A Web Based Interface Metaphor for Presenting Complex Process Knowledge
by
Vuong P. Nguyen

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirement of the Degree of
Bachelor of Science in Computer Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
August 15, 1997

© 1997 M.I.T.
All rights reserved.

Author

Department of Electrical Engineering and Computer Science
August 15, 1997

Certified by

Professor Thomas W. Malone
Thesis Supervisor

Accepted by

Arthur C. Smith
Chairman, Department Committee on Graduate Theses

OCT 29 1997
A Web Based Interface Metaphor for Representing Complex Process Knowledge

by

Vuong Nguyen

Submitted to the

Department of Electrical Engineering and Computer Science

August 15, 1997

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

Abstract

The Process Handbook is a research project currently being developed at the MIT Center for Coordination Science. The goal of the project is to create a repository of business process descriptions which can be used to redesign existing processes, to create new processes, or to learn more about an organization. As this project has matured, the need for a tool that allows multiple users remote access to the Process Handbook has arisen. This thesis focuses on the development of this tool, a Web based graphical user interface (GUI) to the Process Handbook. This interface will allow users in almost any geographic location to navigate and examine the contents of the Process Handbook.

Thesis Supervisor: Thomas W. Malone
Supervisor's Title: Patrick J. McGovern Professor of Information System
Acknowledgments
I would like to thank all of the people at the MIT Center for Coordination Science who have helped me on my thesis. Firstly, I would like to thank my thesis advisor, Professor Thomas Malone, who showed a great deal of interest and enthusiasm for the work I did. I also like to thank Mark Klein and John Quimby for guiding me through the development of the project. They provided much of the knowledge and insight into the design and implementation of the project. Also, I would like to thank George Herman, Steven Buckley, and Abraham(Avi) Berstein for their contributions to my thesis. Finally, I would like to thank my parents for supporting me through my education at this institution.
Table of Contents

CHAPTER 1: INTRODUCTION

1.1 MOTIVATION ................................................................................. 8
  1.1.1 Graphically Displaying Complex Data ........................................... 8
  1.1.2 Providing a Distributed, Collaborative Environment .......... 9
1.2 GOALS ............................................................................................ 9
1.3 SCOPE OF RESEARCH .................................................................. 11

CHAPTER 2: BACKGROUND ................................................................ 12

2.1 THE WORLD WIDE WEB ................................................................. 12
  2.1.1 History of the World Wide Web .................................................. 12
  2.1.2 Hypertext Markup Language ...................................................... 14
  2.1.3 Other Display Technologies Used in the World Wide Web ...... 15
    2.1.3.1 Java .................................................................................. 15
    2.1.3.2 Javascript ......................................................................... 16
    2.1.3.3 Virtual Reality Modeling Language ................................... 16
    2.1.3.4 ActiveX ............................................................................ 17
    2.1.3.5 VBScript ........................................................................... 18
    2.1.3.6 Cookies ............................................................................ 18
  2.1.4 Advantages and Disadvantages of Using the World Wide Web ... 19

CHAPTER 3: DESIGN OF THE INTERFACE ............................................. 21

3.1 DESIGN ISSUES ........................................................................... 21
  3.1.1 Navigational Efficiency ............................................................. 21
  3.1.2 Computer Display Characteristics ............................................ 21
  3.1.3 Choice of Web Browser ........................................................... 22
3.2 WEB TECHNOLOGIES USED .......................................................... 23
3.3 LAYOUT OF THE INTERFACE .......................................................... 23
  3.3.1 The Login Window ................................................................. 23
  3.3.2 The Session Window ............................................................... 26
    3.3.2.1 The “Contents” Screen ....................................................... 26
    3.3.2.2 The “Search” Screen .......................................................... 29
    3.3.2.3 The “Profile” Screen .......................................................... 29
    3.3.2.4 The “Discussion” Screen ................................................... 33
    3.3.2.5 The “Comments” Screen ................................................... 34
    3.3.2.6 The “Help” Screen .............................................................. 35
  3.3.3 The Object Window ................................................................. 37
    3.3.3.1 The Specialization View ..................................................... 38
    3.3.3.2 The Decomposition View .................................................... 40
    3.3.3.3 The Trade-off Matrix View ................................................ 42
    3.3.3.4 The Details View .............................................................. 42
3.3 OTHER FEATURES ................................................................ 48
  3.3.1 Guided Tours ........................................................................... 48

CHAPTER 4: EVALUATION OF THE INTERFACE .................................... 51

4.1 ALTERNATIVE DESIGNS .............................................................. 51
4.2 A COMPARISON OF THE DIFFERENT DESIGNS ......................... 51
  4.2.1 Analytical Comparison of Navigational Efficiency .................. 53
4.3 REVIEWING THE DESIGN GOALS ............................................... 55

CHAPTER 5: FUTURE WORK ................................................................. 58
CHAPTER 6: CONCLUSION ........................................................................................................59

APPENDIX A .......................................................................................................................... 60

A1. THE PROCESS HANDBOOK ............................................................................................. 60
   A1.1 Overview and Goals .................................................................................................. 60
   A1.2 Specializations of Processes .................................................................................. 60
      A1.2.1 Bundles and Trade-off Matrices ..................................................................... 60
      A1.2.2 Advantages of Specialization of Processes .................................................... 62
   A1.3 Coordination Theory ............................................................................................... 62
      A1.3.1 Coordination Processes and Managing Dependencies in the Process Handbook .. 63

APPENDIX B ........................................................................................................................... 64

B.1 TECHNIQUES FOR DISPLAYING HIERARCHICAL INFORMATION ON THE WEB ........ 64
   B.1.1 List Hierarchies ..................................................................................................... 64
   B.1.2 Table Hierarchies .................................................................................................. 65

BIBLIOGRAPHY .................................................................................................................... 67
List of Figures

Figure 2.1 - Diagram of Web architecture from early discussions of the Web in 1990 ........................................ 13
Figure 3.1 - The login window ................................................................. 24
Figure 3.2 - The user registration window ............................................. 25
Figure 3.3 - The session window ............................................................ 26
Figure 3.4 - The contents screen of the session window ......................... 27
Figure 3.5 - A navigation node in the session window ......................... 28
Figure 3.6 - The search form ................................................................. 30
Figure 3.7 - The results of a search of the keywords “produce something” .. 31
Figure 3.8 - The user profile screen ...................................................... 32
Figure 3.9 - The discussion database screen ......................................... 33
Figure 3.10 - The comments screen ...................................................... 34
Figure 3.11 - The table of contents for the help screen ......................... 35
Figure 3.12 - The help page for decompositions .................................. 36
Figure 3.13 - The object window for “Produce automotive component” . 37
Figure 3.14 - The specialization view of “Identify (Gather Info and Select)” .. 39
Figure 3.15 - The decomposition view for “Produce automotive component” ... 41
Figure 3.16 - The trade-off matrix view for “Sell how?” ....................... 42
Figure 3.17 - The description for “Manage usability” ............................ 43
Figure 3.18 - The notes for two different activities named “Send purchase order” ......................................................... 43
Figure 3.19 - The generalization screen for “Identify(Gather Info and Select)” ......................................................... 44
Figure 3.20 - The specialization screen for “Create” ............................... 44
Figure 3.21 - The decomposition for “Produce education and research service” .................. 45
Figure 3.22 - The where used screen for “Acquire automotive component inputs” .... 45
Figure 3.23 - The attributes for “Identify potential customer on telephone” .... 46
Figure 3.24 - The navigational links for “Provide copy capabilities” ........ 46
Figure 3.25 - The other links for “Resolve problems collaboratively - automotive” .... 47
Figure 3.26 - A discussion request link and the form that appears after you click on it .... 47
Figure 3.27 - The trade-off matrix screen for “Sell How?” ....................... 48
Figure 3.28 - A guided tour embedded in the description for the navigation node “Hiring processes” ............................................ 49
Figure A.1 - Summary display showing specializations of the activity “Sell something” .... 61
Figure A.2 - The trade off matrix for “Sell how?” .................................. 61
Figure B.1 - A simple list hierarchy ...................................................... 65
Figure B.2 - The vertical table hierarchy .............................................. 66
Figure B.3 - The horizontal table hierarchy .......................................... 66
List of Tables

Table 3.1 - The object types in the handbook and their geometric symbol........................................... 29
Table 4.1 - Ranking of the designs using the design goals of chapter one........................................... 52
Table 4.2 - Comparisons of the navigational efficiency of each of the designs..................................... 54
Table A.1 - Examples of elementary dependencies between activities and alternative coordination mechanisms for managing them ........................................................................................................ 63
Chapter 1: Introduction
The Process Handbook is a research project currently being developed at the MIT Center for Coordination Science. The goal of the project is to create a repository of business process descriptions from which these processes can be redesigned, used to create new processes, or used to learn more about an organization. Initially, access to the information in the handbook was only available locally to one user at a time, using a software application developed for one operating system. As this project has matured, the need for a tool that allows multiple users remote access to the Process Handbook has arisen. This thesis focuses on the development of this tool, a Web based interface to the Process Handbook. This paper will discuss various techniques used to display information on the World Wide Web and the design choices that were made in the development of the interface. These design choices will be analyzed and future extensions to the current interface will be suggested.

1.1 Motivation
The Process Handbook is intended to aid in the creation of new, innovative business processes. Before new processes can be created or old processes reengineered, users must first be presented with information about existing processes in such a way that it assists them in their understanding of the processes. It would be also beneficial if the information stored in the handbook could be viewed by users, regardless of their geographical location and without requiring special software or computer platforms. Implementing a web-based user interface will help achieve these conditions.

