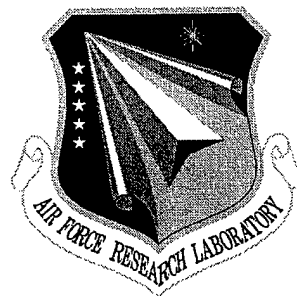


AFRL-IF-RS-TR-1998-96
Final Technical Report
June 1998



EVALUATION METHODS FOR COMPLEX INTELLIGENT INFORMATION SYSTEMS (EMCIIS)

Sterling Software, Inc.

Paul Losiewicz, Kasey Dingman, and Adele E. Howe

19980713 012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

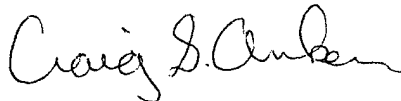
**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

DTIC QUALITY INSPECTED 1

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1998-96 has been reviewed and is approved for publication.

APPROVED:



CRAIG S. ANKEN
Project Engineer

FOR THE DIRECTOR:



NORTHROP FOWLER, III, Technical Advisor
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFTB, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1998		3. REPORT TYPE AND DATES COVERED Final Jun 95 - Dec 97
4. TITLE AND SUBTITLE EVALUATION METHODS FOR COMPLEX INTELLIGENT INFORMATION SYSTEMS (EMCIIS)			5. FUNDING NUMBERS C - F30602-95-C-0257 PE - 62702F PR - 5581 TA - 27 WU - 93	
6. AUTHOR(S) Paul Losiewicz, Kasey Dingman, Adele E. Howe				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Prime Contractor: Sterling Software, Inc. Information Technology Division Beeches Technical Campus, Rt 26N Rome NY 13441-4909 Subcontractor: University of Massachusetts Department of Computer Science Box 34610, LGRC Amherst MA 01003-4610			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFTB 525 Brooks Road Rome NY 13440-4505			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-1998-96	
11. SUPPLEMENTARY NOTES Air Force Research Laboratory Project Engineer: Craig S. Anken/IFTB/(315) 330-4833				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The objective of the EMCIIS effort was to perform research and development studies and experiments to demonstrate improvements in methods for evaluating the behavior and performance of Complex Intelligent Information Systems (CIIS). The critical goals achieved during the life of this effort included the following: *Make evaluation methods and tools accessible to CIIS researchers and developers through an Evaluation of Information Systems (EIS) on-line web site. The hypertext accessible methods were developed using Hypertext Markup Language (HTML) and were made available through the World Wide Web (WWW). *Develop a Hypertext CIIS Evaluation Field Guide. The hypertext Field Guide provided tools for the evaluation of information systems and was made available through the WWW. In addition, the effectiveness of the Field Guide as an aid to CIIS researchers, developers, and evaluators was assessed. *Advance the state of evaluation methods and research in support of CIIS incorporation into C4I and other systems. The primary product is the EIS WWW site which provides a centralized location for AI researchers seeking information on and assistance with, AIL systems evaluation. Development of the site and its on-line tools was carried out by Sterling Software, the University of Massachusetts (UMASS), and Colorado State University (CSU).				
14. SUBJECT TERMS Intelligent Information Systems, Evaluation, Diagnosis, Tools			15. NUMBER OF PAGES 40	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

SECTION 1 INTRODUCTION.....	1
1.1 Identification.....	1
1.2 Contract Overview.....	1
1.3 Document Overview	1
SECTION 2 THE EVALUATION OF INFORMATION SYSTEMS SITE, PRESENT AND FUTURE..	3
The Present.....	3
The Future	11
SECTION 3 EIS SITE STRUCTURE AND MAINTENANCE.....	12
3.1 Site Location	12
3.2 Site Maintenance Recommendations	12
3.3 Contract Completion Issues.....	13
SECTION 4 SOFTWARE INVENTORY AND DELIVERABLES	14
4.1 A001 – Program Progress Report	14
4.2 A002 – Contract Funds Status Report.....	14
4.3 A003 – Technical Information Report	14
4.4 A004 – Technical Information Report	14
4.5 A005 – Technical Information Report	14
4.6 A006 – Technical Information Report	14
4.7 A007 – Presentation Materials.....	14
4.8 A008 – Final Scientific and Technical Report	15
4.9 Software Inventory.....	15
SECTION 5 ACRONYMS	16
APPENDIX A COLORADO STATE UNIVERSITY’S FINAL SUBCONTRACTOR REPORT	17
EVALUATION METHODS FOR COMPLEX, INTELLIGENT INFORMATION	
DRIVEN SYSTEMS	

LIST OF FIGURES

Number	Title	Page
1	EIS Home Page.....	3
2	EIS Field Guide – Selection of Experiment Type.....	5
3	Description of Experiment Type	6
4	Exploration of the Data.....	7
5	On-line Tools for Analysis (Colorado State University)	8
6	Description of Test and Modeling Methods (PDF).....	9
7	Advice Corner	10

SECTION 1 INTRODUCTION

1.1 Identification

This Final Scientific and Technical Report was prepared by Sterling Software for the Air Force Research Laboratory (AFRL), Information Directorate at Rome Research Site, Information Technology Division, Intelligent Information Systems Branch EMCIIS Program Management Office under Contract F30602-95-C-0257 entitled "Evaluation Methods for Complex Intelligent Information Systems (EMCIIS)." This document fulfills Contract Data Requirements List (CDRL) Data Item Number A008 of Contract Line Item Number (CLIN) 0002, Final Scientific and Technical Report.

1.2 Contract Overview

The objective of the EMCIIS effort was to perform research and development studies and experiments to demonstrate improvements in methods for evaluating the behavior and performance of Complex Intelligent Information Systems (CIIS). The critical goals achieved during the life of this effort included the following:

- a. Make evaluation methods and tools accessible to CIIS researchers and developers through an Evaluation of Information Systems (EIS) on-line web site. The hypertext accessible methods were developed using Hypertext Markup Language (HTML) and were made available through the World Wide Web (WWW).
- b. Develop a Hypertext CIIS Evaluation Field Guide. The hypertext Field Guide provided tools for the evaluation of information systems and was made available through the WWW. In addition, we attempted to assess the effectiveness of the Field Guide as an aid to CIIS researchers, developers, and evaluators.
- c. Advance the state of evaluation methods and research in support of CIIS incorporation into Command, Control, Communications, Computers and Intelligence (C4I) and other systems.

The primary product is the EIS WWW site which provides a centralized location for Artificial Intelligence (AI) researchers seeking information on, and assistance with, AI systems evaluation. Development of the site and its on-line tools was carried out by Sterling Software, the University of Massachusetts (UMASS), and Colorado State University (CSU).

1.3 Document Overview

The EMCIIS Final Scientific and Technical Report has been organized into six sections.

Section 1, Introduction, identifies and describes the objective of the EMCIIS program.

Section 2, The Evaluation of Information Systems (EIS) Site, Present and Future provides a description of the capabilities of the EIS site and recommendations for future support.

Section 3, Site Structure and Maintenance, describes the site location and actions necessary to maintain the EIS web site at its current level.

Section 4, Software Inventory and Contract Deliverables, provides an annotated list of all software and deliverables provided to the government under this effort.

Section 5, Acronyms, lists and defines acronyms and abbreviations used throughout this document.

Appendix A, Final Subcontractor Report, Evaluation Methods for Complex, Intelligent Information Driven Systems, Colorado State University's final report.

SECTION 2 THE EVALUATION OF INFORMATION SYSTEMS SITE, PRESENT AND FUTURE

The Present

The Evaluation of Information Systems (EIS) site has a unique collection of on-line tools and resources¹ to assist project managers, researchers, and programmers in the design of evaluative procedures and methods for the software systems they wish to evaluate. Such a site is envisioned to provide DoD project managers a common reference point for statistical procedures, metrics and experimental design advice for software evaluation. The EIS home page is shown in Figure 1.

