The Web-Based Bibliographic Information System “*BibIS*”

Specification of the Database Project for
CIS 4301 “Information and Database Management Systems I”
in the Spring Semester 2008
(Version 1.1)

Markus Schneider

1 Introduction

Bibliographic information is required and important in many application areas and for many different tasks. For example, students write reports, master theses, and PhD dissertations. Researchers write journal articles, conference papers, workshop papers, and books. Managers write sales reports and market analysis studies. All these tasks have in common that they contain *bibliographic references*, which we will call *bibtags*, to some technical literature and that these references have to be managed in a consistent and reliable manner. The main goal of this project is to apply database technology to solve this problem. The result is supposed to be a *web-based bibliographic information system* named *BibIS*.

The handling of bibliographic references includes tasks like acquiring, storing, retrieving, changing, and searching bibliographic data as well as user management, project management, transformation of biographical data into different output formats, and thematic category organization. Further, as a very important extension, a PDF version of each publication (called *bibliographic reference file*) that is specified by a bibliographic reference is supposed to be stored in *BibIS*. This enables the user to view the publication on the screen, to print it out, and to send it by email to others.

2 Requirements

In this section, we describe the main and minimal requirements that *BibIS* should satisfy. This means that this specification is incomplete, vague, and not totally clear in all details. This is intended since it gives you the chance to be creative and find new requirements and solutions.

2.1 Support of Different Publication Types

The system should be able to support different publication types like journal articles, conference papers, master theses, PhD theses, and technical reports. Each publication type has a specific set of associated *attributes* like author names, year of publication, page range, and volume. It is important to understand that these publication types should *not* be hard-coded in the system. The system administrator should be able to define them interactively by using the system. That is, at the beginning, *BibIS* does not know about any publication type. These have to be first entered into the system before they can be used (see also Section 2.12).

Table 1 shows you some examples of publication types together with attributes as they have been defined in BibTeX. BibTeX is a textual format for bibliographic references in combination with LaTeX (LaTeX), which is a document markup language and document preparation system that is very popular in the scientific domain (see also Section 2.6). A minimal amount of information should be kept for all references of each

Publication Type	**Publication Attributes**
article	AUTHOR, TITLE, JOURNAL, YEAR, volume, number, pages
book	AUTHOR, TITLE, PUBLISHER, YEAR, volume, series, edition
booklet	TITLE, author, howpublished, year
inbook	AUTHOR, TITLE, CHAPTER, PUBLISHER, YEAR, volume, series, edition
incollection	AUTHOR, TITLE, BOOKTITLE, PUBLISHER, YEAR, editor, volume, series, edition, chapter, pages
inproceedings	AUTHOR, TITLE, BOOKTITLE, YEAR, pages
manual	TITLE, author, organization, year
masterthesis	AUTHOR, TITLE, SCHOOL, YEAR
misc	author, title, howpublished, year
phdthesis	AUTHOR, TITLE, SCHOOL, YEAR
proceedings	TITLE, YEAR, editor, volume, series
techreport	AUTHOR, TITLE, INSTITUTION, YEAR, type, number
unpublished	AUTHOR, TITLE, NOTE, year

Table 1: Examples of publication types and publication attributes as they are defined in BibTeX.

publication type to ensure a certain degree of consistency among the bibliographic data. For this purpose, we introduce the concepts of *mandatory attributes* (written in upper case in Table 1) that have to be specified and *optional attributes* (written in lower case in Table 1) that can be specified for each reference.

2.2 Bibtags as Unique Identification of References

Each reference must have a unique identifier that we call a *bibliographic tag*, or *bibtag* for short. A bibtag does not only have to be valid within the database but also *outside* the database, that is, it must be a stable identifier. The reason is that when references are transformed into an output file in a particular format (like BibTeX or Endnotes), their bibtags should be maintained so that they can be used in external applications (like LaTeX or Word).

Bibtags consist of (at most) three concatenated parts. The first part, called the *author component*, represents the authors of a publication. The second part, called the *year component*, represents the year when the publication was published. The third part, called the *location component*, represents the place where the publication was published. An important task is that bibtags are constructed *automatically* by *BibIS* such that they are unique.

The author component is constructed according to the following rules:

- If a publication has $2 \leq n \leq 4$ authors, the initials of the last names of all authors are taken (e.g., “ADCB” for “James Adam, Ryan Donovan, Ben Cox, and John Benson”).
- If a publication has $n \geq 5$ authors, the initials of the last names of the first four authors are taken, followed by the character ‘+’ (e.g., “EFGH+”).
- If a publication has 1 author, the initial of the author’s last name is taken, followed by the second and third lower case letters of the last name (e.g., “Smi” for “John Smith”).

