The Report Committee for Preston Brent Landers
certifies that this is the approved version of the following report:

Speakur: a case study of Web Component architecture for composable applications

APPROVED BY

SUPERVISING COMMITTEE:

Adnan Aziz, Supervisor

Christine Julien

Speakur: a case study of Web Component architecture for composable applications

by

Preston Brent Landers, B.A.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2015

Dedicated to my wife Andrea and to my parents.

Acknowledgments

I wish to thank the multitudes of people who helped me. Time would fail me to tell of the multitudes of individuals ...

Speakur: a case study of Web Component architecture for composable applications

Preston Brent Landers, M.S.E.
The University of Texas at Austin, 2015

Supervisor: Adnan Aziz

This report is a case study of applying encapsulation, composition and distributed synchronization techniques to web application architecture through the use of Web Components, a proposed extension to the W3C HTML5 document standard. The author presents Speakur, a real-time social discussion plugin for the mobile and desktop web, as an example of how HTML5 Web Component technologies can enable web designers and programmers to realize engineering principles such as modularity, encapsulation and separation of concerns when composing pages out of many disparate elements sourced from a diverse set of authors.

Web authors can add a Speakur discussion to their page by inserting a simple HTML element at the desired spot to give the page a real-time discussion or feedback system. Speakur uses Google's Polymer framework's implementation of the draft Web Components (WC) standard to achieve encapsulation of its internal

implementation details from the containing page, which only interact through a well defined interface (API).

Web Components are a proposed HTML5 standard for writing custom HTML tags that take advantage of new browser technologies like Shadow DOM, CSS Flexbox and data-bound templates. This report will review these new Web Component technologies and provide a case study of structuring a real-world WC applet that is embedded in a larger page or app. The major research question is whether W3C Web Components provide a viable path towards the encapsulation and composition principles that have largely eluded web engineers thus far. In other words, *are components really the future of the web*? Subsidiary questions include assessing the maturity and suitability of current Web Components technologies for widespread deployment and how to efficiently synchronize component state across a distributed network.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This report presents a case study of applying W3C Web Components to achieve encapsulation and separation of concerns within the context of collaborative web authoring. The author has created Speakur, a real-time discussion social plugin for the web, as an experiment to determine the maturity and viability of using Web Components to develop modern web applications.

Like most web apps, Speakur is primarily based on HTML and Javascript. The Hypertext Markup Language (HTML) document standard has proven wildly successful since its introduction in 1993 by British computer scientist Tim Berners-Lee, with billions and billions of pages served, and millions of public and private web sites forming a major part of our information landscape. [CITE?] More than any other invention other than perhaps email, the World Wide Web has shaped how we see and use the global network.

Those designing and programming applications for the Web platform have long dreamed of the ability to mix and match independent, reusable chunks of functionality (components) in their documents without mutual interference. The original and current Document Object Model (DOM) browser abstraction provided by HTML does not allow for significant decoupling; everything lives to-

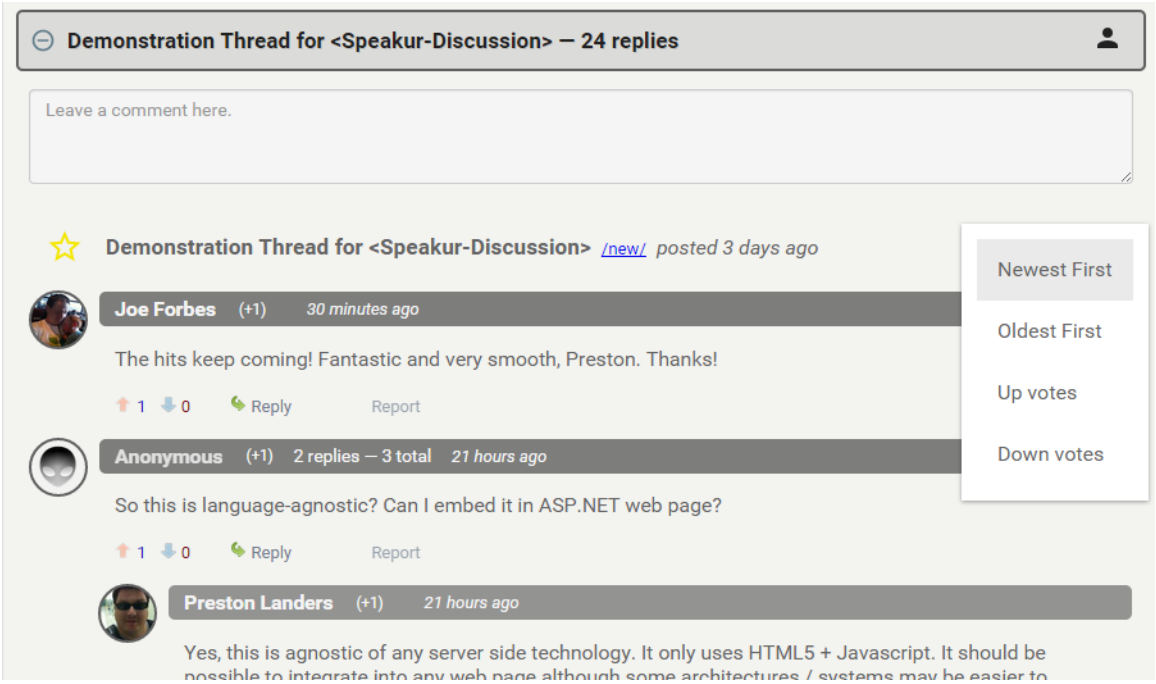

Figure 1.1: A Speakur thread inside a demonstration page.

gether on one big page. Hacks like the `<iframe>` tag have allowed one to work around some of these restrictions, but usually in an elegant and limited way.

