Ensuring Faultless Communication Behaviour in an E-Commerce Cloud Application

Rustem A. Kamun and Ross Horne

Kazakh-British Technical University,
Faculty of Information Technologies,
Almaty, Kazakhstan
r.kamun@gmail.com

Abstract

An increasing scope and complexity of Web services raises a new challenge of controlling their interaction. The goal of this work is to ensure that processes in a production Cloud are correctly interacting according to a specification of their communication behaviour. To accomplish this goal, we employ session types to analyse the global and local communication patterns. Session types represents ”formal blueprints” of how communicating participants should behave and offers a concise view of the message flows.

This work confirms the feasibility of application of session types on “non-linear” business protocols used by an e-commerce Cloud provider and developed in Session-Java, an extension of Java implementing Session-Based programming. Furthermore, we highlight the importance of this approach for services replicated across multiple Cloud providers each of which must correctly cooperate.

Contents

1 Introduction

The growing needs for information availability and accessibility present new challenges for application development. There are two forces working in parallel with regard to the need for integration. First, the necessity of application integration within a company (enterprise), and second, business-to-business integration. There is an increasingly widespread acceptance of Service-Oriented Architectures (SOA) as a paradigm for integrating software applications within and across organizational boundaries. In this paradigm, independently developed and operated applications are exposed as (Web) services which are then interconnected using a stack of standards, which depicted in Figure 1. However there remain open challenges when it comes to managing service interactions that go beyond simple sequences of requests and

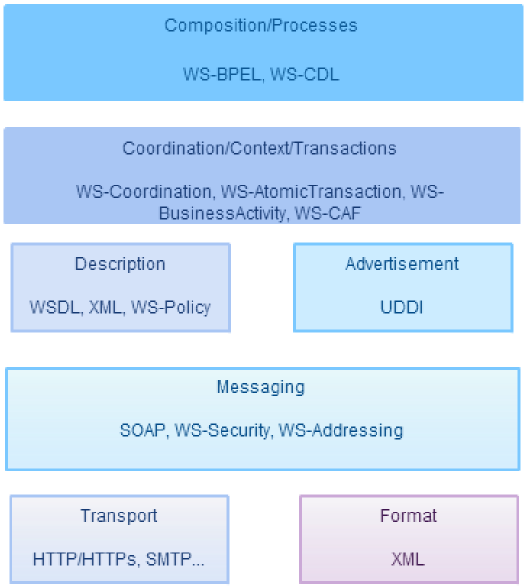

Composition/Processes
WS-BPEL, WS-CDL
Coordination/Context/Transactions
WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, WS-CAF
Description
WSDL, XML, WS-Policy
Advertisement
UDDI
Messaging
SOAP, WS-Security, WS-Addressing
Transport
HTTP/HTTPs, SMTP...
Format
XML

Figure 1: Stack of WS Standards

responses or involve large numbers of participants (multi-party communication). A need arises for new transaction implementations, more suitable for the web. One of those techniques for describing collaboration between a collection of services is a choreography model. It captures the interactions in which the participating services engage and interconnections between these interactions, including control-/data-flow dependencies. However, a choreography does not describe any internal action that occurs within a participating service. In turn, we do not have a clear idea how a global description can be executed, and, therefore, its functionality is not clear.

In this paper we will demonstrate a method of controlling process interactions represented by sessions. The formal theory based on session types ensures communication safety by verifying that session implementations of each engaged participant conform to the defined protocol specification. In order to prove the feasibility of this theory we utilized Session-Java, an extension to Java language. It works by specifying the intended process transaction protocol using session types and implementing the interaction using session operations. The conformance of protocol to its implementation is statically verified in compile mode, while the whole communication between peers is validated at runtime, at negotiation stage.

In this paper, first of all, we outline the main issues in Web-services Choreography Description Language discussed by W3C group [2], then we describe the main concepts of correctly functioning communication of processes (*conformance*, **well-formedness**), after that we show the implementation of Cloud scenarios and highlight possible application of session types into a services operating within multiple cloud vendors (cloud integrator).

Through these we are aiming to confirm the suitability of Session-Java as an implementation of business to business transactions. We want to explore the agility and robustness of the language and the scalability as scenarios vary in size but also complexity. In addition we will be looking for things such as ease of programming in SJ, any limitations, bugs or non-implementable scenarios.

2 Implementation of Choreography

Interacting processes must accomplish the goal of the computation. For this, the processes should have correct functionalities and correct interact with each other. With the interaction becoming more complex, the problems related to specify the interaction of the participants will be harder too, if we still want to do it locally. In addition, it is harder to verify the interaction locally. These are the main motivation under the design of WS-CDL.