The year component takes the last two digits from the publication year represented as a four-digit number (e.g., “99” for “1999” or “01” for “2001”). The location component is an acronym that is defined for each conference and each journal (e.g., SIGMOD for the “ACM Conference on Management of Data” or

"TODS" for the journal "ACM Transactions on Database Systems"). BibTeX definitions of these acronyms are available in the files "conferences.bib" and "journals.bib" on the course web site. It can be that for a particular publication type such an acronym does not exist. For other publication types like books or technical reports, a location component does not exist. Instead, we mark them by two capital letters like "BO" for "book" and "TR" for "technical report".

In most cases, this procedure is sufficient to obtain unique bibtags. If a user wants to insert a reference into *BibIS* with an already existing bibtag (this happens extremely seldom), lower case letters 'b', 'c', etc. have to be added to the current bibtag to avoid identifier collisions. Of course, all user actions must be checked for correctness and uniqueness of the identifiers.

As an example, the publication "Markus Schneider & Thomas Behr. Topological Relationships Between Complex Spatial Objects. ACM Transactions on Database Systems (TODS), 31(1), 39-81, 2006." has the bibtag "SB06TODS".

In some cases, it is impossible to determine a unique bibtag with the above method since part or all of the information is missing. For example, the location component (acronym) does not exist for all publication types. Further, the author component and/or the year component can be missing. For these and other cases, you have to devise a method so that these cases can be handled by *BibIS*.

2.3 Thematic Categories

All references entered into a database should be structured according to *thematic categories*. A thematic category determines a research topic that is partially or completely covered by a particular reference. Several thematic categories can be assigned to the same reference. For example, let us assume references about database technology. The label "Databases" could be the root of the hierarchy. At the next deeper level we could have "Multimedia Databases", "Deductive Databases", "Spatial Databases", etc. This hierarchy can then be refined and continued to deeper levels. We assume that such a *category hierarchy* (*category tree*) is global for the whole *BibIS* system and thus for all users. The global hierarchy can only be defined by system administrators but not by normal users. Note that it must be possible to create and update this hierarchy interactively (see Section 2.12). References can *only* be assigned to leaf nodes but not to inner nodes.

2.4 Reusability

It must be possible to reuse the reference information in the database for different applications. That is, only the pure information about references should be stored in the database but not how they are presented, formatted, or visualized. This allows us to reuse the same bibliographic data for many different applications and purposes.

2.5 Searching Biographical Data

The system should enable sophisticated and versatile searching. It should be possible to search for attribute values (alone, in combination by different Boolean operations), according to thematic categories, and for keywords similar to Google search. A clear search strategy has to be developed. The display of the result should be split up according to the publication types since they have different collections of attributes. From the result, the user should be allowed to select those of interest and to let them transform into a BibTeX output file (see Section 2.6).

2.6 Transformation of Biographical Data into Different Output Formats

An important use of biographical data is as references in documents as described above. Therefore, it must be possible to transform them into output files according to particular output formats. An output format of interest is BibTeX, which works together with LaTeX. LaTeXis a document preparation and high-quality typesetting system. In this project, we will focus exclusively on BibTeX.

Several files can be found on the course web site that help you better understand BibTeX. The file “templates.bib” presents the publication types and attributes of Table 1 in BibTeX notation. The file “my.bib” gives you an example of a BibTeX output file. The files “conferences.bib” and “journals.bib” offer acronyms of conference names and journal titles. These acronyms are used in BibTeX entries.

2.7 Management of Reference Files

Besides storing references, a main task of *BibIS* is to manage reference files. A *reference file* contains an electronic version of a publication in a particular format like PDF, POSTSCRIPT, Word, etc. It should be asked for when the reference is inserted into the system. A reference file allows us to display a publication on the screen, to print it out, and to send it to other people. Reference files can be found in the Internet (e.g. on CiteSeer (*http://citeseer.ist.psu.edu/*)). A reference without its corresponding reference file should be an exception in *BibIS*. That is, we prefer to have complete information about each reference. It must be possible to visualize reference files on the computer screen and to print them out.

Names of reference files leverage the existence of bibtags in order to be unique. If *X* is a bibtag and the reference file is available in PDF, POSTSCRIPT, Word, etc., the name of the reference file is *X.pdf*, *X.ps*, *X.doc*, etc.

2.8 Definition of Group Projects

Documents are usually written in groups with a certain goal in mind. Such a goal can be to write a research paper, a book, a project deliverable, or a report. *BibIS* should support the definition of projects, the assignment of members (users) to groups, and the assignment of papers to projects. The assigned papers form candidate papers that later possibly appear in the bibliography of a publication. It must be possible to transform all or part of the assigned project papers into the BibTeX output format.

