

Design, Data Mining and Analysis of Hyperactive Environments

by
Michael M. Kashambuzi

B.S. Civil Engineering
Massachusetts Institute of Technology, 1997

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Mechanical Engineering
at the
Massachusetts Institute of Technology

June, 1999

© 1999, Massachusetts Institute of Technology. All Rights Reserved.

Signature of Author

.....
Michael M. Kashambuzi
May 24, 1999

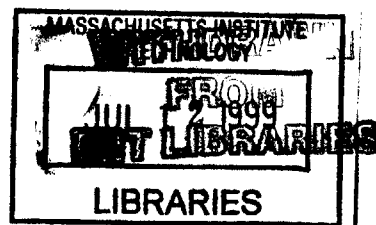
Certified by

.....
Nishikant Sonwalkar
Director, Hypermedia Teaching Facility
Thesis Supervisor

Accepted by

.....
Ain Sonin
Chairman, Graduate Committee

ENG



Design, Data Mining, and Analysis of Hyperactive Environments

by

Michael Kashambu

Submitted to the Department of Mechanical Engineering
on May 24, 1999, in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Mechanical Engineering

ABSTRACT

As the Internet and the World Wide Web continue to grow, new and emerging capabilities have made it possible to design and utilize concepts that were never before imaginable in such an environment. With these new capabilities, the use of the World Wide Web has become applicable in several fields including education.

In this thesis, we present a framework for designing and analyzing web-based, hyperactive learning environments. The proposed present preliminary requirements to assess the utility of hyperactive environments for the particular application, design techniques for the creation of these hyperactive environments using the latest web technologies, and analytical tests and evaluations for scrutinizing the adequacy and performance of such environments in a learning setting.

The model developed was then used in the design, development, and testing of several hyperactive environments. The conclusions of our studies indicate that:

1. Student performance is the same through the use of a web-based environment as it is through the use of an actual lab environment.
2. Students' performance is independent of past exposure to the material of a simulation, year in school, or preference for a lab environment.
3. A majority of students prefer the use of the hyperactive environment on the grounds of flexibility, ease of use, and ease of operation.
4. Students strongly agree that the hyperactive environment is suitable for achieving its predefined learning objectives.
5. Students strongly agree that the hyperactive environment performs well at the interfacial level making it easy to navigate between the instructions and experiment and facilitating the collection of data.

Thesis Supervisor: Nishikant Sonwalkar
Title: Director, Hypermedia Teaching Facility

Acknowledgements

I would like to thank my advisor, Dr. Nishikant Sonwalkar, for providing me with an atmosphere in which I could pursue and achieve my goals and his valuable contributions during the stages of this thesis. Additional thanks go to Dr. Nishikant Sonwalkar, Professor Richard Larson and the Sloan Foundation for their generosity in sponsoring me. Many thanks go to the Hypermedia Teaching Facility staff and students for their daily support.

Above all, this thesis is dedicated to my family which has always encouraged me to do all I can to the best of my abilities, supported me in all the decisions I have made, and sacrificed all else to make me who am I today.

Table of Contents

Acknowledgements	3
Table of Contents	4
List of Tables	9
List of Figures	12
Chapter 1 Introduction	13
Chapter 2 Web-Based Technologies	18
2.1 Introduction.....	18
2.2 Hypertext Markup Language (HTML)	18
2.3 Scripting.....	19
2.3.1 Common Gateway Interface (CGI) Scripting.....	20
2.3.2 JavaScript.....	21
2.3.2.1 Client-side JavaScript.....	21
2.3.2.2 Server-side JavaScript	22
2.4 Web Applications.....	24
2.4.1 Java	24
2.5 Plug-In Technologies	25
2.6 Choice of Technology.....	26
2.7 Application of Web Technologies	26
2.7.1 Carnot Cycle Simulation.....	26
2.7.1.1 Objective.....	26
2.7.1.2 Technology	29
2.7.2 Yield Management Simulation	29
2.7.2.1 Objective.....	29

2.7.2.2 Technology	30
2.7.3 Rotameter Experiment	31
2.7.3.1 Objective.....	31
2.7.3.2 Technology	31
Chapter 3 Data Variables, Collection and Storage Mechanisms.....	33
3.1 Introduction.....	33
3.2 Data Variables.....	33
3.2.1 Characterization of the End User.....	34
3.2.2 Performance Measures of the Hyperactive Teaching Tool.....	34
3.2.3 The Relationship between Student Performance, Usage and Evaluation of the Hyperactive Teaching Tool	35
3.3 Data Storage.....	36
3.3.1 Database.....	36
3.3.2 Data Model	36
3.3.2.1 Entity-Relationship Model	37
3.3.2.2 Model Application.....	38
3.4 Data Collection	41
3.4.1 Collection Mechanism.....	41
3.4.2 Storage Mechanism.....	41
3.4.3 Data Cleaning	41
3.4.4 Data Coding	42
3.4.5 Data Enrichment	43
3.5 Integration of Data Collection and Storage Mechanisms	43
Chapter 4 Statistical Analyses and Data Mining.....	44
4.1 Introduction.....	44
4.2 Statistical Tests	45
4.2.1 Man-Whitney U-Test.....	45

4.2.1.1 Purpose	45
4.2.1.2 Interpretation of Results	46
4.2.2 Chi-Squared (χ^2) Test	46
4.2.2.1 Purpose	46
4.2.2.2 Interpretation of Results	46
4.2.3 Choice of Statistical Test to Perform.....	47
4.3 Data Mining	48
4.3.1 Association Rules	48
4.3.1.1 Purpose	48
4.3.1.2 Interpretation of Results	48
4.3.2 Decision Trees	49
4.3.2.1 Purpose	49
4.3.2.2 Interpretation of Results	49
Chapter 5 Application of Statistical Tests and Data Mining to Hyperactive Environments	50
5.1 Introduction.....	50
5.2 General Statistics	51
5.2.1 Characterization of the End User.....	51
5.2.2 Performance Measures of the Hypermedia Teaching Tool	53
5.2.2.1 Satisfaction of the Learning Objectives.....	54
5.2.2.2 Performance of the Human-Machine Interface	58
5.2.2.3 Overall Ranking of the Tool.....	60
5.3 Mann Whitney Tests	62
5.3.1 Test for Independence of Hypermedia Environment Ranking with Student Prior Exposure to Calibration as the Population.....	62
5.3.2 Test for the Independence of Hypermedia Environment Ranking with Student Year as the Population.....	63
5.3.3 Test for the Independence of Hypermedia Environment Ranking with Student Preference of Lab Environment as the Population	64

5.3.4 Test for the Independence of Student Performance with Student Prior Exposure to Calibration as the Population.....	65
5.3.5 Test for Independence of Student Performance with Student Year as the Population.....	66
5.3.6 Test for Independence of Student performance with Student Preference of Lab Environment as the Population.....	67
5.4 Chi-Squared (χ^2) Tests.....	69
5.4.1 Test for Student Experience with Calibration Regarding Ranking of the Hyperactive Environment.....	69
5.4.2 Test for Student Year in School with Calibration Regarding Ranking of the Hyperactive Environment.....	70
5.4.3 Test for Student Preference for Environment Regarding Ranking of the Hyperactive Environment.....	72
5.4.4 Test for Student Experience with Calibration Regarding Student Performance.....	73
5.4.5 Test for Student Year in School Regarding Student Performance.....	75
5.4.6 Test for Student Preference for Environment Regarding Student Performance.....	76
5.5 Association Rules.....	78
5.6 Decision Trees.....	82
5.6.1 Determination of How Well the End Users Perform.....	82
5.6.2 Determination of How Well the Environment is Ranked.....	85
Chapter 6 Conclusions and Further Work.....	88
Appendix A.....	90
A.1 Carnot Cycle Simulation.....	90
A.2 Airline Yield Management Simulation.....	91
A.3 Chemical Engineering Rotameter Experiment.....	92
Appendix B.....	93
B.1 Mann-Whitney Test.....	93
B.1.1 Variable Requirements.....	93
B.1.2 Testing Procedure.....	93
B.2 Chi-Squared (χ^2) Test.....	94

B.2.1 Variable Requirements.....	94
B.2.2 Calculation of χ^2	95
B.3 Association Rules	96
B.3.1 Description.....	96
B.3.2 Determination of Desired Association Rules.....	96
B.3.3 Data Structure and Algorithm.....	97
B.4 Decision Trees.....	98
B.4.1 Description.....	98
B.4.2 Data Structure and Algorithm.....	100
References	103
Bibliography	104

List of Tables

Table 3-1 Student Attributes	38
Table 3-2 Student Performance Attributes.....	38
Table 3-3 Content of the Evaluation Survey	39
Table 3-4 Entity Representations using Primary Key	40
Table 3-5 Entity Representations using a Foreign Key.....	40
Table 5-1 Responses in Favor of the Use of the Hyperactive Lab.....	53
Table 5-2 Count and Measure Of Responses to Agreement Concerning Objectives.....	55
Table 5-3 Count and Measure of Responses to Agreement Concerning Procedures.....	55
Table 5-4 Count and Measure of Responses to Agreement Concerning Schematic Drawing	56
Table 5-5 Count and Measure of Responses to Agreement Concerning the Plotting of the Calibration Curve	56
Table 5-6 Count and Measure of Responses to Interpretation of the Calibration Curve	57
Table 5-7 Count and Measure of Responses to Identification of Systematic Error	57
Table 5-8 Count and Measure of Responses to Identification of Random Error	57
Table 5-9 Count and Measure of Clarity of Instructions.....	59
Table 5-10 Count and Measure of Navigation between Experiment and Instructions.....	59
Table 5-11 Count and Measure of Ease of Use of Interface	60
Table 5-12 Count and Measure of Ease of Data Collection.....	60
Table 5-13 Count and Measure of Overall Rank of Hyperactive Teaching Tool	62
Table 5-14 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Prior Exposure to Calibration as the Population.....	63

Table 5-15 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Year as the Population	64
Table 5-16 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Preference of Lab Environment as the Population	65
Table 5-17 Results of a Mann Whitney Test for Independence of Student Performance with Student Prior Exposure to Calibration as the Population	66
Table 5-18 Results of a Mann Whitney Test for Independence of Student Performance with Student Year as the Population.....	67
Table 5-19 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Preference of Lab Environment as the Population	68
Table 5-20 Chi Squared (χ^2) Test for Student Experience with Calibration Regarding Ranking of the Hyperactive Environment.....	70
Table 5-21 Chi Squared (χ^2) Test for Student Year in School with Calibration Regarding Ranking of the Hyperactive Environment.....	71
Table 5-22 Chi Squared (χ^2) Test for Student Preference for Environment Regarding Ranking of the Hyperactive Environment.....	73
Table 5-23 Chi Squared (χ^2) Test for Student Experience with Calibration Regarding Student Performance.....	74
Table 5-24 Chi Squared (χ^2) Test for Student Year in School with Calibration Regarding Student Performance.....	76
Table 5-25 Chi Squared (χ^2) Test for Student Preference for Environment Regarding Student Performance.....	78
Table 5-26 Association Rules Relevant to Student Year, Student Experience with Rotameter Calibration, Student Preference for a Laboratory Environment, Student ranking of the Hyperactive Environment, and Student Performance	81

Table A.1-1 Files Used by the Carnot Cycle Simulation	90
Table A.2-2 Files used by the Airline Yield Management Simulation	91
Table A.3-3 Files used by the Chemical Engineering Rotameter Experiment.....	92
Table B.1-1 Steps used in calculating the U-value for a Mann-Whitney Test.....	94
Table B.2-2 Steps used in calculating the χ^2 value for the Mann-Whitney Test	96
Table B.3-1 Steps used in generating association rules	97

List of Figures

Figure 1-1 Design Methodology for Hyperactive Environments.....	15
Figure 2-1 Client- and Server-Side JavaScript.....	23
Figure 2-2 Carnot Cycle as Seen in Conventional Media.....	28
Figure 2-3 Screen Shot of the Carnot Cycle Simulation.....	28
Figure 2-5 Screenshot of the Chemical Engineering Rotameter Experiment.....	32
Figure 3-1 Entity-Relationship Model for Hyperactive Environment Evaluations.....	39
Figure 3-2 Entity-Relationship Model.....	40
Figure 3-3 Integration of Data Collection and Storage Mechanisms.....	43
Figure 5-2 Histogram of Ranking of the Hyperactive Environment with Gaussian Distribution Superimposed.....	61
Figure 5-3 Decision Tree for Determination of Student Performance with Student Year, Preference for Lab Environment, Experience with Rotameter Calibration, and Supposed Ease of Use as Categories.....	84
Figure 5-4 Decision Tree for Determination of Hyperactive Environment Rank with Student Year, Preference for Lab Environment, Experience with Rotameter Calibration, and Supposed Ease of Use as Categories.....	87
Figure A. 1-1 Interactions between Files in the Carnot Cycle Simulation.....	90
Figure A. 3-2 Interactions between Files in the Chemical Engineering Rotameter Experiment ...	92
Figure B.3-1 Data Structure Used in Determining Association Rules.....	98
Figure B.4-1 Data Structure Used in Building Trees.....	101
Figure B.4-2 Algorithm Used in Building Binary Tree [11].....	102

Chapter 1

Introduction

As the Internet and the World Wide Web continue to grow, new and emerging capabilities have made it possible to design and utilize concepts that were never before imaginable in such an environment. With these new capabilities, the use of the World Wide Web has become applicable in numerous fields including education.

In this thesis, we present a framework for designing and analyzing web-based, hyperactive learning environments. We present preliminary requirements to assess the utility of hyperactive environments for the particular application, design techniques for the creation of these hyperactive environments, and analytical tests and evaluations for scrutinizing the adequacy and performance of such environments in a learning setting.

Designers of web simulations tend not to perform an analysis to determine the utility of their work in such learning environments. In order to benefit fully and to understand the efficacy of the World Wide Web in an educational setting, such an analysis needs to be carried out on several levels. First, it is important to understand if the Web environment and its associated capabilities are applicable for the particular learning application. By defining the learning objectives of the application, we step closer to identifying which capabilities are useful for our intended learning goals. Next, with the collection of available programming techniques, languages and tools, it is important to identify which ones make the best use of the resources. Upon making such a

determination, the appropriate tools are then used to design the application. After this stage, the ever-crucial stage of evaluation is carried out to assess the adherence of the hyperactive environment to the initial learning objectives. The simulation is then deployed for use by its expected audience. During this stage, data is collected regarding performance of the hyperactive environment as well as characteristics related to its end users. This data is then analyzed to assess the ability of the hyperactive environment to fulfill its intended learning goal(s). As should be expected, the above methodology is recursive in nature in that each stage must be revised if it fails to meet the desired objectives.

The suggested framework is diagrammed in Figure 1-1 below.