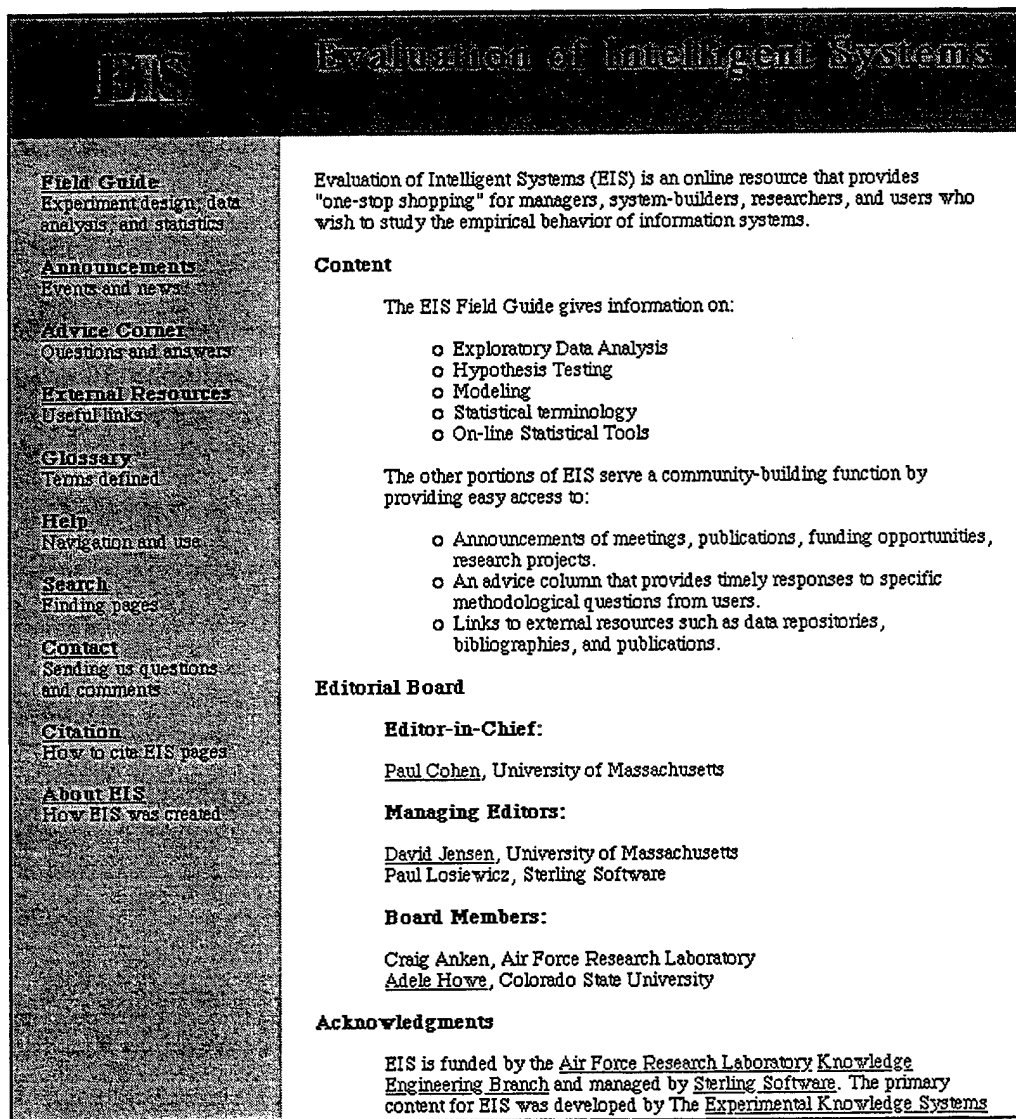


Figure 1. EIS Home Page

¹ As verified in CIIS Evaluation Test Cases and Analysis Report CDRL A004.

The EIS site currently includes:

- 460 distinct items (including HTML pages, GIF images, PDF files, and directories)
- The Evaluation of Information Systems Field Guide
- Advice Corner
- External Resources
- Glossary
- Help
- Announcements
- Search

Additional information is provided on *Evaluation Methods for Artificial Intelligence (EMAI)*, as well as citation methods and contact information. The primary content of the site is provided by the Field Guide. Much of the EIS Field Guide was constructed using substantial portions of Paul Cohen's *Empirical Methods for Artificial Intelligence*, MIT Press, 1995. These portions are used with permission from MIT Press. Material from EMAI is reproduced with modifications to make it suitable for use without the context of the original printed book. In general, deletions remove references to sections surrounding the original section in the book. Additions include necessary details from those surrounding sections.

The Field Guide provides an on-line interactive resource for empirical, experimental evaluation of information systems software.

The Field Guide provides:

- a. 12 experiment types falling under three basic categories: Single System Performance Examination; Multiple System Comparison; and Environmental Factor-based Performance Explanation
- b. 36 evaluative techniques or methods (e.g., three-group resistant line or parametric confidence interval for the mean. The techniques cover: Measures of Central Tendency; Measures of Dispersion; Constructing and Interpreting Scatterplots; Constructing and Interpreting Histograms; Constructing and Interpreting Contingency Tables; Calculating Covariance and Correlation; Randomization Test of Correlation; Three-Group Resistant Line; Calculating Cross-Correlation; r-to-z Transform; Monte Carlo Test (general); Randomization (general); One-Sample t Test; Two-Sample t Test; Paired-Sample t Test; and Constructing Power Curves.

The EIS Field Guide provides information on designing experiments and analyzing the data they produce. For each of several basic experiment types, the Field Guide describes techniques for data preparation, data exploration, hypothesis testing, and modeling. The content of the Field Guide is organized by experiment type. To choose the most appropriate type, you can answer several simple questions from an experiment type Advisor. Alternatively, for the advanced user who may be already familiar with the experiment types, you can browse a List. The description of each technique includes details about its application, as well as warnings about potential pitfalls, suggested follow-up procedures, and definitions of relevant terms. The techniques are linked to warnings and follow-up suggestions and a large technical glossary. The glossary contains crosslinks to other glossary terms and back to the Field Guide.

For any given experiment type, EIS briefly describes the type and presents several analysis options (Figure 2). EIS provides tools for data exploration and analysis. Wherever possible, on-line tools will be provided in preference to off-line packages. For each experiment type, EIS provides visual displays that help explore relationships present in the data. Such exploratory data analysis (EDA) helps users identify complex relationships often missed by statistical tests. They can also suggest follow-up analyses that reveal subtle details of a more general relationship.

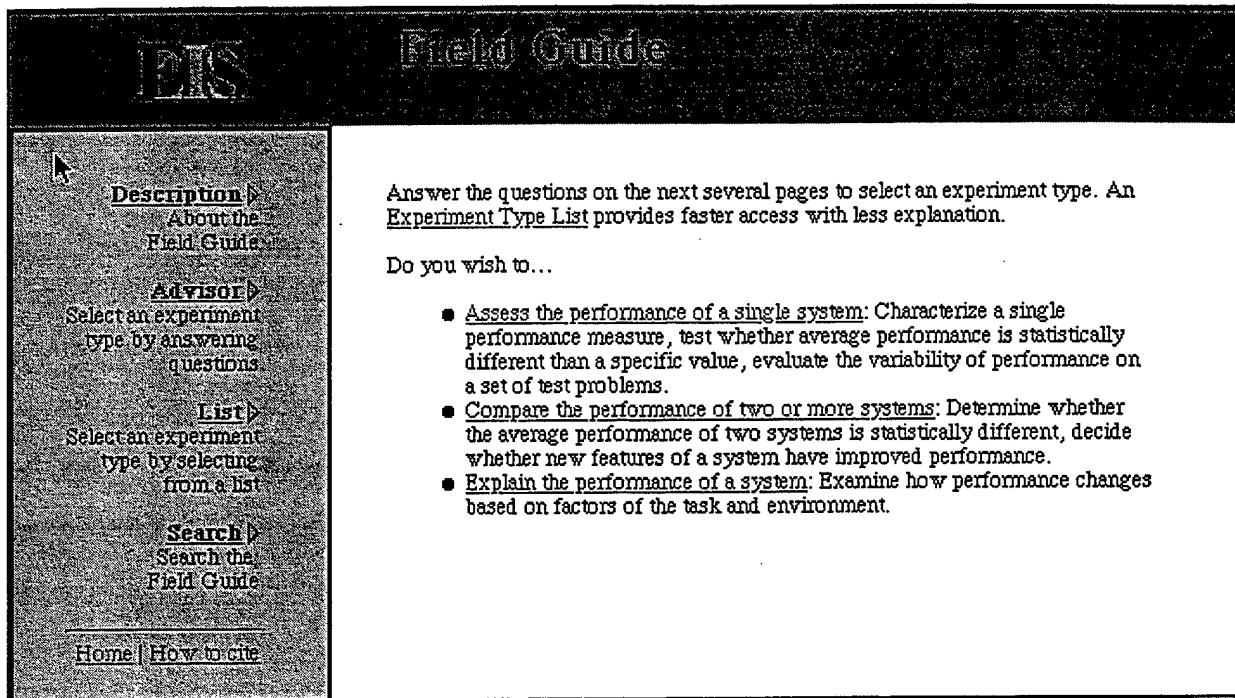


Figure 2. EIS Field Guide – Selection of Experiment Type

For each analysis option, EIS provides a description (Figure 3), and methods for exploration and visualization of the data (Figure 4). Additionally, comprehensive added support which so far has been determined to be unique to EIS is provided, such as on-line software, warnings and advice, details about interpreting and following up on the analysis, and a glossary of relevant terms. Each hypothesis test and modeling technique in EIS is described in relatively non-technical language. When technical terms are required, they are linked to the Glossary. Results of the on-line statistical tools are provided in standard statistical language. Novice users can obtain assistance with interpretation using separate information.

Warnings indicate potential pitfalls in using particular techniques for hypothesis testing or modeling. Many statistical procedures contain important assumptions, some of which are not obvious to many users. Each warning discusses the statistical basis of the assumption and the potential impact of ignoring it. In the future, this section will be dynamically linked to a database that will incorporate contributions from the evaluation community.

Follow-up indicates the next steps a user can take, given particular results from a statistical hypothesis test or modeling technique. Follow-up helps create a chain of experimental procedures and statistical inferences, rather than one-shot analysis.