At the time of HTML's introduction, the concept of quickly and easily composing a static web page, much less a full-fledged dynamic application, out of Lego-like reusable building blocks seemed like a distant dream at best. The introduction to web browsers of the Javascript [1] (JS) programming language in 1995 allowed for a completely new dimension of dynamic behavior that did not exist before. Eventually web apps like Google Docs rivaled traditional desktop applications in

[1] Javascript, also rendered as JavaScript or JS, has no significant relation to Sun's (now Oracle's) popular Java programming language; the name is an unfortunate coincidence at best.

functionality. Still, web apps had to be stitched together 'by hand' in ways that carefully ensured that the different parts worked together in perfect harmony or else disaster frequently ensued. Each component or area of the system could not help but be tied to the others at some level as a result of the programming model imposed by the DOM and HTML.

Over the years, going along with and perhaps helping to drive the growth of the web, Javascript grew in importance and a great many frameworks and libraries sprang up around the JS ecosystem to help manage this complexity and to provide structure to client side web apps. For many years individual JS frameworks seemed to be as ephemeral as teenage pop idols as the industry searched for the Next Big Thing that would make writing high quality web apps less of bug-ridden, messy chore. In recent years Google's Angular [CITE] has emerged as a dominant client framework, due in part to its perceived high quality [CITE] and the fact that it represents an common point for a fragmented industry to rally around. Facebook's React JS library with its Virtual DOM is an up-and-comer focused on high performance that is more complementary in nature to Angular than a true challenger.

Yet despite the recent successes of web frameworks like Angular and React in capturing developer attention, and the emergence of the updated HTML5 standard in 2011, there still did not exist a clear picture of how web apps could achieve the encapsulated component model that had become prevalent in other areas of software engineering. That is, until engineers from Google and Mozilla

[1] and other organizations got together to draft a new standard called Web Components that will extend and enhance HTML5 in interesting ways that could have unforeseen effects.

1.1 Web Component Overview

At its most fundamental level, the Web Component standard consists of 4 core DOM technologies that, if accepted by the World Wide Web (W3C) Consortium and web browser vendors, will eventually become native features provided by browsers and available to all web pages without any additional JS frameworks. The core Web Component technologies are:

- **Custom Elements**: extending HTML with new elements—custom HTML tags
- **Shadow DOM**: encapsulation for the internals of custom elements
- **Templates**: scaffolding for instantiating blocks of HTML from inert templates
- **Imports**: packaging for HTML components

This paper will explore related web standards initiatives that are frequently associated with Web Components but not formally grouped under them such as mutation observers, model driven views, and the CSS Flexible Boxes and CSS

[1] Sponsor of the popular Firefox web browser, this organization grew out Netscape whose Navigator browser helped first popularize the web.

Grid systems. Because these technologies are not yet formally accepted as a W3C standard and not yet widely implemented in commonly used mobile and desktop browsers, Speakur has been implemented with Google's experimental Polymer framework. Polymer provides a Javascript 'polyfill' library to implement many of the new Web Component features in currently available browsers. Eventually this platform polyfill should become unnecessary, in theory, as WC becomes widely adopted in browsers.

The potential componentization of the web is one of the most exciting developments in web engineering in years and follows the overall growth in software-as-a-service (SaaS) and service oriented architectures as a organizational and deployment concept. The conversion of dynamic web logic—not mere snippets of plain HTML—into bundles of reusable, extendable, composable components enables web developers to move to a higher level of abstraction than was previously possible.

The move towards a component-based Web will enable all sorts of interesting new composite services, mashups, and help broaden the potential pool of web developers. It will do this by allowing authors to publish easily reusable 'widgets' that can be easily juggled around and combined in novel ways that previously required a highly integrated (and hugely expensive) development model or lots of tedious glue code.

1.2 Structure of This Paper

The goal of this paper is to demonstrate the application of software engineering design patterns embodied in the W3C proposed Web Components standard such as encapsulation, composition, and automatic synchronization of application state. This paper attempts to explain many of the goals and principles of the Web Components initiative and show how a number of different technologies taken together help raise the overall level of abstraction for web authors and developers.

The Background section of this report provides an introduction some of the architectural problems inherent in modern web authoring and how Web Components (WC) attempts to address them. It also provides some background on software engineering design patterns that are embodied in Web Components such as encapsulation, composition, and inheritance, as well as technologies such as WebSockets and NoSQL databases. It describes some of the motivations behind the development of Speakur and some of the specific software engineering questions it addresses, such as the ability to provide a hassle-free way to host an embedded discussion forum inside an arbitrary web resource in a way that is fully encapsulated.