1.1.1 Graphically Displaying Complex Data
When a user is examining a process in the handbook, that user is required to absorb an immense amount of information in order to get a good understanding of the process. This information needs to be presented in a fashion that is concise yet thorough. A graphical viewer provides substantially more information to a user in a smaller amount of space than a text-based description because the sequential, immediate nature of the spoken language lacks the capacity to communicate the sense of dimensional complexity inherent in a process description [Tufte, 1990].

The information stored in the Process Handbook is a semantic net of very complex, but highly structured data. From any one activity in the information space of the handbook, a user can explore the many different dimensions that originate from that activity, such as
its specializations, generalizations, decomposition, dependencies, and the other activities that utilize this activity. A major challenge of this thesis was thus to develop an appropriate interface to display these dimensions in a way that aids the user in understanding the contents of the handbook, utilizing current Web technologies.

1.1.2 Providing a Distributed, Collaborative Environment
No one person has a clear understanding of every business process that has been developed or that will ever be developed. Group collaboration on process analysis allows different people to contribute their unique perspective and experience. Through this collaboration, the potential for innovative ideas can be greatly increased.

Members of a group should not be limited by geographical locality. Cultural experiences can contribute greatly to innovative process engineering. Laws and customs in one country may influence the way people approach business process design. People from different parts of the world may have special insight of various aspects of a process. By providing a collaborative environment, people can communicate their ideas and learn from others in a controlled forum.

Just as important, within the constraints of their access control, groups of users will all share the same information. This will help eliminate possible inconsistencies that arise when multiple copies of the handbook are initially distributed, and consequently each individual makes their own modifications to the process description.

Another benefit of providing a distributed environment is the ability to easily expand the community of Process Handbook users and contributors. By increasing the number of people involved with the handbook, the amount and diversity of the contents of the handbook will be greatly increased. Concurrently, the number of people familiar with the concepts of the handbook will increase, providing a wider base of people that can collaborate to invent new business processes.

1.2 Goals
The potential benefits of Web interface are many, but if it is not carefully designed, users can find the interface too difficult to use or not powerful enough to be useful. To avoid these pitfalls, the design of the interface should meet the following criteria:
• An intuitive presentation metaphor - The information in the Process Handbook is very complex, highly structured, and multi-dimensional. If the information is not properly presented, a user may have a difficult time comprehending the information. Also, the context in which the user is viewing information must be preserved. When a user is navigating through the complex and multi-dimensional information space of the handbook, users can easily lose track of where they came from and how they can retrace their steps. This loss of context can also be very detrimental to the understanding of the information in the handbook. Thus, the way information is presented is key to an effective and useful interface to the handbook.

• Navigational efficiency - Users should be able to find the activities they are looking for with minimal effort. Also, moving from one place in the handbook to another place should not be difficult.

• Suitability for both “naïve” and “power” users - In order to reach the greatest number of users possible, it is necessary to make the interface easy to learn and use for beginners, but powerful enough for more experienced users. Although the interface will be designed to allow easy entry into the handbook community, support for power users is essential because it is these users that will initially make the greatest contributions to the Process Handbook, so it is necessary to design an interface that would encourage their participation in the handbook project.

• Quick response time - The server generating the information must fill requests in a reasonable amount of time. If there is too great a delay between requests for information, users may be discouraged by this inefficiency in the interface, and thus, refrain from using it.

• Support for focused learning - It would be very beneficial if there were a way to guide users through a specific “path” in the handbook to facilitate their understanding of a specific process. A process can have so many related items in the handbook that a user may stray from the path that is most beneficial to their understanding.

• Providing functionality similar to the single-user application - The single-user application\(^1\) has a rich array of features that would be very desirable for the Web interface to have. The single-user application has the advantage of being developed for one platform, Microsoft Windows, and that this platform has a great deal of software development support. However,

---

it would be ideal if some of the more important features of the single-user application could be emulated by the tools available for the World Wide Web.

- The ability to access the interface on a wide range of computers - Many users use different types of computers with different operating systems, monitors, and Internet connections. By designing the interface to be acceptable to a user with a 14.4k baud modem, monitor that supports a resolution of 1024 pixels by 768 pixels, and any computer platform with a full-featured Web browser, such as Netscape Navigator, the number of potential users will be maximized.

1.3 Scope of Research
Because this interface utilizes the World Wide Web as its medium of communication, it is necessary for the database containing the Process Handbook information to reside on a computer with an Internet connection that is running a Web server which supports common gateway interface (CGI) scripts in addition to the standard hypertext markup language (HTML). It is also assumed that an access control model is implemented at the data storage level to enforce privacy of sensitive information between different groups in the handbook community.

This thesis focuses on the design of the interface. Specifically, the interface will be accessed via an existing Web browser. The research will explore different ways of using existing web technologies to display process information effectively.
Chapter 2: Background

2.1 The World Wide Web
The World Wide Web is defined as the universe of global network accessible information [Berners-Lee, 1996]. It was originally created as a means for scientists to share information, using traditionally incompatible computer systems. However, the popularity of the Web with the general public has grown exponentially each year since its introduction to the public in 1993, and this explosion of popularity has had many social and economic impacts. In the following sections, a brief historical review of the Web is given, and new Web technologies which could be useful in implementing the interface are discussed. Then some of the relative strengths and weaknesses of using the Web for this application are explained.

2.1.1 History of the World Wide Web
The first prototype for the World Wide Web was developed by Tim Berners-Lee in late 1990 at CERN, the European particle physics laboratory. Berners-Lee initially developed a system in 1989 that allowed random links to be created among various sources of data that could access from any machine connected to a local area network, but not a wide area network, so they set out to create a new system that allow simple information sharing across the entire Internet. The initial design of the Web was based upon the following criteria, as stated in [Berners-Lee, 1996]:

- An information system must be able to record random associations between arbitrary objects, unlike most database systems.
- If two sets of users start to use the system independently, making a link from one system to another should be an incremental effort, not requiring unscalable operations such as the merging of link databases.
- Attempts to limit users to particular languages of operating systems are doomed to failure.
- Information must be available on all platforms.
- Attempts to force users to deal with information in the same way computers deal with information are doomed to fail
- To insure accuracy, information must be easy to enter and correct.
Figure 2.1 - Diagram of Web architecture from early discussions of the Web in 1990. Taken from [Berners-Lee, 1996].

The basic architecture of the Web was designed for great flexibility. Figure 2.1 shows a diagram of the Web architecture that was proposed in discussions that took place in early 1990. The specifications of the Web were made independently of the implementation of the Web in order to avoid the interoperability constraints imposed by various computer systems. This allowed other specifications, such as the file transfer protocol (FTP), to be smoothly integrated into the Web specifications. By following this principle of minimal constraint, the Web was easily adopted. People had to make small or incremental changes to use the web, first an a parallel application to existing technologies, then as the primary technology itself.
The first browsers for the Web only displayed text pages. Not until 1993, when Marc Andreessen created Mosaic for the X Windows system, was there a browser that displayed images as well as text. This browser showed the potential of the Web, and became very popular. Since then, Andreessen has gone on to co-found Netscape Communications, maker of the most widely used Web browser today.

The ability of the Web to handle arbitrary data helped its expansion, but also created a major problem. Many organizations were introducing proprietary Web features that created a diverse set of Web browsers. This made Web content more difficult to maintain. In order to provide the most appropriate content, web servers would have to check the browser’s manufacturer in order to send the correct Web content to the browser. This fragmentation of standards was threatening to cripple a technology that many universities and companies had come to rely on. Thus, in 1994, the World Wide Web Consortium was formed. This consortium, based at the Massachusetts Institute of Technology and the National Institute for Computer and Automation Research in France, consists of about 150 members representing all major Web technology developers as well as other organizations that have increasingly used or depended on the Web. The consortium provides a forum in which competing companies can work together to develop common specifications. By forming this consortium, it is hoped that the evolution of the Web will be much smoother and consolidated.

2.1.2 Hypertext Markup Language
The hypertext markup language (HTML) is the basis on which every Web site is built. This language was designed to allow people to do basic publishing over the Internet. Many different software companies have introduced their own extensions to this language, but the only official version of the language can be set by the World Wide Web Consortium. The most current extension to the language is HTML 3.0. This version incorporates many of the extensions that have been suggested previously by companies that implemented these features into their browsers. The two biggest software companies that make Web browsers, Microsoft and Netscape Communications, have been battling over the next extensions HTML, which will include features to allow dynamic behavior on the client side. For now, HTML 3.0 is the de facto standard for the language.
HTML it is based on the Standard Generalized Markup Language (SGML). The main intent for making it compatible to SGML was to help promote the Web's initial acceptance into the document community, which was accustomed to SGML, and the hypertext community, which at the time was only considering SGML-based standards for the language [Berners-Lee, 1996]. Unlike conventional desktop publishing, many details of how documents are laid out are left to the capabilities of individual browsers or reader preferences, rather than being completely specified by authors. This allows documents to be viewed on a very wide range of equipment. The initial version of HTML allowed only simple formatting of text, such as font size and appearance. Some of the new features in version 3.0 include fill-out forms, tables, and mathematical equations.²

2.1.3 Other Display Technologies Used in the World Wide Web
Since the emergence of the World Wide Web, many technologies have been developed that offer more power and flexibility in information presentation than is available in HTML. Below are descriptions of some of the more popular technologies used today to develop Web content.