2.9 User-Friendly User Interfaces for Different User Groups

The three main user groups using *BibIS* are guests, (normal) users, and system administrators. Users of all groups must enter the system through an account and a password. Guests get temporary, read-only access to the database. Users get permanent, read-only access to all references stored in the database and in addition write access to all references entered by them into the system. Administrators obtain full access to all components of the system. Each user group must get its own, tailored and user-friendly user interface providing the functionality that it is allowed to use and that is appropriate for it.

An important aspect is that although *BibIS* is web-based, it is supposed to be a closed system. That is, not everybody is allowed to access the system. The reason is that it takes an enormous effort and is very time-consuming to find and collect all the bibliographic references and reference files in the Internet. Hence, it is regarded as a value that one does not want to share with everybody. Therefore, a person who wants to have guest or normal user status must first apply for an account. The system administrator will then make a decision about the request. A request can be accepted or rejected by sending a corresponding email to the requester. In case of acceptance, the guest or normal user gets an account and a password.

2.10 Pervasive Accessibility

Stand-alone applications have the problem that they are only usable on a local computer. This restricts their usability. A requirement is therefore that *BibIS* is a *web-based* database application and thus accessible world-wide. This requires that you carefully think about a suitable web design and an appropriate distribution of user functions to the single web pages.

2.11 Bulk Loader Facility

Frequently, a user has already created a large collection of references on her own in the BibTeX format. The system should then enable their import by a *bulk loader*. The BibTeX references must be correct with respect to the mandatory attributes and make use of the bibtags for uniquely characterizing a document as well as the acronyms for consistently denoting a conference name or journal title. Reference files must be correctly named (see Section 2.7). A clear strategy has to be developed if, for example, publication types or attributes in external BibTeX files have not been specified in *BibIS*, or if the external BibTeX file contains syntactical errors.

2.12 Extensibility of *BibIS*

BibIS should be highly extensible. Extensibility refers to the addition, deletion, and modification of publication types, users, publications, reference files, projects, thematic categories, etc. by the system administrator and partially by the user. In general, all data in the system must be updatable.

3 Related Work

It may be worthwhile to have a look at some existing systems for our problem domain. Here are a number of useful web sites:

- *http://wiki.services.openoffice.org/wiki/Bibliographic_Software_and_Standards_Information*
- *http://www.biblioscape.com/biblioexpress.htm*
- *http://www.biblioscape.com/biblioweb.htm*
- *http://www.refworks.com/*
- *http://www.pycs.net/users/0000177/categories/biblios/*
- *http://kimura.univ-montp2.fr/ jdutheil/B3/B3.html*

Especially the first URL gives a good overview of existing software packages. Most of them are standalone systems. Only very few of them are web-based.

4 Action Items and Project Deliverables

Project deliverables are supposed to be well structured and well written documents. High readability and comprehensibility is absolutely required. Together, the deliverables constitute 20% of the project grade. There are currently two project deliverables. Each deliverable should have a cover page containing information about the class, the group number, the group members, and the deliverable number. Due dates for deliverables are announced in class, by email, and/or on the class web page.

The first project deliverable includes the following parts:

- *Overall design of BibIS and its functionality*. Based on the requirements sketched in this specification and based on your own ideas, this part describes your overall design of the *BibIS* system. You should delineate your concepts for satisfying the requirements and the functionality you plan to offer. First, you should give an overview of your ideas to design BibIS. Then, for each subsection in Section 2, you have to explain in detail how you plan to satisfy the requirements. Note that the emphasis here is on your *conceptual design*.

- *User interface considerations*. An important question is how the promised functionality is offered at the user interface. You should give a clear description about the flows of action and web pages the user can expect. Hand-drawn screen shots with detailed explanations could be a way to perform this task. Any kind of programming or HTML web page construction is unnecessary. Note that the emphasis is *not* on producing a high gloss pamphlet with nice pictures.

The second project deliverable includes the following parts:

- *Design of an ER diagram*. You should identify the important entity sets, relationship sets (cardinalities!), and attributes that are relevant and have later to be stored in the database. In a separate part of the document, you should explain the important concepts of your design. Note that information (for example, about publication types, attributes, categories) is created during a *BibIS* session and thus unknown now.

- *Transformation of the ER diagram into a database schema*. Derive the database schema according to the rules learnt in class. Give explanations if they are needed.

The only restrictions for your later implementation are that first they must use the Oracle DBMS (versions 9i or 10g) and that SQL commands must be explicitly used, that is, embedded, into your program code. Tools that hide the database programming are *not* allowed. Finally, note that it is your risk to use resources that are not provided by CISE. If there should be a failure of your software solution in your assigned demonstration time slot at the end of the semester, your project demonstration will be graded with 0%. There is no possibility to perform your demonstration at another time then.