Biblet: A portable BibTeX bibliography style for generating highly customizable XHTML

Tristan Miller
German Research Center for Artificial Intelligence (DFKI GmbH)
Postfach 20 80
67608 Kaiserslautern, Germany
Tristan.Miller@dfki.de
http://www.dfki.uni-kl.de/~miller/

Abstract

We present Biblet, a set of BibTeX bibliography styles (`bst`) which generate XHTML from BibTeX databases. Unlike other BibTeX to XML/HTML converters, Biblet is written entirely in the native BibTeX style language and therefore works "out of the box" on any system that runs BibTeX. Features include automatic conversion of LaTeX symbols to HTML or Unicode entities; customizable graphical hyperlinks to PostScript, PDF, DVI, LaTeX, and HTML resources; support for nonstandard but common fields such as `day`, `isbn`, and `abstract`; hideable text blocks; and output of the original BibTeX entry for sharing citations. Biblet's highly structured XHTML output means that bibliography appearance to can be drastically altered simply by specifying a Cascading Style Sheet (CSS), or easily postprocessed with third-party XML, HTML, or text processing tools.

We compare and contrast Biblet to other common converters, describe basic usage of Biblet, give examples of how to produce custom-formatted bibliographies, and provide a basic overview of the implementation details for those wishing to modify the style files.

Introduction

In today's world of ubiquitous Internet access, it is becoming increasingly expected that every researcher, graduate student, professor, and other academic have a personal web page listing one's contact information, qualifications, teaching schedule, ongoing and completed research projects, and publications. Normally such pages are maintained by the academic himself, and thanks to the extensive formatting capabilities of HTML [65], XHTML [81], and CSS [8, 47], authors can easily give their home pages a unique personal style.

Despite these tools, creating and maintaining an online list of publications has traditionally been a troublesome process. Authors must manually enter their bibliography data using the appropriate HTML[1] and CSS markup to ensure that the list's formatting matches the rest of the website. Since many authors already maintain a database of their publications in a format like BibTeX [57], this approach entails maintaining two separate bibliographies which can easily get out of sync.

Furthermore, if the author at some point decides to change the style in which the bibliography is displayed, CSS can help only so much. By altering the list's style sheet, one can change the style of book titles from italicized to bold, or suppress the display of abstracts and annotations. However, CSS cannot make changes such as abbreviating author or journal names, switching the order of volume and issue numbers, or changing the sort order of publications from author to year. To make such changes, the author must tediously edit the individual list entries in the HTML file.

One solution to these problems is to use some tool to automatically convert the author's existing BibTeX database to HTML, possibly employing some intermediate format such as LaTeX [44] or XML [9]. Then only one publication database need be maintained; the author can rerun the conversion whenever the BibTeX database is updated or whenever he wishes to effect a fundamental change in formatting, such as sort order.

In this paper, we present Biblet, one such tool for converting BibTeX databases to HTML web pages. We compare and contrast its features and

[1] Hereinafter, unless otherwise noted, we use the term 'HTML' to refer to HTML and XHTML collectively.

capabilities to those of similar software, and discuss its limitations and the limitations of the underlying BibTeX format.

Background

Using BibTeX BibTeX [56–58] is a bibliography program originally designed to work with Leslie Lamport's LaTeX [44] document preparation system. To use it, the author creates a database of publications he wants to reference in a BibTeX database file with the filename extension `.bib`. The contents of this file are a series of records in a format similar to the following:

```
@ARTICLE{m05toc,
  author  = {Tristan Miller},
  title   = {The Tyranny of Copyright},
  journal = {Imagine},
  year    = {2005},
  month   = may,
  volume  = {4},
  number  = {1},
  pages   = {1,8-11},
  issn    = {1710-5994},
}
```

The `@ARTICLE` token specifies the type of publication, which in turn determines the available fields. Other common publication types include `@MANUAL`, `@PROCEEDINGS`, and `@BOOK`. The text `m05toc` is the database key used to refer to the publication in LaTeX citation commands and elsewhere in the `bib` database. The remainder of the record consists of a comma-delimited list of field-value pairs, where the values can be either predefined macros (such as `may` in the above example) or strings delimited with curly braces or quotation marks.

To use a BibTeX database called `imagine.bib` with a LaTeX document `foo.tex`, the author must write the following commands at the place in the document where the bibliography[2] is to appear:

```
\bibliographystyle{modern}
\bibliography{imagine}
```

The first command tells the system how to format the bibliography and the second command tells it which database to use. To cite documents from the database, the LaTeX `\cite` command is used:

```
... in my article~\cite{m05toc} ...
```

Note that the argument to `\cite` is the database key of the publication to be referenced. LaTeX might typeset the above example as follows:

... in my article [1] ...

[2] In this article we use the terms "bibliography" and "reference list" interchangeably.

At the end of the document in a separate section, the full bibliographic details of every publication cited appear, possibly as follows:

[1] Tristan Miller. The tyranny of copyright. *Imagine*, 4(1):1,8–11, May 2005. ISSN 1710-5994.

The exact formatting of this reference list depends on the argument to `\bibliographystyle`.

In order to properly typeset the citations and references, the author must invoke the LaTeX and BibTeX programs a number of times. Exactly what goes on behind the scenes is illustrated in Figure 1. (In this diagram, files furnished by the user are indicated by light rounded boxes, while computer-generated files are indicated by dark rounded boxes.) First, LaTeX is run on the LaTeX document `foo.tex`, which produces an incomplete typeset version of the document, `foo.dvi`, and an auxiliary data file, `foo.aux`. This auxiliary file contains information for use by BibTeX — namely, the bibliography style, the bibliography database filename, and which publications from said database to include in the reference list. The contents of the `aux` file in our example might look as follows:

```
\relax
\citation{m05toc}
\bibstyle{modern}
\bibdata{imagine}
```

Next, BibTeX is invoked on `foo.aux`. Seeing the `\bibstyle` and `\bibdate` commands, BibTeX searches for and opens the files `imagine.bib` and `modern.bst`. The `bst` file is actually a program which specifies how to convert a BibTeX bibliography — in this case, `imagine.bib` — into LaTeX code. BibTeX scans `imagine.bib` until it encounters the `m05toc` entry, applies to it the transformation rules specified in `modern.bst`, and writes the output in a new file named `foo.bbl`. This `bbl` file contains LaTeX code which, depending on the bibliography style, may contain something like the following:

```
\begin{thebibliography}{1}
\bibitem{m05toc}
Tristan Miller.
\newblock The tyranny of copyright.
\newblock {\em Imagine}, 4(1):1,8--11,
  May 2005.
\newblock ISSN 1710-5994.
\end{thebibliography}
```