2.1.3.1 Java
Java is a powerful programming language developed by Sun Microsystems in an attempt to provide a development environment that would generate programs which would run on any type of computer. Java is an object oriented language loosely based on another popular programming language, C++. According to the Java specifications, the behavior of these applications, known as applets, should be identical on any computer that supports Java. Java applets are different than typical applications in that they reside on the network in centralized servers [English, 1997].

From the corporations' point-of-view, Java will simplify the creation and deployment of applications thus saving money. Applications created in Java can be deployed without modification to any computing platform, thus saving the costs associated with developing software for multiple platforms. And because the applications are stored on centralized servers, there is no longer a need to have people insert disks or ship CD's to update software [English, 1997]. Also, many of the newest Web browsers can run Java applications because they have a

² These features may have already been supported by web browsers, but they did not become a standard part of HTML until version 3.0 was released.
built in Java compiler. This universal compatibility makes Java a very attractive tool in which to create applications for the WWW.

2.1.3.2 Javascript
Javascript is a object-based programming language introduced by Netscape in 1995 to allow designers of Web pages the ability to directly embed more functionality an HTML document more user interaction with the document on the client side. This allowed the contents of Web browsers to be updated without having to send a new request to the Web server.

Another interesting feature of Javascript that is supported by the Netscape Web browser is Javascript’s ability to communicate with Java applets. This provides developers of Web sites with incredible flexibility in creating interactive Web pages. Java applets are confined to a predefined block of space on a Web page. However, a developer can write Javascript functions that process events anywhere on a Web page. By designing the Javascript code and Java to communicate with each other, a Web page can have the full power of Java applets without the overhead of having to download a much larger program to handle a few special user inputs.

2.1.3.3 Virtual Reality Modeling Language
The Virtual Reality Modeling Language (VRML) is a language for describing multi-participant interactive simulations -- virtual worlds networked via the global Internet and hyperlinked with the World Wide Web[Bell, et al, 1995]. All aspects of virtual world display, interaction and Internetworking can be specified using VRML. It is the intention of its designers that VRML become the standard language for interactive simulation within the World Wide Web.

The first version of VRML allows for the creation of virtual worlds with limited interactive behavior. These worlds can contain objects which have hyperlinks to other worlds, HTML documents or other valid MIME types. When the user selects an object with a hyperlink, the appropriate MIME viewer is launched. When the user selects a link to a VRML document from within a correctly configured WWW browser, a VRML viewer is launched. Thus VRML viewers are the perfect companion applications to standard WWW browsers for navigating and visualizing the Web. Future versions of VRML will allow for richer behaviors, including animations, motion physics and real-time multi-user interaction. The real attractiveness of VRML is its three dimensional aspect, which can provide unique opportunities to display
information than can be more easily understood by users because all users are inherently accustomed to processing three dimensional data of some sort.

2.1.3.4 ActiveX
Microsoft defines ActiveX as a set of integration technologies that enables software components to interoperate in a networked environment using any language. Although ActiveX is usable only with the latest release of Internet Explorer, Microsoft is developing other versions that will support operating systems such as the Macintosh. ActiveX has the “cumulative” advantage as well, meaning every time a person visits a different Web page, the controls (or component applications) the person downloads are saved to the person’s hard drive. On the other hand, Java applets have to be downloaded every time the person visits the same Web site.

Underneath ActiveX is a technology called Object Linking and Embedding (OLE), also known as Component Object Model (COM), and the Distributed Component Object Model (DCOM). According to Microsoft, ActiveX components are OLE objects and DCOM extends interactions between components across networks.

The problem with existing OLE controls is their typically large size. This is due both to the inherent complexity of the full OLE interfaces the controls must implement and to the fact that the Microsoft libraries used to create the controls were never optimized for size [Mace and Rubenstein, 1996]. Because any control that’s not resident on the user’s system must be downloaded across the Internet before it can run, controls used on Web pages need to be as small as possible. Few existing OLE controls fit this criteria. Microsoft has taken two approaches to solving this problem.

One approach has been to simplify the definition of ActiveX with regard to the interfaces it must implement. An ActiveX control, technically defined, is any COM object that implements IUnknown, the basic OLE interface that provides access to all the object’s other interfaces. This makes it possible for developers to implement just a subset of the OLE interfaces needed for the task at hand, yet still be within the requirements of the specification. The other approach has been to revamp the frameworks used to develop ActiveX controls.

Although ActiveX controls are familiar to developers, they may be a new concept for some Web site masters. In essence, ActiveX controls provide a way for a programmer to create a high-level,
reusable object with some type of useful generic behavior. The programmer can then pass the control along (or sell it) to another developer, who can use it as a building block inside any programming tool that supports ActiveX controls. (Most major Windows-based tools for C/C++, Basic, Pascal, and many other languages do.) An ActiveX control could be anything from a push button to a fully functional spreadsheet. Thousands of controls are available commercially, from scores of vendors. The richness and breadth of these controls make up the single biggest strength of the ActiveX platform.

2.1.3.5 VBScript
Microsoft Visual Basic Scripting Edition (VBScript) is the native scripting language of Microsoft Internet Explorer and the most obvious choice for programming ActiveX-based Web pages. VBScript is a Web-adapted subset of Visual Basic for Applications (VBA), Microsoft's standard Basic syntax. It uses Basic's familiar object-property dot notation, subs, functions, and flow-control structures, and it has the broad range of intrinsic functions that Basic programmers will expect. Anyone who has programmed in a modern Basic language will feel comfortable with VBScript.

Like VBA, VBScript supports the Option Explicit statement to force variable declaration. But unlike VBA, VBScript is not explicitly typed, and all variables, string or numeric, are stored as Variants. Implicit data types—including integer, long, single, double, string, numeric, date, and financial—exist, but the system only infers these by the format of the data. Fortunately, there are conversion functions and functions for testing the type of a variable (VarType, IsNumeric, for example) that can be employed for pages that implement data validation.

VBScript lacks any support for file input/output and direct memory access. This approach, which VBScript shares with Java and JavaScript, is necessary to prevent the language from becoming a vehicle for programs with subversive or destructive behavior [Mace and Rubenking, 1996].

2.1.3.6 Cookies
Cookies are a general mechanism which server side connections (such as CGI scripts) can use to both store and retrieve information on the client side of the connection. The addition of a simple, persistent, client-side state significantly extends the capabilities of Web-based client/server applications. Web servers can create cookies by appending special information in the header of a Web page. Web clients can now also create cookies using a scripting language such as
Javascript. This idea was originally introduced by Netscape in 1995, but most browsers today, including Internet Explorer, support the use of cookies. Cookies are saved on the user’s computer, so Web servers can maintain information about users without having to commit extra storage space for the information on the server.

### 2.1.4 Advantages and Disadvantages of Using the World Wide Web

Developing a web-based interface provides many advantages over traditional software development. Distributing the latest version of a software interface to all the users in the handbook community would a very costly and time consuming effort, especially considering that one of the effects the interface will have is to expand the population of the community greatly. Supporting and maintaining multiple versions of the same application for different platforms is also an administrative nightmare. It is not reasonable to assume that the developer of an application will have access to every type of computer platform that the users have (the cost of this is not justified by the project), nor its it reasonable to limit the computer platform that users must have. Because there is a common Web standard, the interface can be used on a number of different computer platforms. The problems with distribution and maintenance associated with traditional software applications can be solved by using the Web. When updates and changes are made to the web interface, all users will have immediate access to them when they use the interface again [Rice, et al, 1996].

Another benefit gained by using a Web interface is that users are not required to have very powerful, expensive systems to use the interface. Typical Web usage involves temporary transfer of data from a server to a client. When, the client receives the data, it spends some time processing it to format the data for the user to view. The user will then spend a significant time examining the information before another request is made. This usage scenario does not require high powered computers to be effective, and it is the scenario that is envisioned for the handbook interface. It is not necessary for a user to have a continuous link to the handbook. Users will be spending the vast majority of the time thinking about and the information, not doing continuous, computationally intensive tasks.

There are some disadvantages to making a web interface instead of a local user interface. Most important is that less functionality available to the platforms. Although several new Web technologies are providing functionality that was previously only available to a desktop application, it is difficult to develop applications with these technologies because they incur a
great overhead in network bandwidth, causing the interface to appear to be responding more slowly to the user, making the application less attractive to the user. Also, some of these technologies do not have the widespread acceptance or availability, so using any of them may exclude a significant number of potential users. Therefore, a web interface may not be as sophisticated as a local, computer-specific user interface.

Another concern deals with the possibility that users may feel uncomfortable from the loss of the sense of tight-integration that a user would experience from using an application locally as opposed to the remote, client-server model that the Web uses. However, this last concern is not as great as it may have been in previous years because of the greater exposure people have had to the Internet and the World Wide Web, and thus, their growing acceptance of the medium as a way in which tasks can be done.

Despite its shortcomings, the benefits from using the World Wide Web greatly outweigh its disadvantages. Its inherent attributes provide benefits for both developers and users, making it an appropriate choice for the web interface for the Process Handbook.
Chapter 3: Design of the Interface
In the development process, two major stages occurred. First, the design parameters for the interface was identified, and then the layout and implementation of the interface took place. In this chapter, the design criteria for the interface will be discussed, and the details of the implementation of the interfaced will be covered.

3.1 Design Issues
While developing the Web interface, several issues arose that had a significant effect on the design and implementation of the interface. These issues included: navigational efficiency, the amount of screen space available, and the brand of Web browser the user should be utilizing. These issues are discussed below.