Assume that a choreography C defines random role R, then a process P may be considered as implementation of R, $C \trianglerighteq P$, if

1. P can execute all communication required by R with other roles of C, in suitable time with suitable order.

2. P support all the local activities mentioned in R.

The paper [38] defined a small language Chor, a model of simplified WS-CDL, and a simple process language for the description of roles from a local viewpoint. Based on these models, this paper describes the concept of projections, that map a given choreography C to a set of role processes. Using a projection, we will get a set of processes, where each of them represents a role in the choreography. The main problem here is to give an answer on the following question: “Does process P can play role R defined by choreography C?”.

Figure 2: EasyChair logo

So a reasonable definition of the implementation of choreography is based on the concepts of projection and local conformance. By projection, it should be considered a procedure which takes a choreography in Chor and delivers a set of processes in the role language, while each of the processes corresponds to a role in choreography. In other words, a projection can partition a choreography C into a set of independent processes which realize the behaviour specified by this choreography provided in Eq. 1.

$$[[proj(C,1) \quad || \ldots || \quad proj(C,n)]] = [[C]] \tag{1}$$

3 Implementation of Cloud Scenarios

3.1 Scenario 1: SaaS integration

3.2 Scenario 2: Payment and Wallet Recharging

4 Future Work

The easychair class was designed to be easy to use, and specifically favoring electronic and on-paper publishing by the EasyChair conference system [36]. EasyChair is a free conference management system that is flexible, easy to use, and has many features to make it suitable for various conference models. It is currently probably the most commonly used conference management system [36]. The easychair class was designed according to some requirements, which are described in Appendix A. An article that occupies approximately 15 LNCS-formatted pages takes up approximately 14 easychair pages.

4.1 Main Criterias for Faultless Web-services

4.2 Choreography Implementation and Conformance

5 Typesetting

Typesetting with easychair is, well, easy. Just by using the document class entry in the document's preamble as follows: `\documentclass{easychair}` the typesetting work is nearly done. The easychair class is a relatively conservative extension of the standard `article` class, so most of the environments, section headers, etc. defined by `article` are available.

5.1 Generalities

The following are the general default parameters easychair introduces into the typesetting aspect of articles. If you use easychair for proceedings or other kinds of publishing through EasyChair, do not alter these – papers deviating from the formatting standards will be rejected by EasyChair.

1. The default paper size is US letter. It can be explicitly set to A4 (a4paper) or letter (letterpaper) paper in the document class entry, e.g.:
   \documentclass[a4paper]{easychair}
2. The print area for both letter and A4 paper sizes is 145x224 mm. This size has been selected to allow for inexpensive printing using our current print-on-demand publisher.
3. The base font is Computer Modern, and the sans-serif font is Helvetica. The base font size is 10pt. If you use any other font size, there is no guarantee that the produced document will look nice or fit into our standard page size.
4. The references list is condensed. The default bibliography styles, such as plain, abbrv, and alpha, are suggested.
5. PNG, JPG, and PDF images are supported, i.e., those that are supported by the standard graphicx package [3], and render nicely in online versions of PDF documents. This document shows some examples of JPG and PDF images, for example in Figures 2. If the papers are designed for publishing in print, the images should be at least 300dpi in resolution.

5.2 Front Matter

The front matter of an easychair article follows the article style, augmented with the \titlerunning and \authorrunning commands for use by authors. For the \author command with multiple authors, use \and to separate authors from different institutions, as done in this document. Institutions are defined using \institute in a similar way and affiliations of authors assigned using \inst. Authors must set the \titlerunning and \authorrunning. For example, the front matter of this document defined the authors and title as follows.

```
\title{The {\easychair} Class File \\
Documentation and Guide for Authors}
\titlerunning{The {\easychair} Class File}

\author{
    Serguei A. Mokhov\inst{1}\thanks{Designed and implemented the class style}
\and
    Geoff Sutcliffe\inst{2}\thanks{Did numerous tests and provided a lot of suggestions}
\and
   Andrei Voronkov\inst{3}\thanks{Masterminded EasyChair and created versions
     3.0--3.1 of the class style}\\
\and
   Graham Gough\inst{3}\thanks{Changed author list format.}\\
}

\institute{
  Concordia University,
  Montreal, Quebec, Canada\\
  \email{mokhov@cse.concordia.ca}
\and
```

```
   University of Miami,
   Miami, Florida, U.S.A.\\
   \email{geoff@cs.miami.edu}\\
\and
   University of Manchester,
   Manchester, U.K.\\
   \email{andrei@voronkov.com, graham@cs.man.ac.uk}\\
 }
\authorrunning{Mokhov, Sutcliffe, Voronkov and Gough}
```

5.3 Page Numbering

Page numbers are at the bottom of every page. Authors must leave the page numbers in as-is. When EasyChair proceedings are processed by EasyChair, the correct volume page numbers will be inserted automatically.

