Remaking ACM \LaTeX{} styles

Boris Veytsman

Abstract

The Association for Computing Machinery is one of the largest publishers of computation texts in the world. It publishes more than fifty journals and many more conference proceedings every year. It was among the early adopters of \TeX{}. Unfortunately, over the years ACM styles accumulated many patches and haphazard changes. They diverged to the point when supporting became an impossible task. This warranted a complete refactoring.

This talk discusses the experience of rewriting ACM styles and the lessons learned.

1 Introduction

Five years ago I was asked to update \BibTeX{} styles for the Association of Computing Machinery (ACM). I did not know at that time that this commission would start a very interesting line of work.

The ACM is one of the largest publishers in the computing and information science in the world. It produces dozens and dozens of journals and conference proceedings. Thus I considered the work on this assignment to be a great honor and a large responsibility.

It befits the ACM mission and traditions that it is one of the early adopters of \TeX{}. There are timestamps in the ACM style files going as far back as the middle of 1980s, i.e., even predating \TeX{}3. As any computer specialist knows only too well, code this old requires much care and attention lest it become a crazy quilt of patches upon patches (the integrity of \TeX{} itself over the years is an important exception rather than the general rule). This is especially true when the code is maintained by generations of programmers stressed by deadlines and production requirements.

In the case of the ACM files, both the \LaTeX{} and \BibTeX{} code and the output display the result of many temporary \textit{ad hoc} decisions and show overlapping fingerprints of editors and coders, often with incompatible philosophies and approaches. As one frustrated \TeX{}pert wrote me (name withheld by request),

\[\ldots\text{3 packages copied in with a comment (good!) that they are needed but without taking out }\texttt{\endinput}\text{ that was in the code from the package copied in (bad :-}) so after the first nothing else is ever used\ldots\]

1. Class files:
   (a) \texttt{acm\_proc\_article-sp.cls}
   (b) \texttt{acmlarge.cls}
   (c) \texttt{acmsiggraph.cls}
   (d) \texttt{acmsmall-ec13.cls}
   (e) \texttt{acmsmall.cls}
   (f) \texttt{acmtog.cls}
   (g) \texttt{acmtrans2m.cls}
   (h) \texttt{sig\_alternate-05-2015.cls}
   (i) \texttt{sig\_alternate.cls}
   (j) \texttt{sigchi\_ext.cls}
   (k) \texttt{sigchi.cls}
   (l) \texttt{sigplanconf.cls}

2. \BibTeX{} styles:
   (a) \texttt{ACM\_Reference\_Format\_Journals.bst}
   (b) \texttt{SIGCHI\_Reference\_Format.bst}
   (c) \texttt{acmsiggraph.bst}
   (d) \texttt{acm\_abbrv.bst}
   (e) \texttt{acm\_alpha.bst}
   (f) \texttt{acm\_plain.bst}
   (g) \texttt{acm\_unsrt.bst}

\textbf{Figure 1}: Legacy code base (2015)

\[\ldots\text{and it seems there is a redefinition of }\texttt{startsection} \text{ inside that is broken—last night 30 min before my deadline I found 3 sections dangling at the bottom of columns}\ldots\]

\[\ldots\text{and the footnotes are horror and the fonts too and}\ldots\]

\[\ldots\text{looks worse than your average Word document }\ldots\]

These problems were exacerbated by the amount of copy and paste in the \TeX{} code. Many times over the years whenever the need arose, the original code was cloned, changed in subtle (or not so subtle) ways, and a new class file was released. At the end of 2015 I found that I was dealing with as many as 12 class files and 7 \BibTeX{} styles (Figure 1). Thus any update to the system required dozens of tantalizingly similar but slightly different changes in these files. This was not sustainable.

Another problem with the old styles was that interfaces to the elements like tables or figures were set long before the common standards were adopted. As the result, they looked quite strange for a \LaTeX{} user. The unusual ways to do usual things were confusing to the authors and caused errors.