3.1.1 Navigational Efficiency
A major concern about the interface is the ability for a user to reach a desired point quickly and easily. The sheer volume of information in the handbook can make finding specific items very difficult to do. Thus, features necessary for efficient navigation must be incorporated. A side condition that these navigational features must also satisfy is the preservation of context for the user. If the user decides to quickly move from point to point, it is necessary that the interface provides a mechanism that allows the users to not get “lost” in the information space of the handbook.

3.1.2 Computer Display Characteristics
The computer display was another major consideration while designing the interface. The space available on the screen as well as the resolution of the screen dictate the way in which information will be presented. These two characteristics are interrelated in their significance. If a small screen tries to run at a higher than ideal screen resolution, then it may be actually contain the desired information, but that information may be uncomfortable or undetectable to the human eye. Conversely, if a very large screen runs at too low a resolution, then details may become overemphasized, and users may lose the ability to properly capture the concept that is presented on the screen. The goal was to maximize the amount of information available on the screen while maintaining the comprehensibility of the display. The size and resolution of computer monitors in use vary greatly, and there is no reasonable way to enforce a common monitor size and screen resolution to every user of the handbook. Therefore, it was decided to aim the design at a particular monitor size and resolution, a 17" monitor using a 1024 pixel by 768 pixel resolution.
These specifications are reasonable at this time because the majority of desktop computers sold today come with a 17" monitor and that screen resolution is the supported by all major video card vendors. Those whose use the smaller displays on laptop computers may have a harder time seeing some of smaller details in the interface, while those who are fortunate enough to have access to very large 21" monitors running at higher resolutions will be able to view additional information more easily, if desired. Because laptop users are currently a very small user base, it was not considered a significant drawback to not provide a more accommodating interface to these users.

Another point to note is that users will set individual preferences for the display properties of their web browsers. These settings determine the type and size of the character fonts the browser uses, and combined with the monitor settings, will affect the way information is presented through the web browser. Because the interface does not try to affect these browser settings, it is left to the users themselves to determine the font type and size that will be most comfortable for them to use.

3.1.3 Choice of Web Browser
A third consideration to the design was the choice of which Web browser(s) would be suitable for the interface. Because the interface was to be a very graphical, highly interactive application, a Web browser had to support more than just simple HTML pages. It also had to support the other Web technologies that were used to implement the interface, namely Javascript and frames. It is expected that users who access the handbook Web interface use at least the Netscape Navigator 3.0 Web browser. This browser has the built in support for Javascript as well as being the browser that is available for the most computer platforms. Although Microsoft’s Internet Explorer 3.0 and higher also support a version of Javascript, its version of Javascript has some proprietary features, and therefore, compatibility issues are a concern. Tests of the interface on the version 3.0 of Internet Explorer show that the interface does not work properly because Internet Explorer does not execute Javascript commands in the same manner as Netscape, and Microsoft’s insistence on its own version of Javascript makes it doubtful that its Web browser will ever be fully compatible with the handbook Web interface. Also, Internet Explorer is only available on a very limited number of computer platforms, which limits the potential number of users. Thus, as the development of each browser diverges into more incompatibility, it is satisfactory to provide support for only Netscape browsers because they are available on the
widest number of platforms, including those supported by Microsoft, so we believe the potential number of users are not significantly limited by this decision.

3.2 Web Technologies Used
Taking into consideration the criteria set above, the choice to use a combination of HTML and Javascript to implement the interface was made. For this version of the interface, Javascript’s feature set provided the means to incorporate the additional functionality to the interface that HTML alone could not provide. Also, because Javascript was supported by both Microsoft’s and Netscape’s browsers, its use would not reduce the size of the potential user pool. The current version of VRML does not support the use of scripts, so the amount of interaction allowed in a VRML world would be similar to that in a purely HTML page. ActiveX showed a great deal of potential, but its current lack of native support on Netscape web browsers was seen as a limitation. Java also showed similar potential, but the network overhead that would be caused by the repeated reloading of the same applet does not justify the features it does provide. Also, both ActiveX and Java did not provide any features that could not be emulated using a combination of HTML and Javascript, and the development time required to make components from these technologies was much greater than than developing the interface in HTML and Javascript.

3.3 Layout of the Interface
In this section, the components of the interface will be discussed. Alternative designs and their relative strengths and weaknesses will be detailed in Chapter 4. In this interface, there are basically three types of windows that users will interact with, the login window, the session window, and the object window. Each of these windows serve a different purpose, which are discussed below.

3.3.1 The Login Window
All users must enter the Web interface through the login window (Figure 3.1). This window displays various information, and requests a login name that a user has defined previously when creating a user profile. Taking a closer look at Figure 3.1, a user would first encounter a basic introduction about the Process Handbook is given along with a hypertext link to the working paper [Malone, et al, 1997]. The next thing the users encounters is the license agreement. Because this Web site and the information in the handbook is property of the Massachusetts Institute of Technology, all users must agree to follow basic copyright law when using the Web site.
Welcome to the M.I.T. Process Handbook

The Process Handbook is a tool for managing knowledge of businesses and business processes. To see a working paper that describes the project in more detail, click here.

This information is being provided to you, the LICENSEE, by the Massachusetts Institute of Technology (M.I.T.) under the following license. By obtaining, using and/or copying this information, you agree that you have read, understood, and will comply with the terms and conditions of the license.

Login (Optional): [ ] (no spaces or special characters please)

Enter Process Handbook

You are allowed to define "profiles" to specify how you view the contents of this site. If you would like to use a profile you defined previously, then please enter the login name of the profile above. If you like to define a new profile, then enter a login name for the profile, then click on the button to enter the site. If you would like to use the default profile, leave the field blank and enter the site.

NOTE: In order to view this site, you must have a browser that supports frames and Javascript. We suggest using Netscape Navigator 3.0 or higher. If you don't have this browser, you can download the software here.

Figure 3.1 - The login window.

The next section of the window provides the “login” field and a brief description of user profiles. User profiles contain specific information about each registered user, such as the user’s real name and email address. User profiles have also served to contain various session options that allow users to configure to their preferences. If a user wants to log into the site anonymously, then the user simply has to leave the login field blank. The default profile will be used during this user’s session.

When a user logs on, a cookie is used to save several pieces of information about the user, including the login name. Because of the way cookies function, this only allows one valid user on a computer at any one time.
New User Registration

As you are not on our database, we would like you to complete the following registration form so that your session can be tailored to your needs.

**Personal Information**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login:</td>
<td>john doe</td>
</tr>
<tr>
<td>Full Name:</td>
<td>John Doe</td>
</tr>
<tr>
<td>Email Address:</td>
<td><a href="mailto:jdoe@anon.net">jdoe@anon.net</a></td>
</tr>
<tr>
<td>Starting Activity ID</td>
<td>0</td>
</tr>
</tbody>
</table>

_[Save Profile]_

---

**Figure 3.2 - The user registration window.**

The final part of the login window describes the software requirements needed for the Web interface to function properly. Specifically, a Web browser that supported Javascript and frames is required, and Netscape Navigator 3.0 or higher is suggested.

If a user uses a login name that has never been used at the site previously, then the user is taken to a user registration window (Figure 3.2). This registration window will request some personal information about the user, such as the user’s real name and email address. After the user fills in the information and clicks on the “Save Profile” button, the user is taken to the session window.
3.3.2 The Session Window

The session window can be thought of as the main control center for the Web interface. Many basic functions that are best provided in a central location are in this window. Figure 3.3 shows that the session window is divided into two frames: the top frame being a menu of buttons, each providing a different function, while the bottom frame is used to display the particular screen for each function when one of the buttons is clicked. Currently there are six buttons in the menu: “Contents,” “Search,” “Profile,” “Discussion,” “Comments,” and “Help”.

3.3.2.1 The “Contents” Screen

The “Contents” screen displays the navigational index of the Process Handbook. The navigational index is structured to be hierarchical, so navigating through the index is similar to navigating through a collapsible outline. Objects in the hierarchy can be expanded or collapsed to
reveal or hide the objects that are linked navigationally to the object. This is accomplished by clicking on the arrows next to the hypertext. A red arrow that points to the right next to a hypertext link indicates that there are objects associated with the object named; clicking on the red arrow will reveal the objects. An object with a blue arrow pointing down indicates that this object’s navigational links are currently being displayed; clicking on the blue arrow will hide the navigational links. An object that has no arrows next to it has no navigational links. Clicking on a hypertext link will launch an object window for that particular object, except if the object is a navigation node. Instead, the display frame is filled with a detailed view of the navigation node, including its description and the navigational links associated with it. This allows a user to concentrate the user’s exploration of the handbook. Figure 3.4 shows the navigational hierarchy of the contents screen with several branches expanded.

Figure 3.4 - The contents screen of the session window.
Figure 3.5 shows the contents of the display frame after a user clicks on a navigation node. The description for the node is given as well as a listing of navigational links of this node. This listing uses the same set of geometric symbols that the contents screen uses to indicate the type of object that is linked to this node. If a user clicks on a navigation node link, then the contents of that node will replace the current node in the display frame, otherwise an object window will be opened for the object. Like other objects, navigational nodes can have links to resources outside of the handbook (see “Other Links” in section 3.3.3.4 for more details).

There are also geometric symbols between the arrows and hypertext that indicate the object type of the hypertext object. Table 3.1 summarizes the symbols. The most prevalent objects in the handbook currently are activities and navigation nodes.
3.3.2.2 The “Search” Screen
The “Search” screen allows a user to search for a particular activity or dependency by its name or the name of the contact associated with the object. In the case of dependencies, a user can search in its description also. Figure 3.6 shows the layout of the search screen. After the user enters the keyword(s) to search for, a request is sent back to the server, and the results of the search are then displayed for the user. Figure 3.7 shows the results of a search for activities containing the search string “produce something”. Clicking on one of the hypertext links will launch an object window for that activity.