5.4 Section Headings and Capitalization

Section and paragraph headings in easychair are invoked via the standard commands, such as `\section`, `\subsection`, `\subsubsection`, and `\paragraph`. Generally, every non-trivial word in a heading must be capitalized according to general capitalization guidelines. A reasonable rule to use is that all prepositions, coordinating conjunctions and articles having four or fewer letters should not be capitalized. If you do not know what it means, simply do not capitalize the following words: *amid, anti, as, at, atop, but, by, down, for, from, in, into, like, near, next, of, off, on, onto, out, over, pace, past, per, plus, qua, save, than, till, to, up, upon, via, with, for, and, nor, but, or, yet, so, a, an, the.* For example, if you want to call your paper "oldest but goldest", then the proper title for it is "Oldest but Goldest". "Oldest but goldest" is wrong (since "goldest" is not capitalized) and "Oldest But Goldest" is wrong ("but" should stay lower-cased since it belongs to the list of words above.) Needless to say, "OLDEST BUT GOLDEST" is very wrong.

Paragraph headings should not be capitalized and should have a trailing period. That is, you should write

```
\paragraph{EasyChair is cool.}
```

rather than

```
\paragraph{EasyChair is cool}
```

unless your aim was to write something like

```
\paragraph{EasyChair is cool} when you use it for publishing.
```

Of course, you are welcome to replace the trailing period by any other punctuation mark, for example

```
\paragraph{EasyChair is cool!}
```

See the examples in this document, e.g., Section 5 is a section, this (Section 5.4) is a subsection, and Section 5.4.1 is a subsubsection.

5.4.1 Subsubsection Header

This is a subsubsection.

Paragraph header. This is a paragraph. One way of saving space when hyper-references are not essential is to use paragraphs instead of subsubsections.

5.5 Mathematics

Mathematics can be done inline for simple things, e.g., an equation $x = 0$, possibly with super and subscripts, e.g., $x_k^2 \approx 27$, Greek letters, e.g., $\alpha \cup \Theta \neq \gamma$, etc. Larger formulae must be done using `\[ \]` bracketing, e.g.,

$$\int_0^1 x dx = \left[\frac{1}{2}x^2\right]_0^1 = \frac{1}{2}$$

or using `\begin{equation}` and `\end{equation}` for numbered equations, e.g.,

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{n\to\infty} (1 + x/n)^n \tag{2}$$

Use `\begin{align*}` and `\end{align*}` (or without the `*` include number) to align equations, e.g.,

$$\begin{aligned} x^2 + y^2 &= 1 \\ y &= \sqrt{1 - x^2} \end{aligned}$$

Fonts, using `\matcal` and others can also be used in the math mode: $\mathcal{ALC}$.

5.6 Tables

Table 1 shows an example of a table of data that was conveniently available (i.e., the data has nothing to do with easychair) apart from being related to the research of two of its authors.

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
	/100	time	out	Con.	Eff.	/35	/40	/25
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25
SInE 0.4	64	75.3	64	0.32	8.5	26	13	25
leanCoP-SInE 2.1	35	110.8	35	0.23	3.2	23	1	11
E-LTB 1.1pre	18	63.4	0	0.21	2.8	7	9	2
EP-LTB 1.1pre	18	77.8	18	0.21	2.3	7	9	2
E-KRH'-LTB 1.1.3	0	–	–	–	–	0	0	0

Table 1: LTB division results

5.7 References

References must be provided in a `.bib` file, so that BibTeX can be used to generate the references in a consistent style in a volume. The preferred styles are `plain` and `alpha`. For example, the references for this paper are generated from the lines

```
\bibliographystyle{plain}
\bibliography{easychair}
```

and a way to compose the entires, e.g. citing this class style [13] is below:

```
@misc
{
  easychair-latex-class,
  author       = {Serguei A. Mokhov and Geoff Sutcliffe and Andrei Voronkov},
  title        = {The {\sf easychair} Class File Documentation and Guide
                  for Authors},
  year         = {2008--2011},
  howpublished = {[online]},
  note         = {Available at \url{http://easychair.org/easychair.zip}}
}
```

6 Installation and Usage Instructions

6.1 Installation

The “installation” of the easychair document class is easy. Download the latest version of the `easychair.zip` package from `http://www.easychair.org/easychair.zip` and unzip it in the directory where you will prepare your paper. You will get the following files, out of which you may need to keep only the `easychair.cls` style class if you are familiar with the rest of the files and do not require them to get started. We are also working to make easychair available from CTAN [30], such that it can be installed with the popular TeXLive [20] and MiKTeX [22] LaTeX package management systems.

- `easychair.cls` – the class file that this is all about.
- `easychair-a4.pdf` – the PDF version of this guide rendered using `a4paper` option.
- `easychair.tex` – the LaTeX source of this guide, and `easychair.bib` – the supporting bibliography entries found starting on page 14.
- `logoEC.pdf` – the PDF version of the EasyChair logo rendered in Figure 2 and `throneEC.jpg` – the JPG version of the easy throne rendered in Figure 4.