The Approach section details the specific structures and techniques used when constructing a Web Component, and describes the technology and software architecture choices that went into Speakur. It describes how Speakur uses Web Components to implement encapsulated functionality that is protected

The Implementation section describes the application of Web Component principles to the specific task of providing a flexible and suitably generic discussion forum / commenting system. It describes the overall architecture, code flow, and synchronization process. An important topic in this section is security: how can we implement a largely client-based system while maintaining some kind of data integrity?

This is followed by an Analysis section which discusses some of the outcomes as compared to the original goals and also looks at the impact of the selection of Web Components, Polymer, Firebase and some of the other architectural choices. A few quantitative results are included, I hope.

Finally, the Conclusion section is just all kinds of awesome and wraps up the paper.

1.3 Source Code and Demonstration Resources

The source code for Speakur consists of HTML and Javascript code located in two git version control repositories. The first repository is for the actual `<speakur-discussion>` HTML element that is available for use by web authors, and the second contains additional demonstrations, a standalone application and a management console.

The public documentation to help web authors use Speakur in their own sites can be found here:

`https://github.com/Preston-Landers/speakur-discussion`

Demonstrations of several web pages which show off embedded Speakur discussions are available at the following location:

`https://preston-landers.github.io/speakur-discussion/components/speakur-discussion-dist/demo.html`

Chapter 2

Background

When the Web was first created by Tim Berners-Lee in 1989, web pages were largely envisioned as single author documents, or at least written by a single cohesive team. The idea of gluing together a complex web application out of simpler components like snapping together Legos seemed like a far-off dream at best. The concept of creating your own HTML elements with customized behaviors seemed beyond the capabilities of the web browsers of the day like Mosaic and Netscape Navigator.

The rest of this text is just filler for now.

We are not going to look at the complete set of instructions contained in *Instructions for Preparation of Doctoral Dissertations and Dissertation Abstracts* or *Format For The Master's Thesis and Report* which can be obtained from the Office of Graduate Studies (OGS) or on their web page, `http://www.utexas.edu/ogs`. The doctoral Instructions I am using are dated March, 2001. The master's Format I am using is dated May, 2001.

Here we will look at a few instructions related to the arrangement of the dissertation, thesis, or report and a few other "technical" details, providing some examples of common LaTeX usage and some examples of not-so-common LaTeX

usage.

The following are just a couple of tests for the "quote" and "quotation" environments. The following paragraph is a quote.

> This template package is provided and licensed "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The following paragraph is a quotation.

> This template package is provided and licensed "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The OGS Instructions say prose quotations over four lines should be indented on the left. The Doctoral Degree Evaluator says the quote environment is the correct one to use.

2.1 Arrangement of Dissertation

Always remember that this "fake" dissertation is only intended to be a template for writing your own. Since the ultimate responsibility of making sure your dissertation meets the Graduate School's requirements, however, lies only with you, you ***must*** get the current *Instructions for Preparation of Doctoral Dissertations*

and Dissertation Abstracts from the Office of Graduate Studies or their web page and check everything yourself. If you don't, you may have a very rude awakening from the Lynn Renegar, Doctoral Degree Evaluator (aka, "The Ruler Lady") at a most inopportune time.

Arrange your dissertation as follows (all sections are required unless said otherwise.

1. Fly Page (blank protective page). This page is **not** counted in the numbering. **Note:** This template does not insert a Fly Page; if you are printing an official copy, you must manually insert a blank piece of paper on your own. Electronic documents do not need a fly page.

2. Copyright Legend (optional) - See OGS Instructions Sample Form A. Begin counting **pretext** pages here, but **do not place a number on this page.**

3. Committee Certification of Approved Version. See OGS Instructions Sample Form B. This page is included in the pretext count, but there should be no page number on the page.

4. Title Page - See OGS Instructions Sample Form C. This page is counted, but there should not be a page number on this page.

5. Dedication and/or Epigraph (optional). Included in count, but not numbered.

6. Acknowledgments or Preface (optional) - Begin showing **pretext** page numbers with **lower case Roman numerals** at bottom of page.

7. Abstract (optional) - See OGS Instructions Sample Form D.

8. Table of contents - List ALL sections which follow it. There are too may different ways a table of contents may be done for the OGS to give examples in their Instructions booklet, but do be sure there is agreement between the major headings in your text and their designations in the Table of Contents (fortunately LaTeX does this for you automatically). Please ask the Doctoral Degree Evaluator for assistance if necessary.

9. List of Tables, List of Figures, List of Illustrations, Nomenclature, List of Supplemental Files (such as multimedia files) (optional).

10. Text. The text should be divided into chapters, books or sections. The first page is Arabic numeral "**1**". All sections, **from the first page of text through Vita,** should be numbered consecutively.

11. If you group all Tables, Figures, or Illustrations in one place in your dissertation, the section should be placed here, immediately after the text and before any appendices (optional).

12. Appendix or Appendices (optional).

13. Glossary (optional) - this section may be placed either here or after the Table of Contents, in the area with List of Tables, List of Figures...

14. Bibliography - consult your supervisor about which recognized style to use.

15. Index (optional).

16. Vita - This should be a brief biographical sketch of the author. List in the Table of Contents. See OGS Instructions Sample Form E.