3.3.2.3 The “Profile” Screen
The “Profile” screen allows a user to modify the user’s profile. As mentioned earlier, a user’s profile contains personal information, and can also be used to store session configurations. This screen also displays the user groups that the user belongs to. User groups are part of the mechanism that provides an access control model to the current version of the handbook. Currently, a user cannot choose the user groups in which the user is a member; only an administrator can assign users to groups. Figure 3.8 shows the profile screen for a typical user. Note that this screen resembles the new user registration screen in Figure 3.2. This is because both screens use the same form to create and edit user profiles. The profile screen additionally shows the user group information.
Figure 3.6 - The search form.
Figure 3.7 - The results of a search of the keywords “produce something".
Profile

You may modify the information below to match your personal information.

Personal Information

Login: johndoe
Full Name: John Doe
Email Address: jdoe@anon.net
Starting Activity ID: 0

User Groups

- Group 28
- Group 30

Save Profile

Figure 3.8 - The user profile screen.
3.3.2.4 The "Discussion" Screen

The "Discussion" Screen is provides a link to a Lotus Notes discussion database in which users can discuss any aspect of the Process Handbook, from its theoretical foundings to specific activities within the handbook. The link on the session window starts the user at the general Process Handbook discussion. This discussion database allows users to interact and communicate their thoughts to one another by creating or responding to articles written by other users. These articles are stored in the database for future reference by any user with access to the database. This ability to communicate in a controlled forum is very important to the development of new processes, and using this commercial product was an convenient way to provide this functionality. Figure 3.9 shows the entry point to the discussion database.
3.3.2.5 The “Comments” Screen
This screen provides users with the ability to send comments directly to the administrators of the Web interface. This screen provides a form that can be filled out by the user, and after the user submits the form, the administrator receives the comment by electronic mail. Figure 3.10 shows the comments screen. Comments will allow the administrators to get a sense of the users’ reaction to the interface, helping to enhance the interface, while giving users the opportunity to affect future versions of the interface.
3.3.2.6 The “Help” Screen
A user can obtain help by clicking on the “Help” button. The table of contents for the help pages appear in the display frame of the session window. A user can navigate the help pages by clicking on hypertext, which will jump it from page to page. There are sections that explain the different windows in the Web interface and some that give a little background on the Process Handbook. Figure 3.11 shows the table of contents for the help pages. Figure 3.12 shows the help page explaining the concept of decompositions.

Figure 3.11 - The table of contents for the help screen.
Decompositions

Decomposition involves the breaking down of a process in subparts (or subactivities). Here we find an "and" relationship between the subactivities that make up the process. This view of the process allows you to see the different levels of the process. As with many other process representation techniques, a decomposition in the Process Handbook represents a process as a map of various activities which need to be performed in order to reach a goal. The decomposition can be understood at multiple levels and is similar to the way in which objects can be decomposed in object-oriented hierarchies used for software programming. In a process context, decomposition builds explicit links between high-level strategic descriptions of a process (e.g., "Provide high-quality client service") and low-level operational activities (e.g., "Answer all client questions the same day they are asked"). Decomposition encourages process understanding at a level beyond flowcharts that lays the foundation for productive process redesign.

Figure 3.12 - The help page for decompositions.
Figure 3.13 - The object window for “Produce automotive component”. The specialization view provides the context in which the object is viewed. The contents section provides a persistent view of the most important information related to the object. The details view allows the user to examine more closely the various aspects of the contents view.

3.3.3 The Object Window
An object window provides a focused view on one particular object in the handbook. Depending on the type of the object, the object window will concentrate on the more relevant types of information associated with the object type. As shown in Figure 3.13, the object window is partitioned into three major section: the specialization view, the decomposition/trade-off view, and the details view. Each section is given one-third of the window by default, but the ability to
resize each of the sections provides a great deal of flexibility as to the particular aspect of the object that a user would like to focus on.

The object hyperlinks in each view behave differently when clicked on. When a user clicks on a name of an object in the specialization view, the whole object window is refocused on that window. The specialization view is refocused on the new object, the appropriate middle view is generated, and the details view is set to the new object. In the decomposition and trade-off matrix view, when a user clicks on the hypertext link of an object’s name, the details view (see section 3.3.3.4) is refocused on that object, while the rest of the object window remains unchanged. This allows a user to examine an object in the decomposition more closely without having to create a separate object window for the object. As in the specialization view, clicking on any hypertext of an object’s name in the details view will refocus the whole object window onto the object.

Looking at Figure 3.13, notice how each of the object names has an “@” appended to it. Clicking on this separate hypertext link will open a new object window focused on the object next to the “@”. The contents of the original object window are not changed. All the different views use the “@” to open new object windows.

3.3.3.1 The Specialization View
This view is common among all object windows. It occupies the top third of the object window. The specialization view indicates where in the specialization hierarchy the object is by displaying a particular path through the specialization hierarchy from the root activity to the focused object of the object window. Figure 3.14 focuses on the specialization view of the object window for the activity “Identify (Gather Info and Select)”. The top pull down window allows a user to choose the layout of the view. There are two choices: “outline” and “table”. The outline mode shows the specialization view in a manner similar to the navigation view of the “Contents” screen of the session window (see Figure 3.4). The default view mode is the outline mode. The arrows to expand/collapse nodes are present, as well and the geometric symbols that represent the object type of the object, and the hypertext is formatted similarly. The outline view is shown in Figure 3.14a. The table mode is shown in Figure 3.14b. In this mode, specializations of an object are clustered to the right, while the expanding/collapsing metaphor is maintained by using
the same arrows used in the outline mode. In all views, the name of the focused activity is emphasized by using a bold and italicized font. Also, the name of bundles are encased in brackets.

Figure 3.14 - The specialization view of “Identify (Gather Info and Select)”. A) The outline mode. B) The table mode C) A different path to the activity.

\(^*\) For more information about the specialization hierarchy, see the Appendix A.
Multiple inheritance is the ability for an object to have multiple parents in the specialization hierarchy. Because of multiple inheritance, it is possible for an object can exist in more than one place in the specialization hierarchy if the object or any of its ancestors have more than one parent. This is why a second pull down menu is provided underneath the viewing mode menu. This second menu contains all of the possible paths in the specialization hierarchy from the root activity to the focused object. The activity “Identify (Gather Info and Select)” has two parents, “Gather Info” and “Select”. In Figures 3.14a and 3.14b, the path that passes through “Select” was shown. In Figure 3.14c, the path through “Gather Info” is displayed.

3.3.3.2 The Decomposition View
If the object being viewed in the object window is an activity, then the middle section of the object window will display the decomposition of the activity. The decomposition of “Produce automotive component” is shown in Figure 3.15. The decomposition view is composed of a menu that provides a choice of view modes, similar to the topmost menu of the specialization view. The bottom portion of the decomposition view displays the decomposition of the focused activity of the object window. There are two viewing modes supported in the decomposition view, horizontal and vertical. The horizontal view mode is similar to the table mode of the specialization view. The decomposition is expanded from left to right, with the subactivities of an activity clustered to the right of the activity. The vertical view mode is similar to the horizontal mode, except it is rotated 90 degrees, so that the decomposition is displayed in a top-down manner. Figure 3.15a shows the vertical mode view, while Figure 3.15b shows the horizontal mode view.
Figure 3.15 - The decomposition view for “Produce automotive component”. A) The vertical mode. B) The horizontal mode.

In both views, only one level of the decomposition is initially shown. The user can explore the lower decomposition levels. The blue and red arrows used previously in the specialization view and navigation screen are utilized here to indicate if an activity has a decomposition, and whether or not it is currently displayed. By expanding and collapsing the subactivities in the decomposition, the user can gain the desired decomposition detail. Both Figures 3.15a and 3.15b show an expanded subactivity for “Produce automotive component”.

41
3.3.3 The Trade-off Matrix View

When the focused object of the object window is a bundle, the middle section of the object window will display the trade-off matrix for the object\(^4\). Figure 3.16 shows the trade-off matrix view for the bundle “Sell How?” Each row consists of an activity in the bundle, while each column represents a different dimension that is being compared between the activities in the bundle. The activity names are hypertext links, when clicked, will refocus the details view on that activity. This view also utilizes the “@” next to the hypertext object names in order to provide the user a way to immediately open a new object window focused on that object. The name of the bundle is a hypertext link that will refocus the details view back onto the bundle. This provides a convenient way for the user to get back to the details of the bundle.

![Figure 3.16 - The trade-off matrix view for “Sell how?”](image)

3.3.3.4 The Details View

This view occupies the bottom portion of the object window. The details view provides the user with specific information about an object in the middle contents view of the object window. When an object window is created, the focused object is the default object displayed in the details view. The user can change the contents of this view by clicking in the hypertext of the object’s name in the contents view. As shown in Figure 3.17, the details view consists of a menu of detail choices and a display frame. Depending on the type of object being displayed, the menu of will have different choices. For an activity, the menu of choices includes the following:

- Description - This choice displays a hypertext description of the object. Because it is interpreted as HTML, the description can imbed hypertext links to other resources on the World Wide Web. Figure 3.17 shows the description of the activity “Manage usability”.

\(^4\) For more information about bundles and trade-off matrices, see Appendix A.
Figure 3.17 - The description for “Manage usability”

- Notes - Notes are small details that can be used to distinguish objects that have the same name in the handbook. Figure 3.18 shows two activities named “Send Purchase Order” and their notes. These notes allow a user to differentiate between the two. One deals with a purchase order for one item, while the other deals with the purchase order of multiple orders.

