

Modeling Human Workload in Unmanned Aerial Systems

TJ Gledhill

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Michael A. Goodrich, Chair
Kevin Seppi
Eric M. Mercer

Department of Computer Science
Brigham Young University
April 2014

Copyright © 2014 TJ Gledhill
All Rights Reserved

ABSTRACT

Modeling Human Workload in Unmanned Aerial Systems

TJ Gledhill

Department of Computer Science, BYU
Master of Science

Unmanned aerial systems (UASs) often require multiple human operators fulfilling diverse roles for safe correct operation [1–3]. Although some dispute the utility of minimizing the number of humans needed to administer a UAS [4], minimization remains a long-standing objective for many designers. Reliably designing the human interaction, autonomy, and decision making aspects of these systems requires the use of modeling. We propose a conceptual model which models human machine interaction systems as a group of actors connected by a network of communication channels. We also propose a workload taxonomy derived from a review of the relevant literature which we then apply to the conceptual model. We present a simulation framework implemented in Java, with an optional XML model parser, which can be analyzed using the Java Pathfinder (JPF) model checker. The simulator produces a workload profile over time for each human actor in the system. We conducted case studies by modeling two different UAS. Wilderness search and rescue using a UAV (WiSAR) and UAS integration into the national air space (NAS). The results of these case studies, while inconclusive, are consistent with known workload events and the simple workload metric presented by Wickens.

Keywords: human workload, unmanned aerial system, uas, national air space, unmanned aerial vehicle, modeling human machine interaction

ACKNOWLEDGMENTS

The authors would like to thank the NSF IUCRC Center for Unmanned Aerial Systems, and the participating industries and labs, for funding the work. The authors would like to thank Neha Rungta of NASA Ames Intelligent Systems Division for her help with JPF and Brahms. The authors would also like to thank the NSF IUCRC Center for Unmanned Aerial Systems, and the participating industries and labs, for funding the work. Further thanks go to Jared Moore and Robert Ivie for their help coding the model and editing this paper.

Table of Contents

List of Figures	vii
List of Tables	ix
List of Listings	xi
1 Introduction	1
1.1 Class Options	1
1.2 Preamble Setup Commands	5
2 Parting Shots	7
3 Modeling	9
3.1 Making modeling easier	9
References	11

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

To use this package, you must first declare it as your document class

```
\documentclass[<options>]{byumsphd}
```

then put the `\maketitle` command at the beginning of your document and add the remaining content. This class is meant to ensure that the last part really is the hard part; nobody wants to go through years of research and hard work only to discover that their margins are next to impossible to get right.

The class declaration is, of course, not all that is required. Several things must be specified in the preamble of your document before the appropriate preface material can be generated. Each of these is discussed within its own section.

1.1 Class Options

This class has several options that affect document output in various ways. These are described in detail here.

- Document Type:
 - **phd (default)**: Produce a dissertation
 - **ms** Produce a thesis
 - **electronic**: Make an ETD (sets equalmargins, oneside)
 - **hardcopy**: Make bindable (sets no equalmargins, twoside)

- Font Size:
 - **10pt**
 - **11pt**
 - **12pt (default)**
- Preamble Settings:
 - **lot**: Produce a List of Tables (default off)
 - **lof**: Produce a List of Figures (default off)
 - **lol**: Produce a List of Listings (default off)
- Page Formatting:
 - **equalmargins**: Make all margins equal - no binding offset
 - **twoside (default)**: Alternate margins for even and odd pages
 - **oneside**: One-sided document - margins don't alternate
 - **twosidetoc**: Start two-sided margins at the TOC, don't wait for the document body and arabic numerals to do it. This is a strange requirement that I have seen in some places, especially strange since it means that the page numbering will change suddenly in the middle of the roman numerals (the first four full sheets might have numbers i-iv, but not printed, and the next sheets will each have two page numbers; most people probably won't notice, but it's still a weird requirement).
 - **openright (default)**: Chapters start on an odd page
 - **openany**: Chapters start anywhere (tip: don't use this)
 - **duplexprinter**: If printing two-sided (to save trees while evaluating the state of things, for example), this ensures that a blank page is printed when necessary between the front matter and the first chapter, ensuring that the first chapter

starts on the front of a sheet (the right side). It is important to note that you should *not* plan on using this to generate the final document. Paper that is required for theses is typically much heavier and more opaque than paper you are going to have in your department printer. Print everything one-sided so that text from the back of a page doesn't bleed through when getting copied for the final draft. Here is how I suggest you print the final draft:

- * Print the entire document *without* this option (but *with* the twoside option so that page margins alternate correctly).
- * Take it to the copy center and say the word "thesis" to someone behind the counter.
- * Show her the front matter and say "one-sided". A sticky note is likely to be applied.
- * Show her the rest and say "two-sided".
- * Smile: she has your baby in her hands.
- * Pay and wait until it's done.