2.2 Other Requirements

2.2.1 Margins

The dissertation, after printing, should have left and top margins of 1 1/2 inches, and the right and bottom margins should be 1 1/4 inches. These margins should be consistent throughout the dissertation - including all pages in the appendix. **All page numbers must be *at least* one inch from the edges of the page**. Headers are rarely used in dissertations; if you are considering using them, check with the Doctoral Degree Evaluator first to be sure they will be accepted.

2.2.2 Spacing and Page Arrangement

The document should be double-spaced or space-and-a-half. Exceptions to double-spacing are: the Table of Contents, Lists of Tables, Tables, Figures, Graphs, Captions, Footnotes, Endnotes, Appendices, Glossary, Bibliography and Index; these may be single-spaced. Paragraph indentations are usually five to ten spaces. Prose quotations over four lines should be in block quote (double or single spaced, indented on the left). Do not use quotation marks if the quotation is indented except for quotations within the block quote. Please refer to a style manual for more detailed instructions.

Be sure that each new chapter or major section (i.e., Appendix, Bibliography, Vita) begins on a new page.

2.3 Master's Theses and Reports

Always remember that this "fake" thesis or report — assuming you have followed the instructions in the next chapter about how to format it as such — is only intended to be a template for writing your own. Since the ultimate responsibility of making sure your thesis or report meets the Graduate School's requirements, however, lies only with you, you ***must*** get the current *Format For The Master's Thesis and Report* from the Office of Graduate Studies or their web page and check everything yourself. If you don't, you may have a very rude awakening from the Mike Feissli, Master's Degree Evaluator at a most inopportune time.

That said, the formatting requirements for Dissertations and Reports and Theses are very similar. They are, however, ***not*** identical. The primary differences are in the ordering of the title and signature pages and where the optional index is inserted. For Master's Theses and Reports, the Title Page must be in front of the Signature Page. For Master's Theses and Reports, ***nothing*** is permitted to come between the bibliography and the vita; the index, if used, must be before the bibliography. If you want to use an index, talk with Mike Feissli before your deadline to verify that its inclusion is acceptable. The index can be removed by commenting out one line with a percent sign, if necessary, for producing the "official" copy of your thesis or report, and then inserted for copies for your advisor and you by removing the percent sign.

Chapter 3

How to Use the utdiss2 Package

3.1 Preamble

The preamble of the document starts like this:

```
\documentclass[12pt]{report}
\usepackage{utdiss2}
```

The first line declares "`report`" as the document class, with an option of 12pt for the character size, which is slightly greater that usual (the default is 10pt), but is what the Office of the Graduate School (OGS) recommends. You may include other options, as in any other LaTeX document.

The second line loads the `utdiss2` package, which contains a set of commands intended to produce a document fulfilling the official requirements for a doctoral dissertation or master's thesis or report. Besides that, you may include other packages. For instance:

```
\usepackage{amsmath,amsthm,amsfonts,amscd}
```

for mathematical symbols, or,

```
\usepackage{draftcopy}
```

to have a large "watermark" across each page of your document that says, "DRAFT."

The next few commands in the preamble are required.

`\author{Craig William McCluskey}` Replace my name in the command by your full, official University name. Make it combination of lower and upper-cases.

`\address{9905 Chukar Circle\\ Austin, Texas 78758}` Replace my address with your *permanent* (not local) address. Use `\\` to separate address lines.

`\title{Writing a Doctoral Dissertation with \LaTeX{} at the University of Texas at Austin}` Replace the name of this document in this command by your dissertation title. If the title consists of more than one line, it should be in inverted pyramid form. You may have to specify the line breakings by `\\` commands.

`\supervisor[Isaac Newton]{Johannes Kepler}` This document has two supervisors listed. See the source file (disstemplate.tex) for information on how to have only one supervisor. This command can be broken across lines as it is in the source file and as the `\committeemembers` command is shown below.

```
\committeemembers
        [Erwin Schr\"odinger]
        [Albert Einstein]
        [Charles Townes]
        {Arthur Schawlow}
```

This document shows four non-supervisor committee members. See the source file (disstemplate.tex) for information on how to have a different number.

`\previousdegrees{B.S.}` Replace B.S. with your previous degree.

The next few commands in the preamble are optional.

```
%\graduationmonth{...}
%\graduationyear{...}
%\typist{...}
```

Their use is documented in the source file.

At this point, if you are writing a master thesis or report you must use the optional `\degree` and `\degreeabbr` commands and specify

```
%\degree{MASTER OF ARTS}
%\degreeabbr{M.A.}
%\masterreport
%\masterthesis
```

as documented in the source file. By default the document is formated as a *dissertation*[1]

The default spacing for both text and quoted text is doublespaced. That can be changed with the following self-explanatory commands:

[1] The command `\dissertation` is also provided for symmetry.

```
\oneandonehalfspacing
\singlespacing
\oneandonehalfspacequote
\singlespacequote
```

Some versions of LaTeX in combination with some types of printers produce printed output that has incorrect vertical margins. The command `\topmargin 0.125` is provided to allow easy adjustment if it's needed.

If there are 10 or more sections, 10 or more subsections for a section, etc., you need to make an adjustment to the Table of Contents with the command `\longtocentry`. This command allocates the proper horizontal space for double-digit numbers.