Thus, the decision of the senior staff of ACM to make a radical refactoring of the styles was excellent news. Both the typographic design and the coding were going to change. This was an opportunity to write the styles from scratch.
2 Organization of work

With many stakeholders, it took some effort to organize the writing of the styles and templates. The tasks were distributed as follows. The ACM editors updated the design and fonts selection. I wrote the \LaTeX and Bib\LaTeX code. The company Aptara [1], which does typesetting for the ACM, developed word processor templates for the authors who do not use \TeX, as well as tools for the extraction of metadata.

Since many conference committees (SIGs) wanted to be involved in the process, \LaTeX and Bib\LaTeX code was put in a Github repository (https://github.com/borisveytsman/acmart). Github-based development turned out to be quite efficient for our purposes: the testers and SIG representatives could quickly assess the changes, submit bug reports and even contribute the code. Github seems to be a mature environment for free software development.

Sometimes it was difficult to accommodate the wishes of all the stakeholders, but we tried to keep in the spirit of compromise and consensus.

3 Design features

Instead of many class files (Figure 1) we use one class, acmart, with options corresponding to the output version. I sincerely hope this decision (one document class with options rather than several classes) will prevent the proliferation of copy-and-paste that plagued the old styles.

As suggested by the name, acmart is based on the famous amsart class [4], so all AMS-L\TeX advanced math typesetting features are available by default. You can use environments like cases, gather or multline, commands like \textfrac and \textfrc or \text in math mode, as well as AMS-style theorem definitions (the class itself defines several theorem-like constructs and theorem styles).

There are three journal options: acmsmall for small trim size journals, acmlarge for large trim size journals and acmtog for Transactions on Graphics, which traditionally uses two-column format. There are five proceedings options: sigconf for most conference proceedings, siguap, sigplan and sigchi for specific proceedings with distinct formatting, and sigchi-a for the special SIGCHI Extended Abstract. The latter is quite unusual: it has wide margins with marginal figures and tables. Another option, manuscript, is for a generic manuscript.

In Figures 2, 3, 4 and 5 some examples of the output are shown. Additional samples can be found in the documentation on CTAN (http://ctan.org/pkg/acmart) or in your \TeX distribution.

Another important decision was to eliminate use of proprietary fonts. The Libertine fonts [6] with newtxtm [5] give the pages a clean and crisp look. The footnotes are no longer cramped. In general, we tried to add a little air to the pages, while keeping in mind that the authors are constrained by page count limits.

One of the main principles of the design is the integrity of the interface. While the typesetting of the journals and proceedings is quite different, the interface is the same. The author should be able merely to change acmsmall to sigconf option in the \documentclass command in order to typeset the manuscript in a different category. The only exception are the marginal figures and tables for the sigchi-a option, which have no corresponding material in the other formats.

Another principle is the logical markup with most visual decisions made by L\TeX. This can be demonstrated by the way authors’ information is encoded. In the old design the authors should manually set the number of authors and align their addresses on the page using tabular-like commands. The new design does this automatically.

Since the \TeX file is used both for typesetting and for automatic extraction of metadata by Aptara...
A Multifrequency MAC Specially Designed for Wireless Sensor Network Applications

GANGLI WU, Department of Computer Science, University of Virginia
YANG WANG, College of Information and Computer Science, Shandong University
TANG WANG, College of Information and Computer Science, Shandong University
YING WU, Department of Computer Science, University of Virginia

Abstract

The problem of frequency assignment in wireless sensor networks (WSNs) is a critical issue that affects the performance of the network. Existing solutions, such as the orthogonal frequency-division multiple access (OFDMA) and the time-division multiple access (TDMA), are not sufficient for large-scale WSNs because of their limited frequency resources and the high computational cost required for frequency assignment. In this paper, we propose a new multifrequency MAC protocol that is specifically designed for WSNs. Our protocol uses an opportunistic approach to assign frequencies to nodes, taking into account the dynamic nature of WSNs. We evaluate our protocol through simulations, and the results show that it significantly improves the network's performance compared to existing solutions.

