data structure to capture personal and professional relationships among persons listed in the BIOGRAPHY table. Such relationships are reciprocal in a peculiar way, for in order for them to be meaningful we must specify not only the fact of the relationship but also its *type*. 'Father' implies 'son' or 'daughter'; 'son' implies 'mother' or 'father'. If we state that Alfonso X is the son of St Ferdinand III, then we also want to state that Ferdinand III is the father of Alfonso X. Right now *Philobiblion* requires that both aspects of the relationship be supplied manually for each person involved. We would like it (*a*) to establish automatically the reciprocal relationship whenever the first is specified or (*b*), preferably, generate the relation algorithmically on the basis of the data set down in one of the related records (in order to avoid the necessity of supplying two sets of physical data when one of them can be calculated on the basis of the other).

For family relationships this should not be a problem. Whenever we supply 'son', the system will automatically generate 'father' or 'mother' (depending upon the sex of the person involved) in the related record. Even some professional relationships are unproblematic. 'Student' implies 'teacher'; 'client' implies 'patron'. Unfortunately, some relationships are semantically asymmetrical, in the sense that only one side can be specified. Thus we can state that Juan de Mena was the secretary of John II of Castile, but what is the reciprocal of that relationship? We could avoid the problem by establishing a second data structure with an inverted relationship. Thus in John II's record we could say that Juan de Mena was his secretary; whereas in Juan de Mena's record we could say that he was secretary to John II. But again we introduce what seems to me to be an undesirable level of complexity, undesirable because it sins against Occam's razor: thou shalt not multiply entities, i.e. data fields, unnecessarily. Where suitable terminology is lacking, it would be much simpler to signal the reciprocal of a given relationship through the use of an appropriate symbol, e.g. <.

2.6 *Remote Data Entry*

The ability to merge data entered or updated on machines not linked to the main database will be much more difficult to achieve, although it is not impossible. For existing records it would require the ability to merge records in a given table on the basis of their record key while allowing the user to review manually all records with conflicting data. For new records it would require maintenance of several distinct sets of record keys, and the ability to integrate them manually into a master set. In a project like *BOOST*, however, with collaborators in five different countries, such merge capabilities would be highly desirable.

2.7 *Complexity versus Interactive Usability*

Finally, and perhaps most basically, there remains the question of whether any db can handle the particularities of medieval manuscripts and medieval literary texts (by extension any manuscripts and texts) without being either Procrustean or too complex to use. How complex does such a db need to be in order to handle those particularities; and if it is complex enough to handle them, is it too complex to use? Specifically, can a relational db be sufficiently detailed so that it can offer a codicologically exact as well as indexable description of a codex, its texts, and the people and institutions related to them? Can it serve as the basis for a printed version or for porting into other formats and yet be fast enough and friendly enough to be usable in electronic form as well?

The Scylla of complexity is related directly to the Charybdis of interactive usability. If the dbms lacks data fields for information which the user regards as important or is so schematic that the user cannot find his or her way around, it will not be used. If, on the other hand, implementation of those data fields or the use of mechanisms to enrich the on-screen information environment slows the data base down, then it may not be used or, if it is, may in fact be less cost effective than manual methods. Let us look at a specific example:

One of the basic principles of relational databases is that, in order to facilitate both data entry and maintenance, the same data should never be entered in more than one field. Relations among tables are indicated simply by laying down record keys from one table into the related table. However, to make the database usable in interactive form, symbolic fields must be used in profusion to bring the data from the first table and display it in the entry screen of the second. Thus in the ANALYTIC table, *texid* contains the record key to the related record in UNIFORM.TITLE; and on the basis of that key a program reaches out to UNIFORM.TITLE and assembles a string composed of the author's name, the title of the text, the name of the translator (if necessary), and the date of composition, and lays it down at the top of each page of the ANALYTIC screen. Similarly, *manid* contains the record key to the related record in MS.ED; and on the basis of that key another program lays down the location of the manuscript (city, library, shelfmark, date) or the imprint of the edition (location, printer, date) on the second line of each page. But this process slows the database down considerably, forcing the user to wait while the system collects data from dozens of different fields in the two related tables and flashes it on to the screen. Running under MS-DOS, we have found it necessary to use an 80386-based machine (IBM PS/2 model 70) running at 25 megaherz to achieve acceptable performance, simply because so much background processing was going on as the user moved from screen to related screen. When we moved from a PS/2 model 50 (80286 microprocessor), the speed of data entry increased by about one-third simply because the operator did not have to wait for the machine.

A similar hardware-based limitation is the amount of RAM memory required for interactive use. In principle Advanced Revelation allows the user to navigate from one window to another in hypertext-like fashion—from a given manuscript in MS.ED to a text contained in that manuscript in ANALYTIC, to the UNIFORM.TITLE record for that text, to the author in BIOGRAPHY. ... In practice, in a 640 Kb system the program crashes because of lack of memory after four or five moves. Advanced Revelation can take advantage of both expanded (640 Kb–1 Mb) and extended (above 1 Mb) memory with an appropriate utility; but it is evident that *Philobiblion* is pushing the limits of both the hardware and the software.