When LaTeX is next run on `foo.tex`, it inserts the contents of `foo.bbl` at the exact position where the `\bibliography` command occurs, producing a new version of `foo.dvi` which includes the list of references. Typically, LaTeX must be run once more to resolve references, the `\bibitem` and `\cite` commands

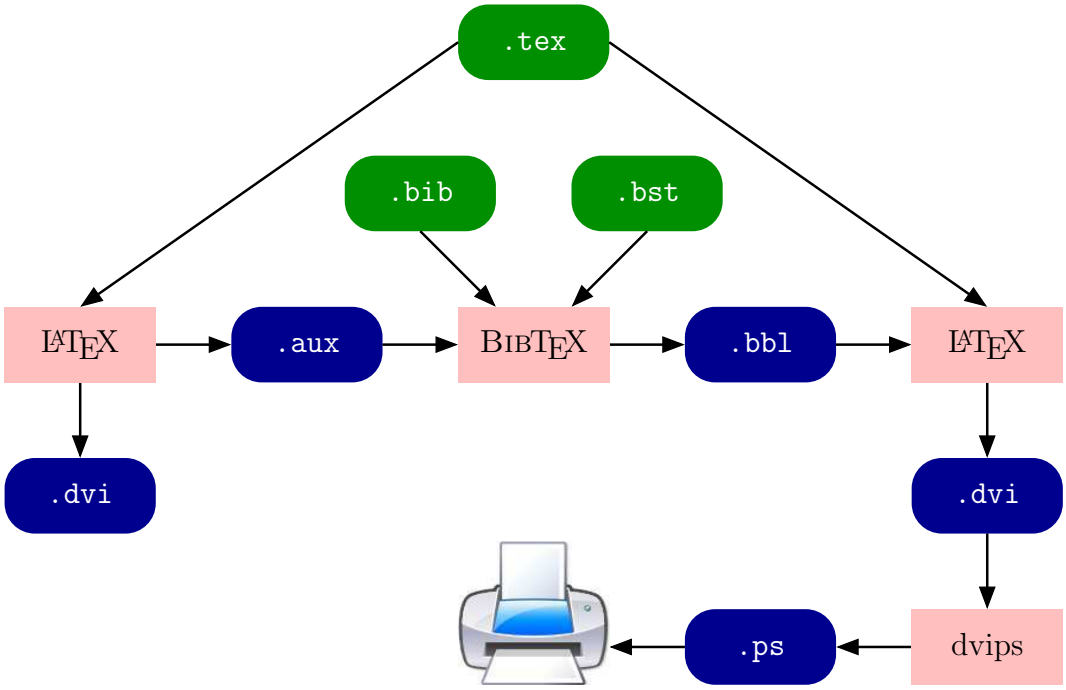

Figure 1: BibTeX workflow

being analogous to the `\label` and `\ref` commands used to create other types of cross-references [44, §4.2]. The resulting DVI file is the final typeset version of the document; it can then be converted to a `ps` file for printing on a PostScript printer, or to a PDF for distribution on the Internet.

Previous work Because BibTeX outputs LaTeX command sequences, authors wishing to create an online list of publications typically have three options. The simplest but least convenient to the casual web page visitor is to simply post the `bib` database itself on one's website, either by linking to the file directly or by embedding it as preformatted text in an HTML page. The former case can cause problems for some web browsers which correctly recognize the database as a file with the MIME media type `text/x-bibtex` [20, 21] but do not know how to display it for the user. The second case is guaranteed to allow users to view the database from within their web browsers, albeit only in the crude original format.

The second option is to create a skeleton LaTeX document citing the desired publications, run a LaTeX-to-HTML converter on it, and then extract the resulting bibliography for use on another web page. Programs implementing this approach include Nikos Drakos and Ross Moore's LaTeX2HTML [16, 24, 26], Eitan Gurari's TeX4ht [25, 31, 59], Luc Maranget's HeVeA [50, 51, 77], and over a dozen lesser-known applications [2, 7, 10–12, 22, 28, 36, 53–55, 62, 69–73, 79, 80, 86]. These programs have the advantage that any existing BibTeX bibliography style can be used.

The third option is to use a program which directly converts the BibTeX database to HTML (or to XML, which is easy to convert to HTML). While there are several such utilities available [4, 17, 23, 27, 29, 32, 33, 35, 42, 43, 48, 60, 61, 66, 68, 74, 75, 82, 85], none of them seems to be as well-known or widely used as their LaTeX-to-HTML counterparts. Indeed, most of them have no presence on CTAN and some of them even share the same name.

There are a number of criteria to consider when evaluating software implementing these latter two approaches:

License Many people expect their software to be free in the sense that they may freely modify and redistribute the program for any purpose [76]. The ability to modify the program necessarily implies that human-readable source code is provided. Most of the software cited above is available under a permissive license such as the GNU General Public License [19] or the LaTeX Project Public License (LPPL) [45], though some packages impose restrictions on commercial use or redistribution, and others have restrictive proprietary licenses and do not include source code.

Portability Part of BibTeX's popularity springs from its availability on a wide variety of computing platforms. To be useful for BibTeX users as a whole, any conversion tool should be available for a large subset of these platforms and should be installable with minimal effort. Most existing converters are implemented in widely available scripting languages such as Awk [67], Perl [83], or Python [49], or in portable compiled languages such as C [39]. Some, like bibtex2html [17], are implemented in relatively esoteric languages for which compilers are not widely available.

Standards compliance In order to be displayed and indexed properly and consistently by all web browsers and other Internet applications, the HTML output by a converter should conform to the official W3C hypertext standards [5, 63–65, 81]. The HTML produced by some converters is not syntactically valid, or conforms to an obsolete standard.

Symbols BibTeX bibliographies often use some LaTeX markup for special characters such as accented letters and typographical symbols. A good converter should transform these into their Unicode/ISO 10646 [41, 78] or HTML [65, §24] equivalents. Unfortunately, even the most common and actively maintained converters often fail spectacularly in this regard, particularly with regard to basic punctuation such as quotation marks and dashes.

Math Handling embedded math mode is a problem, since HTML and Unicode alone are not sufficient to display most mathematical constructs. Some converters transform LaTeX math markup into MathML [3], though the latter is not yet widely supported by web browsers. Most (perhaps forgivably) ignore this problem and simply output the original LaTeX code, though others resort to questionable solutions such as producing bitmap images (which do not scale with the surrounding text) or using deprecated, nonportable hacks [18].