6.2 Required Packages

The easychair class relies only on packages deemed standard and shipped by most LaTeX distributions in the worlds of Linux (current `texlive` [20] or older `tetex`), MacOS X, and Windows (via Cygwin or MiKTeX). If for some reason your distribution is old or doesn’t have the packages listed below, you can always obtain a copy from CTAN [30]. Note that EasyChair loads this packages automatically so you don’t have to use `\usepackage` to use any of these packages.

- `url` [1] (included also by `hyperref` automatically) – to provide URL rendering support for the monospaced font, which takes care of special characters as well as line wrapping.

- `hyperref` [19] – to allow hyperlinking of URLs and cross references within an article. Its options are set to either `letterpaper` or `a4paper`, depending on the `\documentclass` options.

- `graphicx` [3] – the standard package for rendering PNG, JPG, and PDF graphic images, primarily in `figure` environments.

- optional `mathptmx` [23] – Times base font for compactness (use with the `withtimes` easychair option).

- `helvet` [24] – Helvetica as sans-serif.

- `listings` [14] – to allow highlighted source code listing styles.

- `latexsym` [32] – to provide common math and other symbols.

- `amsthm` [29] – to provide $\mathcal{AMS}$ theorem-like environments.

- `empheq` [10] – to provide equation environments, etc.

- `geometry` [34] – to set easychair margins, outlined in Section 5.1.

- `lastpage` [7] – to allow computationally referencing the last page.

- `fancyhdr` [35] – for running heads.

- `footmisc` [4] – to ensure that footnotes are always at the bottom.

- optional `makeidx` [33] – for index generation (use with the `thesis` easychair option).

- `eso-pic` [16] – for draft versions and checking page overlflows vs. a border drawn around the headers, footers, and the main body of the article.

6.3 Recommended Packages

Here is a list of some packages that this guide's authors have experimented with, and which are suitable for inclusion if needed by article authors. These packages must be loaded using `\usepackage`. In general, authors may use any standard packages provided they do not change the basic layout and font settings established by the easychair class. Such packages must be provided with the submission of articles.

- `rotating` [5] – to rotate floats (figures and tables) on the page, when wide tables or figures do not fit in portrait layout.

- `pdflscape` [18] – similar to `rotating`, but also allows rotating text to make it conveniently viewable in a PDF viewer that supports individual rotated pages. A possible disadvantage is that a page break is forced, which may create gaps before or after the landscape page.

- `algorithm2e` [6] – provides a figure-like algorithm environment for formal algorithm presentation with highlighting.

6.4 Compiling

pdflatex [9] is the preferred tool for producing PDF files with easychair class documents. The author kit (easychair.zip) includes some minimal automation that authors can use at their discretion.

- Linux and UNIX-like platforms (also works under Cygwin and MacOS X): A Makefile is provided for the GNU make [26] utility, so this document can be compiled by typing make at the terminal prompt (on the systems where both GNU and non-GNU versions of make are installed, one may need to use gmake).

- Microsoft Windows: TEXnicCenter [37] or LEd [25] and MiKTEX [22] as their backend are common tools for LATEX processing under Microsoft Windows. The former provide a GUI front-end to LATEX, and the latter is the Windows native-compiled binaries and standard packages with a comprehensive package update tool. The easychair.tcp project file is provided for TEXnicCenter users, as well as easychair.lpr for LEd users.

- MacOS X: TeXShop [11] is a tool for LATEX processing under Mac OS X. It provides a GUI front-end to LATEX. The backend can be installed through the fink [31] repository or the Darwin Ports.

Some authors use packages that require the use of latex instead of pdflatex, most notably pstricks and its derivatives. If switching to modern packages, such as tikz is not an option, these authors can use the easychair class with latex.

6.5 Bug Reports

Please report bugs, errors, and omissions you find with the easychair class to its current maintainer, Andrei Voronkov, at andrei@voronkov.com. Any *constructive* feedback is always welcome. If anybody has a very hard question, we might be unable to answer it without Sergei. If anybody knows his whereabouts, Geoff and Andrei have the ransom money ready.

7 Avoiding Text Overflows

When used in EasyChair, the easychair class is intended for publishing, including printing. This implies that the text of your document should not overflow the page dimensions. This section gives some practical advice on how you can avoid text overflows. When you run pdflatex or latex on your files, you sometimes get warnings like

```
LaTeX Warning: Overfull \hbox (7.43138pt too wide) in paragraph at lines 994--994
```

or

```
LaTeX Warning: Float too large for page by 55.21666pt on input line 638.
```

They mean what they say: your text overflows the allowed space. When you prepare a PDF document that will only be used online or as a draft, in many cases you can ignore these warnings, since they only make your document uglier than it could be but it remains still readable. However, for documents that are supposed to be printed, overflows make a lot of difference. There is no universal recipe on how to make your text fit into the page. However, there are some common cases that can be easily fixed using the tricks described here.

