SGML Bibliographies

Tom Gordon

July 5, 1993

This chapter describes an SGML document type definition for bibliographies. The logical structure of these bibliographies is the same as in BibTeX. There is a Unix tool, described in the "Unix Commands" chapter, for translating bibliographies in this SGML format into BibTeX's `bib` format or, if you prefer, into Unix's `refer` format.

Why use this SGML format instead of the `bib` format directly? There may be a few reasons. If you are using SGML to write your articles, then it is probably easier to use this SGML document type for your bibliographies. The syntax of BibTeX's `bib` files is, not surprisingly, completely different. Also, here too SGML offers the possibility of easily translating these bibliographies into some other format, such as Unix's `refer` format. Finally, some may prefer to maintain their bibliographies in a semi-WYSIWYG way, using an SGML structure editor.

This chapter is divided into an overview followed by a reference section. The overview contains a simplified description of the structure of bibliographies; it is correct as far as it goes, but certain details are not discussed. The reference section contains the complete SGML Document Type Definition for bibliographies and should be consulted for further information or clarification.

1 Overview

First, here is an example of a (very short) bibliography in the format of the `biblio` document type.

```
<!doctype biblio public "-//GMD//DTD biblio//EN">
<biblio>

<inproc id="gordon:etal:92:acmml">
<author> T.F. Gordon <and> J. Hertzberg <and> A. Horz
<title> Experiences with ML for Building an AI Planning Toolbox
<btitle> Proc. ACM Workshop on ML and its Applications
<year> 1992
<note> To appear
<annote> The <tt/qwertz/ toolbox is a system of ML modules for
implementing AI planners.  We discuss why ML was chosen to implement
the system, provide a few guidelines for using the language, based on
our experience, and make some suggestions for language extensions and
programming environments.
<keywrds> qwertz AC

<report id="guesgen:hertzberg:92:AP608">
<author> H.W. G&ue;sgen <and> J. Hertzberg (eds.)
<title> On Space and Time
<inst> GMD
<type> Arbeitspapier
```

<no> 608
<year> 1992
<annote> The report is a collection of papers about spatial and
temporal reasoning, stemming from members of GMD's AI division. Some
of them are in German, and some of them have been published previously.
<keywrds> qwertz AC

<report id="guesgen:hertzberg:91:tasso31">
<author> H.W. G&ue;sgen <and> J. Hertzberg
<title> A Constraint Based Approach to Spatiotemporal Reasoning
<inst> BMFT-Verbundprojekt TASSO
<type> TASSO-Report
<no> 31
<year> 1991
<annote> In this paper, we introduce a form of spatiotemporal
reasoning that uses homogeneous representations of time and the three
dimensions of space. The basis of our approach is Allen's temporal
logic on the one hand and general constraint satisfaction algorithms
on the other, where we introduce a new view on constraint reasoning to
cope with the affordances of spatiotemporal reasoning as introduced
here. As a realization for constraint reasoning, we suggest a
massively parallel implementation in form of Boltzmann machines.
<keywrds> qwertz TASSO AC

<incoll id="hertzberg:92:racProbleme">
<author> J. Hertzberg
<title> &Ue;ber Verzweigungen, Kulissen und das Yaler Schie&sz;en --
        Probleme beim reasoning about change
<editor> H.W. G&ue;sgen <and> J. Hertzberg
<btitle> On Space and Time
<publish> Arbeitspapiere der GMD No. 608
<year> 1992
<pp> 143-158
<annote> In diesem Aufsatz stelle ich die klassischen Probleme
zusammen, die auftreten, wenn man &ue;ber die Effekte von Ereignissen logisch
schlie&sz;t, und ich beschreibe, wie sie zusammenh&ae;ngen. Im einzelnen
behandle ich die folgenden Probleme: qualification problem, prediction
problem, persistence problem, frame problem und ramification
problem. Zus&ae;tzlich beschreibe ich kurz das Yale shooting
problem, das allerdings auf einer anderen Ebene liegt als die
vorgenannten.
<keywrds> qwertz TASSO AC