Figure 3.18- The notes for two different activities named “Send purchase order”.

- Generalizations - This choice displays the possible paths in the specialization tree from the root activity. If an object has multiple generalizations, then there will be multiple paths to the root activity. Figure 3.19 shows the generalization screen for “Identify (Gather Info and Select)”. This activity has two generalizations, “Gather Info” and “Select,” so there are two paths to the root activity “Act”.

43
Figure 3.18 - The generalization screen for “Identify(Gather Info and Select)”.

- Specializations - An outline view mode of a specialization view rooted at the activity is shown. By default, the view will also show two levels of specializations below the activity, if they exist. Figure 3.20 shows the specialization screen for the activity “Create”. There are two sublevels of specializations for this activity, so they are displayed initially. As in the specialization view, the name of bundles are encased in brackets.

Figure 3.20 - The specialization screen for “Create”.

- Decomposition - This choice provides the decomposition of the activity in the same vertical mode as the decomposition view section of the object window. Figure 3.21 gives the decomposition for the activity “Produce education and research service” expanded one level. As in the decomposition view, the subactivities that also have decompositions have a red arrow next to their names.
Figure 3.21 The decomposition for “Produce education and research service”.

- Where Used - The “Where Used” choice shows all the activities in which the current activity is a subactivity, i.e. the current activity is contained in the first level of decomposition of these activities. Also, any dependencies that this object is coordinating is shown in this choice. Figure 3.22 shows the where used information for “Acquire automotive component inputs”. This screen shows that this activity is a part of the decomposition of six other activities.

Figure 3.22 - The where used screen for “Acquire automotive component inputs”.

- Attributes - Each object in the handbook has various attributes associated with it. The attributes listed under the “Primary Attributes” table give very specific information about the required components of an activity. Specifically, these attributes include: actors, skills required, location, resources required, and evaluation criteria. User-generated attributes also appear in this section. Other information is also given in this screen, including an activity id, a unique process interchange format (PIF) id, the date it was last modified, and the name of the person that administers the object. In Figure 3.23, the attributes of “Identify potential customer on telephone” are given.
• PH Nav Links - All the objects in which a navigational link has been created are displayed here. All the hypertext links to these objects work in the normal manner, except if the link is a navigation node. If the user clicks on a navigation node, then the details display is filled with the screen for the navigation node (see Figure 3.5 for an example of a navigation node page). The navigational links for “Provide copy capability” are shown in Figure 3.24.

Figure 3.24 - The navigational links for “Provide copy capabilities”.

• Other Links - Links to Internet resources that are related to this activity but are not located inside the Process Handbook reside in this view. There are three classifications of links supported in the handbook: resource, example, and discussion. A resource link can include a published paper or article. An example link can include a link to a company’s Web site that utilizes this particular activity. A discussion link would provide a direct link to a discussion
about the object in a Lotus Notes discussion database or some other type of discussion forum. Figure 3.25 shows a discussion link and a resource link for “Resolve problems collaboratively - automotive”. If an activity does not have any discussion links, then a link to a discussion request form is given. Figure 3.26 shows the request link and the resulting discussion request form for an activity.

Figure 3.25 - The other links for “Resolve problems collaboratively - automotive”.

![Discussion Link and Resource Link](image)

Figure 3.26 - A discussion request link and the form that appears after you click on it.
- Trade-off Matrix - If the activity is a bundle, the “Decomposition” choice is replaced by a “Trade-off Matrix” choice. This choice displays a trade-off matrix in the same manner as the trade-off matrix view. Figure 3.27 shows the trade-off matrix for “Sell How?”

![Trade-off Matrix](image)

**Figure 3.27- The trade-off matrix screen for “Sell How?”**

### 3.3 Other Features

The session and object windows provided a user with the ability to explore the contents of the Process Handbook. However, to provide a more robust interface to the handbook, other features were incorporated to aid users. Because Javascript was used to build an infrastructure for the interface, advanced programming features were provided that allowed additional functionality to be built into the interface. One such function is the guided tour.

#### 3.3.1 Guided Tours

Guided tours are scripted usage scenarios for the handbook. They provide a way to specify a particular path through the process information in the handbook.

Guided tours can have many uses for the interface. They can be used to give an interactive tutorial on the use of the interface, or they can be used to give a presentation about a particular subset of process knowledge. By controlling the sequence in which process information is accessed, important aspects of the information that may not be evident to users can be emphasized as the guided tour progresses. This feature helps create a common understanding among a group of users, providing a base in which can be built upon through the collaboration of these users.
A guided tour can be created by embedding calls to the Javascript functions designed specifically for the Web interface in the description of a navigation node. This functions can create new object windows, or refocus existing windows to a new object. The functions also provide the ability to update the various sections of an object window. This granularity of functionality provides a powerful set of features that gives developers explicit control over the behavior of guided tours.

Figure 3.28 - A guided tour embedded in the description for the navigation node “Hiring processes”.

Figure 3.28 shows a guided tour embedded in the description of the navigation node “Hiring processes”. By linking the Javascript commands to HTML buttons, a user can click on the buttons to generate the events described in the buttons. For example, when a user clicks on the button labeled “View ‘Hire human resources’” in Step 1, an object window will appear focused
on the activity "Hire human resources". If the user then clicks on the button in Step 2 labeled "View ‘Select human resources’," the object window created in Step 1 will be refocused onto "Select human resources". This guided tour allows a user to access the activities relevant to "Hiring processes" without having to make the effort of searching for the activities.
Chapter 4: Evaluation of the Interface

This chapter analyzes the design described in the previous chapter. A set of alternative designs will be discussed, and the features of the chosen design will be compared to the alternatives to show why the design was chosen over the others. This will include an analytical comparison, in which measurements of how much effort a user would have to make in order to perform certain tasks in each design are made. Finally, a review of the design goals and how they were satisfied will be given.

4.1 Alternative Designs

There were several different designs considered before the one described above was chosen. Each design had its strengths and weaknesses, and after a great deal of deliberation, the design described in chapter three was chosen. The two other interface designs that received much consideration were:

- A single frame object window - This design is similar to the chosen design because it also utilizes a session window. However, the object window only has a details view. The behavior of all the links in the details view is the same as those stated in the chosen design.

- A two frame object window - This is an extension on the single frame object window design, in which the object frame will have two details views, one on top of the other. When an object window is initially created, the top details view will be inhabited by the selected activity. When a user clicks on a hypertext link for an activity, the bottom details view will set for that activity. When a hypertext link is clicked on in the bottom details view, the top details view is overtaken by the object, and the bottom details view remains unchanged. The "@" links still function in the same manner, creating a new object window when a user clicks on them.

4.2 A Comparison of the Different Designs

Each design has desirable and undesirable characteristics. By comparing the designs, it was determined that the chosen design had the best overall characteristics that satisfied the goals stated in section 1.2. Table 4.1 summarizes the comparisons made between the designs. The designs were ranked in order of which design satisfied the goals most adequately, with a "1" indicating the most proficient. Overall, the chosen design outperformed the other designs in several important categories, including context preservation and navigational efficiency.
<table>
<thead>
<tr>
<th>Intuitive presentation metaphor/maintains context</th>
<th>Chosen Design</th>
<th>Single Frame Object Window</th>
<th>Two Frame Object Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigational efficiency</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Suitability for both naïve and experienced users</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Quick response time</td>
<td>3</td>
<td>T1</td>
<td>T1</td>
</tr>
<tr>
<td>Support for focused learning</td>
<td>T1</td>
<td>T1</td>
<td>T1</td>
</tr>
<tr>
<td>Providing functionality similar to the single user interface</td>
<td>1</td>
<td>T2</td>
<td>T2</td>
</tr>
</tbody>
</table>

Table 4.1 - Ranking of the designs using the design goals of chapter one. The chosen design is superior or equal to the other designs in the majority of the goals. The goals in which the chosen design trail are considered less relevant. Note: A "T" in the rankings denotes a tie.

Although the overall rankings seemed to favor the chosen design only slightly, the chosen design is superior in the goals that are considered the most important - preservation of context and navigational efficiency. The preservation of context helps users maintain an awareness of their location in the handbook. The chosen design accomplishes this by constantly providing the location of the activity in the specialization tree. The other designs don't address this issue very well. The single-frame object window completely disassociates the activity from the context in which it can be found in the handbook. The two-frame object window doesn't provide much more support for context preservation than the single-frame design. Because one activity was displayed because a user clicked on the link to it in the details view of another activity, the user is aware that there is some immediate relationship between the two activities in the object window. However, the exact relationship between the two will be lost once the user begins to explore the other details of one of the activities, thus losing the detail screen in which the original relationship was shown. In section 4.2.1, an analysis of the designs will show that the chosen design is the most navigationally efficient design.

There are two goals in which the chosen design does not perform as well as the other designs - a quick response time for each request and ease of use for new ("naïve") and experienced("power") users. The slower average response time is mainly due to the fact that an object window for the chosen design requests a great deal more data when it is created or refocused than the object windows in the other designs. However, this extra initial overhead is considered reasonable.

because the information it is retrieving does serve other purposes. The specialization data is used to help preserve context, and can be used to access other related activities more quickly, and the decomposition and trade-off matrix are considered constantly relevant; rarely would a user skip this information for an activity or bundle if they are seriously examining these objects. These other criteria outweigh the need for the quickest possible response time.

The chosen design received a lower ranking in the goal of ease of use for naïve and power users because the more complicated layout of the object window may make for a steeper learning curve for new users, but once new users make the transition to "power" user status and have learned how to utilize the layout, it will become more functional to this users than the simpler object windows in the other designs.