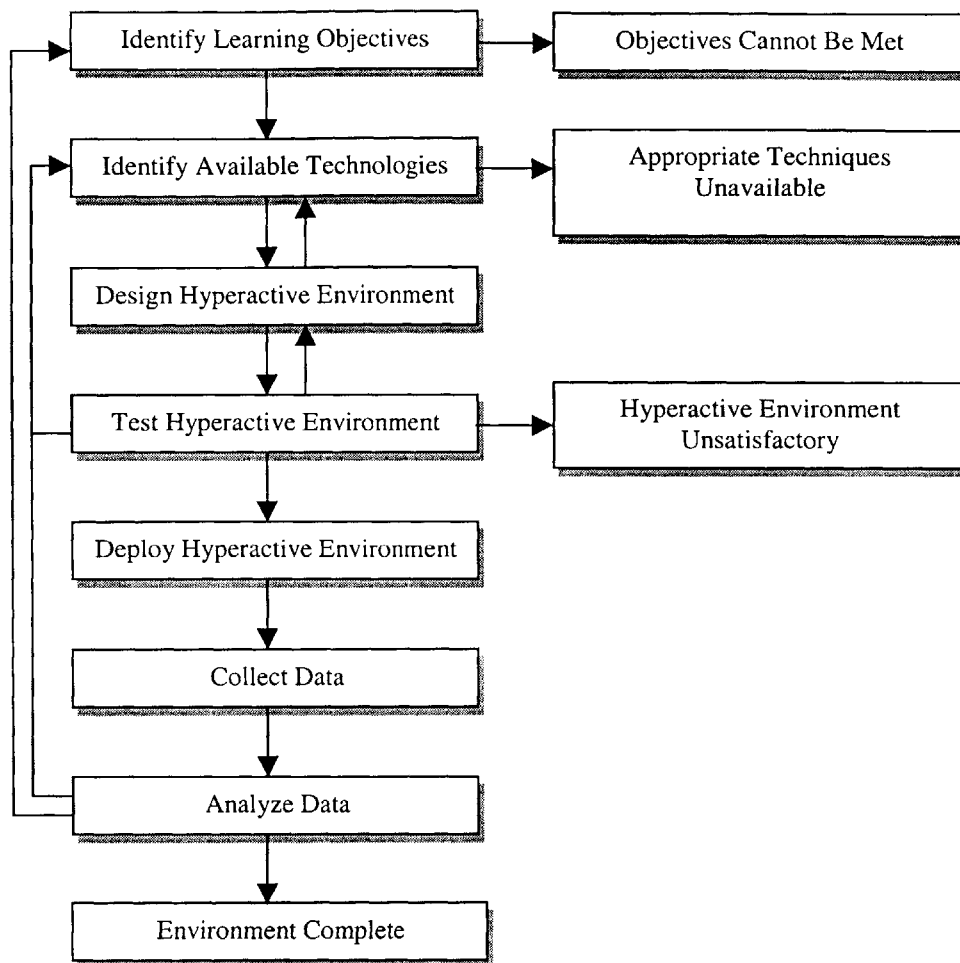


Figure 1-1 Design Methodology for Hyperactive Environments

As shown in Figure 1-1, the process of developing a hyperactive environment starts with the identification of the learning objectives. Knowledge of the objectives this early in development is critical for subsequent stages. If the objectives cannot be made through the use of a web-based environment, the choice is made to cease development. The second step involves the identification of the available technologies that may be suitable for development purposes. If no technique exists which can successfully meet the requirements of the environment, the decision to discontinue development is made. The third phase of development is in the actual design of the hyperactive environment implementing the techniques chosen in the second stage. Upon

completion of the design, testing is carried out to assess the ability of the environment to perform on a functional level. If the environment fails to meet the requirements at this stage, a re-evaluation of the chosen technologies and design methods used is carried out. If adjustments to the technologies and design methodologies can be made, they are carried out. Otherwise, an unsatisfactory environment leads to the cessation of development. If the fifth stage, the deployment stage, is reached, the hyperactive environment has satisfied the requirements at a purely functional level. Once deployed to end users, data is collected to perform a more detailed analysis of the hyperactive environments. The analysis of this detailed data marks the end of the methodology for the design of hyperactive environments. However, results from the analysis should be used for future development.

The directional arrows represent the sequence of steps between the identification of needs and the analysis of the hyperactive environment. Upward-pointing arrows represent a revision of the step at which the arrow points; forward-pointing arrows represent a completion of the step at their tail. Arrows that point horizontally represent a decision to end the design. The lowest box in the diagram represents a successful completion of the design of a hyperactive environment.

The remainder of this thesis is organized as follows. Chapter 2 provides an overview of programming standards and techniques available for use on the World Wide Web. These techniques will be described and cases of their use will be provided. Chapter 3 describes the data that will be used in the analysis of the hyperactive environments along with the methods of collection and storage. Chapter 4 concerns the statistical analyses and data mining techniques that will be used on the data collected for analysis of the hyperactive environment. Chapter 5 employs a case study to highlight the proposed methodology with an example of a hyperactive environment in use. Chapter 6 presents a conclusion of the thesis and areas for further work.

The conclusions of our studies indicate that:

1. The hyperactive environment is well rated by its end users
2. The ranking of the hyperactive environment is independent of student year, student experience, and student preference for lab environment implying that end user experience does not affect the ranking of the hyperactive environments.
3. The end users' performances through the use of the hyperactive environments are comparable to those using actual lab environments.
4. The end users' performances are independent of student year, student experience, and student preference for lab environment implying that all end users are capable of using hyperactive environments.

Chapter 2

Web-Based Technologies

2.1 Introduction

The number of technologies available for use on the World Wide Web has grown substantially since its beginning in 1994-1995. Some of the technologies most commonly available include HTML, JavaScript, Java and plug-ins. Several of these technologies can be applied to hypermedia teaching environments. In order to properly choose which technology suits an application, it is important to understand the capabilities of the technology. With such knowledge in mind, it is easier to determine which technology is most suitable. This chapter describes some web technologies used in the design of hyperactive environments followed by three educational implementations.

2.2 Hypertext Markup Language (HTML)

The Hypertext Markup Language (HTML) is probably the most apparent technology associated with the World Wide Web. "HTML is a document-layout and hyperlink-specification language. It defines the syntax and placement of special, embedded directions that aren't displayed by the browser, but tell it how to display the contents of the document, including text, images, and other supported media" [1].

HTML is the framework upon which other web document constituents sit. It serves as the container in which other components and languages subsist. Using HTML, one can construct an environment that delivers information-rich content to viewers on the web more so than through the use of conventional means of information propagation. HTML provides the content and structure while the World Wide Web serves as the mechanism for delivery.

At its beginning, "HTML was designed to structure documents and make their content more accessible, not to format documents for display purposes" [1]. However, over time, HTML has become a medium for formatting documents through the inclusion of better tags for textual, image, tabular and other elemental manipulations.

The latest version HTML is more commonly referred to as Dynamic HTML (DHTML). DHTML provides the user with an environment that permits better textual formatting, better positioning of content, dynamic page building and much improved interactive content.

2.3 Scripting

In order to perform server and web application programming, scripting is used. Scripting languages are those languages that provide programmers with the behaviors and associated objects needed to run applications on the web. Two different methods of scripting exist. These are:

1. Server-side scripting. In this form, programs run on the machine that is serving the information. This is done for several reasons.
2. Client-side scripting. In this case, the programs run on the client-side machine. The main advantage of client-side scripting is its reduction of the load on the server. Another

advantage is its ability to create dynamic content for the client with relative ease and speed as compared with server-side scripting.

It is through scripting that any kind of program is able to run on the web.

2.3.1 Common Gateway Interface (CGI) Scripting

"CGI is the part of the Web server that can communicate with other programs running on the server" [2]. The Common Gateway Interface is the layer that gives the web the ability to transfer information from web clients to programs on the server and then relay the results back to the users. In doing so, "CGI turns the web from a simple collection of static hypermedia documents into a whole new interactive medium" [2].

CGI scripting allows applications to be run which take advantage of

- Forms. Forms are used to collect information from users and respond appropriately.
- Gateways. Gateways provide a mechanism for communication between a web server and some other application that cannot be accessed directly from the Web.
- Dynamic documents. Dynamic documents are that which are built on the fly and contain information that is relative to a particular user session or request.

CGI scripting can be done in practically any programming language. Those most popularly used are Perl, C, C++, Shell, Tcl, Visual Basic and AppleScript. The choice of which language to use depends on several factors. However, according to Gundavaram, some considerations to make when choosing a language include

- Ease of text manipulation
- Ability to interface with other software libraries and utilities

- Ability to access environment variables (in UNIX)

2.3.2 JavaScript

JavaScript is a technology developed by Netscape Communications Corporation™ but implemented by most popular browsers. “JavaScript is a lightweight interpreted programming language with rudimentary object-oriented capabilities” [3]. JavaScript gives programmers the ability to create programs with the intention of running them on the web. Two forms of JavaScript exist: client-side JavaScript and server-side JavaScript. Although both of these are built on the same core language corresponding to ECMA-262, client- and server-side JavaScript employ it differently. These differences in employment are highlighted below.

2.3.2.1 Client-side JavaScript

Client-side JavaScript has associated with it functionalities that are meant for use from within a web browser. Typically, client-side JavaScript is found embedded in an HTML page and is interpreted at runtime (i.e. as the page is loading).

Client-side JavaScript is useful for several reasons but on three major levels which are the

1. Programmer level,
2. Document level, and
3. End users level.

Client-side JavaScript gives the programmer control over the appearance of the content in a document. Content control ranges from text and image formatting to the generation of dynamic HTML. JavaScript is also capable of controlling the browser behavior from navigation between documents to creation and manipulation of browser windows. In addition, client-side JavaScript

allows interaction between the content in a document. The different components can interact with each other for different intended purposes. As a simple example of component interactivity, clicking a button can manipulate an image. Interaction between the user and the browser is easily facilitated using JavaScript. Users can control components and, likewise, the browser can keep track of user actions through the use of cookies.

2.3.2.2 Server-side JavaScript

Server-side JavaScript has associated with it functionalities that are meant for use on a web server. Server-side JavaScript is also found embedded in an HTML page but is compiled into an executable file before deployment.

Server-side JavaScript allows web scripts to perform actions found in client-side JavaScript as well as other actions that are not. The four main standouts of server-side JavaScript are

1. the ability to maintain state,
2. the ability to access files,
3. the ability to access databases, and
4. the ability to access code written in another language

A major advantage of server-side JavaScript is its ability to maintain a particular state. The state refers to the status of the

1. Server (what request have been made, which applications are in use, which applications are sharing information, etc.),
2. Application (how many users are connected, what other applications have requested information from this application, etc.),

3. Client (what actions has the client taken, what are the client's characteristics, etc.),
and
4. Request (what most recent information has the client requested)

Server-side JavaScript allows for interaction with files on the server. This allows information to be retrieved from or stored in files for later use.

Server-side JavaScript integrates the server with databases. The use of databases allows large repositories of information to be stored and retrieved. Server-side JavaScript makes it simple to perform such actions on a database while overstepping the difficulty of having to write low-level programs.

In the case where one needs access to functions written in another language, one can employ server-side JavaScript to fulfill this requirement. By compiling the functions as libraries and installing them on the web server, native functions can be accessed through server-side JavaScript.

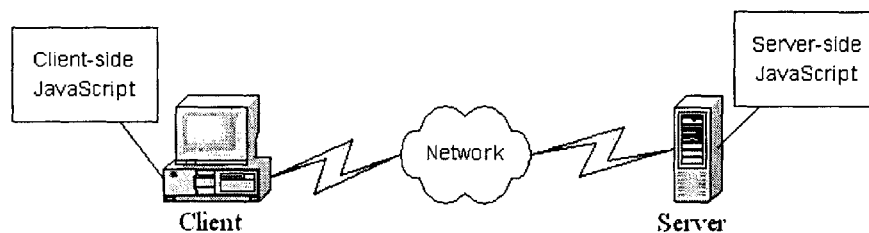


Figure 2-1 Client- and Server-Side JavaScript

2.4 Web Applications

At times, scripting is unsuitable for developing complex applications. For these cases, Java is a powerful alternative.

2.4.1 Java

Java, an object-oriented programming language developed by Sun Microsystems, is best known for its use on the World Wide Web and its platform independence. However, Java is much more than that. “Java is a language for creating safe, portable, robust, object-oriented, multithreaded, interactive programs” [4].

According to Naughton[4], Java is notable for many characteristics. Some attributes of Java are that it is

1. Simple and powerful. Using Java, one can easily write a complex program “without having to expose of all the dangerous inner workings of the underlying system” [4].
2. Safe. Since the use of Java is apparent on the Internet, Java developers strongly integrated the issue of security into the language making it difficult to access system resources, spread viruses, etc.
3. Object oriented. Java employs the object model.
4. Robust. Java is considered robust because it debugs code as you write it and run it, manages memory well, and provides a mechanism for handling exceptions.
5. Interactive. Java makes it easy to build interactivity between the program and the end user but also between different elements of the program using multithreading.

6. Architecture neutral. Because they run on the Java virtual machine and are not dependent on the operating system, Java is a portable programming language that is independent of the machine architecture.
7. Interpreted and high performance. Through the use of bytecode, Java programs can be both interpreted programs for speed or, for high performance situations, can be optimized using the Java runtime engine to create native machine code.
8. Easy to learn. Because of its widespread use, Java has been made so that it is easier to learn compared to other programming languages.

Java programs can exist as standalone programs or as applets. Standalone programs run at the operating system level in their own window whereas applets run in a container within a web browser.

2.5 Plug-In Technologies

Several companies began developing applications that run from the web document container. These programs which allow users to run programs that are not supported by a browser are called plug-ins.

Plug-ins give capabilities to the web browser that would not otherwise be possible. Some examples of multimedia plug-ins are Quicktime™ for viewing movie clips and LivePicture™ and CosmoPlayer™ for viewing 3-D worlds.

Plug-ins, in some cases, also provide application programming interfaces (APIs) that allow the programmer to extend the functionality of the application.

2.6 Choice of Technology

Here are a few tips on deciding which technology is best for use.

1. For basic formatting of text and images, use HTML.
2. For more customized formatting including positioning of content, font sizing, and layering, use dynamic HTML (DHTML).
3. For quick and easy scripts that require user interaction and dynamic text (formatting and positioning) and do not carry security or privacy concerns, use client-side JavaScript. For scripts that contain file and database access as well as access to functions written in other native languages and scripts that require maintenance of state, use server-side JavaScript.
4. For programs that require network programming, complicated layouts and formatting, complex computation, and definition of objects not provided in any of the other scripting languages, use Java.

2.7 Application of Web Technologies

Given the above technologies available for the design of hyperactive environments, the following three applications were developed.

2.7.1 Carnot Cycle Simulation

2.7.1.1 Objective

The objective of this simulation is to provide a visual environment for understanding the concept of the Carnot cycle [12].