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

Figure 3: Original table and tables with `tabcolsep` set to 5pt, 3pt, and 1pt

7.1 Using the `debug` Option to Check for Overflows

A general guideline is that you should first check how the alleged overflows looks in reality. For example, some LaTeX overflows are invisible or hardly noticeable for people and overlows by less than 4pt look OK anyhow. The easychair class has an option `debug` that adds to each page of your documents a red frame corresponding to the allowed page dimensions. Figure 5 on page 17 shows an example of a page created using this style. With such a red frame, you can clearly see any overflows apart from very minor ones.

7.2 Tables

Many page overflows happen because of large tables. In many case these overflows can be easily removed by slightly reducing padding added by LaTeX to every column. It is controlled by the LaTeX command `\tabcolsep` whose value by default is 6pt. Even small changes in the value of this command may give drastic reductions in the width of tables. This is illustrated in Figure 3 on page 11. Note though that there is no free lunch: smaller values for this command may result in lower redability.

7.3 Images

Images included using `\includegraphics` are easy to resize since one can specify the size of the result explicitly. For example, Figure 4 shows three copies of the same image having different sizes obtained using the following commands:

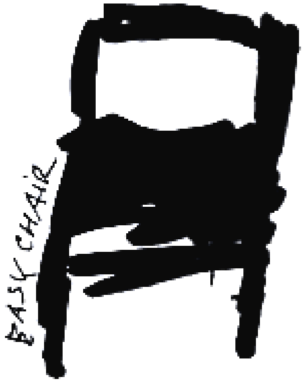
```
\includegraphics[width=0.5\textwidth]{throneEC.jpg}
\includegraphics[width=0.3\textwidth]{throneEC.jpg}
\includegraphics[width=0.15\textwidth]{throneEC.jpg}
```

7.4 A Universal Recipe

LaTeX has a very powerful weapon for reducing the size of almost anythings. More precisely, it can reduce anything producing what LaTeX considers a box. This weapon is called `\scalebox`.

Consider an example (check the source of this file to see how it was produced).

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054



The number of users of EasyChair and one of its logos,
scaled to the number of users in 2010

This is what happens when we put (almost) the same LaTeX code in `\scalebox{0.55923}{...}` to scale it down to the number of users in 2009:

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

The number of users of EasyChair and one of its logos,
scaled to the number of users in 2009

We can scale it down even further to the 2008 figure using `\scalebox{0.30856}{...}`:

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

The number of users of EasyChair and one of its logos,
scaled to the number of users in 2008

or further down to 2007:

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

The number of users of EasyChair and one of its logos,
scaled to the number of users in 2008

This size reduction technique is very efficient: using the right scale you may post your whole article on Twitter in a single tweet. However, it may also may parts of your text virtually unreadable with an unfortunate side effect of annoying reviewers.

8 Submitting Your Article for EasyChair Proceedings

This section is intended only for the authors and editors of EasyChair proceedings.

When you prepare an article for EasyChair proceedings, it should be submitted through EasyChair. EasyChair automates the submission process as much as possible and goes to a great length to ensure that your article can be published and printed. Publication for EasyChair means much more than just putting a PDF of your article online. It collects a some meta-information about the article to classify it, find similar articles, make it easily searchable, and index it in various Web services, such as DBLP. This section explains how EasyChair processes your article.

8.1 Submitting the Article

Submitting the article is easy. All you should do is to put the source of your article in a single zip file. The source must contain all auxiliary files required to create a PDF file of your article: this includes images, bibliography, and all non-standard LaTeX packages you used[1] For example, suppose that your main LaTeX file is `main.tex`, it inputs another file `macros.tex` and uses the file `biblio.bib` to produce the bibliography. Suppose it also uses two images `images/easy.jpg` and `images/easy.jpg`. Then you should create a zip archive containing all these files. Suppose all these files are put in a directory `mypaper` on your computer, where `images` is a subdirectory of `mypaper`

On almost any operating system (Linux, Windows, or Mac) you can achieve this by using the following sequence of commands:

```
cd mypaper
zip -r mypaper.zip *
```

This will create a zip archive `mypaper.zip` including all files in the directory `mypaper` and its subdirectories.

9 Future Work

We plan to further strengthen the easychair class and promote it for electronic publishing for EasyChair-powered conferences and workshops, and take over the world, as shown in Figure 4. We aim at creating a new model of *affordable publishing*, where anybody can become a low-cost publisher.

9.1 Acknowledgments

- Aleksander Kosenkov for the graphics that are used here and on the EasyChair website [36].
- The CTAN [30] and LaTeX communities [37, 22].
- Leslie Lamport for LaTeX [12].
- Peter Grogono for his neat kickstart LaTeX introduction [8].
- Guilin Qi, Jasmin Christian Blanchette, Leslie Lamport, Uwe Pfeiffer, and others for constructive feedback on the style, most of which got incorporated into the version 2 of the class style.

[1] A non-standard LaTeX package is a package that is not included in CTAN.