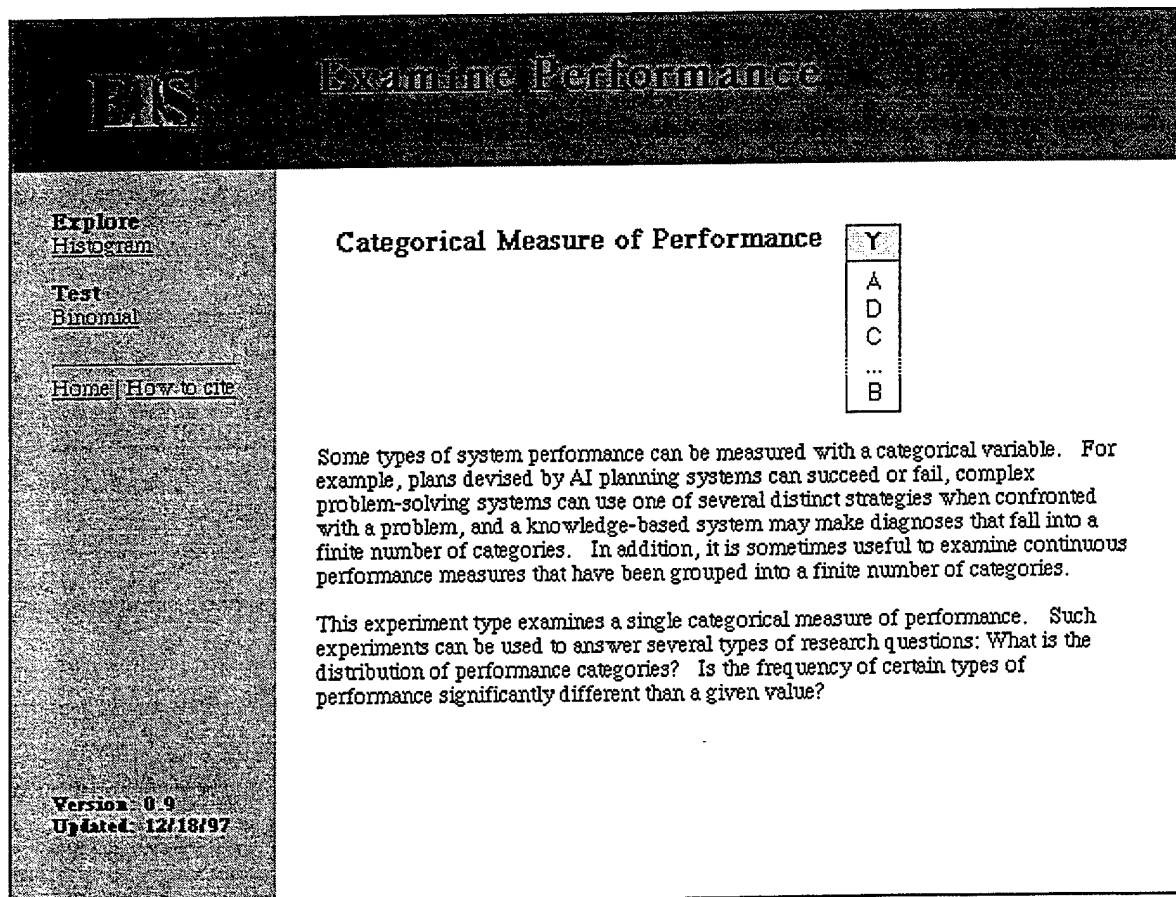


Figure 3. Description of Experiment Type

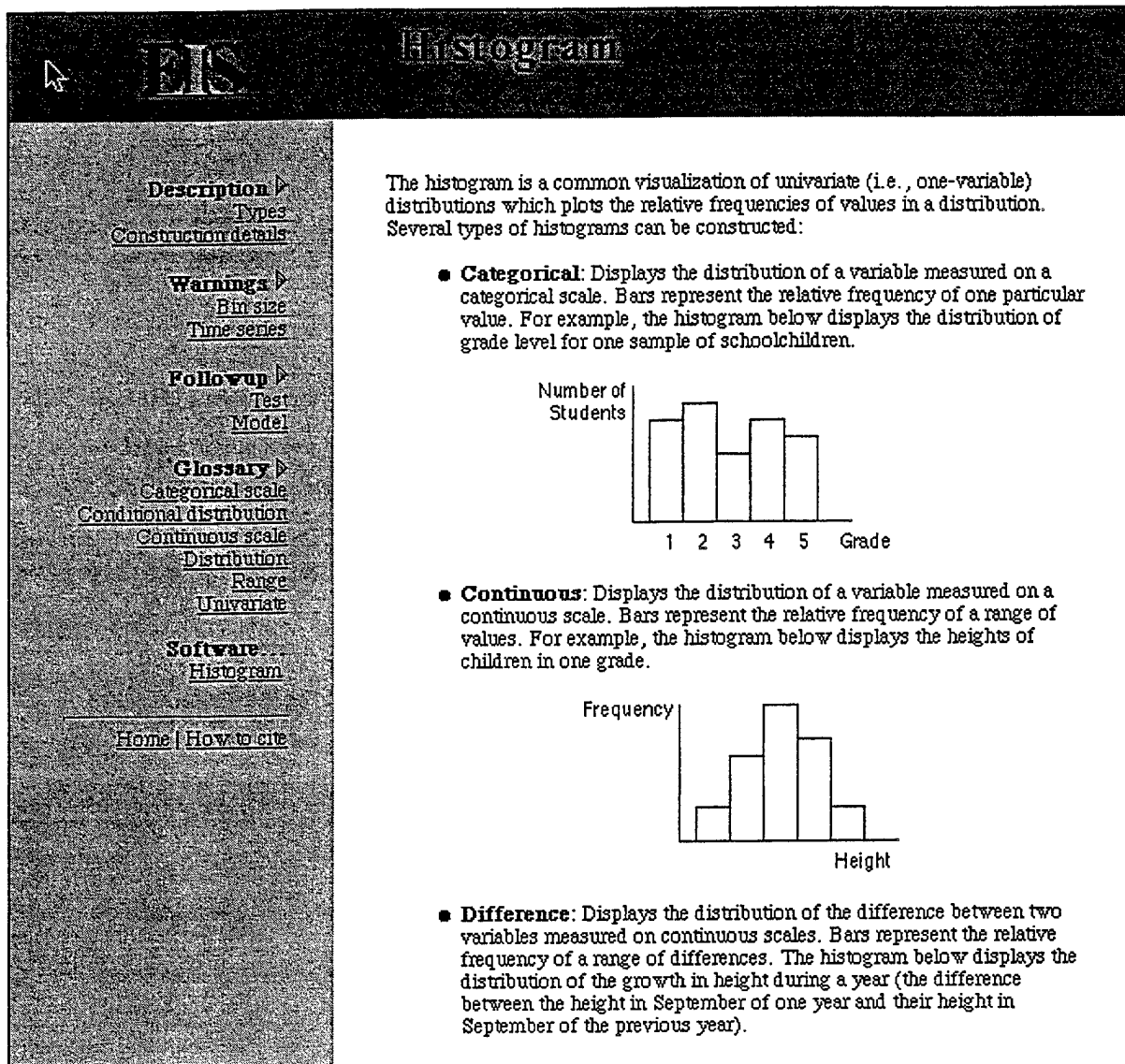


Figure 4. Exploration of the Data

The on-line tools (Figure 5) provide more than just statistical calculations. They will provide users the ability to explore their data interactively. Lastly, Test and Modeling descriptions (Figure 6) are provided for each type of statistical hypothesis test and modeling technique. They indicate what follow-on inferences a user can make and test for based on the outcomes of particular tests or modeling techniques.

The glossary provides an on-line dictionary of statistical terms. We are constructing the initial version from terms in Empirical Methods for Artificial Intelligence (EMAI). Other terms will be added in response to user requests and questions.

The screenshot shows the CLASPWeb Histogram tool interface. At the top left is the CLASP logo, and next to it is the title "CLASPWeb Histogram". Below the title is a horizontal line. The main heading is "Histogram". Underneath is the label "Distribution:" followed by a text input field containing the sequence "1 2 2 3 3 3 4 4 4 4 4 5 5 5 5 5 5 5". To the right of the input field are up and down arrow buttons. Below the input field is a horizontal bar with a slider and left/right arrow buttons. Underneath the bar is the label "Bin Width:" followed by a text input field containing the value "2". Below that is the label "Graph Size:" followed by two text input fields, both containing "300". Below the graph size fields is the label "Action:" followed by two buttons: "Reset" and "Submit". At the bottom of the interface is a navigation bar with five buttons: "Home", "Statistics Help", "CLASP Help", "CLASPWeb Help", and "About". Below the navigation bar is a footer line that reads "This page is maintained by dahlman@cs.colostate.edu".