The Carnot cycle describes the reversibility and transfer of heat between two reservoirs at different temperatures. For a Carnot cycle, a system undergoes a cyclical process involving four

reversible changes. Two of these changes are isothermal meaning the process involves constant temperature; the other two are adiabatic meaning no heat is transferred into or out of the system during the process. The position of the piston in the cylinder is associated to the Carnot cycle with the pressure-volume (P-V) and temperature entropy (T-S) diagrams of the ideal gas contained in the cylinder.

Most textbooks will present the Carnot cycle using a diagram relating piston position to the aforementioned diagrams using a numerical code representing the point in the cycle as in Figure 2-2. A description of the Carnot cycle using these diagrams would be described in the following four steps:

1. The gas expands adiabatically along the path 1-2 where, at point 2, the gas reaches the temperature of the cold temperature, T_c .
2. The gas is compressed isothermally at T_c and rejects an amount of heat Q_c along the path from 2-3.
3. The gas is compressed adiabatically along the path from 3-4 until it reaches the temperature of the high temperature reservoir, T_h .
4. The gas expands isothermally along the path from 4-1, thereby absorbing an amount of heat, Q_h .

The simulation developed accomplishes the same task; however, it displays the graphs in real-time while showing a description of each stage of the cycle above the graphs being plotted (Figure 2-3).

This simulation represents the possibilities of connectivity between all of the above technologies.

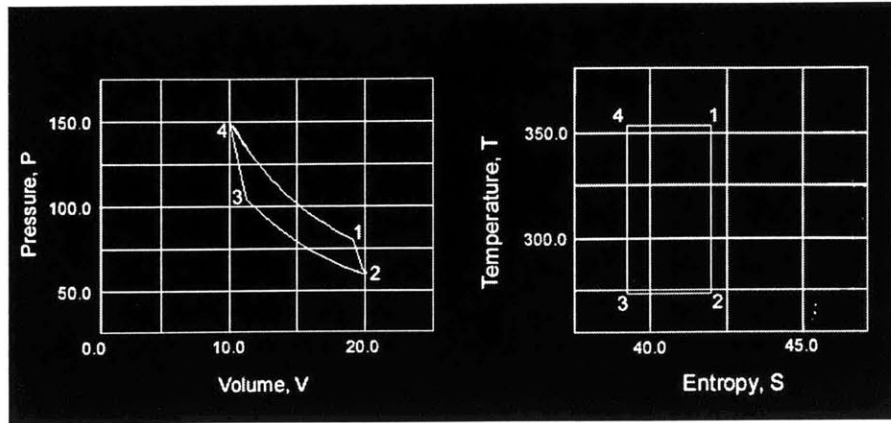


Figure 2-2 Carnot Cycle as Seen in Conventional Media

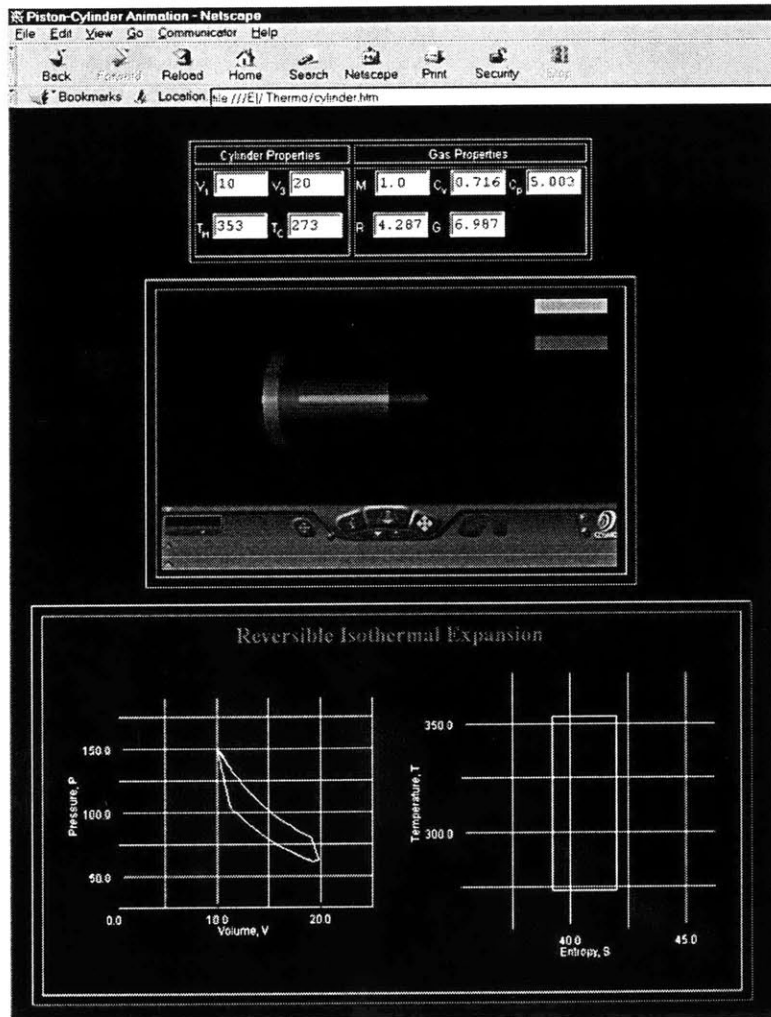


Figure 2-3 Screen Shot of the Carnot Cycle Simulation

2.7.1.2 Technology

The Carnot cycle simulation (Figure 2-3) takes advantage of all the above mentioned technologies. HTML is used to create the interface. JavaScript is used on the client-side to get parametric input from the user. A plug-in (CosmoPlayer™) is used to display the motion of the piston in the cylinder as a 3-D animation. Java is used to display the P-V and T-S diagrams as well as interact with the plug-in to control the motion of the piston-cylinder apparatus.

Using LiveConnect™, the parameters entered by the user in the HTML page are sent to the Java applet. The Java applet, through the use of class APIs provided by CosmoPlayer, receives information about the position of the piston in the cylinder and calculates the values shown in the P-V and T-S diagrams. For more detailed information on the implementation, see Appendix A.1.

2.7.2 Yield Management Simulation

2.7.2.1 Objective

The objective of the yield management simulation is to provide students with the ability to make decisions concerning booking practices to yield the most profit in the airline industry.

Heuristics, which employ complex mathematical expressions, probability distributions, and calculations, are commonly used to simulate yield management in the airline industry. An application to actively display the reservation of airline avoiding the use of the above complex methods is desirable.

2.7.2.2 Technology

The yield management simulation (Figure 2-4) relies on the use of HTML and Java. The HTML serves to provide end users with instruction on yield management and the operation of the application; Java is used to program the booking process heuristics and display the reservation of the seats as request come in.

Java was chosen due to the complex nature of the heuristics and the intensive graphical display of information. The layout also contributed to the choice of Java as the suitable programming language for the application. For more detailed information on the implementation, see Appendix A.2.

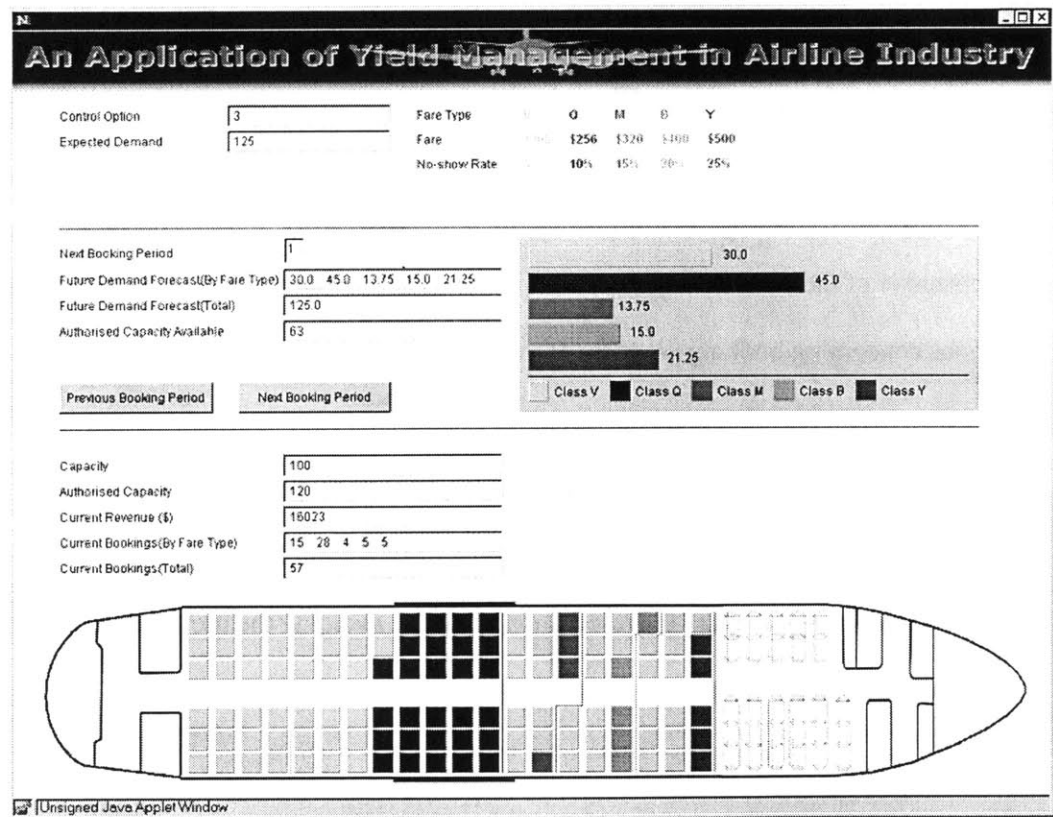


Figure 2-4 Screen Shot of the Airline Yield Management Simulation

2.7.3 Rotameter Experiment

2.7.3.1 Objective

A common problem faced in chemical engineering research as well as in industrial process plants is calibration of flow meters. It is a classic problem in which one assesses the accuracy of results and precision of equipment involved in an experiment by using an independent set of standards.

The objective of the rotameter experiment is to calibrate a rotameter, analyze the data, plot the results, and write a brief technical report. The environment for doing this is provided through the Web.

2.7.3.2 Technology

Although it is rather complex, the rotameter experiment (Figure 2-5) relies only upon the use of HTML and JavaScript. HTML is used to provide instruction and the container for the experiment. DHTML is used for both static and dynamic positioning of content. Client-side JavaScript is used to control the interaction between elements of the experiment and user input. For more detailed information on the implementation, see Appendix A.3.

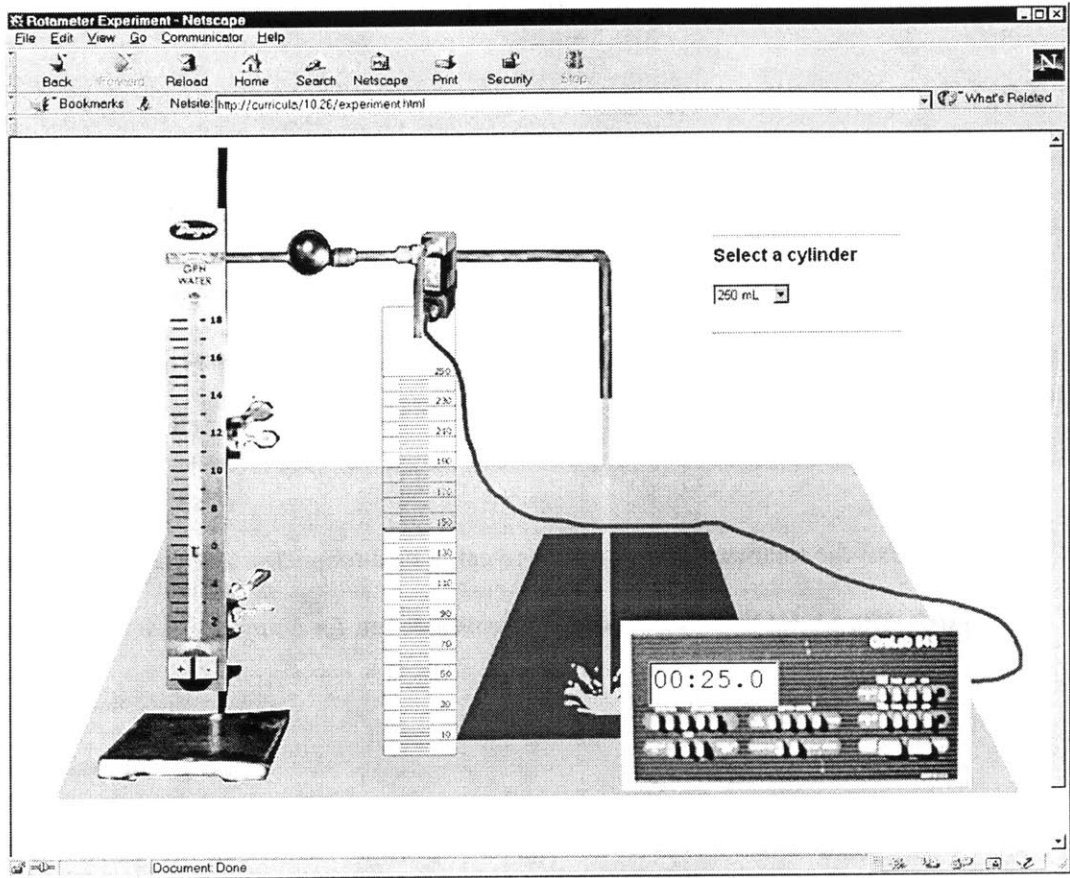


Figure 2-5 Screenshot of the Chemical Engineering Rotameter Experiment

Chapter 3

Data Variables, Collection and Storage Mechanisms

3.1 Introduction

In order to evaluate hyperactive environments and their utility in web-based learning, it is important to first determine which variables are required to understand the problem. In hypermedia teaching environments, the most important variables are those that relate to the end users of the technology and the attributes of the technology that are critical for evaluating its performance. Additionally, the method by which this data is collected and stored is critical for performing evaluative tests. For this purpose, we turn to the first critical steps in data mining which relate to the manner by which data is collected and stored for further analysis.

3.2 Data Variables

To fully evaluate the performance of hyperactive teaching tools, measurement needs to be made on three scales:

1. Characterization of the end users
2. Performance measures of the hyperactive tool and
3. The relationship between student performance, student usage of the hyperactive tool and evaluation of the simulation in a learning environment.

3.2.1 Characterization of the End User

Knowledge of certain characteristics about the end users helps in the validation of results achieved from analytical and evaluative tests used to assess the strengths and weaknesses of hyperactive teaching tools. End user data that is collected allows for the determination of populations that make use of the designed hyperactive learning environments. For our purposes, population can be characterized on the following three scales:

1. **The student's year.** It can be argued that the year of a student (e.g. freshman, sophomore, etc.) may have an overall effect on his/her performance. By tracking this variable, a determination can be made as to how critical the student's year is with regard to the student's performance on various levels.
2. **The student's course.** The experience of the student sometimes plays a role in the determination of a student's performance. For students who are familiar with the material presented in a course module, it might be expected that they should perform better than students without any prior exposure to the course material should.
3. **The prerequisites satisfied by the student.** To further distinguish between performance levels on the basis of experience on the part of the student, information as to whether or not certain prerequisites have been satisfied by the student is gathered.