Custom LaTeX macros Some bibliographies use custom LaTeX macros, defined in the BibTeX `@PREAMBLE` or in a separate LaTeX document. The wisdom of employing such macros is questionable, though in some cases they are the only way to get around BibTeX's inherent limitations.[3] Unfortunately, LaTeX is notoriously difficult to parse by anything except LaTeX itself, so most converters offer no or partial solutions to this problem.

Hyperlinks The principal advantage of HTML is that it allows the inclusion of links to other documents. In the case of online bibliographies, it would be useful if each entry included a link to the document itself (for example, as a PostScript or PDF file) where available. Some converters cooperate with packages such as `url` and `hyperref`, or with bibliography styles such as `natbib` which support `url`, `ps`, and `pdf` fields. Others simply have no support for hyperlinks.

Anchors HTML also allows links to certain points, called *anchors*, within documents. Since users may wish to link to individual entries in their publication list from other web pages, it would be useful if the converter would associate a unique anchor with each bibliography entry. Few existing converters implement this feature.

Styling Perhaps most importantly, the converter's output should be adjustable by the user. The converter should allow at least as much variation in reference formatting as BibTeX itself, but should ideally output its HTML in such a way that the appearance can be further customized via CSS. In this manner the online bibliography can be made an integral part of the author's web page rather than a generic-looking computer-generated list.

Introducing Biblet

Though the existing converters we examined typically excelled in some of the above-mentioned criteria, none of them provided good support across the board. This lack of a combination of good features in existing software was the primary impetus for the development of a new tool, Biblet.[4] Biblet sports the following features:

- It is freely available and redistributable under the terms of the LPPL.
- It outputs valid XHTML 1.0, making its pages viewable by any conforming web browser and facilitating postprocessing by XML applications.
- It makes use of an extensive LaTeX-to-Unicode translation table, ensuring proper display of most typographical symbols.
- It has extensive support for internal anchors and external hyperlinks.
- Virtually every element of a bibliography entry is encapsulated in its own named HTML tag, allowing for extensive styling with CSS.

[3] An example of this is the oft-used `\noopsort` kludge to force proper sorting of non-English names [57, pp. 4–5].

[4] According to the Oxford English Dictionary, "biblet" is an archaic word of uncertain origin and meaning, though it is thought to refer to a small library.

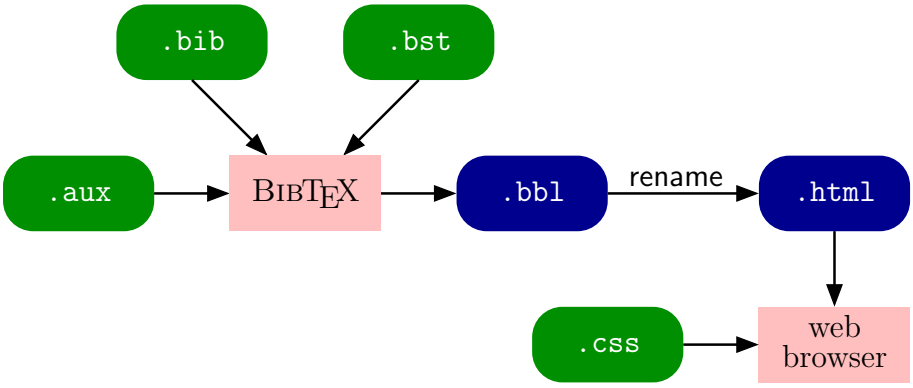

Figure 2: Biblet workflow

```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN'
  'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en' lang='en'>
  <head><title>My publications</title></head>
  <body>
    <h1>My publications</h1>
    <div class='bib-bibliography'>
      <h2 class='bib-year' id='year-2005'>2005</h2>
        <ul>
          <li class='bib-bibitem' id='cite-m05toc'>
            <div class='bib-article'>
              <p>
                <span class='bib-author'>Tristan Miller.</span>
                <span class='bib-title'>The tyranny of copyright.</span>
                <em>Imagine</em>, 4(1):1,8&ndash;11, May 2005.
                ISSN 1710-5994.
              </p>
        ...
</html>
```

Listing 1: Sample output of Biblet (abridged)

Significantly, Biblet is written entirely in the BibTeX stack language [56], making it portable to any system that can run BibTeX itself. It is, in effect, simply another bibliography style (`bst`) file, just like the standard `plain`, `abbrv`, and `alpha` styles. The difference is that while the latter output LaTeX code, the Biblet produces HTML.

The basic Biblet workflow is illustrated in Figure 2. Note that, unlike in the regular BibTeX workflow of Figure 1, there is no `tex` file as input; rather, the user creates the `aux` file directly. In this file, the user issues a `\citation{`⟨*key*⟩`}` command for each bibliography item, or `\citation{*}` to include all items. This is followed by a `\bibstyle` command indicating which Biblet `bst` style to use and a `\bibdata` command indicating which `bib` database to process. A sample `aux` file for use with Biblet might look as follows:

```
\citation{m05toc}
\bibstyle{blplain}
\bibdata{imagine}
```

The user runs BibTeX on this `aux` file as usual. The resulting `bbl` file, however, is actually an HTML document. It can be renamed and opened in any web browser or HTML editor.

Listing 1 shows the abridged contents of the `bbl` file produced in the above example, and Figure 3 shows how this file appears when opened in a web browser.

Sprucing things up The reader will note from Listing 1 that Biblet has enclosed most of the important parts of the bibliography in their own HTML