3.2 Document

Next comes the actual text. It could be a sequence of chapters divided into sections, subsections, etc., all in the main file:

```
\chapter{...}     % The first chapter. The
                  % \chapter command is of the form
                  % \chapter[..]{..} or \chapter{..} where
    ... text ...  % [...] is the entry in table of contents
                  % and {...} is the chapter heading printed
                  % in the body of the document.
\section{...}     %
                  % IMPORTANT: If your chapter heading consists
```

```
                % of more than one line, it will be auto-
    ... text ...  % matically broken into separate lines.
                % If you don't like the way LaTeX breaks the
                % chapter heading into lines, however, use
\section{...}   % `\newheadline' command to break lines.
                % NEVER USE \\ IN SECTIONAL (E.G., CHAPTER,
    ... text ...  % SECTION, SUBSECTION, SUBSUBSECTION) HEADINGS!
                %
\chapter{...}   % This is Chapter 2.
    ... text ...
\section{...}
    ... text...
\subsection{...}
    ... more text ...
\subsubsection{...}
    ... more text ...
\appendix       % The appendix begins here.
% \appendices   % If more than one appendix chapters,
                % use \appendices instead of \appendix
\chapter{...}   % First appendix chapter, i.e., Appendix A.
\section{...}   % This is appendix section A.1.
    ................
```

Or, the chapters can be written in different files like this document and be loaded by `\include` commands:

```
\include{chapter-introduction}
\include{chapter-instructions}
\include{chapter-howtouse}
\include{chapter-makingbib}
\include{chapter-tables+figs}
\include{chapter-math}
\appendices
\index{Appendices@\emph{Appendices}}%
\include{chapter-appendix1}
\include{chapter-appendix2}
\include{chapter-appendix3}
```

Having the chapters in separate files makes the main .tex file simpler and allows chapters to be easily re-ordered (just swap the order of the include commands) or left (commented) out for draft copies.

Note: If you have only one appendix, in addition to using `\appendix` instead of `\appendices`, you must leave out the `\chapter` definition at the start of the appendix's text. Putting it in will cause the insertion of an extra page with only the word Appendix on it and will cause the appendix to be labeled Appendix 1, both of which are poor form if there is only one appendix.

If you are writing a short dissertation that does not require chapters, you need to use the command \nochapters just before the first section:

```
\nochapters


\section{...}     % First section.
    ... text ...
\section{...}     % Second section.
    ... text ...
        (...)
```

Next comes the bibliography. It can be made by hand like this:

```
\begin{thebibliography}{foo}
\bibitem ...
\end{thebibliography}
```

Or it can also be generated with BiBTEX, as explained in chapter 4.

Finally the vita is produced like this:

```
\begin{vita}
     % Insert your brief biographical sketch here.
     % Your permanent address and the name of the
     % typist(s) are generated automatically.
\end{vita}
```

Chapter 4

Making the Bibliography with BiBTEX

BiBTEX allows one to generate automatically the bibliography from a database of bibliographic items. You need to do the following:

1. Create the bibliographic database, which is a file whose name ends in `.bib`. Let us call it `diss.bib`. Entries in this file are like this:

```
@BOOK{knuth:tb,
  author = "Donald K. Knuth",
  title = "The \TeXbook",
  publisher = "Addison-Wesley",
  year = "1984",
}
@TECHREPORT{poorten:sp,
  author = "Alf~J.~van der Poorten",
  title = "Some problems of recurrent interest",
  institution = "School of Mathematics and Physics,
                 Macquarie University",
  address = "North Ryde, Australia 2113",
```

```
  number = "81-0037",
  month = "August",
  year = "1981",
}
@ARTICLE{erdos:oap,
 author = "Paul Erd{\"o}s and Paul Turan",
 title = "On a problem in the theory of uniform
          distribution, {I}",
 journal = "Indag. Math.",
 volume = "10",
 year = "1948",
 pages = "370--378",
}
```

2. Include a `\bibliographystyle` command in your LaTeX file, say

   `\bibliographystyle{plain}` and a `\bibliography` command to load the bibliography, in this case `\bibliography{diss}`, at the point of your document where the bibliography should be inserted.

   The document at this point will look like this:

   ```
   \bibliographystyle{plain}
   \bibliography{diss}
   ```

3. Run LaTeX on your main file, say `foo.tex`: `latex foo`. This generates an auxiliary file `foo.aux` with a list of `\cite` references.

4. Run BiBTEX on your file: `bibtex foo`. BiBTEX reads the auxiliary file, looks up the bibliographic database (`diss.bib`), and writes a `.bbl` file with the bibliographic information formated according to the bibliographic style file (`.bst`, say `plain.bst`) specified. Messages about resources used and error messages are written to a `.blg` file (in the case of this template, disstemplate.blg).

5. Run LATEX again: `latex foo`, which now reads the `.bbl` reference file.

6. Run LATEX for a third time: `latex foo`, resolving all references.

This includes all bibliographic items that have been cited in the document with a `\cite` command. In order to include non cited items in the bibliography, use the command `\nocite`. For example, `\nocite{knuth:tb}` anywhere in the document (after `\begin{document}`) includes in the bibliography the item with label `knuth:tb`. In order to include *all* items of the bibliographic database, use the command `\nocite{*}`.