3.2.2 Performance Measures of the Hyperactive Teaching Tool

Simulation data seeks to evaluate the hyperactive simulation on the following three scales:

1. **The performance of the simulation.** Performance of the hyperactive simulation is judged on two scales:

- i. **Technical merit.** Evaluation of technical performance seeks to address how well the hyperactive environment represents the actual simulation, mainly in terms of its operation.
 - ii. **Presentation.** Presentation relates to how well information regarding the hyperactive simulation is relayed to the end user. Issues of concern include:
 - visual similarity between the actual simulation and the hyperactive simulation
 - navigability between entities of the simulation
2. **The satisfaction of the learning objectives on the part of the simulation.** Each simulation is designed with certain learning objectives in mind. Another means of evaluating a hyperactive simulation is to assess how well it actually meets these learning objectives.
3. **Performance of the man-machine interface.** A major part of the design of hyperactive teaching tools is in the creation of a user-friendly interface. Without such, the teaching tools are as good as defeated. Evaluation seeks to determine how well the interface of the simulation performs.

3.2.3 The Relationship between Student Performance, Usage and Evaluation of the Hyperactive Teaching Tool

It is not merely enough to characterize the end users and evaluate the performance of hyperactive simulations in isolation. Since end users are those who evaluate the tools, it is critical to understand the relationships that identify how the various populations of end users assess the performance of the hyperactive teaching tools. For example, students who have never performed

a particular experiment in an actual lab setting have no prior knowledge of its look and/or its functionality. In such a case, particular attention needs to be paid to their evaluation of these aspects of the tools. We will, therefore, attempt to establish the relationship between performance variables and simulation characteristics through the use of statistical methods and data mining techniques described in Chapter 4.

3.3 Data Storage

To understand the concepts behind data storage, it is necessary to define the idea of a database and that of a data model.

3.3.1 Database

In order to facilitate the collection and storage of data, a database is employed. “A database is a logically coherent collection of data with some inherent meaning” [5]. A database is made up of files, records and fields which taken together represent a logical representation of the information within.

3.3.2 Data Model

To best represent the information to be contained in a database, it is necessary to employ a data model. A data model is graphical means of representing the information in a database. “The function of the data model is to clearly convey data and its relationships, attributes, definitions...that govern the data” [6].

Several different data models exist. The choice of which data model to use depends on the information being stored in the database. For the information being stored regarding the

evaluation of hyperactive teaching tools, the most appropriate data model is the relational data model, sometimes referred to as the entity-relationship model.

3.3.2.1 Entity-Relationship Model

The entity-relationship model is one that defines the relationship between the items in the database. The entity-relationship model is identified by the following terms:

1. **Entities.** “An entity is a principal object that is of significant interest to the user” [7].

In order for an object to be identified as an entity, it must adhere to the following criteria:

- Each entity must be significant.
- Each entity must be generic.
- Each entity must be fundamental.
- Each entity must be unitary.

Entities can either be strong or weak entity types. Weak entity types “are identified by being related to specific entities from another entity type in combination with some of their attribute values.... it may not be possible to identify a weak entity without an owner entity” [5]. The strong entity is that entity without which the weak entity cannot exist.

2. **Attributes.** Attributes are characteristics that define entities. “An attribute is a fact or nondecomposable piece of information about an entity” [7].
3. **Relationships.** In most cases, a database will contain more than one entity. For these cases, the relationship serves as the association between the different entities. If relationships exist, foreign keys, unique identifiers, tie together the different entities.

3.3.2.2 Model Application

In light of the entity-relationship model, for application towards hyperactive teaching tools, the significant entities to be stored in the database are

- the student information,
- the evaluation of the tools, and
- the student performance.

The attributes of each entity are given in the tables below.

Student Information	
Username	Primary key which identifies each student individually
Department	Department in school (Mechanical Engineering, etc.)
Year	Year in school (Freshman, Sophomore, etc.)
Prerequisite(s)	Prerequisites for a given class (defined by professor)
Prior Lab Experience	Experience with lab work related to hyperactive tool (Yes/No)

Table 3-1 Student Attributes

Student Performance	
Username	Foreign key which identifies individual student from student information
Score	Numeric and/or Letter grade

Table 3-2 Student Performance Attributes

Evaluation Survey	
Username	Foreign key which identifies individual student from student information
Ranking Questions	Rankings on achievement of learning objectives by tool
Freeform Questions	Comments on the achievement of learning objectives by tool
Preferences	Preference for hyperactive or actual experiment
Overall Ranking	Rank of tool on the scale of 1-10

Table 3-3 Content of the Evaluation Survey

The relationships that tie together the entities are described in the statements below.

- Students respond by giving an evaluation of the hypermedia tools
- Students are judged based on their performance

Given the information contained in the entities and their attributes and relationships, Figure 3-1 represents the prescribed entity-relationship model.

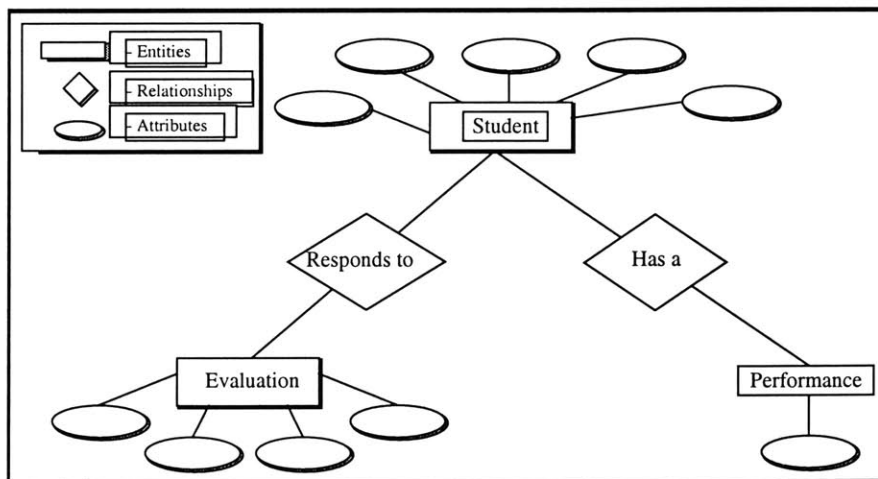


Figure 3-1 Entity-Relationship Model for Hyperactive Environment Evaluations

To convert the entity-relationship model to that required in the database, tables are used to represent the entities and primary and foreign keys are used to both identify and relate the different entities. Given the following relational model in Figure 3-2,

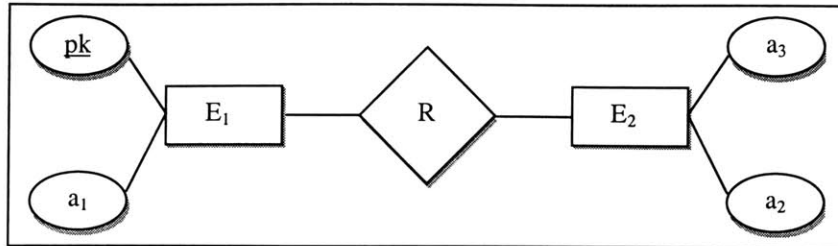


Figure 3-2 Entity-Relationship Model

the following tables highlights the translation of entity types to the database representation of the same information.

<u>pk</u> ¹	a ₁
------------------------	----------------

Table 3-4 Entity Representations using Primary Key

<u>pk</u>	a ₂	a ₃
-----------	----------------	----------------

Table 3-5 Entity Representations using a Foreign Key

This model is applicable in our case because the student is the strong entity whereas their evaluation and performance are the weak entity types.

¹ pk refers to the primary key

3.4 Data Collection

Having specified the use of a database as the storage mechanism, the next challenge is the collection and storage of variables to take advantage of database functionality.

3.4.1 Collection Mechanism

The primary means of data collection is through the use of questionnaires. These questionnaires are filled out by the end-users of the hyperactive tools and address the same questions highlighted in §3.3.2.2. The questionnaire is provided to the end-users using a web interface.

3.4.2 Storage Mechanism

To address the issue of the storage mechanism, the issues of data cleaning, coding and enrichment need to be grasped.

3.4.3 Data Cleaning

Data cleaning is the process of removing pollution from the data. This process “can be executed in advance while others are invoked only after pollution is detected at the coding or discovery stage” [8] in order to prevent pollution from ever entering the database.

Data pollution can arise in the following situations:

1. **Duplicate records.** Duplicate records can arise as the result of outright duplication, negligence and deliberate misrepresentation of information.

2. **Domain inconsistency.** Domain inconsistency represents information that is not plausible given the domain, “an expression for the permissible values for the column” [9].

To prevent data pollution of the above forms in advance, questionnaires are presented in a manner that provide end-users with all possible responses to many questions. For example, rank-based questions allow single responses through the use of radio buttons; some other questions restrict the values of responses by providing pull-down lists.

The only questions that are difficult to filter pollution from are free-form questions. For these cases, data cleaning is pursued once all data has been stored in the database.

3.4.4 Data Coding

Data coding is the procedure by which data is transformed from its initial state to some other form. “The way in which we code the information will, to a great extent, determine the type of patterns we find. Coding, therefore, is a creative activity that has to be performed repeatedly in order to get the best results” [8].

As Adriaans mentions, the manner in which data is coded can severely affect the informative patterns that we can extract from it. With this knowledge, it is more befitting to maintain the data in as complete a state as possible. With the data in its most general form, the analyst can convert it to a more appropriate format in order to carry out a particular test.

3.4.5 Data Enrichment

In many cases, “new information can easily be joined to the existing client records” [8].

Additional information can be added to records in the database to enrich the value of the data.

For the data collected, enrichment is provided regarding student performance. Additional information, made available by course instructors regarding student performance, is combined with existing records in the database.

3.5 Integration of Data Collection and Storage Mechanisms

The integration of data collection and storage involves several different mechanisms. As previously mentioned, a web interface is used to collect the information from end users.

Information is presented using web-based forms on the client computer. Data cleaning is accomplished using Server-side JavaScript, a Netscape™ technology. Within the Server-side JavaScript, structured query language (SQL) is used to insert information into a remote database.

The diagram below represents the interconnection between these mechanisms.

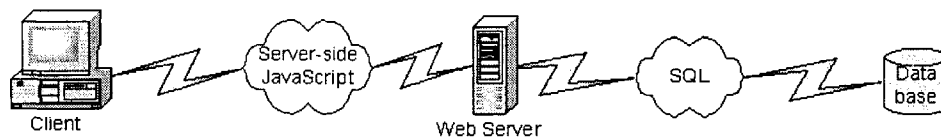


Figure 3-3 Integration of Data Collection and Storage Mechanisms

Chapter 4

Statistical Analyses and Data Mining

4.1 Introduction

It is not enough to merely design hyperactive teaching tools and to assume their utility in an educational setting. It is crucial to understand what factors are critical in determining the success of hyperactive educational tools. A variety of tests can be carried out to aid in making the above determination. We turn to two areas to assist us in answering this question.

The first category is statistics. Statistical tests are applicable because they help in "providing evidence for judging the validity of a hypothesis or inference" [10]. For our analyses, we choose to focus on two statistical tests that are commonly used to test variables or populations for the possibility of significant differences.

The other category is a relatively new area known as data mining. Data mining, sometimes referred to as knowledge discovery in databases (**KDD**), employs an independent series of tests to provide clues about data that may not be visible through the use of statistical tests or other analyses. The tests we choose to guide us in our factorial analysis are the generation of association rules and decision trees.

4.2 Statistical Tests²

During the initial stages of knowledge discovery, it is prudent not to make any assumptions about the distributions in the data. With this in mind, the statistical tests that we choose to perform are non-parametric tests, characterized as such due to their treatment of distributions in the data.

"...Nonparametric tests make fewer assumptions about distributions" [10].

Nonparametric tests rely on the use of nominal and ordinal data.

Given these constraints, the Mann-Whitney U-Test and the Chi-Squared Test are the non-parametric tests that will be performed.

4.2.1 Man-Whitney U-Test

4.2.1.1 Purpose

When it is necessary to test for the possibility of significant differences between two samples, a Mann-Whitney U-test is performed. The Mann-Whitney U-Test seeks to test either of two hypotheses:

- The null hypothesis which states that there is no significant difference between two populations
- The alternative hypothesis which states that the difference between two populations is real.

² See Appendix B for detailed information about the following statistical tests

The null hypothesis is the goal of carrying out the test. The goal of this particular test is to determine whether or not the two samples are from the same population.

4.2.1.2 Interpretation of Results

The smaller value of U is then compared against an expected value determined with the knowledge of n_1 , n_2 and α (the significance level) and the z-statistic. If the calculated value is greater than the expected value, it can be said that the groups from which the two samples were taken are significantly different at the chosen significance level thereby satisfying the alternative hypothesis.

4.2.2 Chi-Squared (χ^2) Test

4.2.2.1 Purpose

Given sets of sample data from independent populations where each trial permits several outcomes, it is sometimes necessary to tell whether or not similar frequencies in data can be obtained in other cases. The χ^2 test determines whether or not expectations are valid or due to chance. The χ^2 test asks the question, "Are obtained and expected frequencies different and, if so, with what probability?"

4.2.2.2 Interpretation of Results

The χ^2 value is a direct indication of whether or not observed frequencies are close to those expected. Larger values of χ^2 indicated greater differences between observed and expected frequencies and smaller values indicate the opposite.

An empirical value for χ^2 is calculated based on

1. the number of rows and columns in the contingency table providing the number of degrees of freedom $[(r-1)(k-1)]$ and
2. a pre-specified significance level, $\alpha=0.10$

The calculated value of χ^2 is then compared to this empirical value calculated [using the $(r-1)(k-1)$ degrees of freedom] and, if greater, signifies a significant difference across the two categories in the sample data at the chosen level of significance.

4.2.3 Choice of Statistical Test to Perform

Two considerations are taken when deciding which statistical test to perform, these being

1. the selection of which variables from your sample are the independent and dependent variables, and
2. the determination of the class of the variables in your sample. Variables can be classified as nominal, ordinal or interval.

With knowledge of the above two data, it becomes more manageable to decide which statistical tests to apply. If necessary, it is possible to convert variable types to those required by a particular statistical test.

For the current study of technology-enabled learning, with no prior knowledge of the distribution of data, the nonparametric tests seem to be an appropriate choice for determining the significance of hyperactive environments.