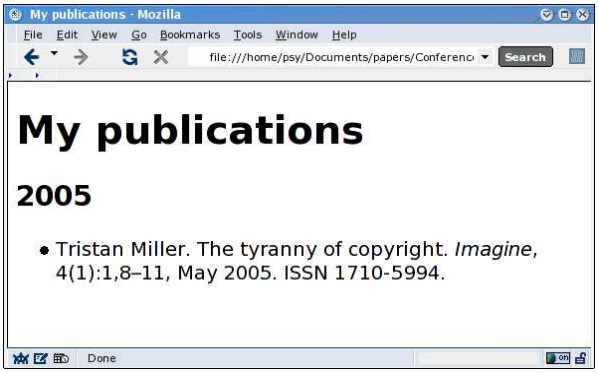

Figure 3: Biblet output as viewed by the Mozilla web browser

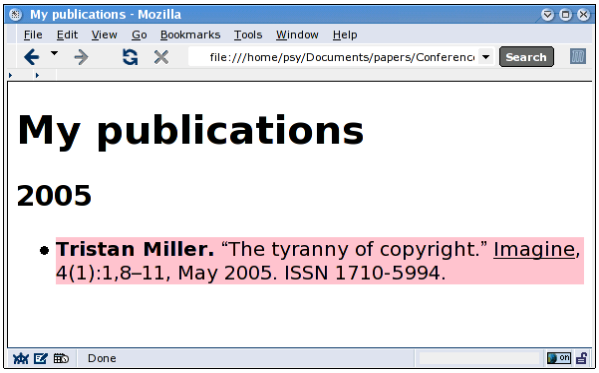

Figure 4: Biblet output with CSS styling

elements. This makes it easy to alter the appearance of the list by applying CSS styles. For example, say we wish all entries of type `@ARTICLE` to be displayed with a pink background, author names to be printed in bold type, journal names to be underlined instead of italicized, and article titles to be enclosed in quotation marks. Rather than manually editing the HTML file produced by Biblet, or even the `bst` file which generates the HTML, we simply write a file `mystyle.css` as follows:

```
.bib-article {
  background-color: pink;
}
.bib-author {
  font-weight: bold;
}
.bib-article em {
  font-style: normal;
  text-decoration: underline;
}
.bib-title:before {
  content: "\201C";
}
.bib-title:after {
  content: "\201D";
}
```

(In this example, 201C and 201D are the hexadecimal values for left and right double quotation marks in Unicode.) To apply this style to our HTML file, we insert the following line into the `<head>` element:

```
<link href='mystyle.css' type='text/css'
      rel='stylesheet' />
```

When the browser view is refreshed, the new formatting styles take effect, as shown in Figure 4.

Because Biblet is aimed at producing online document catalogues, it recognizes a number of special fields in `bib` databases. Among these are `ps`, `pdf`, `dvi`, `html`, `tex`, and `txt`, whose values contain URLs [6] referencing respectively PostScript, PDF, DVI, HTML, (LA)TEX, and plain-text versions of the publication. Biblet converts these fields to textual or graphical hyperlinks within the bibliography entry. Other common fields such as `abstract`, `isbn`, and `issn` are also supported.

It is also possible for a Biblet `bst` style to output arbitrary HTML at the beginning and end of a bibliography, as well as within and in between entries. In fact, the default `bst` files distributed with Biblet produce far more extensive HTML than is actually shown in Listing 1. For example, styles which provide icon hyperlinks to online versions of the document will typically include a legend explaining the purpose of each icon. Other styles allow the option of outputting the original `bib` entry for researchers to copy and paste into their own BIBTEX bibliographies, plus JavaScript [37] code allowing the user to toggle the display of the BIBTEX entry and/or abstracts.

Biblet comes with several predefined CSS styles and icon sets for use with the publication lists it generates. Figure 5 gives a sampling of the available styles; note also the formatting of legends, abstracts, hyperlinks, original BIBTEX data, and toggles.

Implementation

As mentioned previously, Biblet is implemented as a BIBTEX style (`bst`) in the nameless BIBTEX stack language. The idea for this approach came through the observation that the `bst` styles were solely responsible for the producing the content of the `bbl` files; that is, the BIBTEX program itself did not write anything to these files unless directly instructed to do so by the `bst` style. The initial steps in the development of Biblet therefore involved going through Patashnik's original `plain.bst` style and replacing all the LATEX markup it outputted with analogous HTML markup. Thus, `{\em ...}` was changed to `<em>...</em>`, the `thebibliography` environment