Beyond the specific problems lies that common to all software developers. How much is enough? Each new solution raises new problems. And as we solve them the system grows more and more complicated. We would like to believe that the end is in sight, but it is quite clear that *Philobiblion* has only begun to capture the complex realities of medieval texts and the society which created them.

Notes

1. *BOOST* will also form part of *ADMYTE: Archivo Digital de Manuscritos y Textos Españ*oles, a CD-ROM database which will also contain a large corpus (c.100 MB) of medieval Spanish texts originally transcribed for the Madison Dictionary project or newly transcribed from incunabula in the Biblioteca Nacional (Madrid), a version of TACT, the text analysis program developed at the University of Toronto by John Bradley and Lidio Presutti, specifically designed to work with the Madison texts, and UNITE, Francisco Marcos Marín's program to automate certain aspects of textual criticism. See Faulhaber and Marcos-Marín 1990. *ADMYTE* has been accepted as an official project of the Sociedad Estatal para la Ejecución de Programas para el Quinto Centenario (Spanish Quincentennial Commission) and is scheduled for release in the first half of 1992.
2. The *connect* field is not limited to the types given above but in fact can be used to specify *any* relationship between the text and a person associated with it. Diacritics are represented with lower ASCII digraphs (e.g. o\ = ó) which can be mapped to any output device as required.

References

Cárdenas, A., Nitti, J. J., and Gilkison, J. (1975). *Bibliography of Old Spanish Texts*, 1st edn. Madison, WI: Hispanic Seminary of Medieval Studies.

—— and Anderson, E. (1977). *Bibliography of Old Spanish Texts*, 2nd edn. Madison, WI: Hispanic Seminary of Medieval Studies.

Concheff, B. (1985). *Bibliography of Old Catalan Texts*. Madison, WI: Hispanic Seminary of Medieval Studies.

FAMULUS: Reference Manual for the 1110 Incorporating Change Notices A, B & C (1977). Madison, WI: MACC Academic Computing Center. University of Wisconsin.

Faulhaber, C. B. (1983). *Medieval Manuscripts in the Library of the Hispanic Society of America: Religious, Legal, Scientific, Historical, and Literary Manuscripts*, 2 vols. New York: The Hispanic Society of America.

—— (forthcoming *a*). 'Bibliography of Old Spanish Texts: Evolution of a Data Base'. *International Conference on Data Bases in the Humanities and Social Sciences (Montgomery, 11–13 July 1987.*

—— (forthcoming *b*). 'Cataloguing Manuscripts with the Aid

of the Computer: The Case of The Hispanic Society of America.' *Homenaje a Maria Teresa Cardó.*

Gómez Moreno, G., Mackenzie, D., Nitti, J. J. and Dutton, B. (1984). *Bibliography of Old Spanish Texts*, 3rd edn. Madison, WI: Hispanic Seminary of Medieval Studies.

—— and Marcos-Marín, F. (1990). 'ADMYTE: Archivo Digital de Manuscritos y Textos Españoles.' *La Corónica* 18.2: 131–45.

Gorman, M. and Winkler, P. W. (1988). *Anglo-American Cataloguing Rules*. 2nd edn., 1988 revision. Ottowa: Canadian Library Association; Chicago: American Library Association.

Kish, K. (1979). 'The Wisconsin Seminary of Medieval Spanish Studies', *La Corónica*, 8: 84–7.

McCrank, L. J. and Batty, C. D. (1978). 'The Mt. Angel Abbey Manuscript and Rare Books Project: Cataloguing with FAMULUS', *Computers and the Humanities*, 12: 215–22; bibliography p. 221.

Marcos-Marín, F. (forthcoming). 'Computers and Text Editing. A Review of Tools, an Introduction to *Unite*, and Some Observations Concerning its Application to Old Spanish Texts', *Romance Philology*, 45.

[Minel, J. L. *et al.*] (1986). *Medium: Base de données sur le manuscrit médiéval*. Paris: Institut de Recherche et d'Histoire des Textes.

Nitti, J. J. 1978. 'Computers and the Old Spanish Dictionary.' *Computers and the Humanities* 12: 43–52.

Olsen, M. (1988). 'Theory and Applications of Inexact Pattern Matching: A Discussion of the PF474 String Co-Processor', *CHUM*, 22: 203–15.

Small, J. P. (1987). 'Computer Index of Classical Iconography', *ACH Newsletter*, 9.1: 12–14.

—— (1988). 'A Database for Classical Iconography', *Art Documentation* 7.1: 3–5.

Sperberg-McQueen, C. M. and Burnard, L. (eds) (1990). *Guidelines for the Encoding and Interchange of Machine-Readable Texts*. Document Number: TEI P1. Draft: Version 1.0. N.p.: The Association for Computers and the Humanities; The Association for Computational Linguistics; The Association for Literary and Linguistic Computing.

Working Group on the International Standard Bibliographic Description for Older Monographic Materials. 1980. *ISBD(A): International Standard Bibliographic Description for Older Monographic Publications (Antiquarian)*. London: IFLA International Office for UBC.