CLASPWeb Histogram

Histogram

Distribution:

1 2 2 3 3 3 4 4 4 4 4 5 5 5 5 5 5 5

Bin Width: 2

Graph Size: 300 X 300

Action:

Home	Statistics Help	CLASP Help	CLASPWeb Help	About
----------------------	---------------------------------	----------------------------	-------------------------------	-----------------------

This page is maintained by dahlman@cs.colostate.edu

Figure 5. On-line Tools for Analysis (Colorado State University)

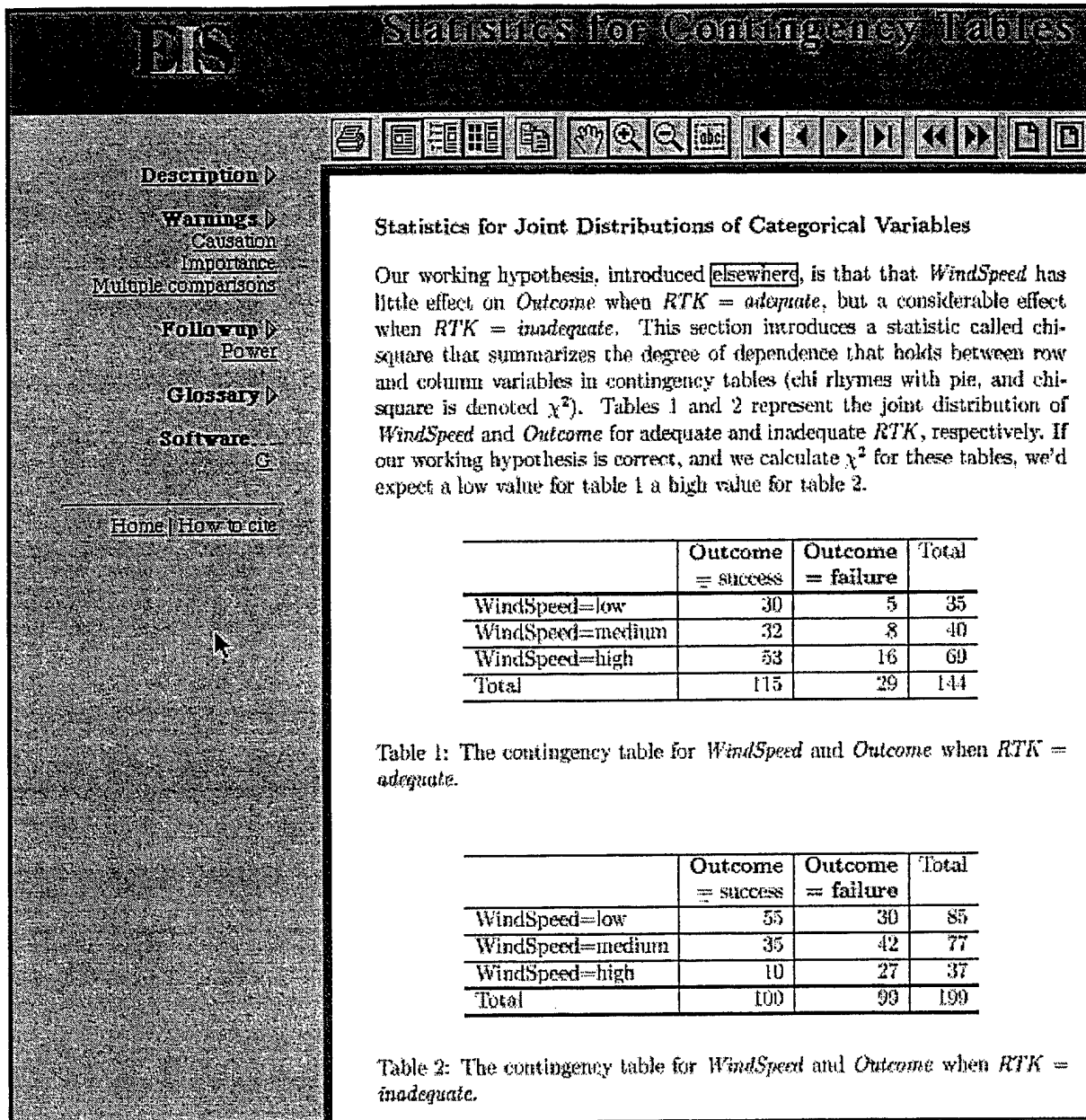


Figure 6. Description of Test and Modeling Methods (PDF)

The advice corner (Figure 7) is a constantly growing, indexed set of short articles in question-and-answer format. The Advice Corner Editor will take questions submitted by users, ask members of the Editorial Board to supply answers, and then produce edited versions to be added to EIS. Each question/answer pair will contain a title, short description, keywords, and full text. The keywords will be used to facilitate searching and dynamic page creation.

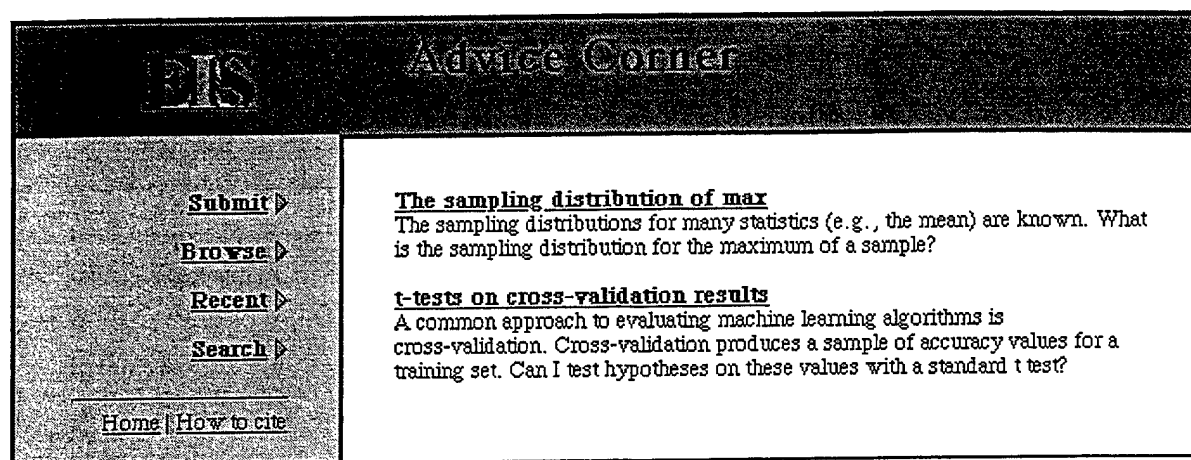


Figure 7. Advice Corner

Underlying EIS is an architecture to support a constantly growing set of statistical resources and “community wisdom” about empirical methods for information systems. Many EIS resources — including the advice corner, warnings, announcements, and glossary — will be partially composed of entries submitted by outside researchers and screened and organized by area editors and referees. In this way, EIS can grow to reflect the problems and solutions developed by researchers in the field.

At the time of this writing, several items have been uploaded to the UMASS beta EIS site, `<http://eksl-www.cs.umass.edu:80/~jensen/EIS/beta3/EIS-home.html>`, including:

- The addition of two more experiment types (analyzing execution traces and time series), and the completion of 16 additional techniques: Statistics on contingency tables; Binomial and Multinomial; Z test; Regression; Parametric confidence intervals for regression; Bootstrap confidence intervals for regression; Parametric confidence intervals; Bootstrap confidence intervals; Bootstrap (general); Bootstrap two-sample t test; One-way ANOVA; Two and Three way ANOVA; and four techniques for execution traces and time series.
- The addition of more items to the announcements, external resources, and advice corner.
- Providing citation instructions for each page so that users will know how to cite information they incorporate into publications.
- Making references similar to the glossary and announcements. This will make additions (and external submissions) easier to handle. An easy-to-use set of references will draw users to the site.
- Improving the help page. In particular, instructions on configuring PDF viewers.

- Adding links to a library of statistical code. This will make it easier for users to incorporate (correct) statistical code into their software.

The Future

We strongly encourage continued DoD support for the site, and see it as a potential clearinghouse of evaluative lessons learned for DARPA, DoD Research Laboratories, and Data and Analysis Center for Software (DACS): <http://www.dacs.com/index.shtml>. However, hosting and support by AFRL is uncertain. Due to the lack of maintenance at AFRL, it is recommended that the latest version of the EIS site be delivered but not uploaded on kbsa4. The site content will continue to be locally hosted at the University of Massachusetts at Amherst and additional on-line tools at Colorado State University. They will be updated on a non-DoD funded basis. The EIS site can be accessed at:
<http://ekslwww.cs.umass.edu:80/~jensen/EIS/beta3/EIS-home.html>

However future development and currency of the content is not guaranteed by voluntary university support. For this reason, Sterling Software recommends future DoD hosting and support of the site via DACS. The objective of DACS is to promote the use of existing software and software technology information by undertaking activities focused on the identification, access, analysis, processing and dissemination of software information. This includes the establishment of analysis procedures, statistical methods and routines to support studies, and conduct investigations of various software engineering/technology issues. EIS is well within the scope of this objective as an on-line resource that provides a single resource for managers, system builders, researchers, and users who wish to study the empirical behavior of information systems.