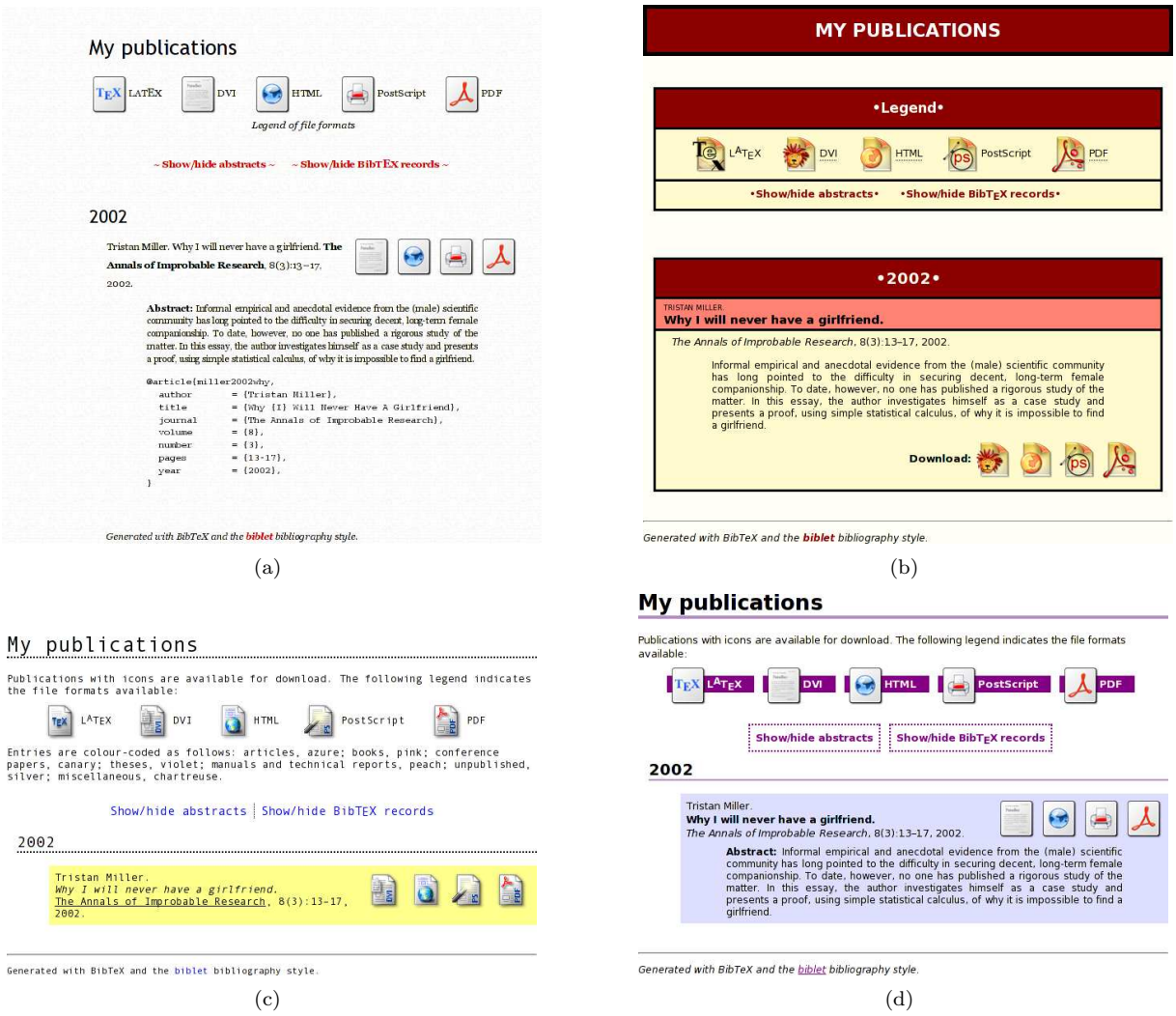

Figure 5: A gallery of four Biblet bibliographies: (a) uses the "Traditional" style with the "Nuvola" icon set; (b) uses the "Boxy" style with the "Noia Warm" icon set; (c) uses the "Fruity Typewriter" style with the "Slick" icon set; and (d) uses the "Amethyst" style with the "Nuvola" icon set. All examples were typeset with the `blplain.bst` BibTeX style; only the CSS styles and icon sets were altered.

to `<ul>...</ul>` tags, and so on.

These simple substitutions resulted in a `bbl` file containing the bibliography as an HTML list (`<ul>`) which, while not a complete HTML document itself, could be cut and pasted into an existing web page. To make the `bbl` file stand on its own as a web page, it was necessary to modify the Biblet `bst` so it outputted some additional HTML markup at the beginning and end of the file. We also added some more HTML code to the bibliography entries themselves, wrapping parts of them in various named containers so that the user could later customize their appearance with CSS.

The next step was to add some custom sorting routines typical of those seen on hand-crafted author publication lists. Most academics sort their publications by year or by publication type (book chapter, article in journal, article in conference proceedings, *etc.*). Besides the actual sorting code, it was necessary to output HTML headers marking a change in the value of a sort key.

Finally, we had to write the code to convert any LaTeX symbols contained in the BibTeX bibliography itself to Unicode or HTML entities. This was the most difficult and time-consuming of all the development tasks. The `bst` language is extremely crude, having been designed principally for ease of implementation on the computers of 1988; features that programmers take for granted in modern general-purpose programming languages, such

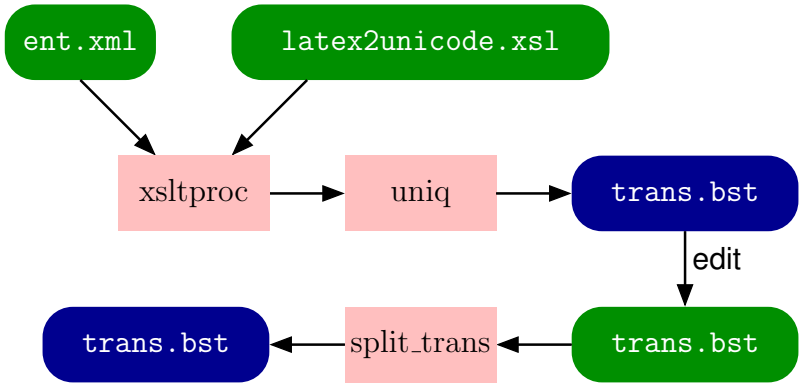

Figure 6: XML to bst

```
<char pos="127">
  <entity name="para" set="iso-8879-num">
    <desc>=pilcrow (paragraph sign)</desc>
  </entity>
  <entity name="para" set="html4-lat1">
    <desc>pilcrow sign = paragraph sign</desc>
  </entity>
  <unicode value="00B6">
    <desc>PILCROW SIGN</desc>
  </unicode>
  <latex>
    <seq>\P</seq>
    <seq req="textcomp">\textparagraph</seq>
    <seq req="textcomp">\textpilcrow</seq>
  </latex>
  <plain value="B6" set="iso-8859-1" glyph="¶"/>
</char>
```

Figure 7: Sample entry from `ent.xml`

as function arguments, arrays, local variables, string manipulation, and dynamic memory allocation are poorly supported or even completely absent. The features of Biblet previously discussed were accomplished with relative ease as they made use of the `bst` language's built-in output and sorting routines. For the symbol conversion task, however, what appeared to be a simple search-and-replace routine ended up being a programmer's nightmare.

For one thing, the language's string operations are limited to concatenating two strings, returning the first n characters of a string, and returning the string length, though the latter treats strings containing special characters idiosyncratically and cannot be used directly for our purposes. It was therefore necessary to code our own string-length (`string.length`) and find-replace (`find.replace`) routines.

Because the `bst` language also does not support arrays, it was not possible to simply enter a static translation table pairing LaTeX symbols with HTML entities, and then have some loop iterate over the table cells, calling `find.replace`. Instead, each symbol mapping had to be entered as a separate function call. This gave rise to another problem: function definitions are statically limited to 100 tokens. Since it takes three tokens to do a find-and-replace (one for the search string, one for the replacement text, and one for the call to `find.replace`, a maximum of 33 substitutions can be performed in one function. Since we needed to make over 500 symbol substitutions, we had to split the code over sixteen separate functions, and add a seventeenth function whose purpose was simply to call the others in sequence.

Rather than tediously coding all this by hand, we wrote a number of support programs to generate the code. Their operation is illustrated in Figure 6. For the LaTeX-to-HTML mappings we used Vidar Bronken Gundersen and Rune Mathisen's comprehensive database [30, 52] which they have kindly made available for any purpose. The database, which is distributed as an XML file named `ent.xml`, has entries for every SGML character [38, 40], giving data such as its name, Unicode value, and, where known, LaTeX macro(s). A sample entry from `ent.xml` is shown in Figure 7. We used an XSL transformation (XSLT) [13] to convert the information in this database to a sequence of `find.replace` calls in a new file, `trans.bst`. The XSLT will opt to convert symbols to named HTML entities when possible; otherwise it will output numbered hexadecimal entities. Here are a few lines from `trans.bst`:

```
"\textparagraph"  "&para;"  find.replace
"\textpilcrow"  "&para;"  find.replace
"\checkmark"  "&#x2713;"  find.replace
```

For a number of reasons, the `trans.bst` file output by the XSLT is not directly usable. First, as mentioned before, it needs to be split up into functions of no more than 33 lines each. Second, `ent.xml` is sometimes a little too pedantic for our

purposes, including some glyphs (*e. g.*, the 'fi' and 'fl' ligatures) which we would rather not convert to HTML entities. It is also missing some other glyphs and common LaTeX macros we would indeed like to convert — examples include the TeX logo (`\TeX`) and the breakable slash (`\slash`). To remedy these last two problems we must edit `trans.bst` by hand; the first can then be rectified by a short shell script, `trans_bst`, which wraps consecutive sets of 33 lines in their own `bst` function definitions. The output of this script can then be inserted into a Biblet `bst` style.