- Miscellaneous Layout:

- **prettyheadings (default)**: Allow larger font sizes for chapter, part, and section headings. If you can get away with it, use this.
- **compliantheadings**: Use the same font size for everything – ugly, but compliant with the minimum style requirements.
- **simplechapterheading**: Chapter headings are "simple", meaning that they are of the format "1 Title" instead of "Chapter 1(newline)Title".
- **chapterleft (default)**: Chapter headings are left justified
- **chaptercenter**: Chapter headings are centered
- **partmiddle (default)**: Part headings are vertically centered

- **parttop**: Part headings are near the top of the page, appropriate if each part contains introductory text.

- Visualization Aids (all default to off):
 - **layout**: Display dotted lines for help with layouts. Shows the margins (helpful for finding overfull hboxes, etc.)
 - **grid**: Display a 1/2-inch grid on every page. When printing your document, use this on one of the pages to find out whether the printer shifts things down or not. My printer shifts everything down by 0.03 inches, which doesn't sound like a lot, but can make a huge difference, especially considering that paper is not actually 8 1/2 by 11, but a little smaller than that in each dimension.
 - **separator**: Output an extra instruction page in between the preamble and the body of the document. Even if the document is two-sided, the preamble *never* is, so this page reminds you or the printing service that this is the case while giving appropriate instructions for the remainder of the document. It *will not print* if the document is all one-sided (there's no point, really), and it should *never* be included in your final copies of the document, but discarded during printing. If it is not discarded, it is probably harmless since you can just throw it away after getting copies made.

1.2 Preamble Setup Commands

In order to produce a correct preamble, several things must be set, and others can optionally be added. The settings described next affect the behavior of `\maketitle`. These are required:

- `\title{<your document title>}`
- `\committeechair{<your adviser's name>}`
- `\committeemembera{<second member>}`
- `\committeememberb{<third member>}`
- `\committeememberc{<fourth member (PhD only)>}`

- `\committeememberd{<fifth member (PhD only)>}`
- `\monthgraduated{<month of graduation>}`
- `\yeargraduated{<hopefully this year>}`
- `\documentabstract{<your thesis abstract goes here>}`

The following are optional or have defaults suitable for CS majors:

- `\yearcopyrighted{<defaults to year graduated>}`
- `\acknowledgments{<the text of your acknowledgments goes here>}`
- `\university{<default: Brigham Young University>}`
- `\department{<default: Computer Science>}`
- `\graduatecoordinator{<default: Parris K. Egbert>}`
- `\college{<default: Physical and Mathematical Sciences>}`
- `\collegedean{<default: Thomas W. Sederberg>}`
- `\collegedeantitle{<default: Associate Dean>}`
- `\settextwidth{<default: 6in>}`
- `\setbindingoffset{<default: 0.5in (for non-equalmargins)>}`

Chapter 2

Parting Shots

If you are interested in getting a handle on what the class is doing, read the source; it is heavily commented and frankly one of the easiest-to-understand classes I have ever seen. It uses a minimum of deep \TeX trickery to accomplish its designs.

If you would like an enhancement made to it, you are welcome to do it yourself or to contact the author; information is available in the class file comments. License information is also available there, but boils down to basically this: you can do what you like with this file, but if you change it, change the name. If you have a change that you think belongs in the main distribution, contact the author and it will most likely be included in the official class file.

Please enjoy, and note all kinds of feedback are welcome, especially the kinds that involve deposits into the author's bank account.

Chapter 3

Modeling

By far the most challenging aspect of this work was the creation of the models. By constructing the simulation framework as a set of Java interfaces and abstract classes it meant that the framework was almost entirely extensible in any direction. We found this very refreshing during the first few development iterations of the framework and model. We created different Actor types and different sub-Actors within those types. Each transition used an anonymous method containing logic for enabling the transition. Eventually as our model grew in size and complexity the freedom of the Java language became a dual edged sword. Our model was full of implicit declarations, inconsistencies, duplicated code, and anonymous methods. It became extremely difficult to maintain the model let alone add to it.

3.1 Making modeling easier

To help improve the modeling experience we chose to

References

- [1] M. L. Cummings, C. E. Nehme, J. Crandall, and P. Mitchell. *Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles*, volume 70 of *Studies in Computational Intelligence*, pages 11–37. Springer, 2007.
- [2] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey. Supporting wilderness search and rescue using a camera-equipped mini UAV. *Journal of Field Robotics*, 25(1-2):89–110, 2008.
- [3] R. Murphy, S. Stover, K. Pratt, and C. Griffin. Cooperative damage inspection with unmanned surface vehicle and micro unmanned aerial vehicle at hurrican Wilma. IROS 2006 Video Session, October 2006.
- [4] R. R. Murphy and J. L. Burke. The safe human-robot ratio. In M. Barnes and F. Jentsch, editors, *Human-Robot Interaction in Future Military Operations*, chapter 3, pages 31–49. Ashgate Publishing, 2010.