SECTION 3 EIS SITE STRUCTURE AND MAINTENANCE

The current EIS web site will be hosted at UMASS, while the original EIS web site hosted at AFRL on kbsa4, will have a link pointing visitors to the UMASS version. At the completion of the EMCIS contract, the EIS web site maintenance will officially be turned over to the government. If AFRL chooses to successfully maintain an EIS web site at AFRL, the delivered tar tape (A006) will have to be installed on the server. It would then be essential that the EIS web site administrator continue to monitor and maintain various aspects of the site. While UMASS and CSU will continue to update local versions of the EIS site, the AFRL version would soon become dated without maintenance. For this reason, it is recommended that the AFRL site be made inactive and temporary use of maintenance by UMASS and CSU be continued until permanent maintenance is arranged. The following sections detail the completed web site and what is necessary to continue maintenance.

3.1 Site Location

The final version of the EIS web site is located at UMASS and can be found at:

`<http://eksl-www.cs.umass.edu:80/~jensen/EIS/beta3/EIS-home.html>`.

The original web site is located at AFRL/IFTB on the kbsa4 server and is stored in the `/http/htdocs/EIS` directory.

3.2 Site Maintenance Recommendations

EIS was produced using GoLive CyberStudio™, NetObjects Fusion™, AdobeAcrobat™, Adobe PageMill™, BBEdit, Alpha, GIF Builder, ClarisWorks™, and Apple Macintosh™ computers. NetObjects Fusion™ was purchased under this contract and it is being turned over to the government. However, as the site grew more complex, it was found that NetObjects Fusion™ had serious limitations:

1. Complex files: Files are necessarily Fusion-defined, using tables which are difficult to interpret and will be nearly impossible to import into other packages in the future.
2. Cannot easily add to the site: Fusion generates some aspects of pages, such as the title graphics and buttons. The only automatically-generated options are poor textual buttons, with very limited format options.
3. Poor user interface: Important options are often hidden within dialog boxes and confusingly named. The operation of Fusion is not at all integrated with the operating system (e.g., no drag and drop, no use of user-defined folders to store portions of a site, etc.
4. Incorporation of outside files: Importing files is a tedious and time-consuming process, because files must be converted to Fusion's proprietary format. Files are stored as a Fusion database, and the HTML files generated from that database are nearly impossible to import into other packages.

For the above reasons, UMASS began use of GoLive CyberStudio™ which allowed for easier global link checking and had fewer import difficulties. CyberStudio™ is recommended for future maintenance and update of the site.

To date, update of the announcements and replies to Q&A are being handled by UMASS and will continue based upon availability of time and personnel at UMASS. It is recommended that maintenance of time sensitive links and pages be augmented by a dedicated database.

3.3 Contract Completion Issues

Both UMASS and CSU have asked to voluntarily maintain the EIS site, and permission has been granted by Mr. Craig Anken to do so. Sterling Software has uploaded a pointer on the AFRL EIS page on the kbsa server to the UMASS site:

`<http://eksl-www.cs.umass.edu:80/~jensen/EIS/beta3/EIS-home.html>.`

On the kbsa server, the top level EIS directory is owned by user dingman. If this directory is maintained, it is recommended that it be changed to a generic EIS account, root, or the user name of the person responsible for continued maintenance of the site.

In addition to the EIS directory, the web site also maintains a crontab file that executes three scripts (daily_stats, monthly_stats, and expire.pl). These were for the purpose of collection of site statistics and maintaining outdated links. See A005 for a detail description of each of these scripts. The crontab file (dingman) is located in the /var/spool/cron/crontabs directory. It is recommended this be disabled because the newest version of the EIS site is no longer on this server.

An EIS_webmaster@kbsa4.AI.RL.AF.MIL email account was set up to receive any EIS related mail. It is currently aliased to kasey_dingman@itd.sterling.com. If the site continues to be maintained at UMASS, this account can be deleted; however, at a minimum, the alias needs to be removed by the AFRL system administrator at contract completion.

SECTION 4 SOFTWARE INVENTORY AND DELIVERABLES

Following is a comprehensive list of all contractual deliverables as a result of this contract.

4.1 A001 – Program Progress Report

Program Progress Reports were submitted monthly.

4.2 A002 – Contract Funds Status Report

Contract Funds Status Reports were submitted quarterly.

4.3 A003 – Technical Information Report

CIIS Evaluation Tools and Methodologies Report was submitted on 04/04/97. This document describes our investigation of Complex Intelligent Information Systems (CIIS) technologies. The investigation of the general CIIS technologies assisted in the formulation of a strategy and method for creation of the evaluation tools and interfaces to be provided by the Evaluation of Intelligent Systems Internet site and the Empirical Methods for Evaluation of Information Systems Field Guide. Included was the Taxonomy of Intelligent Systems provided by the State University of New York Institute of Technology.

4.4 A004 – Technical Information Report

CIIS Evaluation Test Cases and Analysis Report was submitted on 08/27/97. This document provides the results of a web site survey and conducts a comparative analysis of the EIS web site with other existing web sites.

4.5 A005 – Technical Information Report

CIIS Evaluation Field Guide Design Document was submitted on 10/24/97. This document identifies and describes the objective of the EMCIIIS program. It discusses the web site development process, scripts, and configuration management procedures.

4.6 A006 – Technical Information Report

CIIS Evaluation Field Guide was submitted on 12/31/97. This 8mm tape contains the final version of the EIS web site. The EIS web site was also delivered on-line at:

`<http://eksl-www.cs.umass.edu:80/~jensen/EIS/beta3/EIS-home.html>.`

4.7 A007 – Presentation Materials

CIIS Evaluation Field Guide Workshop Materials were submitted at various intervals during the contract.

- EMCIIIS Status Briefing, 11/4/96, at AFRL, Rome, N.Y.
- EMCIIIS Design Workshop, 2/24/97, at UMASS, Amherst, Mass.
- EMCIIIS Status Review and Field Guide Demo, 06/26/97, at AAAI 97, Providence, R.I.
- Evaluation of Intelligent Information Briefing for INFO2000, 10/7/97, Vernon, N.Y.

4.8 A008 – Final Scientific and Technical Report

The Final Scientific and Technical Report was submitted on 12/31/97. This document identifies and describes the objectives of the EMCIS program. It details EIS site recommendations, site structure and maintenance, and a list of software and contract deliverables provided by Sterling Software under the contract.

4.9 Software Inventory

The following is a comprehensive list of software delivered as a result of this contract.

- Prototype EIS web site, dated 04/04/97, delivered on 8mm tape
- Final EIS web site, 12/31/97, delivered on 8mm tape
- NetObjects Fusion For Macintosh 2.0, delivered 12/31/97

SECTION 5 ACRONYMS

This section lists and defines acronyms used throughout this report.

AF	Air Force
AFRL	Air Force Research Laboratory
AI	Artificial Intelligence
C4I	Command, Control, Communications, Computers and Intelligence
CDRL	Contract Data Requirements List
CIIS	Complex Intelligent Information Systems
CLIN	Contract Line Item Number
CSU	Colorado State University
DACS	Data and Analysis Center for Software
DARPA	Defense Advanced Research Projects Agency
DoD	Department of Defense
EDA	Exploratory Data Analysis
EIS	Evaluation of Intelligent Systems
EMAI	Evaluation Methods for Artificial Intelligence
EMCIIS	Evaluation Methods for Complex Intelligent Information Systems
GIF	Graphics Interchange Format
HTML	Hypertext Markup Language
PDF	Page Description Format
UMASS	University of Massachusetts
URL	Uniform Resource Locators
WWW	World Wide Web

**APPENDIX A COLORADO STATE UNIVERSITY'S FINAL SUBCONTRACTOR REPORT
EVALUATION METHODS FOR COMPLEX, INTELLIGENT
INFORMATION DRIVEN SYSTEMS**

The Final Subcontractor Report for Evaluation Methods for Complex, Intelligent Information Driven Systems, prepared by Colorado State University, follows.

Final Subcontractor Report
Evaluation Methods for Complex, Intelligent
Information Driven Systems
F30602-95-0257

Adele E. Howe, P.I.

December 15, 1997

1 Overview of Effort

Given my group's expertise with developing Web-based applications, we undertook as our primary responsibility the development of a Web-based version of a statistics package for on-line demonstration of statistical evaluation methods on the Field Guide. We selected the CLASP (Common Lisp Analytical Statistic Package) system developed at University of Massachusetts as the statistical basis and focused on creating a code interface between CLASP and Web browsers. In addition, to showcase the system, we developed companion documents for accompanying text.

Thus, the project was divided into three major tasks:

1. the design and construction of a system for user directed interaction with the statistics package,
2. development of code for encapsulating statistical methods (i.e., calls) within HTML documents, and
3. conversion of companion documents for text of ClaspWeb.

2 ClaspWeb Interface

We constructed two versions of ClaspWeb, our Web-based statistical system. The proof of concept system used CGI and Perl scripts to collect input from and display data to web pages. Although we were able to write code for the basic functionality quickly, the system was cumbersome (it required three languages and two servers to run), fragile, lacking in sophisticated display capabilities (it relied on Unix utilities for constructing encapsulated graphics) and limited in its handling of user data (e.g., it did not allow partitioning).