Unresolved issues Biblet's approach to building online publication lists is certainly entirely portable, though it does have its drawbacks, most of which stem from the limitations of BibTeX itself.

The first and most apparent problem is Biblet's execution speed — using the interpreted `bst` language to perform extensive string manipulation. With LaTeX-to-HTML symbol conversion enabled, running Biblet on a `bib` file with only a few dozen publications can take several minutes even on a reasonably fast (1.4 GHz) machine. By sacrificing portability, this problem could be solved by writing the symbol conversion routine in an interpreted text-processing language such as Sed [15] or Perl [83], or as a compiled lexical analyzer using Lex and C [39, 46].

A second problem — actually a class of related problems — is Biblet's extensibility. The root of this problem is that there is no way to pass to BibTeX any information besides the `bst` and `bib` files to use. Thus there is no way to specify the title of the HTML document produced by Biblet; likewise the user cannot tell Biblet to include a link back to his home page at the end of the bibliography. To effect such changes, the user must either edit the HTML output by Biblet, or edit the `bst` styles himself — either is a potentially daunting task for someone not familiar with HTML or the `bst` language.

The rigid syntax of `bib` files also poses a problem for Biblet's extensibility. While users and style developers are free to create new publication types and fields, some applications require an extra level of specification that BibTeX simply does not support. An example of such an application is found in Biblet's hyperlink fields. Our `pdf` field, for example, specifies the URL of a PDF file; Biblet might convert the value of this field into an HTML hyperlink as follows:

```
<a href="⟨url⟩"
   type="application/pdf"
   title="⟨title⟩">...</a>
```

The value of the `type` attribute, a MIME content type [20, 21], tells the browser what kind of file to expect when the user follows the link, in case such information is not provided by the server hosting the PDF file. Thus, when processing `html` and `ps` fields, Biblet substitutes the appropriate MIME type — in this case, `text/html` or `application/postscript`.

However, this mapping of fields to MIME types must be hard-coded in the Biblet `bst`. Should a user wish to provide a link to some other kind of file type — say, a Rich Text File (RTF) — he will have to edit the `bst` source. A better solution would be for BibTeX databases to support paramaterized fields so that users could specify unusual MIME types for document links in the `bib` file itself. For example, if a user wanted to provide links to an HTML, an RTF, and a sound recording version of an article, he could specify the URLs as follows:

```
@ARTICLE{m05toc,
  title = {The Tyranny of Copyright},
  ...
  url[type="text/html"] = {http://...},
  url[type="text/rtf"] = {http://...},
  url[type="audio/mp4"] = {http://...},
}
```

There are some extensions to and replacements for BibTeX which go some way towards solving these extensibility issues [14, 34, 84], though none of them are yet popular or stable enough to wholly supplant BibTeX. Patashnik himself has been planning to extend the "official" version of BibTeX to allow for better communication between BibTeX and its environment [58], so it is possible that Biblet's extensibility problems may one day be solved without having to compromise its portability.

Development status and availability

At the time of this writing, Biblet is under active development, and while the interface is not yet stable, the program is nevertheless very usable. A preliminary version of Biblet is available for download at the project's web page, `http://www.nothingisreal.com/biblet/`. Apart from this article, no formal documentation is yet available. By publication time a beta version of a complete Biblet package, including several `bst` and CSS styles and a user's guide, may be available on CTAN.

Bibliography

[1] Tristan Miller. The tyranny of copyright. *Imagine*, 4(1):1, 8–11, May 2005. ISSN 1710-5994.

[2] Romeo Anghelache. *Hermes — A semantic XML+MathML+Unicode E-publishing/Self-archiving Tool*

for LaTeX-authored Scientific Articles, July 2005. URL http://hermes.aei.mpg.de/.

[3] Ron Ausbrooks, Stephen Buswell, David Carlisle, Stéphane Dalmas, Stan Devitt, Angel Diaz, Max Froumentin, Roger Hunter, Patrick Ion, Michael Kohlhase, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical Markup Language (MathML) version 2.0 (second edition). W3C recommendation, W3C, October 2003. URL http://www.w3.org/TR/2003/REC-MathML2-20031021/.

[4] Michael Auth. *The BibTeX-XML-HTML Bibliography Project*, February 2004. URL http://www.authopilot.com/xml/.

[5] Tim Berners-Lee and Dan Connolly. HyperText Markup Language specification — 2.0. Request for Comments 1866, Network Working Group, November 1995. URL ftp://ftp.rfc-editor.org/in-notes/rfc1866.txt.

[6] Tim Berners-Lee, L. Masinter, and M. McCahill. Uniform resource locators (URL). Request for Comments 1738, Network Working Group, December 1994. URL ftp://ftp.rfc-editor.org/in-notes/rfc1738.txt.

[7] Rik Blok. *bbl2html.awk v1.3*, December 2000. URL http://www.zoology.ubc.ca/~rikblok/scripts/bbl2html.awk.

[8] Bert Bos, Tantek Çelik, Ian Hickson, and Håkon Wium Lie. Cascading style sheets, level 2 revision 1: CSS 2.1 specification. W3C working draft, W3C, June 2005. URL http://www.w3.org/TR/2005/WD-CSS21-20050613.

[9] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (XML) 1.0 (third edition). W3C recommendation, W3C, February 2004. URL http://www.w3.org/TR/2004/REC-xml-20040204.

[10] Sergey Brin. *HtmlTeX Home Page*. URL http://www-db.stanford.edu/~sergey/htmltex/.

[11] Otfried Cheong. *Hyperlatex Manual*, July 2005. URL http://hyperlatex.sourceforge.net/html/hyperlatex.html.

[12] Eric Chopin. *LaTeX4Web 1.2 Manual*. URL http://perso.wanadoo.fr/eric.chopin/latex/latex_subset.htm.

[13] James Clark. XSL transformations (XSLT). W3C recommendation, W3C, November 1999. URL http://www.w3.org/TR/1999/REC-xslt-19991116.

[14] Fabien Dagnat, Ronan Keryell, Laura Barrero Sastre, Emmanuel Donin de Rosière, and Nicolas Torneri. BibTeX++: Toward higher-order BibTeXing. *TUGboat*, 24(3):472–489, 2003. Proceedings of EuroTeX 2003.

[15] Dale Dougherty and Arnold Robbins. sed *and* awk. O'Reilly, second edition, February 1997. ISBN 1-56592-225-5.