4.3 Data Mining

The area of data mining provides information that is generally unnoticeable in simple analyses such as the aforementioned statistical tests. Knowledge discovery in databases reveals hidden patterns and associations between variables. For our study, data mining allows the performance of a factorial analysis to highlight the associations between the variables collected.

The two applicable data mining methods used are association rules and decision trees.

4.3.1 Association Rules

4.3.1.1 Purpose

Often, one needs to answer the question, “When one thing is observed, is there another thing that is observed along with the first?” A further refinement may also seek to answer with what magnitude and assurance such an observation holds. When it is necessary to determine what factors imply the presence of others, the generation of association rules becomes prevalent.

4.3.1.2 Interpretation of Results

The rule body, rule head, minimum support, and minimum confidence describe generated association rules and must be taken together. Therefore, any attempt to extract other rules from those generated is inconclusive as a result of the pre-specified conditions.

4.3.2 Decision Trees

4.3.2.1 Purpose

When one strives to achieve a goal, there are certain factors that one must overcome before one can reach that end. With regards to hypermedia teaching tools, the end goal is discovering whether or not the tool is suitable and effective for use in learning environments. The factors that are used to determine such a resolution include ratings of the tool by end users, performance by its end users, etc.

4.3.2.2 Interpretation of Results

Given a decision tree, the root node represents the attribute that provides the most information in determining the goal attribute. The leaves immediately below the root node are the values that appeared for the attribute in the transactions. The successive nodes and branches represent the non-goal attributes and their distinct values that have not already appeared in the decision tree. Lastly, the leaf nodes represent the values of the goal attribute. Overall, a decision tree represents the order of non-goal attributes that lead to the possible values of the desired outcome.

Chapter 5

Application of Statistical Tests and Data Mining to Hyperactive Environments

5.1 Introduction

In Chapter 4, we highlighted techniques that would be useful in evaluating the performance of hypermedia teaching tools. In this chapter, we apply these tests and analyses to the data variables that were described in Chapter 3 to the Chemical Engineering Rotameter Experiment³.

A common problem faced in chemical engineering research as well as in industrial process plants is calibration of flow meters. It is a classic problem in which one assesses the accuracy of results and precision of equipment involved in an experiment by using an independent set of standards.

The objective of the rotameter experiment is to calibrate a rotameter, analyze the data, plot the results, and write a brief technical report. The environment for doing this is provided through the Web.

³ The choice of which simulation to use was governed by the availability of data and no other factors.

5.2 General Statistics

The following information provides a synopsis of the end user population and response to the performance and ranking of the hypermedia teaching tool.

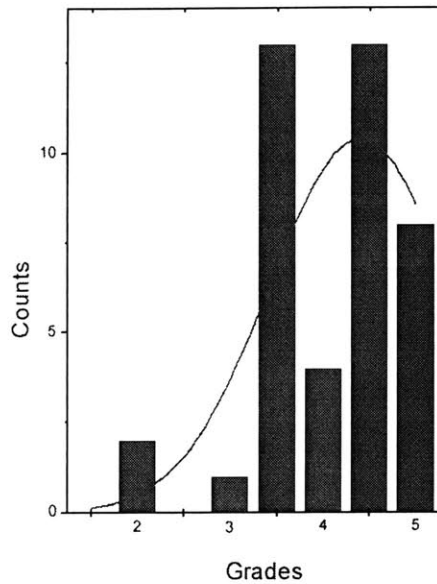


Figure 5-1 Histogram of Student Grades with Gaussian Distribution

5.2.1 Characterization of the End User

As mentioned earlier in Section 3.2.1, knowledge of certain characteristics as regards the end users is instrumental towards the validation of results achieved from analytical and evaluative tests used to assess the strengths and weaknesses of hyperactive teaching environments. For the Chemical Engineering Rotameter Experiment (herein referred to as the hyperactive environment or "Web Lab"), the information regarding end users is given below.

The total number of students surveyed was 41 out of a possible 54 of whom 100% were Chemical Engineering students. 26 of the students were third year undergraduates whereas the other 15 surveyed students were fourth year. Approximately 100% of all students had fulfilled the prerequisites for the web lab. Contrary to this and, despite the fact that 100% of all participants had worked in an actual laboratory setting prior to using the web lab, a vast majority (93%) of students using the web lab had never calibrated a rotameter. The mean score on the lab reports was a 4.1 out of 5 with a standard deviation of 0.76.

The average student score on the lab report (of those surveyed) was a 4.1 out of a possible 5.0. The distribution of grades is given in Figure 5-1.

As one can see, the performance of the students does not follow a normal distribution and hence, non-parametric tests should be used when using their scores as a test variable (described in Chapter 3).

When asked to make a choice regarding which form of the lab they would prefer (actual or web-based), half of the students opted for the use of the web environment as a means for gathering data to write a laboratory report. The following are characteristic responses for choosing to use a web environment.

Factor	Response
Flexibility	"...this can be done on our own time, as opposed to having a TA always being around."
Time	"I would rather use the Web lab because it makes gathering the data much faster."
Ease	"It's easier to use the web lab-you learn the same thing anyway." "With such a tedious task as calibrating a rotameter, working on a computer is

	much easier since you don't have to set up and take down the experimental setup"
Operation	"The 10.26 web lab was nice because everything was set up well and went smoothly. I remember doing it in a real lab and it was messy (spills) and the rotameter was hard to get it exactly where you wanted it."

Table 5-1 Responses in Favor of the Use of the Hyperactive Lab

70% of the remaining students would have preferred to do the laboratory in its actual setting. The tendency for people to prefer the actual laboratory setting is due to their desire for tangibility. Statements similar to those found below capture this feeling.

- "I would prefer a hands on experience, not pushing buttons on a mouse."
- "An actual lab setting gives a more "hands on" feel than doing it over a computer."

The remaining students were unsure of their choice between the actual and web-based versions of the lab. For these students, a preference for a mixture of both actual and simulated experimentation is tantamount.

- "... a good mix of the two methods would be ideal."
- "Both have advantages and disadvantages. Using the actual equipment gives valuable hands-on experience, but using the web setup is much more expedient."

5.2.2 Performance Measures of the Hypermedia Teaching Tool

Performance measures determine how well the hypermedia environment performs on various levels. These levels are articulated further below.

5.2.2.1 Satisfaction of the Learning Objectives

As was mentioned in the previous chapter, each simulation is designed with certain learning objectives in mind. The rotameter experiment has the following as learning objectives:

1. Explanation of the objectives of the rotameter calibration
2. Explanation of the procedures involved in running the experiment
3. Drawing a schematic diagram of the setup
4. Drawing and interpretation of the calibration curve
5. Identification of systematic and random error

A majority of the students either agree or strongly agree that, through the use of the web-based environment, they are able to explain the objectives and procedures associated with the experiment (Tables 5-2 and 5-3).

Rank ⁴	Count
-------------------	-------

⁴ The following is a breakdown of the coded values for rank:

1 – Strongly Disagree
2 – Disagree
3 – Agree and Disagree
4 – Agree
5 – Strongly Agree

Rank ⁴	Count
1	1
2	5
3	6
4	22
5	7

Mean	3.7
Standard Deviation	0.98

Table 5-2 Count and Measure Of Responses to Agreement Concerning Objectives

Rank	Count
2	3
3	3
4	22
5	13

Mean	4.1
Standard Deviation	0.83

Table 5-3 Count and Measure of Responses to Agreement Concerning Procedures

In addition, a large number of students are able to easily draw a schematic diagram of the setup and plot a calibration curve (Tables 5-4 and 5-5).

Rank	Count
2	3
3	7
4	23

5	8
---	---

Mean	3.87
Standard Deviation	0.81

Table 5-4 Count and Measure of Responses to Agreement Concerning Schematic Drawing

Rank	Count
2	5
3	8
4	22
5	6

Mean	3.71
Standard Deviation	0.87

Table 5-5 Count and Measure of Responses to Agreement Concerning the Plotting of the Calibration Curve

However, when it comes to the interpretation of the calibration curve and the identification of systematic and random error (Tables 5-6 - 5-8), the rating of the ability of the web lab to facilitate these aspects is conflicting between end users in that a great number agree that it is useful whereas some disagree with that statement.

Rank	Count
2	3
3	17
4	20
5	1

Mean	3.46
Standard Deviation	0.67

Table 5-6 Count and Measure of Responses to Interpretation of the Calibration Curve

Rank	Count
1	3
2	7
3	15
4	16

Mean	3.07
Standard Deviation	0.93

Table 5-7 Count and Measure of Responses to Identification of Systematic Error

Rank	Count
1	3
2	8
3	13
4	17

Mean	3.07
Standard Deviation	0.96

Table 5-8 Count and Measure of Responses to Identification of Random Error

The first three objectives, being that they are the easier portions of the lab report, are well rated. However, when it becomes necessary for students to interpret results, the ratings of the ability of the lab to facilitate such go down. This result is understandable in that both the actual and web labs do not explicitly make such interpretations known; these are left up to the student. We could expect such a rating regardless of the use of the actual lab or the hyperactive environment.

5.2.2.2 Performance of the Human-Machine Interface

This section seeks to determine how well the human-machine interface of the simulation performs. This performance is judged on the following scales.

1. Clarity of the instructions
2. Navigation between experiment and instructions
3. Ease of use of the interface
4. Ease of data collection

With regard to all the above criteria regarding performance of the human-machine interface, an overwhelming number of students agree and strongly agree that the hyperactive environment for running the calibration experiment is a suitable environment.

Rank	Count
Null	2
1	2
3	8
4	18

Rank	Count
5	11

Mean	3.92
Standard Deviation	0.98

Table 5-9 Count and Measure of Clarity of Instructions

Rank	Count
Null	2
2	1
4	20
5	18

Mean	4.41
Standard Deviation	0.64

Table 5-10 Count and Measure of Navigation between Experiment and Instructions

Rank	Count
Null	2
2	1
3	7
4	20
5	11

Mean	4.05
Standard Deviation	0.76

Table 5-11 Count and Measure of Ease of Use of Interface

Rank	Count
Null	2
3	5
4	27
5	7

Mean	4.05
Standard Deviation	0.56

Table 5-12 Count and Measure of Ease of Data Collection

5.2.2.3 Overall Ranking of the Tool

The overall ranking of the tool represents the culmination of each of the above rankings. Overall, the hyperactive teaching environment for the Chemical Engineering Rotameter experiment was given a rank of 6.7 on a scale of 10. The graph below provides the necessary information.

As one can see from the figure below (Figure 5-2), the ranking of the hyperactive environment is certainly non-normal and hence, non-parametric tests should be used when using the ranking as a test variable.

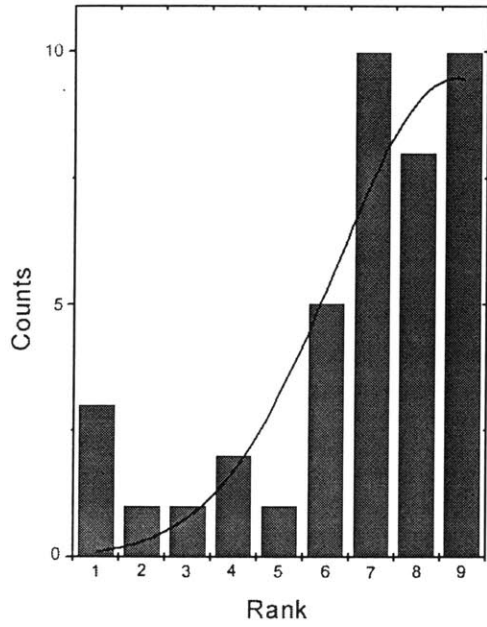


Figure 5-2 Histogram of Ranking of the Hyperactive Environment with Gaussian Distribution Superimposed

Rank	Count
1	3
2	1
3	1
4	2
5	1
6	5
7	10
8	8
9	10

Mean	6.7
------	-----

Standard Deviation	2.35
---------------------------	------

Table 5-13 Count and Measure of Overall Rank of Hyperactive Teaching Tool

5.3 Mann Whitney Tests

As mentioned in Appendix B.1, in order to carry out a Mann Whitney test, the variables must be non-parametric. Given the distributions of student performance and ranking of the hyperactive environment, we are justified in running a Mann Whitney test.

To reiterate, Mann Whitney tests are used to test whether or not a significant difference exist between populations. The tests below which scrutinize different end users are used to determine whether significant differences exist among populations that rank the hypermedia environment and their resulting performance.

5.3.1 Test for Independence of Hypermedia Environment Ranking with Student Prior Exposure to Calibration as the Population

The following evaluation tests for independence between the ranking of the hypermedia environment and past experience with calibration of a rotameter.

Populations	n	ΣR	U
Calibrated a rotameter	3	100.5	19.5
Never calibrated a rotameter	38	94.5	94.5

α	0.10
z	
Calculated	-1.88
Critical	1.96

Table 5-14 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Prior Exposure to Calibration as the Population

Since the magnitude of the calculated z value is less than that of the critical value, there is no statistically significant difference between the ranking of the hyperactive environment between students who have and have never before calibrated a rotameter; the ranking is independent of this experience.

5.3.2 Test for the Independence of Hypermedia Environment Ranking with Student Year as the Population

The following evaluation tests for independence between ranking of the hypermedia environment and the year of the student.

Populations	n	ΣR	U
Third-year undergraduates	26	578.5	162.5
Fourth-year undergraduates	15	282.5	227.5

α	0.10
z	
Calculated	-0.88
Critical	1.96

Table 5-15 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Year as the Population

Since the magnitude of the calculated z value is less than that of the critical value, there is no statistically significant difference between the ranking of the hyperactive environment between third- and fourth-year students. The ranking is independent of year in school.

5.3.3 Test for the Independence of Hypermedia Environment Ranking with Student Preference of Lab Environment as the Population

The following evaluation tests for independence between ranking of the hypermedia environment and preference for a lab environment.

Populations	n	ΣR	U
Actual lab	14	194	205
Web lab	21	436	89

α	0.10
z	
Calculated	-1.95
Critical	1.96

Table 5-16 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Preference of Lab Environment as the Population

Since the magnitude of the calculated z value is less than that of the critical value, there is no statistically significant difference between the ranking of the hyperactive environment between students who prefer an actual lab setting and those who prefer running the lab in a hyperactive environment. The ranking is independent of preference for a particular lab environment.

The three tests conducted all indicate that the ranking of the hyperactive environment is independent of student year in school, experience with rotameter calibration and preference for laboratory environment.

5.3.4 Test for the Independence of Student Performance with Student Prior Exposure to Calibration as the Population

The following evaluation tests for independence between student performance and past experience with calibration.

Populations	n	ΣR	U
Calibrated a rotameter	3	47.5	72.5
Never calibrated a rotameter	38	813.5	41.5

α	0.10
----------	------

z	
Calculated	-0.78
Critical	1.96

Table 5-17 Results of a Mann Whitney Test for Independence of Student Performance with Student Prior Exposure to Calibration as the Population

Since the magnitude of the calculated z value is less than that of the critical value, there is no statistically significant difference between student performance between students who have and have never before calibrated a rotameter. The score on the lab report is independent of this experience.