1. Introduction

Wireless sensor networks (WSNs) are becoming increasingly important in various applications, such as environmental monitoring, home automation, and military surveillance. A major challenge in WSNs is the frequency assignment problem, which is to assign frequencies to nodes in a way that avoids collisions and maximizes the network's throughput.

2. The Body of the Paper

2.1 Frequency Assignment

We propose a suboptimal distribution to be used by each node, which is based on the number of neighboring nodes. The distribution is such that the number of frequencies assigned to a node is inversely proportional to the number of neighboring nodes. This ensures that nodes with more neighbors receive fewer frequencies, which reduces the chance of interference.

2.2 Protocol Design

Our protocol is designed to work with any existing physical layer (PHY) and medium access control (MAC) protocol. It is straightforward to implement and does not require any modifications to the physical layer.

3. Evaluation

We evaluate our protocol through simulations using the ns-3 network simulator. The results show that our protocol significantly improves the network's performance compared to existing solutions, especially in dense environments.

4. Conclusion

In conclusion, we have proposed a new multifrequency MAC protocol that is specifically designed for WSNs. Our protocol uses an opportunistic approach to assign frequencies to nodes, taking into account the dynamic nature of WSNs. We believe that our protocol has the potential to significantly improve the performance of WSNs.

References


Figure 3: Journal output, two columns: acmtog

Figure 4: Proceedings output: sigconf

Figure 5: SIGCHI Extended Abstracts Sample File

Figure 6: Example of author information commands tools, the commands are highly structured. For example, the authors’ information is typed using the commands like \streetaddress or \city (Figure 6). There are special commands for grant sponsors and grant numbers, etc.

The class offers a number of useful features like Remaking ACM \LaTeX\ styles
canned copyright statements (vetted by the ACM lawyers), review mode with line numbers printed, anonymous mode with the information about the authors, affiliations, grants and acknowledgments suppressed (for a blind review), etc. This anonymous mode is just one of the options for conditional typesetting: the authors could also have different versions for the online and hard-copy; for example, the online version could include supplementary materials. There are provisions to include CCS “concepts”: hierarchical keywords generated by the ACM website.

The class uses standard \LaTeX\ interfaces to common elements, such as figures and tables, as much as possible. The only area with ACM-specific commands is the front matter: unfortunately all publishers use their own systems to indicate the authors and their affiliations, and ACM is no exception here.

4 Bibliography

Historically some ACM publications used author-year citations, while other used numbered cites. Even the author-year ones were not uniform: some used \texttt{natbib}, while some used their own interface. There were pervasive differences in bibliography formatting. This led to a large number of “official” ACM \texttt{BibTeX} styles (see Figure 1).

The new \texttt{acmart} package uses only one \texttt{BibTeX} style, which is \texttt{natbib}-compatible and defaults to numeric citations. Fortunately, the \texttt{natbib} package \cite{Daly:2010} allows the user to choose either author-year or numbered citations, thus allowing SIGs to customize their bibliographies. Even when the citation style is numeric, commands like \texttt{\citeyear} and \texttt{\citeauthor} are allowed.

Another interesting feature of the citation style is that the bibliographic output is highly structured for use by the cross-referencing software. This is done transparently to the user, creating the entries like the one shown on Figure 7.

5 Conclusions and acknowledgments

This large work of creating the new ACM styles would not be possible without the help of many people. I would like to express my gratitude to:

- ACM editors: Craig Rodkin, Bernard Rous.
- Aptara: Neeraj Saxena, Sehar Tahir.
- Authors of the early versions of ACM \TeX\ and \texttt{BibTeX} styles.

The new ACM styles are available on CTAN and the ACM web site, as well as in the major \TeX\ distributions like \TeX\ Live and Mik\TeX. As mentioned above, development is hosted at Github, \url{https://github.com/borisveytsman/acmart}. The Github interface is the best way to send me bug reports or feature suggestions.

References


\textcopyright{} Boris Veytsman

Systems Biology School and Computational Materials Science Center
MS 6A2
George Mason University
Fairfax, VA 22030 USA
borisv (at) lk dot net
\url{http://borisv.lk.net}