<report id="rutten:91">
<author> E. Rutten
<title> A Temporal Non-linear Planner: TRIPTIC
<inst> GMD
<type> Arbeitspapier
<no> 582
<year> 1991
<annote> The problem addressed in this report is that of the coupling
of a non-linear planning algorithm with a time map manager. We give a
description of a solution integrating a planner on top of a time map

```
manager, determining a particular share of the tasks between the two
levels.
<keywrds> qwertz TASSO

</biblio>
```

Supposing this bibliography is in a file named “mybib.sgml”, you can check that it is syntactically correct by following this simple procedure:

1. Make sure the `/home/qwertz/format/bin` directory is included in your Unix `PATH` variable.
2. execute this command:

   ```
   $ biblio -c mybib.sgml
   ```

That’s it! The `biblio` command will write easy to understand error messages to your terminal, if there are any.

Once you’ve corrected your errors, the bibliography can be converted into BibTeX, if you’d like, with this command:

```
$ biblio mybib.sgml > mybib.bib
```

The first entry of our sample bibliograhy starts with the <inproc id="gordon:etal:92:acmml"> *tag*. It is used to state the type of the entry, here an article in the proceedings of a conference, and give it a unique identifier.[1] There are many other types of entries, in addition to articles, such as books, reports and Ph.D. theses. A complete list will follow shortly.

This entry has seven *fields*: author, title, journal, volume, number, pages, and year. (The names of some of these fields are abbreviated in the entry.) The fields required for an entry depends on the type of the entry. In the case of articles, the `author`, `title`, `journal` and `year` fields must be included.[2] The `vol`, `no`, and `pp` fields are optional.

Within each field, just about every ASCII character can be used. There are only a few exceptions to be aware of. The most important in practice is &, the ampersand sign. Use the `amp` entity to obtain this character. Foreign language — that is, non-English — characters must be entered using SGML entities. For example, German characters can be entered as follows: ä (`ae`), Ä(`Ae`), ö (`Oe`), Ö (`Oe`), ü (`ue`), Ü (`Ue`), and ß (`sz`). There are also entities for several general purpose symbols, such as £, ¶ and §. See the *General Purpose Characters* table in Chapter 4 for a complete list. The table also include entity names for several ASCII symbols. Use these entities if some ASCII is not being printed as expected. Mathematical formulas cannot be used within bibliographies.

Multiple authors or editors are separated by `and` elements, as in this example:

```
<book id="Smith88">
<author>  Joan M. Smith <and> Robert Stutely
<title> SGML: the user's guide to ISO 8879
<publish> Ellis Horwood
<year> 1988
```

Altogether, there are almost 30 different fields available. Which of these are required or optional depends on the entry type. A book, for example, requires different fields than a Ph.D. thesis. An alphabetical list describing all of the fields follows. After this list, the role these fields play in each of the entry types will be explained. First, here are the fields:

[1] Conveniently, the SGML parser will complain if the identifier is not unique.
[2] The order of fields is no longer fixed; they may appear in any order, so long as each required field is present.

`address`

The address of the publisher. For large publishers, just provide a city. For smaller publishers, type the full address.

`annote`

An annotation. However, it is not included in the list of references, using any of the standard styles.

`author`

The name of the author or authors. Each author's name can be typed in full without using tags to separate the first and last names. However, sometimes BibTeX needs help parsing names correctly, in which case you can use the `first` and `last` tags to unambiguously delimit the parts of the name. If there are mulitple authors, use the `and` tag to separate them, as in the previous example.

`btitle`

The book title, when the entry is for a part of the book, such as a chapter. If the entry is for the book itself, its title is given using the `title` element.

`chap`

The *number* of the chapter in a book.

`edition`

The edition of the publication, such as "second".

`editor`

The editor (or editors) of the publication. See the description for the `author` field above for information about how to enter names.

`howpub`

To describe how the document was published, if not by a publisher or some institution.