[16] Nikos Drakos and Ross Moore. *The LaTeX2HTML Translator*, March 1999.

[17] Jean-Christophe Filliâtre and Claude Marché. *BibTeX2HTML — A translator of BibTeX bibliographies into HTML*, February 1999. URL http://www.lri.fr/~filliatr/bibtex2html/.

[18] Alan J. Flavell. Using FONT FACE to extend repertoire?, April 2005. URL http://ppewww.ph.gla.ac.uk/~flavell/charset/fontface-harmful.html.

[19] Free Software Foundation. GNU General Public License. In Joshua Gay, editor, *Free Software Free Society: Selected Essays of Richard M. Stallman*, pages 195–202. GNU Press, Boston, first edition, 2002. ISBN 1-882114-98-1.

[20] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) part one: Format of Internet message bodies. Request for Comments 2045, Network Working Group, November 1996. URL ftp://ftp.rfc-editor.org/in-notes/rfc2045.txt.

[21] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) part two: Media types. Request for Comments 2046, Network Working Group, November 1996. URL ftp://ftp.rfc-editor.org/in-notes/rfc2046.txt.

[22] *Documentation of Tralics*. The French National Institute for Research in Computer Science and Computing, 2003. URL http://www-sop.inria.fr/miaou/Jose.Grimm/tralics/doc-start.html.

[23] Stéphane Galland. *Documentation of Bib2HTML*, 3.0 edition, January 2005. URL http://www.arakhne.org/bib2html/doc/.

[24] Michel Goossens, Sebastian Rahtz, Eitan M. Gurari, Ross Moore, and Robert S. Sutor. *The LaTeX Web Companion: Integrating TeX, HTML, and XML*, chapter "The LaTeX2HTML translator", pages 83–154. Addison-Wesley Series on Tools and Techniques for Computer Typesetting. Addison-Wesley, June 1999. ISBN 0-2014-3311-7.

[25] Michel Goossens, Sebastian Rahtz, Eitan M. Gurari, Ross Moore, and Robert S. Sutor. *The LaTeX Web Companion: Integrating TeX, HTML, and XML*, chapter "Translating LaTeX to HTML using TeX4ht", pages 155–194. Addison-Wesley Series on Tools and Techniques for Computer Typesetting. Addison-Wesley, June 1999. ISBN 0-201-43311-7.

[26] Michel Goossens and Janne Saarela. From LaTeX to HTML and back. *TUGboat*, 16(2):174–214, 1995.

[27] Norman Gray. *Bibhtml Documentation*, September 2000. URL http://www.astro.gla.ac.uk/users/norman/distrib/bibhtml.html.

[28] José Grimm. Tralics, a LaTeX to XML translator. *TUGboat*, 24(3):377–388, 2003. Proceedings of EuroTeX 2003.

[29] Vidar Bronken Gundersen and Zeger W. Hendrikse. *BibTeXML Documentation*, May 2005. URL http://bibtexml.sourceforge.net/details.html.

[30] Vidar Bronken Gundersen and Rune Mathisen. *ISO Character Entities and their LaTeX Equivalents*, January 2001. URL `http://www.bitjungle.com/~isoent/isoent-ref.pdf`.

[31] Eitan M. Gurari. *TeX4ht: LaTeX and TeX for Hypertext*, February 2005. URL `http://www.cse.ohio-state.edu/~gurari/TeX4ht/`.

[32] Felix Hauser and Philip Schaffhauser. Database-driven XML-enabled bibliography management system. Diploma thesis, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology Zurich, March 2003. URL `http://dret.net/netdret/docs/da-ws2002-hauser-schaffhauser.pdf`.

[33] Johannes Henkel. *How to Compile and Use bib2xml*, October 2002. URL `http://www-plan.cs.colorado.edu/henkel/stuff/bib2xml/README-OR-DIE`.

[34] Jean-Michel Hufflen. MlBibTeX: Beyond LaTeX. In Karl Berry, Baden Hughes, and Steve Peter, editors, *Preprints for the 2004 Annual Meeting*, pages 77–84, Portland, OR, USA, April 2004. TeX Users Group.

[35] David Hull. *bib2html*. URL `http://pertsserver.cs.uiuc.edu/~hull/bib2html/`.

[36] Ian Hutchinson. *TtH: a "TeX to HTML" translator*, 3.40 edition. URL `http://hutchinson.belmont.ma.us/tth/manual/`.

[37] ECMA International. *ECMA-262: ECMAScript Language Specification*. ECMA International, Geneva, Switzerland, third edition, December 1999. URL `http://www.ecma-international.org/publications/standards/Ecma-262.htm`.

[38] International Organization for Standardization. *ISO 8879:1986: Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*, chapter D. International Organization for Standardization, Geneva, Switzerland, August 1986.

[39] International Organization for Standardization. *ISO/IEC 9899:1990: Programming languages — C*. International Organization for Standardization, Geneva, Switzerland, 1990.

[40] International Organization for Standardization. *ISO/IEC TR 9573-13:1991: Information technology — SGML support facilities — Techniques for using SGML — Part 13: Public entity sets for mathematics and science*. International Organization for Standardization, Geneva, Switzerland, 1991.

[41] International Organization for Standardization. *ISO/IEC 10646:2003: Universal Multiple-Octet Coded Character Set (UCS)*. International Organization for Standardization, Geneva, Switzerland, December 2003.

[42] David Kotz. bib2html, January 2003. URL `http://www.cs.dartmouth.edu/~dfk/bib2html/bib2html.html`.

[43] Marco Kuhlmann. *BibTeXML*, January 2004. URL `http://www.ps.uni-sb.de/~kuhlmann/bibtexml/`.

[44] Leslie Lamport. *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, second edition, 1994. ISBN 0-201-52983-1.

[45] LaTeX3 Project. The LaTeX Project Public License, version 1.3a, October 2004. URL `http://www.latex-project.org/lppl/lppl-1-3a.html`.

[46] John Levine, Tony Mason, and Doug Brown. *lex and yacc*. O'Reilly, second edition, October 1992. ISBN 1-56592-000-7.

[47] Håkon Wium Lie and Bert Bos. Cascading style sheets, level 1. W3C recommendation, W3C, January 1999. URL `http://www.w3.org/TR/1999/REC-CSS1-19990111`.

[48] Brenno Lurati and Luca Previtali. *BibTeXML*. Diploma thesis, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology Zurich, March 2001. URL `http://dret.net/netdret/docs/da-ws2000-lurati-previtali.pdf`.