The current version is based on CL-HTTP, a Common Lisp Web manager in the public domain. This version is cleaner, requiring only a single language and server. Additionally, the enhanced access to the Common Lisp runtime system afforded by CL-HTTP allows greater

flexibility in user data manipulation and is more extensible. The cost of these advantages was a longer development time to build simple parsers for the interface, work out security issues and debug some interaction effects between the systems.

ClaspWeb allows a user to download his/her own data to the Web site, partition the data by variables, display the data in a variety of ways, and run statistics on variables or partitions. The home page for the demonstration version is available at <http://satchmo.cs.colostate.edu:4936/>. This page includes links to basic statistics, visualization methods, documentation and examples of HTML documents with embedded statistics (see Section 3 for a description of that aspect of the system).

Each ClaspWeb page contains input forms for data, pull down menus or buttons for options on the methods, and a set of buttons for obtaining help information. For example, the Two Sample T-test page requests the data for the two samples (input windows) and allows the user to indicate which tails should be used (as in Figure 1). When the user clicks on the submit button, the system runs the statistic and creates an output page as shown in Figure 2.

The T-test example shows how the user might enter and analyze a small amount of data. For large amounts of data, the user may download their own datasets into ClaspWeb. First, the user logs in; the login facility was added to allow a single user to manipulate multiple datasets and to prevent one user from accessing another's dataset. Then, a dataset stored on the user's machine can be downloaded by clicking on "Load Dataset" on the homepage and following the directions. Datasets should be in Clasp format. At this point, the user can partition the data and perform statistical and visualization operations on it.

The current version of ClaspWeb includes the following basic capabilities:

Common Statistical Tests

- Two sample T-test
- G-test (contingency table test)
- Mean, Minimum, Maximum, Median, Mode, Quantile, Standard Deviation
- Chi-Square 2x2 and RxC
- One-Way ANOVA
- Linear Regression

Visualizations

- Histogram
- Scatter/Line plot

Evaluation Utilities

- Absolute Order Dependency Detection
- State Transition Diagram Dependency Detection (STDD)

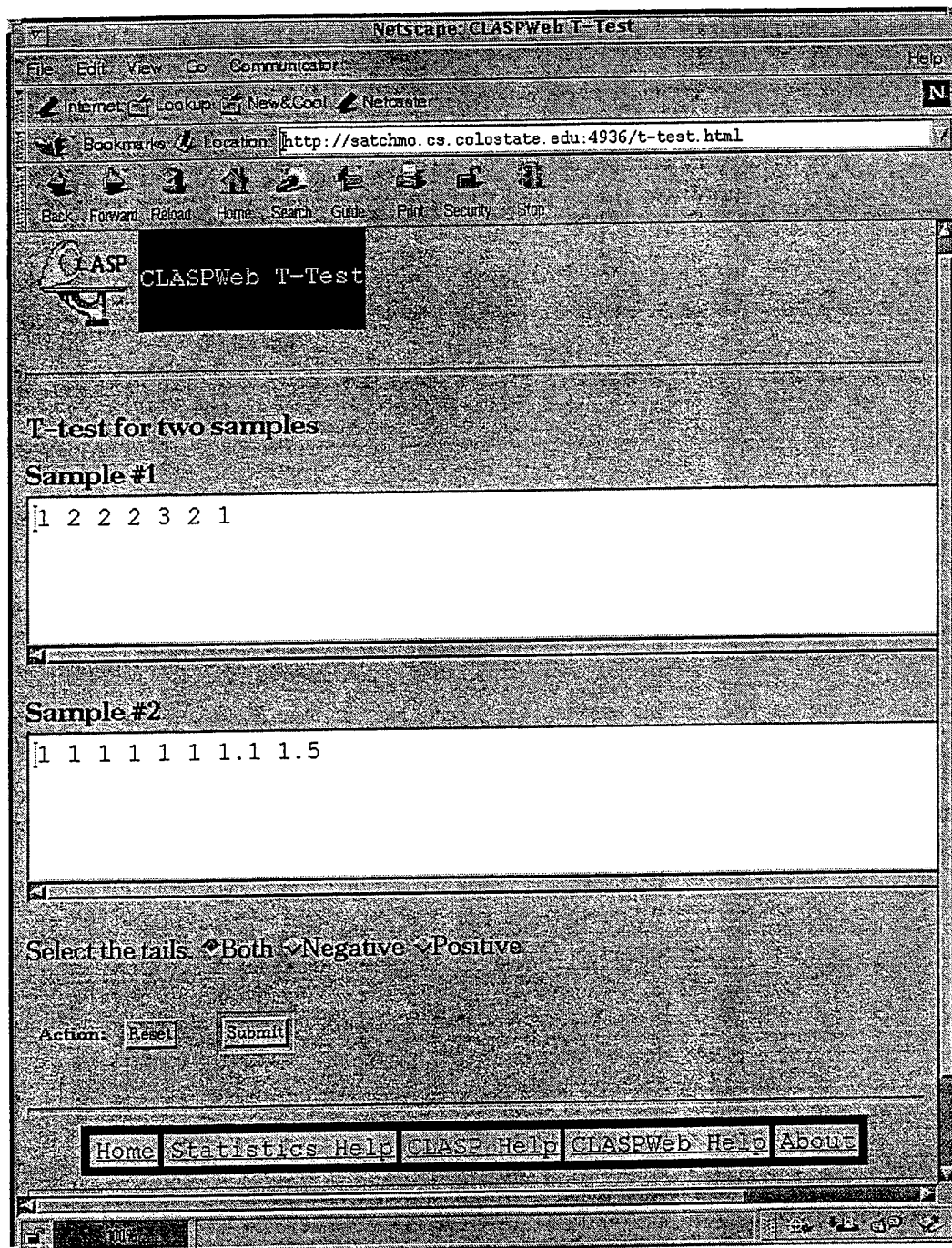


Figure 1: Example ClaspWeb page for T-test

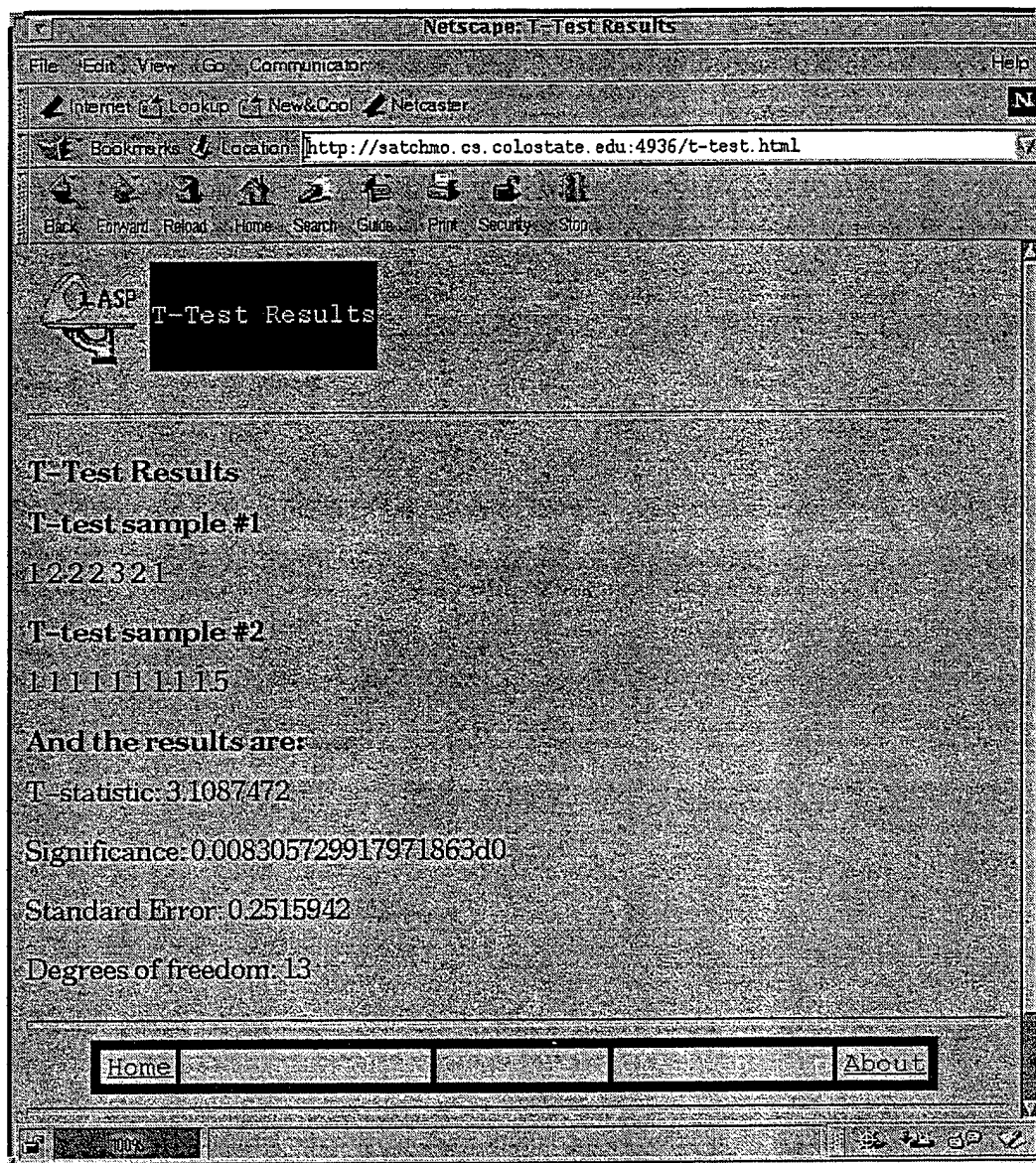


Figure 2: Web page that results from submitting the T-test request from Figure 1

The evaluation utilities were developed at Colorado State for analyzing execution traces. They show how new methods can be integrated as well as existing statistical techniques.