5.3.5 Test for Independence of Student Performance with Student Year as the Population

The following evaluation tests for independence between student performance and student year.

Populations	n	ΣR	U
Third-Year Undergraduates	26	497.5	243.5
Fourth-Year Undergraduates	15	363.5	146.5

α	0.10
----------	------

z	
Calculated	-1.31
Critical	1.96

Table 5-18 Results of a Mann Whitney Test for Independence of Student Performance with Student Year as the Population

Since the magnitude of the calculated z value is less than that of the critical value, there is no statistically significant difference between student performance between third- and fourth-year students. The score on the lab report is independent of year in school.

5.3.6 Test for Independence of Student performance with Student Preference of Lab Environment as the Population

The following evaluation tests for independence between student performance and preference for a lab environment.

Populations	n	ΣR	U
Actual Lab Environment	14	250.5	148.5
Web Lab Environment	21	379.5	145.5

α	0.10
----------	------

z	
Calculated	-0.051
Critical	1.96

Table 5-19 Results of a Mann Whitney Test for Independence of Hypermedia Environment Ranking with Student Preference of Lab Environment as the Population

Since the magnitude of the calculated z value is less than that of the critical value, there is no statistically significant difference between student performance between students who prefer an actual lab setting and those who prefer running the lab in a hyperactive environment. The score on the lab report is independent of preference for a particular lab environment.

The three tests conducted all indicate that student performance is independent of student year in school, experience with rotameter calibration and preference for laboratory environment.

5.4 Chi-Squared (χ^2) Tests

“The χ^2 test tells you whether the two independent samples...have significantly different distributions across the two categories...and may thereby be considered to have been drawn from different populations” [10]. Again, because our variables are non-parametric, we are justified in our use of the Chi-Squared test.

5.4.1 Test for Student Experience with Calibration Regarding Ranking of the Hyperactive Environment

The following evaluation tests whether students who have and have not calibrated a rotameter come from significantly different population using their ranking of the hypermedia environment as the category for evaluation.

Populations	Hypermedia Environment Ranking									Total
	1	2	3	4	5	6	7	8	9	
Calibrated a rotameter	0	0	0	0	0	0	0	1	2	3
Never calibrated a rotameter	3	1	1	2	1	5	10	7	8	38
Totals	3	1	1	2	1	5	10	8	10	41

DOF	8
α	0.10
χ^2	
Calculated	4.5
Critical	13.36

Table 5-20 Chi Squared (χ^2) Test for Student Experience with Calibration Regarding Ranking of the Hyperactive Environment

Since the magnitude of the calculated χ^2 value is less than that of the critical value, there is no statistically significant difference between the distribution of students who have and have never calibrated a rotameter regarding the rankings of the hyperactive environment. The rankings of the hyperactive environment are similar between these two populations.

5.4.2 Test for Student Year in School with Calibration Regarding Ranking of the Hyperactive Environment

The following evaluation tests whether third- and fourth-year undergraduates come from significantly different population using their ranking of the hypermedia environment as the category for evaluation.

n	41
---	----

Count Populations	Hypermedia Environment Ranking									Total
	1	2	3	4	5	6	7	8	9	
Third-year undergraduates	1	0	0	2	1	3	6	7	6	26
Fourth-year undergraduates	2	1	1	0	0	2	4	1	4	15
Total	3	1	1	2	1	5	10	8	10	41

DOF	8
α	0.10

χ^2	
Calculated	8.49
Critical	13.36

Table 5-21 Chi Squared (χ^2) Test for Student Year in School with Calibration Regarding Ranking of the Hyperactive Environment

Since the magnitude of the calculated χ^2 value is less than that of the critical value, there is no statistically significant difference between the distribution of third- and fourth-year students regarding the rankings of the hyperactive environment. The rankings of the hyperactive environment are similar between third- and fourth-year students.

5.4.3 Test for Student Preference for Environment Regarding Ranking of the Hyperactive Environment

The following evaluation tests whether students who prefer an actual versus simulated environment come from significantly different population using their ranking of the hypermedia environment as the category for evaluation.

$$n \mid 41$$

Population	Hypermedia Environment Ranking									
	1	2	3	4	5	6	7	8	9	Total
Preference for actual lab	1	1	0	2	1	2	4	1	2	14
Preference for web lab	2	0	1	0	0	1	4	6	7	21
Undecided	0	0	0	0	0	2	2	1	1	6
Total	2	1	1	2	1	5	10	8	10	41

$$\text{DOF} \mid 16$$

$$\alpha \mid 0.10$$

$$\chi^2$$

$$\text{Calculated} \mid 16.42$$

Critical | 23.54

Table 5-22 Chi Squared (χ^2) Test for Student Preference for Environment Regarding Ranking of the Hyperactive Environment

Since the magnitude of the calculated χ^2 value is less than that of the critical value, there is no statistically significant difference between the distribution of students who prefer actual versus simulated environments regarding the rankings of the hyperactive environment. The rankings of the hyperactive environment are similar between the two populations.

These three tests indicate that the ranking of the hyperactive environment has similar distributions among populations of student year in school, experience with rotameter calibration and preference for laboratory environment.

5.4.4 Test for Student Experience with Calibration Regarding Student Performance

The following evaluation tests whether students who have and have not calibrated a rotameter come from significantly different population using their performance as the category for evaluation.

n | 41

Count | Scores

Populations	1.8	2.0	3.2	3.3	3.5	3.6	3.7	3.8	4.0
Calibrated a rotameter	0	0	0	0	1	0	0	0	1
Never calibrated a rotameter	1	1	1	1	8	1	2	1	1
Totals	1	1	1	1	9	1	2	1	2

Count Populations	Scores									Total
	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	5.0	
Calibrated a rotameter	1	0	0	0	0	0	0	0	0	3
Never calibrated a rotameter	0	3	2	1	2	5	2	4	2	38
Totals	1	3	2	1	2	5	2	4	2	41

DOF 17

α 0.10

χ^2

Calculated 20.52

Critical 24.77

Table 5-23 Chi Squared (χ^2) Test for Student Experience with Calibration Regarding Student Performance

Since the magnitude of the calculated χ^2 value is less than that of the critical value, there is no statistically significant difference between the distribution of students who have and have never calibrated a rotameter regarding their performance. The performance of students is similar between these two populations.

5.4.5 Test for Student Year in School Regarding Student Performance

The following evaluation tests whether third- and fourth-year undergraduates come from significantly different population using their performance as the category for evaluation.

n | 41

Count Populations	Scores								
	1.8	2.0	3.2	3.3	3.5	3.6	3.7	3.8	4.0
Third-year Undergraduates	0	1	1	1	7	0	2	1	1
Fourth-year Undergraduates	1	0	0	0	2	1	0	0	1
Totals	1	1	1	1	9	1	2	1	2

Count Populations	Scores									Total
	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	5.0	
Third-year Undergraduates	1	0	2	1	1	4	1	1	1	26
Fourth-year Undergraduates	0	3	0	0	1	1	1	3	1	15

Totals	1	3	2	1	2	5	2	4	2	41
DOF	17									
α	0.10									
χ^2										
Calculated	18.99									
Critical	24.77									

Table 5-24 Chi Squared (χ^2) Test for Student Year in School with Calibration Regarding Student Performance

Since the magnitude of the calculated χ^2 value is less than that of the critical value, there is no statistically significant difference between the distribution of students by year regarding their performance. The performance of students is similar between these two populations.

5.4.6 Test for Student Preference for Environment Regarding Student Performance

The following evaluation tests whether students who prefer an actual versus simulated environment come from significantly different population using their performance as the category for evaluation.

$$n \mid 41$$

Count Populations	Scores								
	1.8	2.0	3.2	3.3	3.5	3.6	3.7	3.8	4.0
Actual Lab Environment	0	1	1	0	3	0	2	0	1
Web Lab Environment	1	1	0	1	5	1	0	1	1
Undecided	0	0	0	0	1	0	0	0	0
Totals	1	1	1	1	9	1	2	1	2

Count Populations	Scores									Total
	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	5.0	
Actual Lab Environment	0	0	0	0	1	3	0	1	1	14
Web Lab Environment	1	2	2	0	1	1	1	2	1	21
Undecided	0	1	0	1	0	1	1	1	0	6
Totals	1	3	2	1	2	5	2	4	2	41

DOF 34

α 0.10

χ^2

Calculated 28.34

Critical | 44.90

Table 5-25 Chi Squared (χ^2) Test for Student Preference for Environment Regarding Student Performance

Since the magnitude of the calculated χ^2 value is less than that of the critical value, there is no statistically significant difference between the distribution of students who prefer different environments regarding their performance. The performance of students is similar between these two populations.

These three tests indicate that student performance has similar distributions among populations of student year in school, experience with rotameter calibration and preference for laboratory environment.

5.5 Association Rules

As mentioned in the Chapter 3, association rules are useful for determining which variables are dependent on each other. They represent connections between factors.

The following are the significant rules that were discovered in the Chemical Engineering Rotameter Experiment database. These were accomplished using a minimum support of 5 and a minimum confidence of 25%. The attributes used were student year, student lab preference, student experience with rotameter calibration, student ranking of the hyperactive environment, and student performance.

The rules are of the form $X \rightarrow Y$ (support, confidence) where X is the rule body, Y is the rule head, minsupp is the minimum support, and minconf is the minimum confidence.

Some of the rules we discovered are shown below. As an example of the usefulness of association rules, Rule 1 states that if a student prefers a hyperactive lab, there is a 62% chance he or she is a junior and 13 of the 41 cases are in support of this claim. Rule 10 states that, with a 67% assurance, one can claim that if a student is a junior who prefers an actual lab environment, he or she has no experience with calibration and will receive a score of a 'B' on the lab report. 6 of the 41 cases attest to this claim.

Rule	X	Y	Support	Confidence
1	Lab Environment = Hyperactive	Student Year = Junior	13	62%
2	Student Year = Junior	Lab Environment = Hyperactive	13	50%
3	Experience with Calibration = 'No' AND Student Performance = 'B'	Student Year = Junior AND Lab Environment = Hyperactive	8	29%
4	Experience with Calibration = 'No' AND Lab Environment = Actual	Student Year = Junior AND Student Performance = 'B'	6	43%
5	Experience with Calibration = 'No' AND Lab Environment = Hyperactive	Student Year = Junior AND Student Performance = 'B'	8	44%
6	Experience with Calibration = 'No' AND Lab Environment = Hyperactive	Student Year = Senior AND Student Performance = 'B'	5	28%

Rule	X	Y	Support	Confidence
7	Student Year = Junior AND Student Performance = 'B'	Experience with Calibration = 'No' AND Lab Environment = Actual	6	29%
8	Student Year = Junior AND Student Performance = 'B'	Experience with Calibration = 'No' AND Lab Environment = Hyperactive	8	38%
9	Student Year = Senior AND Student Performance = 'B'	Experience with Calibration = 'No' AND Lab Environment = Hyperactive	5	50%
10	Student Year = Junior AND Lab Environment = Actual	Experience with Calibration = 'No' AND Student Performance = 'B'	6	67%
11	Student Year = Junior AND Lab Environment = Hyperactive	Experience with Calibration = 'No' AND Student Performance = 'B'	8	62%
12	Student Year = Senior AND Lab Environment = Hyperactive	Experience with Calibration = 'No' AND Student Performance = 'B'	5	63%
13	Student Year = Junior AND Experience with Calibration = 'No'	Lab Environment = Actual AND Student Performance = 'B'	6	26%

Rule	X	Y	Support	Confidence
14	Student Year = Junior AND Experience with Calibration = 'No'	Lab Environment = Hyperactive AND Student Performance = 'B'	8	35%
15	Student Year = Senior AND Experience with Calibration = 'No'	Lab Environment = Hyperactive AND Student Performance = 'B'	5	33%
16	Student Performance = 'B'	Student Year = Junior AND Experience with Calibration = 'No' AND Lab Environment = Hyperactive	8	26%
17	Lab Environment = Actual	Student Year = Junior AND Experience with Calibration = 'No' AND Student Performance = 'B'	6	43%
18	Lab Environment = Hyperactive	Student Year = Junior AND Experience with Calibration = 'No' AND Student Performance = 'B'	8	38%

Table 5-26 Association Rules Relevant to Student Year, Student Experience with Rotameter Calibration, Student Preference for a Laboratory Environment, Student ranking of the Hyperactive Environment, and Student Performance

The reliability of the association rules strongly depends on the number of records available and the perceived assurance on the part of the researcher or instructor. The above results were generated based on a conservative confidence and should be reassessed with further use of the

rotameter experiment. In order to get precise associations, more data is required to get more acceptable estimates of the minimum support and minimum confidence.

5.6 Decision Trees

Decision trees are used to determine how the goal is attained given other factors. With regards to hyperactive environments, the goal is two-fold:

1. determination of how well end users perform and
2. determination of how well the environment is ranked.

As mentioned in the previous chapter, the sequence of the non-goal attributes from the root of the decision tree represents an ordering of information that is most critical in the determination of the goal attribute. Those that follow are ordered in terms of importance.

Part of the above determinations involves an understanding how well end users perform and how well the environment is ranked given such factors as past exposure to calibration, year in school, preference for a lab environment, and expected ease of use of the actual experiment.

5.6.1 Determination of How Well the End Users Perform

One goal in creating hyperactive learning environments is to assess their ability at allowing users to perform as well as they would in a more traditional learning environment. The most critical factor in the determination of student performance is the student's year.

Regardless of the year of the student, the past experience with calibration of a rotameter, the choice of web lab, and the notion that the use of the web lab would facilitate the use of the actual lab, the most likely score is a B. The only case in which this does not hold is for third-year undergraduates who were undecided about their preferred choice of lab. See Figure 5-3 for more details.