When it comes to providing a mechanism for focused learning, all three designs are equal adept at this goal because all the designs are assumed to have been implemented using the same Javascript framework that allows in depth control of the behavior of the interface. This allows the creation of equally effective guided tours for each design.

4.2.1 Analytical Comparison of Navigational Efficiency
The second major advantage to the chosen design is that it was more navigationally efficient than the other designs. An analytical comparison of the different designs is made to show this point. Specifically, the amount of effort user would have to make in order to perform specific tasks is measured in the number of mouse clicks and window movements necessary to perform the task. Table 4.2 summarizes the results stated below.

In order to get a reasonable sense about the navigational efficiency of each design\(^6\), a set of tasks were chosen that represents those tasks that would commonly occur in a typical session. These tasks include:

- Side by side comparison of the decomposition for two specialization siblings - When the user is trying to determine the best specialization to use in a particular process, knowing the decomposition of these two specializations may have a significant influence on the user's decision. This task can be accomplished more quickly using the chosen design than either of

\(^6\) Because the alternative designs were not completely implemented, measurements were hypothesized using their design specifications.
the other two designs because the decomposition of an activity is immediately displayed when an object window is created. Thus, to see the decomposition for two specialization, a user only needs to click on the name of one of the desired activities in the session window to launch a new object window for the first activity and then move it to a desired location on the screen before clicking on the other activity’s name to open an object window for it. In the other two designs, a user must select “Decomposition” from the menu in the “details” view in order to retrieve the decomposition. This will require an extra click for every activity the user wants to compare. Thus, for two activities, it would require two clicks and one move to create and place two object windows next to each other, and one click for each window to display the activity’s decomposition.

<table>
<thead>
<tr>
<th>Side by side comparison of the decomposition for two specialization siblings, starting from the session window</th>
<th>Chosen Design</th>
<th>Single-Frame Object Window</th>
<th>Two-Frame Object Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 clicks</td>
<td>4 clicks</td>
<td>4 clicks</td>
<td></td>
</tr>
<tr>
<td>1 move</td>
<td>1 move</td>
<td>1 move</td>
<td></td>
</tr>
</tbody>
</table>

| View the description for all 5 subactivities in a decomposition of the focused object in the same object window, starting from the default view | 5 clicks | 10 Clicks | 6 clicks |

| Refocus an object window on a generalization of the currently focused object, starting from the default view of the object window | 1 click | 2 clicks | 2 clicks |

| Refocus an object window on a specialization of the currently focused object, starting from the default view of the object window | 1 click | 2 clicks | 2 clicks |

| Refocus an object window on a subactivity of the currently focused object, starting from the default view of the object window | 2 clicks | 2 clicks | 2 clicks |

Table 4.2 - Comparisons of the navigational efficiency of each of the designs. This table gauges the amount of effort a user has to put in to perform the various tasks, as measured by the number of mouse clicks and times a window has to be moved,

- Viewing the description of the subactivities in an activity’s decomposition - If a user is interested in a particular activity, that user would likely examine the details of its subactivities individually, to get a top level understanding of the activity. To perform this task in the chosen design, the user simply has to click on the subactivities in the decomposition view of the object window to get the details of the subactivity into the details view of the object window. For the single frame object window design, the decomposition of the activity must first be selected from the menu, then a new window for each subactivity
must be created. Although this takes only one extra click in this design, the extra windows created provide more overhead and may muddle the screen with unwanted windows. Thus, there is an additional cost to the user in the form of window management of one click to close these extra windows. For the two-frame object window design, it will take an extra click to first bring up the decomposition, then the subactivities can be displayed in the bottom frame, one at a time.

- Refocusing the current object window or creating a new object window on a specialization, generalization, subactivity in its decomposition, or another activity in which it was used in the decomposition - These tasks represent the simplest navigation to a closely related activity, and should occur very often during a user's session. As shown in Table 4.2, a move to a subactivity or generalization in the chosen design takes half as many clicks as it would in the other designs - only one. Moving to a specialization or "where used" activity takes as many clicks as the other designs - two.

Overall, the chosen design is either more efficient as at least as efficient in each of the tasks listed in Table 4.2. This shows that the chosen design is the most navigationally efficient design of the three.⁷

4.3 Reviewing the Design Goals
This interface was designed on a set of goals specified in chapter one. The chosen design satisfies these goals in the following way:

- Intuitive presentation metaphor - By presenting the specialization and decomposition information in a hierarchical manner, users can quickly understand the relationships between activities. Context is preserved by simultaneously displaying the location of an activity in the specialization hierarchy while a user examines its details. This provides an “anchor” to a

---

⁷ Although the chosen design was determined to be superior to the other proposed designs, the other designs still had some advantages over the chosen design. Thus, an experimental new design is currently being tested that allows a user to switch between the chosen design and the single frame object window design. This will allow users to adjust the interface to fit their navigational needs. Because at the time of this thesis the new design was at its infancy, it was not evaluated in this thesis. However, we believe that the new design will allow novice users to adapt to the concepts of the handbook more easily if they use the single frame object window mode initially, and then as they become more proficient with the interface, they can switch to the more robust mode of the current chosen design.
specific location in the handbook for users to attach themselves to in order not to get lost in the handbook.

- Suitability for both “naïve” and “power” users. The learning curve for new users may be a bit steeper for the chosen design because of the more complicated layout of the object window, but naïve users can still easily learn the interface because it is implemented in a way that is analogous to normal Web browsing. New users don’t have to learn new techniques or pre-configure the interface in order to use it; they simply log into the handbook and click on links to retrieve information from the server. Support for power users is given through the user profile. Power users can modify the behavior and appearance of the interface by changing the values in their profiles. Although this version of the interface lacks many configurable features, future versions are expected to contain many new enhancements that will allow users to configure their behavior, and the mechanism to store user configurations is in place now.

- Quick response time - Although the overall response time is slower in the chosen design, than in the other designs, enhancements were made to the interface to speed up response times. In order to improve response time, the server caches into memory much of the information it retrieves from the handbook database, including basic activity information and the various hierarchical relations between activities, thereby reducing the number of disk accesses and processing time necessary to retrieve this information. Also, the object window is designed to provide incremental information disclosure, reducing the amount of new information retrieved in each click. Excluding searches, the greatest amount of processing time required is when a new object window is created or an existing object window is refocused on a different object, but once this overhead is taken, navigating in an object window will produce very quick responses because the design of the interface promotes the incremental retrieval of information, so only these small increments of information have to be retrieved from the harddrive. Then this new information will be added to the cache of information already retrieved when the window was opened/refocused. Tests were performed to measure the benefits of implementing the cache on the server. The results showed an average time improvement of 20% when initially opening a window.

---

8 To measure the benefits of caching, the amount of time needed for creating new object windows twice for ten random objects in the handbook was recorded. The first time a window was opened for an object, that object’s information was not cached yet. The second time an object window was created, the information
• Navigational efficiency - By providing a search function, users can quickly move between activities without having to use the “Contents” screen to find the activities in the navigational or specialization hierarchies.

• Support for focused learning - This goal can is satisfied through the use of guided tours.

• Providing functionality similar to the single-user application - Features of the single-user interface that have been incorporated into the Web interface including the searching function and the view of an activity in the specialization tree. The Web interface surpasses the single-user interface by providing profiles that allow users to customize their sessions.

• The ability to access the interface on a wide range of computers - Because this interface uses a Web browser and standard HTML to present the data, any person with a computer and access to the Internet can use the interface. There are Web browsers available for just about any computer platform, so getting a browser that will work properly can be done easily.

for the object was cached because of the first object window that was created for it. Before each object was tested, the server was restarted to insure that the cache was completely empty when the first window was created for the object. The server used in the testing had an Intel Pentium Pro 200Mhz processor with 64 megabytes of memory, a SCSI harddrive, and an ethernet connection to a T3 network. The client used an Intel Pentium 166Mhz processor with 32 megabytes of memory.

9 The average time for creating windows with non-cached objects was 7.67 seconds. The average time for cached objects was 6.15 seconds.
Chapter 5: Future Work

The current state of the Web interface allows multiple users to view the information in the Process Handbook in an intuitive, highly functional manner. However, there are several possible enhancements to the interface that can be made that would increase its utility even further. These enhancements include:

- Editing features - The current Web interface does not allow editing of any of the handbook information. In order to have a truly robust collaborative environment, users must be able to modify the information in the handbook. These editing features would include workgroup collaboration, in which multiple users can simultaneously view and edit a common process in real time. This would maximize the collaborative efforts of a group of users.

- Interactive dynamic VRML displays - The virtual reality markup language (VRML) presents the opportunity to explore the information space of the handbook in three dimensions. However, version 1.0 of VRML does not provide a effective means of interaction with the user. The Web interface requires much greater dynamic behavior than VRML 1.0 supports. Still, there may be hope that VRML can still be incorporated into the interface in a useful manner. VRML 2.0, which very few Web browsers supported at the time of this thesis, has the ability to interact with users in a more powerful, programmatic fashion. When support for this version becomes widespread, it may be useful to integrate VRML 2.0 views into the handbook Web interface.

- As computer technology improves, concerns about limited computing resources and communications bandwidth will become non-issues, so the use of Java and ActiveX in the interface should provide a faster, more robust interactive display.