Chapter 5

Making Tables and Including Figures

The *tabular* environment allows us to create complex tables and figures, and draw boundaries around and within it. The following example illustrates this:

Table 5.1: An example of a table.

Gegenwart		*Imperfekt*		*Perfekt*		
ich	bin	ich	war	ich	bin	gewesen
du	bist	du	warst	du	bist	gewesen
er		er		er		
sie	ist	sie	wart	sie	ist	gewesen
es		es		es		
wir	sind	wir	waren	wir	sind	gewesen
ihr	seid	ihr	wart	ihr	seid	gewesen
sie	sind	sie	waren	sie	sind	gewesen
Sie	sind	Sie	waren	Sie	sind	gewesen

Note: The assistance of Herr Professor Lothar Frommhold
in generating this table of German declensions
is gratefully acknowledged.

This table was created with the following sequence of commands:

```
\begin{table}[h]
\begin{center}
\caption{An example of a table.}
```

```
\vskip 10pt
\begin{tabular}{|ll|l|ll|l|lll|}
\cline{1-2} \cline{4-5} \cline{7-9}
\multicolumn{2}{|c|} {\textsl{Gegenwart}} & &
\multicolumn{2}{|c|} {\textsl{Imperfekt}} & &
\multicolumn{3}{|c|} {\textsl{Perfekt}} \\
\cline{1-2} \cline{4-5} \cline{7-9}
ich & bin  & & ich & war   & & ich & bin  & gewesen \\
du  & bist & & du  & warst & & du  & bist & gewesen \\
er  &      & & er  &       & & er  &      &         \\
sie & ist  & & sie & wart  & & sie & ist  & gewesen \\
es  &      & & es  &       & & es  &      &         \\
\cline{1-2} \cline{4-5} \cline{7-9}
wir & sind & & wir & waren & & wir & sind & gewesen \\
ihr & seid & & ihr & wart  & & ihr & seid & gewesen \\
sie & sind & & sie & waren & & sie & sind & gewesen \\
\cline{1-2} \cline{4-5} \cline{7-9}
Sie & sind & & Sie & waren & & Sie & sind & gewesen \\
\cline{1-2} \cline{4-5} \cline{7-9}
\end{tabular} \\[10pt]
Note: The assistance of Herr Professor Lothar Frommhold \\
in generating this table of German declensions \\
is gratefully acknowledged.
```

```
\vskip -20pt
\end{center}
\end{table}
\index{commands!environments!table}%
```

The argument `h` indicates the position for the table, in this case "here if possible". Other values of this argument are: `t` (top of the page), `b` (bottom of the page), `p` (on the page of floats) and `H` (HERE! - requires using the package float.sty. Note: When this option is used, LaTeX ignores all of its formatting rules and does what you say, putting the entire float exactly where it is defined. Check your output to make sure it is what you want! If you are having trouble with LaTeX wanting to put a figure that's larger than roughly half-a-page, as well as all of the figures following it, at the end of a chapter, try using the command `\clearpage` immediately following the large figure — and maybe a `\newpage` later.) It is possible to combine several arguments, such as `ht` ("here if possible, otherwise on top of the page"). The default is `tbp`.

Figure 5.1 is a typical example of inclusion of a figure contained in an encapsulated PostScript file. In order to use it, it is necessary to include the command `\usepackage{psfig}` at the beginning of the document.

You can see the commands that generated this figure in the source file. Look for the line `\begin{figure}[htb] % Imported eps example`.

The command that imports the file is `\psfig`, and it also controls its size (`height` and `width`), and can rotate the figure (`angle`).

Figure 5.1: An example of an imported jpg file.

Figures can also be drawn by using LaTeX commands. Figure 5.2 is an example (taken from [?]).

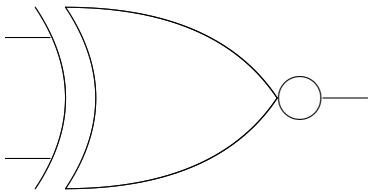

Figure 5.2: An example of a picture

The commands that generated this picture are in the source file following the line `\begin{figure}[htb] % Picture example.`

The commands used have rather obvious meanings. In particular, the command `\qbezier` draws a quadratic Bezier curve, defined by its two ending points, and a third point (whose coordinates are in the middle) that is used as control

point. Figure 5.3 illustrates the effect of the control point:

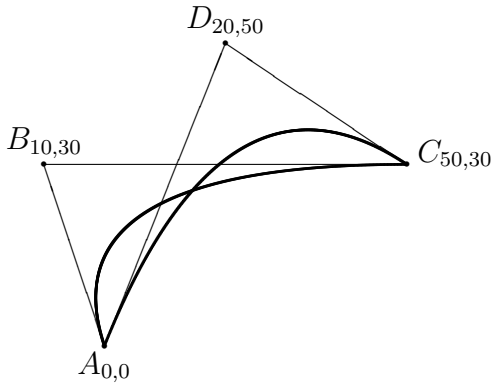

Figure 5.3: Bezier curves

This figure has been generated with the following commands:

```
\begin{figure}[htb] % Bezier curves example.
\begin{center}
   \setlength{\unitlength}{.8mm}
   \begin{picture}(55,55)(-15,0)
      \linethickness{1pt}
      \qbezier(0,0)(-10,30)(50,30)
      \qbezier(0,0)(20,50)(50,30)
      \thinlines
      \put(0,0){\line(-1,3){10}}
      \put(50,30){\line(-1,0){60}}
      \put(0,0){\line(2,5){20}}
      \put(50,30){\line(-3,2){30}}
      \put(0,0){\circle*{1}}
```

```
      \put(0,-1){\makebox(0,0)[t]{$A_{0,0}$}}
      \put(-10,30){\circle*{1}}
      \put(-10,31){\makebox(0,0)[b]{$B_{10,30}$}}
      \put(50,30){\circle*{1}}
      \put(58,29){\makebox(0,0)[b]{$C_{50,30}$}}
      \put(20,50){\circle*{1}}
      \put(20,51){\makebox(0,0)[b]{$D_{20,50}$}}
   \end{picture}
\caption{Bezier curves}
\label{f:qb}
\end{center}
\end{figure}
```

Chapter 6

An Example of Mathematical Writing

6.1 Generalized Fatou's Lemma

Here we show an application of the following lemma:

Lemma 6.1.1 (Generalized Fatou's Lemma)**.** *Let A be a Dedekind ring and F a rational series in $A[[X]]$, i.e., $F = p/q$ for some $p, q \in A[X]$. Then there exist two polynomials $P, Q \in A[X]$ such that $F = P/Q$, where P and Q are relatively prime and $Q(0) = 1$.*

Proof. See [**?**], p. 15, theorem 1.3. □

Theorem 6.1.2. *Let $\{c_n\}_{n=-\infty}^{\infty}$ a set of elements from K such that $c_n \in k'$ for every $n \geq n_0$, and verifying the following recurrence relation of order M:*

$$c_n \;=\; r_1\, c_{n-1} + r_2\, c_{n-2} + \cdots + r_M\, c_{n-M} \tag{6.1}$$

for every $n \in \mathbb{Z}$, where $r_1, r_2, \ldots, r_M$ are in K, $r_M \neq 0$. Then:

(i) The coefficients $r_1, r_2, \ldots, r_M$ are in k', and for every $n \in \mathbb{Z}$, $c_n \in k'$.

(ii) If $c_n \in \mathcal{O}_{k',v}$ for every $n \geq n_0$, then the coefficients $r_1, r_2, \ldots, r_M$ are all in $\mathcal{O}_{k',v}$.

Proof.

(i) Let C_n and R be the matrices:

$$C_n \;=\; \begin{pmatrix} c_n & c_{n+1} & \dots & c_{n+M-1} \\ c_{n+1} & c_{n+2} & \dots & c_{n+M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n+M-1} & c_{n+M} & \dots & c_{n+2M-2} \end{pmatrix} \tag{6.2}$$

and

$$R \;=\; \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ r_M & r_{M-1} & r_{M-2} & \dots & r_1 \end{pmatrix} \tag{6.3}$$

We have that $C_{n+1} \;=\; R\, C_n$. Since the recurrence relation is of order M, C_n is non singular. On the other hand, $R \;=\; C_{n+1}\, C_n^{-1}$. Since the elements of C_n are in k' for $n \;\geq\; n_0$, the entries of R, and those of R^{-1}, will be in k'. Since $C_{n-1} = R^{-1}\, C_n$, we get that the entries of C_n will be in k' also for $n < n_0$.

(ii) For each $t \geq n_0$ define the formal power series

$$F_t(X) \;=\; \sum_{n=0}^{\infty} c_{t+n}\, X^n \tag{6.4}$$

which is in $\mathcal{O}_{k',v}[[X]]$. We have $F_t(X) = p_t(X)/q(X)$, where $p_t(X), q(X) \in k'[X]$ are the following:

$$p_t(X) \;=\; \sum_{j=0}^{M-1} \Big(c_{t+j} - \sum_{i=1}^{j} r_i\, c_{t+j-i} \Big)\, X^j \tag{6.5}$$

$$q(X) \;=\; 1 - r_1\, X - r_2\, X^2 - \dots - r_M\, X^M \tag{6.6}$$

This can be checked by multiplying $F_t(X)$ by $q_t(X)$ and using the recurrence relation, which gives $F_t(X)\, q(X) = p_t(X)$ (see [**?**]).

Now we will prove that $p_t(X)$ and $q(X)$ are relatively prime. To do so, we will see that they cannot have any common root (in $\overline{k'}$). In fact, assume that α is a common root of $p_{t_0}(X)$ and $q(X)$ for some $t_0 \geq n_0$, i.e.: $p_{t_0}(\alpha) = q(\alpha) = 0$. Since $q(0) = 1$, then $\alpha \neq 0$. Now we have:

$$X\, F_{t_0+1}(X) = F_{t_0}(X) - c_{t_0} \tag{6.7}$$

so:

$$\begin{aligned} X\, p_{t_0+1}(X) &= X\, q(X)\, F_{t_0+1}(X) \\ &= q(X)\,(F_{t_0}(X) - c_{t_0}) = p_{t_0}(X) - c_{t_0}\, q(X) \end{aligned} \tag{6.8}$$