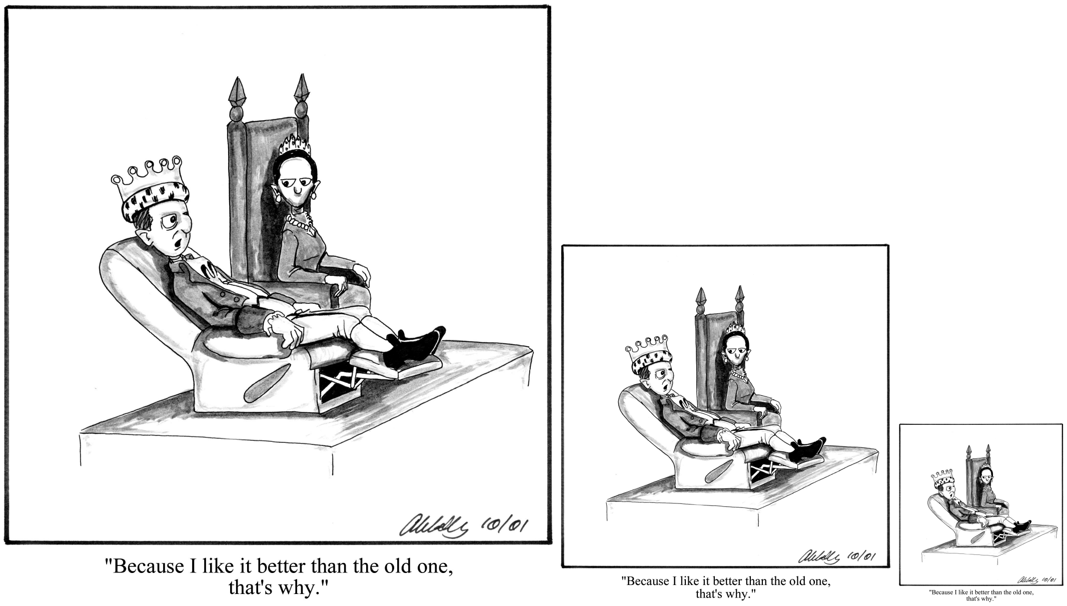

"Because I like it better than the old one,
that's why."
"Because I like it better than the old one,
that's why."
"Because I like it better than the old one,
that's why."

Figure 4: Why one should use EasyChair

9.2 History

- easychair version 3.1 – May 2011, several changes intended for automatic processing by EasyChair.
- easychair version 3.0 – May 2011, changed to use a 10pt font.
- easychair version 2.0 – April 2010
- easychair version 1.0 – June 2008, initial release, used in ESARM'08 [28, 27] and 5 other workshops [36].

References

[1] Donald Arseneau. url: Verbatim with URL-sensitive line breaks. `http://www.ctan.org/tex-archive/help/Catalogue/entries/url.html`, last viewed April 2010, 1986–2011.

[2] D. Marlon B. Alistair and O. Phillipa. *A critical overview of the web service choreography description language*. BPTrends, 2005.

[3] David Carlisle. graphicx: Enhanced support for graphics. `http://www.ctan.org/tex-archive/help/Catalogue/entries/graphicx.html`, last viewed April 2010, 1995–1999.

[4] Robin Fairbairns. footmisc: A range of footnote options. `http://www.ctan.org/tex-archive/help/Catalogue/entries/footmisc.html`, last viewed April 2010, 1986–2009.

[5] Robin Fairbairns and Sebastian Rahtz. rotating: Rotation tools, including rotated full-page floats. `http://www.ctan.org/tex-archive/help/Catalogue/entries/rotating.html`, last viewed April 2010, 2001–2009.

[6] Christophe Fiorio. algorithm2e: Floating algorithm environment with algorithmic keywords. `http://www.ctan.org/tex-archive/help/Catalogue/entries/algorithm2e.html`, last viewed April 2010, 1986–2009.

[7] Jeffrey Goldberg. lastpage: Reference last page for Page N of M type footers. `http://www.ctan.org/tex-archive/help/Catalogue/entries/lastpage.html`, last viewed April 2010, 1986–2010.

[8] Peter Grogono. *A LaTeX2e Gallimaufry. Techniques, Tips, and Traps.* Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, March 2001. `http://www.cse.concordia.ca/~grogono/Writings/gallimaufry.pdf`, last viewed May 2008.

[9] Carl Gutwin. Instructions for pdflatex. `http://www.cs.usask.ca/~gutwin/gi/pdflatex.htm`, last viewed April 2010, 2006.

[10] Morten Høgholm. empheq: EMPHasizing EQuations. `http://www.ctan.org/tex-archive/help/Catalogue/entries/empheq.html`, last viewed April 2010, 2002–2007.

[11] Richard Koch, Max Horn, Gerben Wierda, and Various Contributors. TeXshop. `http://www.texshop.org`, 2001–2011.

[12] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, 1986.

[13] Serguei A. Mokhov, Geoff Sutcliffe, and Andrei Voronkov. The easychair class file documentation and guide, for authors and editors. [online], 2008–2011. Available at `http://easychair.org/easychair.zip`.