`id`

The library identification number of the book, for *your* library. This will not be included in entry as it is printed in the list of references, at least not when using one of the standard styles.

`inst`

The *institution* which published the item, such as a university or research institute, if not a commercial publisher.

`journal`

The name of the journal or magazine in which an article was published.

`key`

Used for alphabetizing and creating a label when there is no `author` or `editor`.

`keywrds`

A list of *key words* for categorizing the item. These words are not included in the list of references when using one of the standard styles.

`loc`

The current *location* of the document. Useful for keeping track of who has borrowed your books! Not printed in the list of references when using one of the standard styles.

`month`

The month the work was published or, if unpublished, written.

`note`

Additional information which may be useful to the reader. Shorter than an annotation, using `annote`.

`no`

The *number* of a journal, magazine or technical report. An issue of a journal is usually identified by both its *volume* (using the `vol` element described below) and number.

`organ`

The *organization* sponsoring a conference. Used in the entries for proceedings.

`pp`

One or more page numbers or range of page numbers, such as

```
<pp> 34,58,62
```

or

```
<pp> 56-108
```

Notice that it is not necessary here to use the `ndash` entity between the page numbers. An ordinary hyphen will be interpreted here by BibTeX as a dash for number ranges.

`private`

For reminders or comments to yourself, that you do not want to appear in an annoted bibliography (using `annote`).

`publish`

The name of the publisher.

`school`

The name of the university where a thesis or disseration was written.

`series`

The name of a series or set of books.

`title`

The item's title.

`type`

The type of a technical report; for example "Research Note".

`vol`

The volume of a journal or multivolume book.

`year`

The year the entry was published, or if it was never published, written. This should consist only of numbers, such as "1984".

Including articles and books, there are thirteen types of bibliographic entries. Each type is described below, including a list of required and optional fields. The order in which the fields are listed is no longer significant. A question mark behind the field name means the field is optional. Alternative fields are connected by a `|` symbol, as in `(author | editor)`. Finally, every entry, regardless of its type, can include any of these optional fields: `note?, annote?, keywrds?, loc?, id?, private?`.

`article`

For an article from a book, magazine, or journal. Use `inproc` for article in conference proceedings, and `incoll` for a chapter or part of a book with its own title. The fields of an article are: `author, title, journal, vol?, no?, pp?, month?, year`

`book`

For published books, with a known publisher. If the book has yet to be publised, or the publisher is unknown, use an `unpub` or `booklet` element. The fields of a book entry are: `(author | editor), title, edition?, publish, series?, vol?, address?, month?, year`

`booklet`

For documents without a named publisher or sponsoring institution. If the `author` is not known, a `key` is required so that the entry can be ordered alphabetically in the list of references. Its fields are: `(author | key), title, howpub?, address?, month?, year?`

`inbook`

A part of a book, either a chapter or some range of pages. The fields are: `(author | editor), title, ((chap, pp?) | pp), edition?, publish, series?, vol?, address?, month?, year`

`incoll`

A part of a book with its own title. Its fields are: `author, title, editor?, btitle, chap?, edition?, publish, address?, month?, year, pp?`

`inproc`

An article in the proceedings of a conference. An `inproc` entry consists of: `author, title, editor?, btitle, organ?, publish?, address?, month?, year, pp?`

`manual`

For user manuals and other technical documentation. Its fields are: `(author | key), title, edition?, address?, month?, year?, organ?`

`masters`

For a master thesis. The fields are: `author, title, school, address?, month?, year`

`misc`

For documents which don't seem to be one of the other entry types: `(author | key), title?, month?, year?, howpub?`

`phd`

For a Ph.D. thesis. The fields are the same as for `masters` entries: `author, title, school, address?, month?, year`

`proc`

The proceedings of a conference: `(editor | key), title, organ?, publish?, address?, month?, year` `report`

For technical reports published by a university or research institute. These are usually numbered within a series. Its fields are: `author, title?, inst, address?, type?, no?, month?, year`

`unpub`

Unpublished documents: `author, title, month?, year`