Hence $p_{t_0+1}(\alpha) = 0$, which means that α is also a root of $p_{t_0+1}(X)$. By induction we get that $p_t(\alpha) = 0$ for every $t \geq t_0$. Grouping the terms of $p_t(X)$ with respect to $c_t, c_{t+1}, \ldots, c_{t+M-1}$, we get:

$$p_t(X) = \sum_{j=0}^{M-1} a_j(X)\, c_{t+j} \tag{6.9}$$

where

$$a_j(X) = X^j \left(1 - \sum_{i=1}^{M-j-1} r_i\, X^i\right) \tag{6.10}$$

Note that $a_0(X), a_1(X), \ldots, a_{M-1}(X)$ do not depend on t. On the other hand $p_t(\alpha) = 0$ implies

$$\sum_{j=0}^{M-1} a_j(\alpha)\, c_{t+j} = 0 \tag{6.11}$$

for every $t \geq t_0$. Note that $a_{M-1}(\alpha) = \alpha^{M-1} \neq 0$, so $a_0(\alpha), a_1(\alpha), \ldots, a_{M-1}(\alpha)$ are not all zero, and (6.11) means that the columns of the matrix C_{t_0} are linearly

dependent, so $\det C_{t_0} = 0$, which contradicts the fact that C_{t_0} is non singular. Hence, the hypothesis that $p_t(X)$ and $q(X)$ have a common root has to be false. This proves that $p_t(X)$ and $q(X)$ are relatively prime.

By (generalized Fatou's) lemma 6.1.1, and taking into account that $\mathcal{O}_{k',v}$ is a Dedekind ring, we get that there exist two relatively prime polynomials $P_t(X)$ and $Q_t(X)$ in $\mathcal{O}_{k',v}[X]$ such that $F_t(X) = P_t(X)/Q_t(X)$ and $Q_t(0) = 1$. Hence: $p_t(X)\,Q_t(X) = q(X)\,P_t(X)$. By unique factorization of polynomials in $k'[X]$, there is a $u \in k'$ such that $P_t(X) = u\,p_t(X)$ and $Q_t(X) = u\,q_t(X)$. Since $Q_t(0) = q(0) = 1$, we get that $u = 1$, so $P_t(X) = p_t(X)$ and $Q_t(X) = q(X)$. Hence, the coefficients of $q(X)$ are in $\mathcal{O}_{k',v}$.

□

6.2 Other Examples of Mathematical Writing

6.2.1 An Example of a Commutative Diagram

The following is an example of a commutative diagram. It requires the `amscd` package.

$$\begin{array}{ccc} S^{\mathcal{W}_\Lambda} \otimes T & \xrightarrow{\ j\ } & T \\ \Big\downarrow & & \Big\downarrow{\scriptstyle \operatorname{End} P} \\ (S \otimes T)/I & = & (Z \otimes T)/J \end{array}$$

That diagram has been made with the following commands:

```
\newcommand{\End}{\operatorname{End}}
```

```
\begin{CD}
S^{{\mathcal{W}}_\Lambda}\otimes T   @>j>>   T\\
@VVV                                    @VV{\End P}V\\
(S\otimes T)/I                  @=      (Z\otimes T)/J
\end{CD}
```

6.2.2 Using AMS Fonts

To use AMS fonts it is necessary to choose from an assortment of LaTeX packages. For instance the command `\usepackage{amsfonts}` calls in the *amsfonts* package, which provides blackboard bold letters (e.g. $\mathbb{R}$) and some math symbols. A superset of that package is *amssymb*. Other packages are *eufrak* for Frankfurt letters (e.g. $\mathfrak{R}$) and *eucal* for Euler script (e.g. $\mathcal{R}$). Consult the LaTeX documentation about this subject for additional information.

Appendices

Appendix A

Lerma's Appendix

The source LaTeX file for this document is no longer quoted in its entirety in the output document. A LaTeX file can include its own source by using the command `\verbatiminput{\jobname}`.

Appendix B

My Appendix #2

B.1 The First Section

This is the first section. This is the second appendix.

B.2 The Second Section

This is the second section of the second appendix.

B.2.1 The First Subsection of the Second Section

This is the first subsection of the second section of the second appendix.

B.2.2 The Second Subsection of the Second Section

This is the second subsection of the second section of the second appendix.

B.2.2.1 The First Subsubsection of the Second Subsection of the Second Section

This is the first subsubsection of the second subsection of the second section of the second appendix.

B.2.2.2 The Second Subsubsection of the Second Subsection of the Second Section

This is the second subsubsection of the second subsection of the second section of the second appendix.

Appendix C

My Appendix #3

C.1 The First Section

This is the first section. This is the third appendix.

C.2 The Second Section

This is the second section of the third appendix.

Vita

Preston Brent Landers was born in Texas and attended high school on the Nevada side of Lake Tahoe. He received his Bachelor of Arts in English from the University of Texas at Austin. He works as a web and mobile software engineer for Journyx, Inc.[*] in Austin, Texas and began graduate studies in Software Engineering at the University of Texas at Austin in August 2012.

Permanent address: planders@utexas.edu

This report was typeset with LaTeX[†] by the author.

[*] http://www.journyx.com

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.