[14] Brooks Moses and Carsten Heinz. listings: Typeset source code listings using LaTeX. `http://www.ctan.org/tex-archive/help/Catalogue/entries/listings.html`, last viewed April 2010, 2006–2007.

[15] G. Ritzinger N. Kavantzas, D. Burdett and Y. Lafon.

[16] Rolf Niepraschk. eso-pic: Absolute positioning of graphics and ship out actions. `http://www.ctan.org/tex-archive/help/Catalogue/entries/eso-pic.html`, last viewed April 2010, 1998–2009.

[17] OASIS. Business transaction protocol. `https://www.oasis-open.org/committees/download.php/1184/2002-06-03.BTP_cttee_spec_1.0.pdf`, 3 June, 2002.

[18] Heiko Oberdiek. pdflscape: Make landscape pages display as landscape. `http://www.ctan.org/tex-archive/help/Catalogue/entries/pdflscape.html`, last viewed April 2010, 2001–2008.

[19] Heiko Oberdiek and Sebastian Rahtz. hyperref: Extensive support for hypertext in LaTeX. `http://www.ctan.org/tex-archive/help/Catalogue/entries/hyperref.html`, last viewed April 2010, 2001–2010.

[20] Sebastian Rahtz, Karl Berry, Manuel Pégourié-Gonnard, Norbert Preining, Peter Breitenlohner, Reinhard Kotucha, Siep Kroonenberg, Staszek Wawrykiewicz, Tomasz Trzeciak, Vladimir Volovich, and TeXuser groups. TeX Live. `http://tug.org/texlive/`, last viewed April 2010, 1996–2009.

[21] Vadim V. Kotov Rustem A. Kamun, German Ilyin. Cloud platform for optimizing your business performance. [online], 2013. Available by `http://v3na.com`.

[22] Christian Schenk and MiKTeX Contributors. MiKTeX. `http://miktex.org`, last viewed April 2010, 2008–2011.

[23] Walter Schmidt, Alan Jeffrey, Sebastian Rahtz, and Ulrik Vieth. mathptmx: Use Times as default text font, and provide maths support. `http://www.ctan.org/tex-archive/help/Catalogue/entries/mathptmx.html`, last viewed April 2010, 1986–2009.

[24] Walter Schmidt and Sebastian Rahtz. helvet: Font support for common PostScript fonts. `http://www.ctan.org/tex-archive/help/Catalogue/entries/psnfss.html`, last viewed April 2010, 1995–2005.

[25] Adam Skórczyński, Sebastian Deorowicz, and LEd Contributors. LEd (LaTeX editor). `http://latexeditor.org`, last viewed April 2010, 2004–2011.

[26] Richard Stallman, Roland McGrath, Paul Smith, and the GNU Project. GNU Make. Free Software Foundation, Inc., [online], 1997–2006. `http://www.gnu.org/software/make/`.

[27] Geoff Sutcliffe, Simon Colton, and Stephan Schulz, editors. *Empirically Successful Automated Reasoning for Mathematics (ESARM)*. CEUR-WS.org, July 2008. http://CEUR-WS.org/Vol-378, last viewed April 2010.

[28] Geoff Sutcliffe, Simon Colton, and Stephan Schulz. Empirically successful automated reasoning for mathematics (ESARM) - 27th july 2008. [online], July 2008. http://events.cs.bham.ac.uk/cicm08/workshops/esarm/, last viewed April 2010.

[29] The American Mathematical Society. amsthm: LaTeX package for theorem setup (AMS style). http://www.ctan.org/tex-archive/help/Catalogue/entries/amsthm.html, last viewed April 2010, 1995–2004.

[30] The CTAN team. CTAN: the comprehensive TeX archive network. ctan.org, 1992–2011.

[31] The Fink Project. Fink. [online], 2001–2011. http://www.finkproject.org/.

[32] The LaTeX Team. latexsym: Base sources of LaTeX. http://www.ctan.org/tex-archive/help/Catalogue/entries/latex-base.html, last viewed April 2010, 1986–2009.

[33] The LaTeX Team. latexsym: Standard package for creating indexes. http://www.ctan.org/tex-archive/help/Catalogue/entries/makeidx.html, last viewed April 2010, 1986–2009.

[34] Hideo Umeki. geometry: Flexible and complete interface to document dimensions. http://www.ctan.org/tex-archive/help/Catalogue/entries/geometry.html, last viewed April 2010, 1996–2008.

[35] Piet van Oostrum. fancyhdr: Extensive control of page headers and footers in LaTeX2e. http://www.ctan.org/tex-archive/help/Catalogue/entries/fancyhdr.html, last viewed April 2010, 1986–2009.

[36] Andrei Voronkov. EasyChair conference system. easychair.org, 2004–2011.

[37] Sven Wiegand and TeXnicCenter Contributors. TeXnicCenter. http://texniccenter.org, last viewed April 2010, 1998–2011.