3 Hypertext Evaluation Documents with ClaspWeb

From an evaluation perspective, the real attraction of Web-based statistics is the possibility of embedding data and analyses into hypertext documents. Up until now, papers summarized results, with authors occasionally providing datasets by request. With ClaspWeb, the data files and analytical techniques can be embedded within the paper, allowing users to check results, examine data, re-run tests, confirm claims, test alternatives and know *exactly* what was done.

To demonstrate, we annotated a recent paper by Howe and Cohen on evaluating planning with tests that correspond to the text. In Figure 3, the text describes an $nx2$ G-test with its companion ClaspWeb form for running the test on the data in the text. The course code indicates the input fields primed already with data and directions for submitting the request to ClaspWeb, as shown in Figure 4. To exploit ClaspWeb then, users need only add the appropriate forms to their documents. These forms can be obtained by viewing source on the desired example from the home page.

We seriously considered security issues in both the interactive and the embedded forms of ClaspWeb. In each case, we restrict the set of allowable actions to those already defined within ClaspWeb. We do not permit users to define their own Lisp functions for partitioning data, but rather severely restrict their access to Lisp functions. We added the login facility to avoid conflicts with data and to provide privacy.

4 Companion Documents

ClaspWeb includes integrated help. The help text came from existing documents that were converted from latex to html with gifs for the equations. In particular, the Clasp manual and portions of Paul Cohen's text *Empirical Methods for Artificial Intelligence* (Chapters three and four) were converted to provide help appropriate to the statistical, visualization and evaluation capabilities included in ClaspWeb. The Clasp manual describes the mechanics of the utilities. Cohen's book provides the background and insight into the function and usage of the utilities.

We also converted two research papers for demonstrating hypertext capabilities. As with the other documents, the papers were converted from latex to html. In this case, they were also annotated with ClaspWeb calls to show how contingency table tests, T-tests and a specialized evaluation method (called *Dependency Detection*) that was developed in-house can be integrated into text.

5 Maintenance of ClaspWeb

We have developed code for the basic functionality and structure of ClaspWeb. We can provide a Unix tar file with the source code. In this section, we describe how to install, start and extend our version.

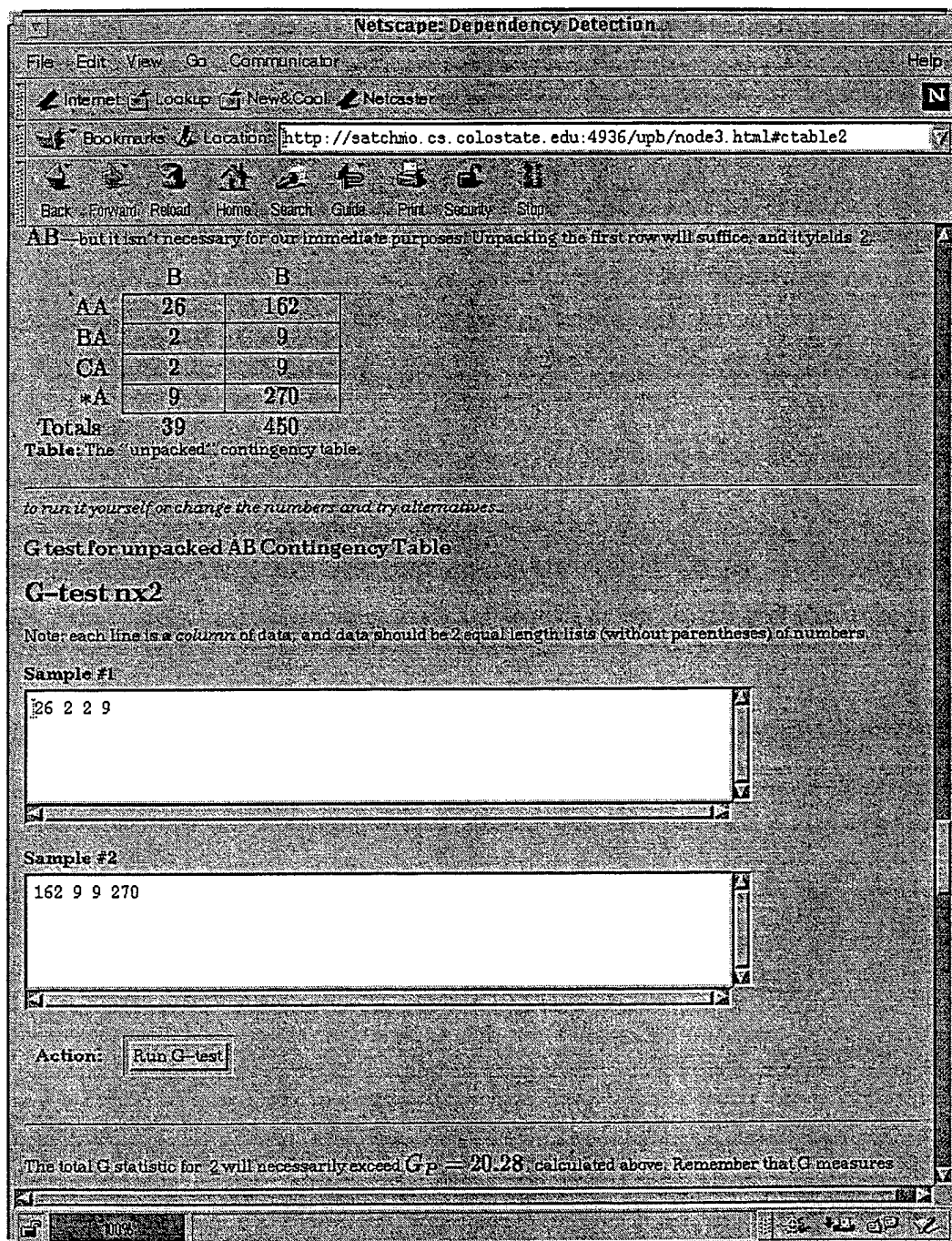


Figure 3: Web page for showing a test (contingency table using G-test) embedded within a document

```

<!-- ----- G-test code ----- -->
<hr width="100%">
<i>to run it yourself or change the numbers and try alternatives...</i>
<h3><a name="ctable2">G test for unpacked AB Contingency Table</a></h3>
<h1>G-test nx2</h1>

Note: each line is a <i>column</i> of data, and data should be 2 equal
length lists (without parentheses) of numbers. <br>

<FORM ACTION="/g-test.html" METHOD="POST" ENCTYPE="application/www-url-form-encoded">
<P><B>
Sample #1</B><br>
<TEXTAREA NAME="SAMPLE-1" ROWS=5 COLS=72>26 2 2 9</TEXTAREA>
</P>
<P><B>
Sample #2</B><br>
<TEXTAREA NAME="SAMPLE-2" ROWS=5 COLS=72>162 9 9 270</TEXTAREA>
<TABLE CELLS=2 COLS=2>
<TR><TD ALIGN="LEFT"><B>Action:</B></TD>
<TD ALIGN="CENTER"><INPUT TYPE="submit" NAME="Submit" VALUE="Run G-test"></TD></TR>
</TABLE>
</P>
</FORM>
<hr width="100%">
<!-- ----- G-test code ----- -->

```

Figure 4: HTML source for producing embedded G-test from Figure 3

5.1 Installation Instructions

ClaspWeb has been developed in Allegro Common Lisp on Sun/Unix-based workstations and requires Allegro CL, CL-HTTP and CLASP. Allegro CL is a commercial software package available from Franz, Inc. CL-HTTP is a public domain package produced at MIT. CLASP is a public domain package produced by Paul Cohen's laboratory at University of Massachusetts¹.