[49] Mark Lutz. *Programming Python: Object-Oriented Scripting*. O'Reilly, second edition, March 2001. ISBN 0-596-00085-5.

[50] Luc Maranget. HeVeA, un traducteur de LaTeX vers HTML en Caml. URL `ftp://ftp.inria.fr/INRIA/moscova/maranget/hevea.ps.gz`.

[51] Luc Maranget. *HeVeA User Documentation — Version 1.08*, May 2005. URL `http://pauillac.inria.fr/~maranget/hevea/doc/`.

[52] Rune Mathisen and Vidar Bronken Gundersen. *SGML/XML Character Entity Reference*, August 2000. URL `http://www.bitjungle.com/~isoent/`.

[53] MicroPress. *The MicroPress TeXpider*. URL `http://www.micropress-inc.com/webb/wbstart.htm`.

[54] Bruce R. Miller. *LaTeXML: A LaTeX to XML Converter; Preview Version 0.3.0*. URL `http://dlmf.nist.gov/LaTeXML/LaTeXML.html`.

[55] David Mosberger. *The dlh Manual*, September 1996. URL `http://www.mostang.com/~davidm/dlh.html`.

[56] Oren Patashnik. *Designing BibTeX Styles*, February 1988.

[57] Oren Patashnik. *BibTeXing*, February 1988. Documentation for BibTeX 0.99b.

[58] Oren Patashnik. BibTeX yesterday, today, and tomorrow. *TUGboat*, 24(1):25–30, 2003. Proceedings of the 2003 Annual Meeting [of the TeX Users Group].

[59] Fabrice Popineau. Affichez vos documents LaTeX sur le Web avec TeX4ht. *Cahiers GUTenberg*, (37–38):5–43, December 2000. URL `http://www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/37-popineau.pdf`.

[60] Luca Previtali, Brenno Lurati, and Erik Wilde. BibTeXML: An XML representation of BibTeX.

In *Poster Proceedings of the Tenth International World Wide Web Conference*, pages 1090–1091, 2001. ISBN 962-85361-3-3. URL http://www10.org/cdrom/posters/1090.pdf.

[61] Chris Putnam. *Bibutils — Bibliography Conversion Utilities*. The Scripps Research Institute, February 2005. URL http://www.scripps.edu/~cdputnam/software/bibutils/.

[62] Russell W. Quong. *Ltoh: A Customizable LaTeX to HTML converter*, April 2000. URL http://quong.best.vwh.net/ltoh/.

[63] Dave Raggett. HTML 3.2 reference specification. W3C recommendation, W3C, January 1997. URL http://www.w3.org/TR/REC-html32.

[64] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. HTML 4.0 specification. W3C recommendation, W3C, April 1999. URL http://www.w3.org/TR/1998/REC-html40-19980424.

[65] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. HTML 4.01 specification. W3C recommendation, W3C, December 1999. URL http://www.w3.org/TR/1999/REC-html401-19991224.

[66] Ali Rahimi. *BibTeX to HTML converter*, July 2002. URL http://people.csail.mit.edu/rahimi/bibtex/.

[67] Arnold Robbins. *Effective awk programming*. O'Reilly, third edition, 2001. ISBN 0-596-00070-7.

[68] Hartmut Seichter. *Simply BibTeX*, February 2005. URL http://www.technotecture.com/?node=projects/simplybibtex/main.

[69] Dorai Sitaram. *TeX2page*, June 2005. URL http://www.ccs.neu.edu/home/dorai/tex2page/.

[70] Julian Smart. *Manual for TeX2RTF 2.0: A LaTeX to RTF and HTML converter*, November 1999. URL http://www.wxwindows.org/tex2rtf/docs.htm.

[71] Kevin Smith. *pyLaTeX Developer's Guide*, October 2001. URL http://pylatex.sourceforge.net/pylatex/.

[72] Kevin Smith. *pyLaTeX HTML Renderer*, October 2001. URL http://pylatex.sourceforge.net/HTML/.

[73] Kevin Smith. *pyldriver User's Guide*, October 2001. URL http://pylatex.sourceforge.net/pyldriver/.

[74] Diomidis Spinellis. *bib2xhtml*, July 2005. URL http://www.spinellis.gr/sw/textproc/bib2xhtml/.

[75] Sara Sprenkle. *bibtex2html*, January 2003. URL http://www.cs.duke.edu/~sprenkle/bibtex2html/README.

[76] Richard M. Stallman. Free software definition. In Joshua Gay, editor, *Free Software Free Society: Selected Essays of Richard M. Stallman*, chapter 3, pages 41–44. GNU Press, Boston, first edition, 2002. ISBN 1-882114-98-1.

[77] Nicolas Tessaud. HeVeA: Traduction de LaTeX. Rapport de stage d'option scientifique, The French National Institute for Research in Computer Science and Computing, July 1999. URL http://pauillac.inria.fr/~maranget/hevea/papers/nicolas/.

[78] The Unicode Consortium. *The Unicode Standard, Version 4.0*. Addison-Wesley, 2003. ISBN 0-321-18578-1. Amended by *Unicode 4.1.0* — URL http://www.unicode.org/versions/Unicode4.1.0/.

[79] Petr Toman. *Selathco 0.91 Documentation*, 2000. URL http://dione.zcu.cz/~toman40/selathco/.

[80] Jürgen Vollmer. *LaTeX2man — A Documentation Tool*, October 2004. URL http://www.ctan.org/tex-archive/support/latex2man/latex2man.html.

[81] W3C HTML Working Group. XHTML 1.0: The extensible hypertext markup language (second edition). W3C recommendation, W3C, August 2002. URL http://www.w3.org/TR/2002/REC-xhtml1-20020801.

[82] Kiri Wagstaff. *bib2html Documentation*, July 2002. URL http://www.litech.org/~wkiri/bib2html/bib2html.html.

[83] Larry Wall, Tom Christiansen, and Jon Orwant. *Programming Perl*. O'Reilly, third edition, July 2000. ISBN 0-596-00027-8.

[84] Thomas Widmann. Bibulus — a Perl/XML replacement for BibTeX. *TUGboat*, 24(3):468–471, 2003. Proceedings of EuroTeX 2003.

[85] Erik Wilde. Towards federated referatories. In *SINN03 eProceedings: Proceedings of the Conference on Wordwide Coherent Workforce, Satisfied Users*, September 2003. URL http://physnet.physik.uni-oldenburg.de/projects/SINN/sinn03/proceedings/wilde.html.

[86] Peter R. Wilson. *LTX2X: A LaTeX to X Autotagger*, January 1997. URL http://www.tug.org/tex-archive/support/ltx2x/ltx2x.html.