[38] X. Zhao Z. Qiu, C. Cai and H. Yang. *Exploring the essence of choreography.* Proceedings of the 16th international conference on WWW, 2007.

A easychair Requirements Specification

The following high-level requirements were set for the development of the easychair class, and were refined as development went along.

1. The style should be easy to use. The average LaTeX user should not need to read a long manual.

2. It should be economical in space but the text should be nice-to-read.

3. It should use fonts producing a reasonable-quality PDF.

4. The bibliography should produce hyperlinks.

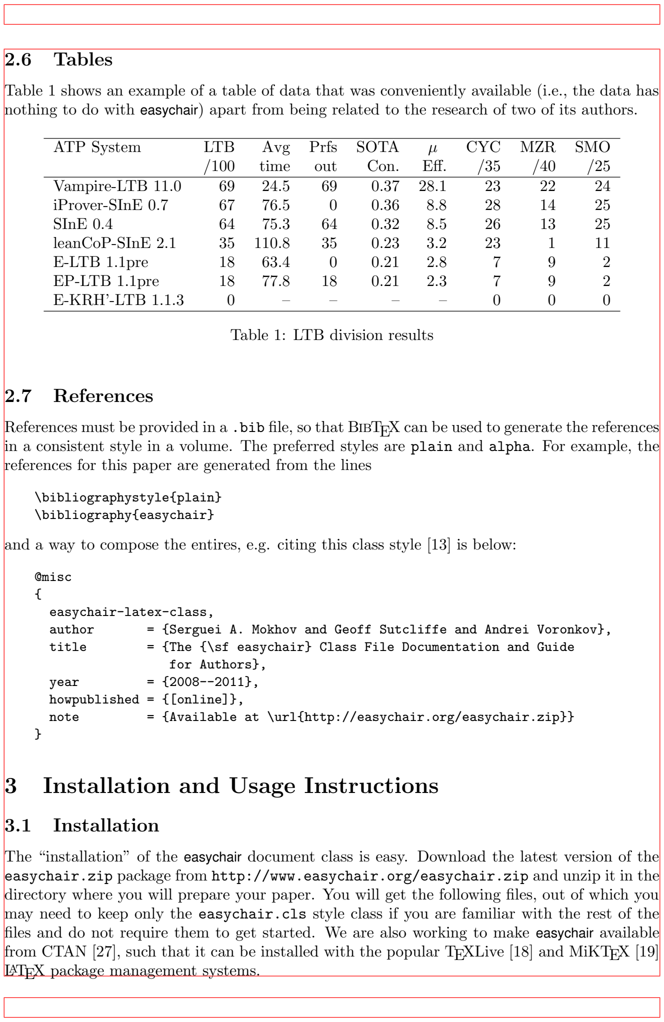
5. Sections should produce menu sections in PDF.

6. The text should look good on both A4 and letter paper.

7. The style should be single-column for convenience of scrolling.

8. The print area should be convenient for printing using print-on-demand publishers.

9. Running heads.



2.6 Tables

Table 1 shows an example of a table of data that was conveniently available (i.e., the data has nothing to do with easychair) apart from being related to the research of two of its authors.

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
	/100	time	out	Con.	Eff.	/35	/40	/25
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25
SInE 0.4	64	75.3	64	0.32	8.5	26	13	25
leanCoP-SInE 2.1	35	110.8	35	0.23	3.2	23	1	11
E-LTB 1.1pre	18	63.4	0	0.21	2.8	7	9	2
EP-LTB 1.1pre	18	77.8	18	0.21	2.3	7	9	2
E-KRH'-LTB 1.1.3	0	–	–	–	–	0	0	0

Table 1: LTB division results

2.7 References

References must be provided in a `.bib` file, so that BibTeX can be used to generate the references in a consistent style in a volume. The preferred styles are `plain` and `alpha`. For example, the references for this paper are generated from the lines

```
\bibliographystyle{plain}
\bibliography{easychair}
```

and a way to compose the entires, e.g. citing this class style [13] is below:

```
@misc
{
  easychair-latex-class,
  author      = {Serguei A. Mokhov and Geoff Sutcliffe and Andrei Voronkov},
  title       = {The {\sf easychair} Class File Documentation and Guide
                 for Authors},
  year        = {2008--2011},
  howpublished = {[online]},
  note        = {Available at \url{http://easychair.org/easychair.zip}}
}
```

3 Installation and Usage Instructions

3.1 Installation

The "installation" of the easychair document class is easy. Download the latest version of the `easychair.zip` package from `http://www.easychair.org/easychair.zip` and unzip it in the directory where you will prepare your paper. You will get the following files, out of which you may need to keep only the `easychair.cls` style class if you are familiar with the rest of the files and do not require them to get started. We are also working to make easychair available from CTAN [27], such that it can be installed with the popular TeXLive [18] and MiKTeX [19] LaTeX package management systems.

Figure 5: A page of a document created using the `debug` option