The installation of ClaspWeb is fairly straightforward. Although it does require some Lisp modifications, the directions for installation are as follows:

1. Install Clasp and CL-HTTP per their instructions
2. Uncompress and untar ClaspWeb into the directory where you want to install it. We'll refer to this directory as \$CW.
3. Edit the file \$CW/loader.lisp so that the following lines reflect the paths where you have installed the different components needed for ClaspWeb.

```
;; The Path to the directory of the Clasp Loader file
(defvar *clasp-loader-path*
        "/s/chopin/d/proj/meps/clasp/clasp/clasp-1.4.3")
```

```
;; The Path to the base directory for CL-HTTP
(defvar *http-directory*
        "/s/chopin/d/proj/meps/cl-http-60-57/")
```

```
;; The Path to the ClaspWeb base directory
(defvar *clasp-web-path*
        "/s/chopin/d/proj/meps/clasp-web-frozen/")
```

4. Also edit the line containing:

```
(http:set-standard-http-port 4936)
```

to reflect the port on which you want ClaspWeb to operate. One important thing to note is that if this port has a number lower than 1024, you must set up CL-HTTP to use the bind80 helper program; this is covered in the documentation for CL-HTTP.

5. Now to load ClaspWeb and start the server start a Lisp image in the \$CW directory and evaluate:

```
(load "loader.lisp")
```

at the Lisp prompt. If all of the paths are correct this should load Clasp, CL-HTTP and ClaspWeb and then start the server. At this point the server is up and running.

¹These packages are available from <http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html> and <http://eksl-www.cs.umass.edu/research/clip-clasp-details.html>, respectively.

6. The first time the code is run, it will be automatically compiled. Should you make changes later and wish to compile it; at the Lisp prompt enter:

```
(compile-system :clasp-web)
```

This will compile ClaspWeb and replace the running interpreted version with a compiled one.

7. To exit ClaspWeb and stop the server, evaluate the following form at the Lisp prompt:

```
(http-user::exit)
```

8. A final note: it may be desirable to run ClaspWeb in the background or as a daemon, this can be done by using the Unix `nohup` command. An example of this usage can be found in the `$CL-HTTP/acl/http` script which comes with CL-HTTP.

Everything (Lisp image and connection to Web) is encapsulated within the single server. The default setup includes five threads, which means that it can handle five requests simultaneously.

5.2 Restarting ClaspWeb

When it is necessary to restart ClaspWeb after the initial installation, for example after a system reboot, it is only necessary to perform step 5.

5.3 Adding New Statistical Utilities to ClaspWeb

The simplest way to add a new statistical function to ClaspWeb is to modify an existing form with a similar function. For instance, the majority of forms that are in the current implementation were created by modifying the t-test form.

In general, each form is kept in a separate lisp file; so the first step in creating a new form is to make a copy of an existing file. In this new file, the four basic structures that need to be modified to support a new test are:

1. The form generating function.
2. The answer exporting function.
3. The URL export function.
4. The page registration function.

The first of these two functions define the structure and content of the new test; the last two make the new test available in ClaspWeb. Finally, once a test has been completed it should be entered into the `sysdcl.lisp` file to be compiled and loaded with the rest of the system. The loader file is a standard `defsystem` file which handles the automatic compilation.

To illustrate the steps involved in modifying the form generation function, we will describe what we would have to do to modify the t-test form. The method that generates the t-test form is called *compute-t-test-form*; the beginning of it is as follows:

```
(defmethod compute-t-test-form ((url url:http-form) stream)
  (with-claspweb-page ((format nil "CLASPWeb T-Test")
                        :stat-help "/emai/ch4/node16.html"
                        :clasp-help "/clasp-docs/node14.html"
                        :clasp-web-help "/home.html#cwhelp"))
```

First, the method should be renamed to, for example, *compute-my-form*, so that the new function does not override the functionality of the original t-test. Then the string "CLASPWeb T-Test" should be changed to match our new function. Similarly, the strings in the key word list (i.e., :stat-help, :clasp-help, clasp-web-help) should be changed to match the locations of help files for your new function.

The rest of this function generates the html to create the form for the test itself. For the most part, we have followed a standard format; you should consult the documentation for CL-HTTP to determine how to make any radical modifications. For simple modifications, you only need to modify the following type of lisp form:

```
(html:accept-input 'html:multi-line-text
  "SAMPLE-1"
  :default "1 2 2 2 3 2 1"
  :stream stream))
```

which creates a multi-line input box called "SAMPLE-1" on the form. This same function can be used to create any type of input field. You can quickly put together a form by simply cutting and copying the accept-input invocations from other forms without understanding the subtleties of what is being done. The worst problem will most likely be malformed or ugly html. The important thing to keep in mind is that all of the different fields should have unique names.

Once the *compute-my-form* function is done, you must create a response function in a similar manner. The response function for the t-test example starts as follows:

```
(defmethod respond-to-t-test-form ((url url:http-form) stream query-alist)
  (with-claspweb-response ("T-Test Results"
    (sample-1 sample-2 tails))
```

The list (sample-1 sample-2 tails) corresponds to the names of the fields in the t-test form. These symbols will be bound to the values of the respective fields as entered by the user. These values are strings, which may necessitate parsing these strings before their use. However, in many cases, simply passing them to *read-from-string* is sufficient. Once the user's input is parsed, the t-test form simply calls the t-test function from clasp and outputs the results to the user using the *write-string* function as follows:

```
(write-string
  (format nil "T-statistic: ~A"
    t-statistic) stream))
```

Again, CL-HTTP is able to generate very complex html to dress up your results; the documentation for CL-HTTP covers how to do that.

The server now needs to be told how to serve the new functions. This is done with the *export-url* function. For example, the t-test form is exported as follows:

```
(export-url #u"/t-test.html"
           :html-computed-form
           :form-function #'compute-t-test-form
           :expiration '(:no-expiration-header)
           :response-function #'respond-to-t-test-form
           :public t
           :language :en
           :keywords '(:cl-http :demo)
           :documentation "These two forms do the t-test stuff.")
```

To export your new test replace the names of the t-test function with those that you defined and `#u"/t-test.html"` with `#u"/address/of/your/new-url.html"`. If you are creating a form to operate on datasets, the *export-url* needs to be copied from another form that operates on datasets since it includes additional information about access control which is used to manage the datasets.

At this point, the url for the new statistical test should be up and running. The final step is to add it to the list of available tests; this is done with the *register* function. For instance, t-test is registered by the following:

```
(register #u"/t-test.html" "t-test" 'simple-functions)
```

where `#u"/t-test.html"` represents the test's url, `"t-test"` is the text that will appear on the list of functions and `'simple-functions` is the list on which to include it. The other possible choice for this last argument is `'dataset-functions`, which causes the function to be added to the set that can handle datasets instead of data entered via the web page.

If the test function is in a file of its own, the file name will need to be added to the loader file. If it has been loaded during development, then it may be necessary to restart the system because the list of available functions may have been corrupted.

6 Prioritized Recommended Extensions

In our development, we focused on constructing infrastructure. If extensions to the existing system are a possibility, the highest priority is extending the repertoire of statistical utilities available through the interface. We focused on building a Web infrastructure. That infrastructure is now in place, but needs to be populated with utilities.

In decreasing order of priority, I suggest the following extensions to the work that has been conducted so far:

1. Add more statistical methods to ClaspWeb. Clasp includes a large number of statistical methods that should be integrated.

2. Add more visualization methods to ClaspWeb. At present, ClaspWeb includes only three visualization methods: histogram, scatter/line plot and state transition diagram graphing (this last was added to present the results of STDD). Plot coloring by partition would be enormously useful as would some color and 3-D techniques.
3. Augment the help documentation. The help documentation is rudimentary. It was borrowed from existing documents. Additionally, ClaspWeb should be integrated with the Field Guide with cross-references to examples on that site.
4. Enhance the ability of users to manipulate their data via ClaspWeb. The partitioning capabilities are limited by security concerns that were mentioned later. More sophisticated partitioning will require development of a restricted partitioning language and careful parsing and interpretation.
5. Tutorial and guidelines on embedding ClaspWeb in HTML documents. Embedded ClaspWeb shows great promise for significantly changing the nature of evaluation documents; these papers can become active research notebooks in which the community can participate in the analysis and interpretation of systems evaluations.
6. Development of additional evaluation methods. Basic statistical techniques work well for traditional experiments. Many areas of AI have developed their own techniques for analyzing data. A facility such as the Field Guide and ClaspWeb serve to highlight some of the gaps in the state of the art that need to be filled.

7 Research Papers that Reference this Grant or ClaspWeb

- A.E. Howe and E. Dahlman. "Characterizing Domain Specific Effects in Flaw Selection for Partial Order Planners." (abstract), In *Poster Session Abstracts for Fifteenth International Joint Conference on Artificial Intelligence*, p. 47, August 1997, Nagoya, Japan.
- A.E. Howe. "Statement of Interest: Comparing Apples and Oranges". In *Working Notes from Workshop on Empirical AI*, Nagoya, Japan, August 1997.
- A.E. Howe and G. Somlo. "Modeling Discrete Event Sequences as State Transition Diagrams". In *Advances in Intelligent Data Analysis: Reasoning about Data, Proceedings of the Second Conference on Intelligent Data Analysis*, Springer-Verlag, pp. 573-584, August 1997, London, England.
- A. E. Howe and G. Somlo. "Modeling Intelligent System Execution as State Transition Diagrams to Support Debugging". In *Proceedings of the Second International Workshop on Automated Debugging*, Linkoping, Sweden, May 1997.