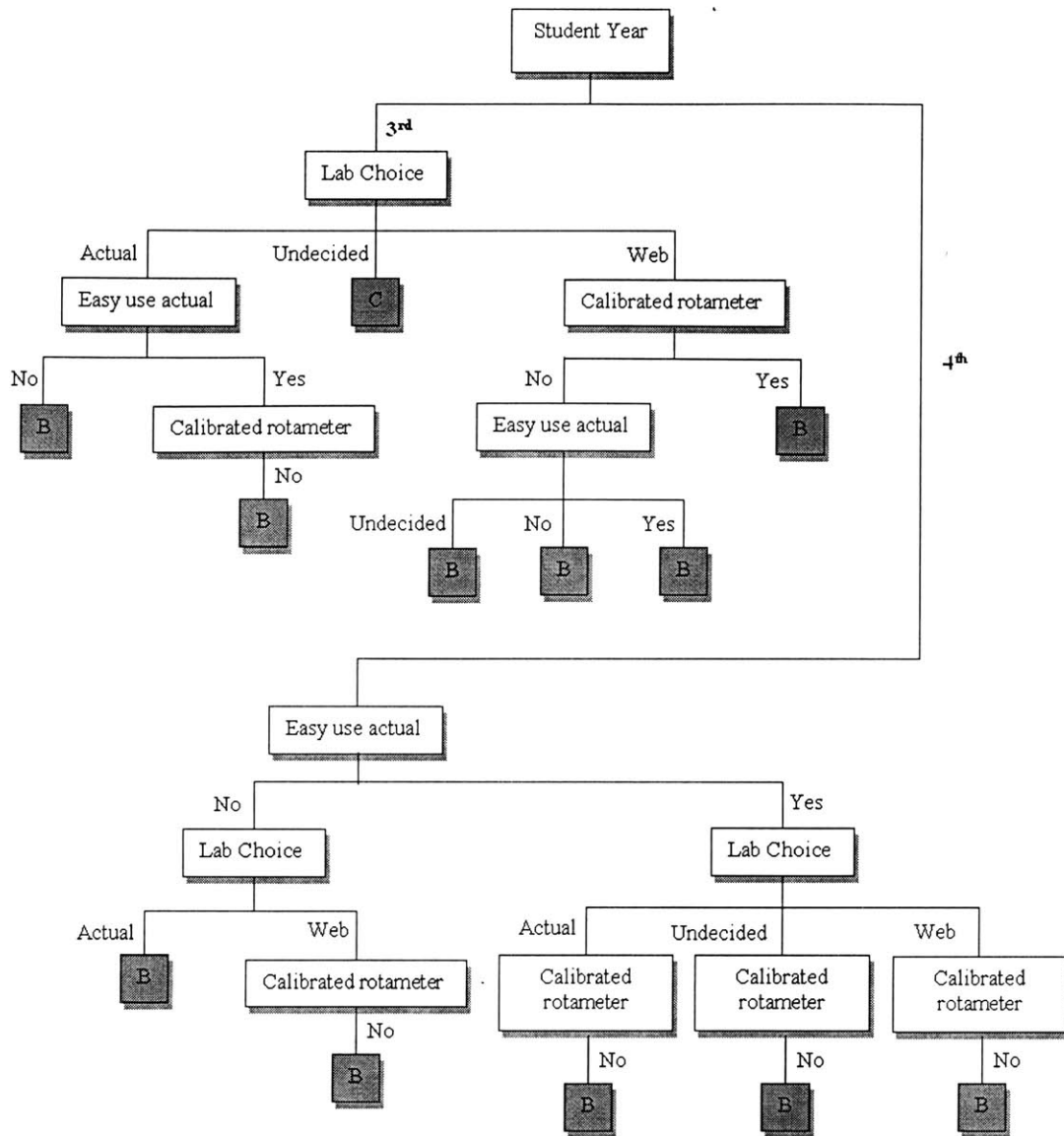


Figure 5-3 Decision Tree for Determination of Student Performance with Student Year, Preference for Lab Environment, Experience with Rotameter Calibration, and Supposed Ease of Use as Categories

5.6.2 Determination of How Well the Environment is Ranked

The basis for determining how well the hyperactive environment is ranked is critically dependent on the end users. In the case of the ranking of the hyperactive environment, the most critical attribute is the preference for a lab environment.

For students who prefer to use the actual lab, third- and fourth-year students who thought that using the web lab would make it easier to use the actual lab and who had never before calibrated a rotameter ranked the hyperactive environment a 7 on a scale of 10. Third-year students who did not think the web lab would make it easier to use the actual lab gave the hyperactive environment a rank of 7 as well; one fourth-year students who did not think the web lab would make the actual lab easier to perform gave it the lowest rank of a 1.

For students who prefer to the web lab, third-year students who thought it would be easy to calibrate an actual rotameter and had never before calibrated a rotameter gave the hyperactive environment an overall rank of 8. Those who had calibrated a rotameter prior to the use of the hyperactive environment ranked it a 9 out of 10. For third-year students who did not think using the web lab would make it easy to use the actual lab nevertheless gave the hyperactive environment a rank of 8 out of 10. Fourth-year students who had never calibrated a rotameter gave a low ranking of 3. Students who were undecided as to whether or not the web lab would make it easy to calibrate an actual rotameter ranked the hyperactive environment as 1.

Lastly, both third- and fourth-year students who had never before calibrated a rotameter but nevertheless thought the hyperactive environment would make it easier to use the actual experiment gave it a rank of 7.

With the above information in mind, the decision tree in Figure 5-4 represents the ranking of the hyperactive environment with end user characteristics and preferences as the defining attributes.

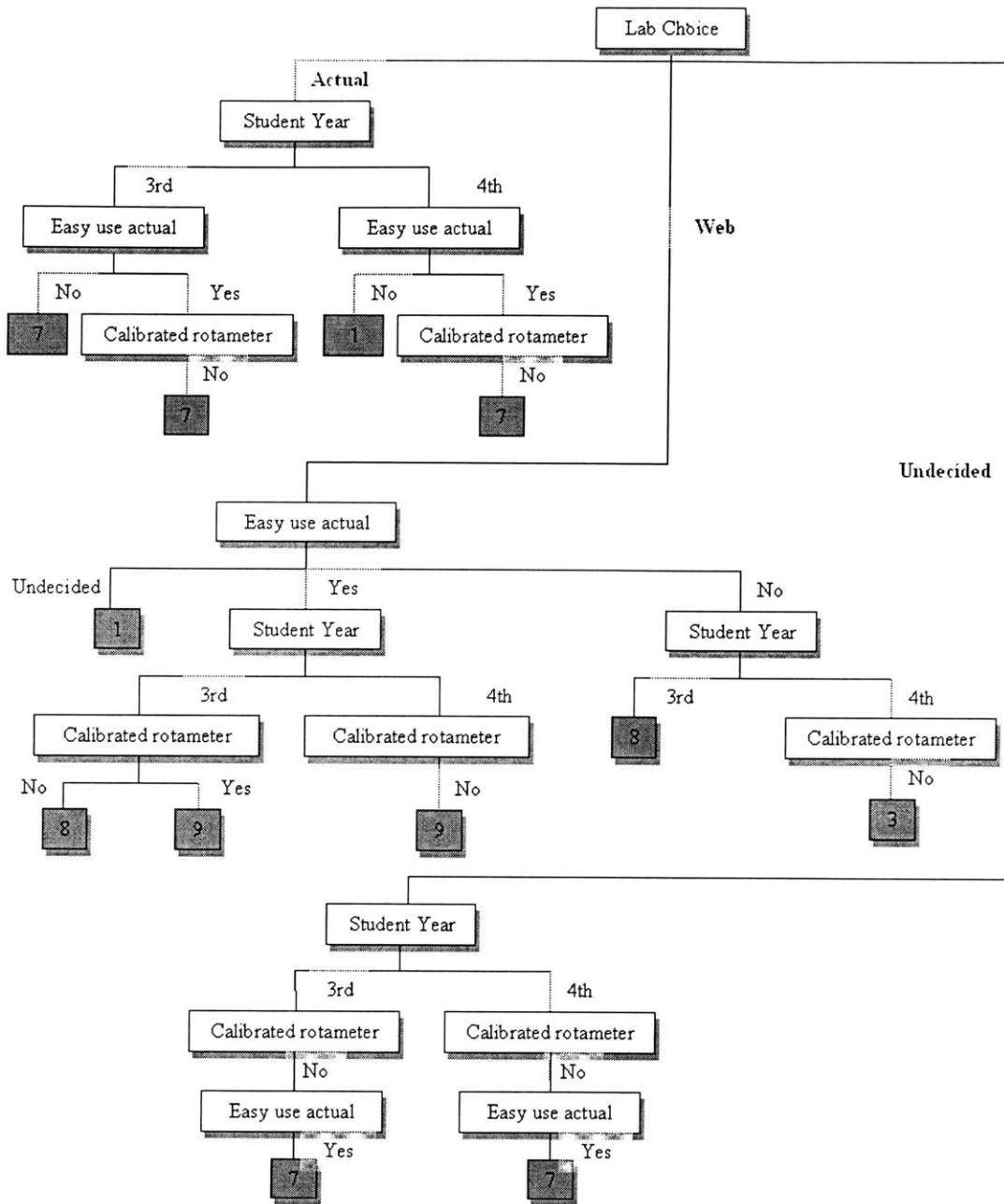


Figure 5-4 Decision Tree for Determination of Hyperactive Environment Rank with Student Year, Preference for Lab Environment, Experience with Rotameter Calibration, and Supposed Ease of Use as Categories

Chapter 6

Conclusions and Further Work

The methodology and analyses proposed were carried out on the Chemical Engineering Rotameter Experiment. Although the sample space was limited and the methodology was employed for the first time, favorable findings resulted.

Some of the conclusions we were able to discover are:

1. Performance on the part of student to generate a technical report through the use of the hyperactive environment is as good as that of students who would do the same using an actual lab setting.
2. For students that have not had prior exposure to the material of a particular simulation or experiment, they are still able to perform as well as those who have through the use of the hyperactive environment. The same result held for third-year undergraduates as compared to fourth-year students.
3. A majority of the students prefer the use of the hyperactive environment to the use of the actual lab environment. This preference is grounded on issues concerning the flexibility associated with the use of a hyperactive environment with regards to time and data collection, ease of use of the environment, and the relative ease of operating the hyperactive environment.
4. In most cases, students are in strong agreement with the idea that the web-based environment is a suitable one for achieving the given learning objectives. In cases where students are not in favor of the use of the hyperactive environment, the same

feelings would probably hold for the actual lab setting, these being rooted in the ability or lack thereof, on the part of the student to interpret results and extract information upon completion of the experiment.

5. Students strongly agree that the hyperactive environment is suitable at the interfacial level, making it easy to navigate between the experiment and instruction as well as facilitating the collection of data.

Through the use of the methodology described in Chapter 1 and the techniques applied throughout the thesis, better design and analysis can be employed to improve the utility and effectiveness of web-based learning environments as shown for the Chemical Engineering Rotameter Experiment.

The proposed methodology may be extended in the future to support the identification of which statistical tests to run on given data sets, the automated discovery of the critical values of minimum support and confidence for association rules, and better graphical representation of the decision trees. In addition, the statistical analyses used were all non-parametric given the distributions of data variables; the development of parametric counterparts needs to be addressed.

The philosophy of the Hypermedia Teaching Facility is that a balance between actual and simulated experimentation maximizes the information retained by the students thereby improving their capacity for retention and comprehension. However, the continual use of web-based experiments may make them the mode of choice in the future.

Appendix A

This appendix contains information relevant to the description of the hyperactive environments developed and their file structure.

A.1 Carnot Cycle Simulation

Filename	Description
Cylinder.html	consists of HTML which takes input from the user, displays the VRML plug-in and the applet for displaying the P-V and T-S diagrams
DataEntry.js	consists of HTML and JavaScript code for passing along the user input to the Java applet
Cylinder3.wrl	consists of the VRML code for displaying and interacting with the piston-cylinder apparatus
ExternalApplet.java	contains the Java code for interacting with the VRML world and calculating and plotting the data points for the P-V and T-S diagrams

Table A.1-1 Files Used by the Carnot Cycle Simulation

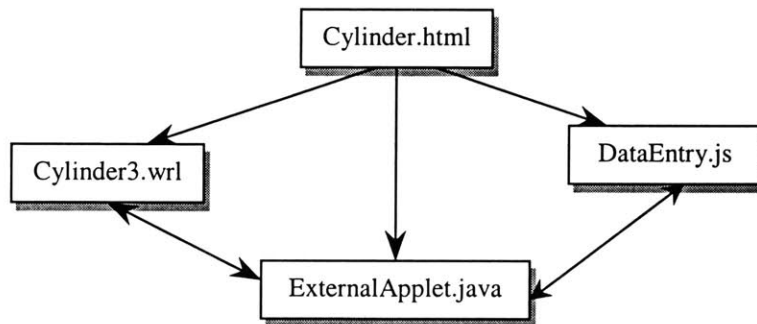


Figure A. 1-1 Interactions between Files in the Carnot Cycle Simulation

A.2 Airline Yield Management Simulation

This appendix contains information related to the file structure and file interactions for the Airline yield Management Simulation.

Filename	Description
BApplet.java	Java file that spawns the application from a browser
BookingApplication.java	Java file that contains user interface and main code
Arguments.java	contains Java code that takes arguments from the user and runs the appropriate control level
Booking.java	Java code that keeps track of the bookings made for a given simulation
GraphCanvas.java	Java code that presents graphical information regarding the bookings made to the user
InfoCanvas.java	Java code that displays information relevant to the booking process to the user
PlaneCanvas.java	Java code that displays color-coded seats during the booking periods
IntermediatePanel.java	Java code that provides the user with information necessary to make a decision concerning restricted bookings in current and future booking periods
ResultsPanel.java	Java code that displays final results (i.e. total reservations made, total revenue, etc.) of the booking periods to the user
FrontBodyPanel.java, RearBodyPanel.java, BodyPanel.java	Java code that displays images of the airplane
TitlePanel.java	Java Code that displays the title of the simulation

Table A.2-2 Files used by the Airline Yield Management Simulation

A.3 Chemical Engineering Rotameter Experiment

Filename	Description
Index.html	HTML code that contains objective of the experiment, instructions for use, and description of equipment
Experiment.html	HTML code that positions content in the browser window and refers to the JavaScript file for handling user interactions
Rotameter.js	JavaScript file that responds to user interaction and contains programming logic

Table A.3-3 Files used by the Chemical Engineering Rotameter Experiment

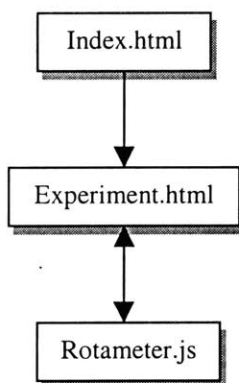


Figure A. 3-2 Interactions between Files in the Chemical Engineering Rotameter Experiment

Appendix B

B.1 Mann-Whitney Test

B.1.1 Variable Requirements

The Mann-Whitney U-Test requires the use of a nominal independent or control variable and an ordinal dependent variable.

B.1.2 Testing Procedure

The first step in carrying out the Mann-Whitney U-Test is the selection of the two samples. Having done this, the dependent variable from each sample is chosen.

The worth of the U-Test is in comparing the ranks of the dependent variables from each sample. In order to do so, the sample dependent variables are combined into one group and assigned ranks. The two samples are then regrouped and a sum is calculated for each group rank.

Next, the U-value for each group is calculated. This calculation is outlined below.

The U value is given by

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1 \quad (\text{Eq. B.1-1})$$

and

(Eq. B.1-2)

$$U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2$$

where

n_1 = size of the first sample,

n_2 = size of the second sample,

R_1 = sum of the ranks for the first sample, and

R_2 = sum of the ranks for the second sample

These above steps are represented in the table below.

Step 1	Selection of two sample groups
Step 2	Selection of dependent variables from each sample group
Step 3	Combine dependent variables
Step 4	Rank order combined variables
Step 5	Separate rank-ordered variables in respective groups
Step 6	Calculate U-values for each sample group

Table B.1-1 Steps used in calculating the U-value for a Mann-Whitney Test

B.2 Chi-Squared (χ^2) Test

B.2.1 Variable Requirements

In most cases, nominal variables are used in the χ^2 test.

B.2.2 Calculation of χ^2

The first step in the calculation of χ^2 is the establishment of a contingency table. This table contains observed frequencies in the data arranged in r rows and k columns. The variable r represents the number of categories in the outcomes; the variable k represents the number of categories for the populations.

Second, the obtained frequencies are calculated for each cell of the contingency table. The obtained frequency is the number of occurrences observed in the data samples.

Third, the marginals, which are the sums of the frequencies across the rows and columns, are calculated.

Having done this, the expected frequencies, which relate the frequencies of observation to the total frequency, are calculated. The expected frequency for each cell is equal to the product of its marginals divided by the total frequency of all observations (the sum of all obtained frequencies).

With this information, χ^2 can be calculated as

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^k \frac{(f_{ij} - e_{ij})^2}{e_{ij}} \quad (\text{Eq. B.2-1})$$

where

f_{ij} = observed frequencies, and

e_{ij} = expected frequencies

These above steps are represented in the table below.

Step 1 Determine the number of categories and populations and establish an $r \times k$

	contingency table
Step 2	Calculate the observed frequency for each cell in the contingency table
Step 3	Calculate the marginals of the contingency table
Step 4	Calculate the expected frequencies for each cell in the contingency table
Step 5	Calculate the χ^2 value

Table B.2-2 Steps used in calculating the χ^2 value for the Mann-Whitney Test

B.3 Association Rules

B.3.1 Description

Given a set of attributes $I = \{i_1, i_2, \dots, i_n\}$, a set of attribute-containing transactions

$T = \{t_1, t_2, \dots, t_n\}$, and a set of variable length transactions $D = \{d_1, d_2, \dots, d_n\}$, an association rule

is an expression of the form $X \rightarrow Y$ where X is the rule body, Y is the rule head, $X, Y \subset I$ and

$X \cap Y = \emptyset$.

An association rule is defined by its

1. Support. A given percentage, s %, of the transactions fulfill the requirement that the statement contain $X \cup Y$.
2. Confidence. A given percentage, c %, of transactions contain $X \cup Y$ as a percentage of those that contain X .

B.3.2 Determination of Desired Association Rules

In generating association rules, the objective is in determining those that meet minimum required support and confidence. These terms are defined as follows:

- **Minimum support.** The minimum number of transactions that contain both the rule body and the rule head. The generated association rules must have $\text{supp}(X \cup Y) \geq \text{minsupp}$.
- **Minimum confidence.** The reliability that the number of records that contain both X and Y is a certain percentage of those that contain X. The generated association rule must have $\text{supp}(X \cup Y) / \text{supp}(X) \geq \text{minconf}$.

By setting low values for the minimum support and confidence, more association rules will be generated; increasing these values will generate fewer rules.

In addition to the above approach, another means of narrowing down the number of generated rules is by selecting which attributes to include in the association rules. There are cases in which there are no associations between attributes that will fulfill the minimum support and confidence requirements (e.g. associations between attributes username and score by virtue of the username being a unique identifier).

These steps taken to generate association rules are reiterated in the table below.

Step 1	Provide the required minimum support and confidence
Step 2	Provide the attributes desired in the generated association rules

Table B.3-1 Steps used in generating association rules

B.3.3 Data Structure and Algorithm

The following data structure defines the association between the different attributes and attribute distinct values. This diagram further defines the algorithm that will be used to generate association rules.

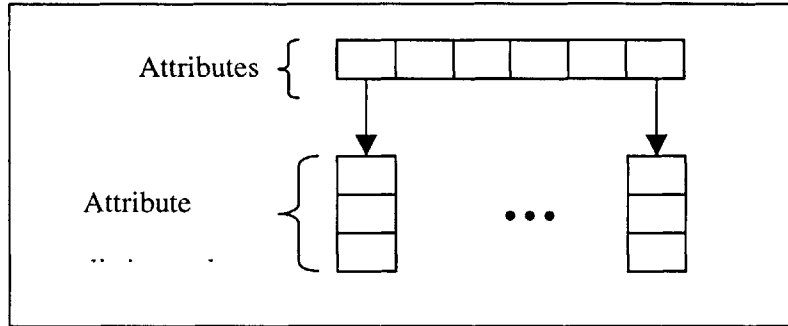


Figure B.3-1 Data Structure Used in Determining Association Rules

The algorithm used in determining which association rules to generate is based on combinatorics. Essentially, it determines all combinations of attribute distinct values for combinations of the different attributes taken k at a time where k can be any value from 1 to $(n-1)$ where n is the number of distinct values.

B.4 Decision Trees

B.4.1 Description

A decision tree is a tree structure in which each node represents a non-goal attribute and each branch signifies the value of that attribute. The leaves of the tree represent the values of the goal attribute.

Given a set of transactions $T = \{t_1, t_2, \dots, t_n\}$ each containing the same number of attributes from the itemset $I = \{i_1, i_2, \dots, i_n\}$, it is desired to produce a decision tree that determines which “non-goal attributes predict correctly the value of the goal attribute” [11].

A key aspect in determining the best decision tree is the selection of the non-goal attributes in the order that provides the most critical information first. In order to do so, the concepts of entropy and gain are introduced.

Before defining entropy and gain, we define the probability distribution, P, as the magnitude of each value of the attribute. P is given by

$$P = \left(\frac{|c_1|}{|T|}, \frac{|c_2|}{|T|}, \dots, \frac{|c_n|}{|T|} \right) = (p_1, p_2, \dots, p_n) \quad (\text{Eq. B.4-1})$$

where

c represents a value of the goal attribute,

T represents the transactions and

the magnitude, | |, represents the number of attributes with a given value.

Entropy is the information contained in the probability distribution and is given by:

$$I(P) = -(p_1 * \log(p_1) + p_2 * \log(p_2) + \dots + p_n * \log(p_n)) \quad (\text{Eq. B.4-2})$$

where

log(p) is taken in base 2

The decision about which non-goal attribute to choose as the current node in the decision tree is determined from the calculation of the gain among the remaining attributes left to add to the tree.

The gain is calculated as:

$$\text{Gain}(X,T) = I(T) - I(X,T) \quad (\text{Eq. B.4-3})$$

and

$$I(X,T) = \sum_{i=1}^{n.} \frac{|X_i|}{|T|} * I(X_i) \quad (\text{Eq. B.4-4})$$

where

X is a non-goal attribute,

I(T) is the information calculated with all values of X and

I(X) is the information calculated with T being the records with attribute X

The non-goal attribute with the highest gain is chosen as the current node in the decision tree.

The leaves of the node are the attribute values.

B.4.2 Data Structure and Algorithm

Given that each attribute may have a variable number of values, the most practical choice for data structure is an n-ary tree. This allows for the representation of the variable number of branches outward from a node in the tree. The diagram below represents an n-ary tree where the nodes represent attributes and branches represent attribute values.

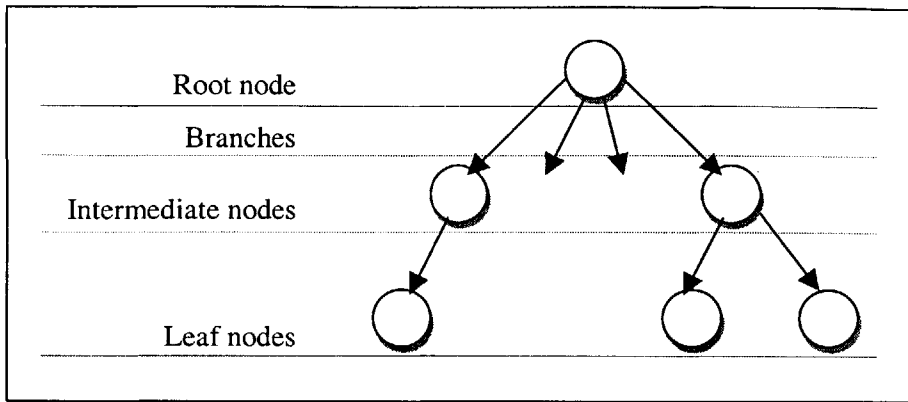


Figure B.4-1 Data Structure Used in Building Trees

The algorithm used in generating the decision tree is the ID3 algorithm. Given a set of non-goal attributes, the goal attribute, and a data set, the ID3 algorithm creates a tree structure that presents a decision process for arriving at the goal attribute. The ordering of the non-goal attributes represents a sequence of attributes that convey the most information, otherwise referred to as entropy, in achieving values of the goal attribute. This algorithm is given below.

```

function ID3( R: a set of non-goal attributes,
              C: the goal attribute,
              S: a training set) returns a decision tree;

begin
  If S is empty, return a single node with value Failure;
  If S consists of records all with the same value for the goal
  attribute, return a single node with that value;
  If R is empty, return a single node with as value the most
  frequent of the values of the goal attribute that are
  found in records of S; [note that then there will be
  errors, that is, records that will be improperly
  classified];
  Let D be the attribute with the largest Gain(D,S) among
  attributes in R;
  Let {dj | j=1,2, ..., m} be the values of attribute D;
  Let {Sj | j=1,2, ..., m} be the subsets of S consisting
  respectively of records with value dj for attribute D;
  Return a tree with root labeled D and arcs labeled
  d1, d2, ..., dm going respectively to the trees

  ID3 (R-{D}, C, S1), ID3 (R-{D}, C, S2), ..., ID3 (R-{D}, C, Sm);

end ID3;

```

Figure B.4-2 Algorithm Used in Building Binary Tree [11]

References

- [1] Musciano, C, and Kennedy B, *HTML: The Definitive Guide*, First Edition, Sebastopol, CA: O'Reilley & Associates, Inc., 1996
- [2] Gundavaram, S, *CGI Programming on the World Wide Web*, First Edition, Sebastopol, CA: O'Reilley & Associates, Inc., 1996
- [3] Flanagan, D., *JavaScript: The Definitive Guide*, Second Edition, Sebastopol, CA: O'Reilley & Associates, Inc., 1997
- [4] Naughton, P., *The Java Handbook*, Berkeley, CA: Osborne McGraw Hill, 1996
- [5] Elmasri, R., and Navathe, S.B., *Fundamentals of Database Systems*, Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc., 1989
- [6] Sanchez, A., *Data Warehousing with Informix: Best Practices*, Upper Saddle River, NJ: Prentice Hall, Inc., 1998 MA: Kluwer Academic Publishers, 1997
- [7] Informix Software, Inc., *Informix Guide to Database Design and Implementation*, Menlo Park, CA: INFORMIX Press, 1991-1998
- [8] Adriaans, P., and Zantinge, D., *Data Mining*, Harlow, England: Addison-Wesley Publishing Co., 1996
- [9] Harrington, J.L., *SQL Clearly Explained*, Chestnut Hill, MA: Academic Press, 1998
- [10] Tuckman, B.W., *Conducting Educational Research*, Fort Worth, TX: Harcourt Brace & Company, 1994
- [11] Ingargiola, G., *Building Classification Models: ID3 and C4.5*,
<http://yoda.cis.temple.edu:8080/UGAIWWW/lectures95/learn/C45/index.html>
- [12] Cravalho, E.G., and Smith, J.L., *Engineering Thermodynamics*, Cambridge, MA: Third Printing, August 1992

Bibliography

- Agrawal, R., Imielinski, T., Swami, A., "Database Mining: A Performance Perspective", San Jose, CA: IBM Almaden Research Center
- Amir, A., Feldman, R., and Kashni, R., "A New and Versatile Method for Association Generation", *Information Systems*, Vol. 22, No. 6/7, pp. 333-347, Great Britain: Elsevier Science Ltd., 1997
- Bhandari, I., Colet, E., Parker, J., Pines, Z., Pratap, R., Ramanujam, K., "Advanced Scout: Data Mining and Knowledge Discovery in NBA Data", *Data Mining and Knowledge Discovery*, Boston, MA: Kluwer Academic Publishers, 1997
- Doe, C., *Informix-Online Dynamic Server Handbook*, Upper Saddle River, NJ: Prentice Hall, Inc., 1997
- Duncan, L., *Official Netscape Server-Side JavaScript for Database Applications*, Research Triangle Park, NC: Ventana Communications Group, 1997
- Fayyad, U.M., "Editorial", *Data Mining and Knowledge Discovery*, 1, 5-10, Boston, MA: Kluwer Academic Publishers, 1997
- Freedman, D., Pisani, R., Purves, R., Adhikari, A., *Statistics*, Second Edition, New York, NY: W.W. Norton & Company, Inc., 1991
- Glymour, C, Madigan, D., Pregibon, D., Smyth, P., "Statistical Themes and Lessons for Data Mining", *Data Mining and Knowledge Discovery*, 1, 11-28, Boston, MA: Kluwer Academic Publishers, 1997
- Goodman, D., "The JavaScript Apostle: Java vs. JavaScript – The Impact of Dynamic HTML", http://developer.netscape.com/viewsource/goodman_dhtml.htm
- Kearsley, G.P., *Artificial Intelligence & Instruction: Applications and Methods*, Reading, MA: Addison-Wesley Publishing Company, 1987
- Rosen, K.H., *Discrete Mathematics and Its Applications*, Fourth Edition, McGraw-Hill, Inc., New York, 1995
- Sanchez, A., *Data Warehousing with Informix: Best Practices*, Upper Saddle River, NJ: Prentice Hall, Inc., 1998 MA: Kluwer Academic Publishers, 1997
- Sonwalkar, N., "The Hypermedia Teaching Facility: An Overview", Cambridge, MA: MIT Hypermedia Teaching Facility, 1997
- , "Hypermedia Teaching Facility: Concept, Technology, and Methodology", Cambridge, MA: MIT Hypermedia Teaching Facility, 1997

---, "Statistical Analysis of Hypermedia Teaching facility Usage", Cambridge, MA: MIT Hypermedia Teaching Facility, 1997

---, "Revolutionizing Mechanical Engineering Education via Internet: The Hypermedia Teaching Facility (HTF) Revolutionizing Mechanical Engineering Education via Internet: The Hypermedia Teaching Facility (HTF)

---, "Hypermedia Teaching Facility Changing The Interface Of Higher Education", *The MIT Report*, Vol. XXIV, #9, 3, Cambridge, MA: MIT Industrial Liaison Program, December 1996

Wiggins, R. and Sonwalkar, N. "Formative Evaluation for the Beta Test Phase of the Chemical Hygiene Hypercourse: An Interim Report", Cambridge, MA: MIT Hypermedia Teaching Facility, 1999