Before enhancements are added to the interface, a task that needs to be performed first is extensive user testing of the interface. Although the analysis given in chapter four provides a rough estimate about the usability of the chosen design, extensive testing by a large number of real users will be very beneficial to the future development of the interface. By soliciting evaluations from real users, unpredictable problems may be discovered and the development of the interface can be tailored to fit the needs of the users. Features not initially believed to be relevant may be considered essential by the user community, and new features may be conceived.
Chapter 6: Conclusion

This thesis was focused on the development of a Web based interface to the Process Handbook. In designing the interface, different Web technologies were considered, including standard HTML, Java, Javascript, ActiveX, and VRML, and it was determined that standard HTML and Javascript could provide the necessary functionality for the interface, although the other technologies present interesting opportunities to expand the interface even further.

The layout of the interface was designed to provide the most information possible while maintaining the context of the session for the user. Multiple windows could be viewed simultaneously, with each window anchored to a specific activity, and when each window is partitioned into subsections containing different aspects of the focused activity. By focusing each window on a particular activity, the context of the session is less ambiguous to the user. Providing several different views simultaneously for the same activity in one window allows more relevant information to be presented to the user without the need of the user to navigate between these views. This should help the user gain a better understanding of the activity without being distracted by the need to switch between views. Comparing this layout with other possible designs shows that this choice had the best overall characteristics of the alternatives.

In conclusion, this interface has been designed to display a type of 'loosely interconnected data that is rarely seen on the World Wide Web, and should have a significant effect on the use of the Process Handbook project. By developing a presentation metaphor for the complex, highly structured process data in the handbook, it will help a user comprehend more easily the information in the handbook. Also, by allowing multiple users in remote locations access to the repository of information in the handbook, the Process Handbook community will be able to expand and benefit from the knowledge of others. This knowledge sharing will further advance the innovation in process design that the Process Handbook project promotes.
Appendix A


A1.1 Overview and Goals
The Process Handbook is a research project that is currently be developed at the MIT Center for Coordination Science. The goal of the project is to build a repository of process knowledge that can aid in the innovation of new business processes. Even though a process handbook could have many different uses, the project has focused primarily on four possible uses: (1) imagining new organizations; (2) redesigning existing organizations; (3) sharing ideas and "best practices" about organizational processes; and (4) generating software for supporting and analyzing processes.

The key intellectual challenge for the handbook has been developing a new approach to analyzing and representing processes that explicitly represents the similarities and differences between related processes. In order to accomplish this, the representation of processes in the handbook capitalizes on two key ideas: (1) the concept of specialization of processes based on ideas about inheritance from object-oriented programming, and (2) concepts about managing dependencies from coordination theory.

A1.2 Specializations of Processes
Most process representation techniques use the concept of decomposition (breaking an activity into its sub components), including this one. However, this representation also incorporates the concept of specialization. The specialization of processes can be thought of as determining the different ways in which a process can be done. Using this concept, processes can be arranged in a hierarchical network with very generic processes at the top and increasingly specialized processes at lower levels. As in object-oriented programming, the specialized processes can automatically inherit properties of their more generic "parents", except where they explicitly add or change a property. Unlike traditional object-oriented programming, however, the inheritance of specializations is organized around a hierarchy of increasingly specialized processes not objects.

A1.2.1 Bundles and Trade-off Matrices
As software tools were being developed for the handbook, it was determined that it would be very useful to group related specializations into “bundles”. Figure A.1 shows how bundles are used in the representation. In the figure, one set of related specializations of “Sell something” is bundled under “Sell how?” and another set is bundled under “Sell what?”
Figure A.1 - Summary display showing specializations of the activity "Sell something". Items in brackets (such as "[Sell how?]") are "bundles" which group together sets of related specializations. Items in bold have further specializations. (Taken from [Malone, et al, 1997])

<table>
<thead>
<tr>
<th>Sell by direct sales</th>
<th>High</th>
<th>Long</th>
<th>High</th>
<th>High margin, tailored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sell by mail order</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Specialty items</td>
</tr>
<tr>
<td>Sell by retail store</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Low margin commodities</td>
</tr>
<tr>
<td>Sell over internet</td>
<td>Low</td>
<td>Fast</td>
<td>Low-improving</td>
<td>Commodities, Specialty items</td>
</tr>
</tbody>
</table>

Figure A.2 - The trade off matrix for "Sell how?". (Taken from [Malone, et al, 1997])

Bundles have two uses in the representation. First, comparing alternative specializations is usually appropriate only within a bundle of related alternatives. It is reasonable to compare "retail store front sales" with "direct mail sales," but comparing "retail store front sales" to "selling automotive components" is not. Where there are related alternative specializations in a bundle, the handbook can include comparisons of the alternatives on multiple dimensions. These comparisons can explicitly state the trade-off between these dimensions. Figure A.2 shows a "trade-off matrix" that compares alternatives in terms of their ratings on various criteria; different specializations are the rows and different characteristics are the columns. Items in the cells of this matrix can be associated with detailed justifications for the various ratings. For very generic processes, the cells would usually contain rough qualitative comparisons (such as "High", "Medium", and "Low"); for specific process examples, they may contain detailed quantitative performance metrics for time, cost, or other factors.

A second use of bundles is in restricting certain kinds of inheritance. Alternatives in a bundle automatically inherit alternatives from the other bundles but not the other alternatives in their own bundle. For example, it makes sense for someone selling beer to be automatically presented with alternatives for direct mail, retail storefront, and telemarketing, but it does not make much sense for this person to be automatically presented with alternatives of selling computers, newspapers, and consulting.
A1.2.2 Advantages of Specialization of Processes

By representing processes in a specialization hierarchy, a number of significant benefits over previous process representation techniques are gained. Two of the most important potential benefits are:

- Conciseness of representations - Specialization can substantially reduce the amount of work necessary to represent a new process. Inheritance allows the easy creation of new processes by identifying a more general process that the new process is intended to specialize and only making the changes needed to distinguish it. This helps support a rapid assessment of the basic features of a process, rather than requiring very tedious detailing. Users can express the gist of an existing process very quickly and add details to the extent that they are relevant to the purpose at hand.

- Generative representations - The specialization hierarchy provides a framework within which users can generate process alternatives. By clustering related alternatives close together in the hierarchy and by providing a mechanism in which these alternatives can be compared, users can easily find the most appropriate instance of a particular process to use to generate other processes. Merely collecting descriptions of how different companies sell consulting services, for example, would probably identify numerous examples of direct sales and perhaps mail advertising techniques. But organizing these examples in the specialization hierarchy along with other kinds of sales processes quickly leads users who are interested in more radical alternatives to options like retail storefront selling, even if no cases of consulting firms using this method had been observed.

A1.3 Coordination Theory

Coordination theory is an emerging research area that focuses on the interdisciplinary study of coordination. Research in this area uses and extends ideas about coordination from disciplines such as computer science, organization theory, operations research, economics, linguistics, and psychology [Malone and Crowston, 1994]. In a general sense, coordination can be defined as the act of working together. More specifically, coordination can be viewed as managing dependencies between activities.

A benefit of the latter definition of coordination given above is that it separates dependencies from the processes that coordinate them. Thus, advancement in this area of research can be achieved by identifying the different types of dependencies and the processes that coordinate them. One possibility lies in identifying and analyzing various dependencies and their coordinating processes, creating a repository of coordination processes that can be used to facilitate the interdisciplinary transfer of knowledge [Malone and Crowston, 1994]. Table A.1 lists three types of dependencies and some possible coordination processes for each dependency. This table is not comprehensive, but it gives an idea of how coordination processes can be organized by the dependencies they coordinate.
<table>
<thead>
<tr>
<th>Dependency</th>
<th>Coordination Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisite (&quot;right time&quot;)</td>
<td>Make to order vs. make to inventory (&quot;pull&quot; vs.&quot;push&quot;).</td>
</tr>
<tr>
<td></td>
<td>Place orders using &quot;economic order quantity&quot;, &quot;Just In Time&quot; (kanban system), or detailed advanced planning.</td>
</tr>
<tr>
<td>Accessibility (&quot;right place&quot;)</td>
<td>Ship by various transportation modes or make at point of use</td>
</tr>
<tr>
<td>Usability (&quot;right thing&quot;)</td>
<td>Use standards or ask individual users (e.g., by having customer agree to purchase and/or by using participatory design)</td>
</tr>
<tr>
<td>Sharing</td>
<td>&quot;First come/first serve&quot;, priority order, budgets, managerial decision, market-like bidding</td>
</tr>
<tr>
<td>Fit</td>
<td>Boeing's total simulations. Microsoft's daily build</td>
</tr>
</tbody>
</table>

Table A.1- Examples of elementary dependencies between activities and alternative coordination mechanisms for managing them

**A1.3.1 Coordination Processes and Managing Dependencies in the Process Handbook**

The idea of separating dependencies from the processes that coordinates them is one aspect that gives the Process Handbook methodology a unique approach to process representation. This feature allows a more succinct representation while giving users the opportunity to generate other possible alternatives to managing dependencies.

The succinct nature of a process representation comes from the fact that one coordination process can be associated with a particular dependency. Thus, in every process description that uses that dependency, the coordination process does not have to be stored in the process description. Instead, that dependency can be thought as being coordinated by an instance of the coordination process, and other instances of the coordination process could manage the same dependency found in other process descriptions.

Another benefit of this dependency model lies in the ability of the handbook to determine the set of processes that can manage the various dependencies in a process description. When a process is being analyzed, alternative coordination processes may be suggested that are not obvious to the users, allowing the users to view the process from a new point of view.
Appendix B

B.1 Techniques for Displaying Hierarchical Information on the Web

The ability to display hierarchies in the interface is a very important. The information in the Process Handbook is very hierarchical. An activity's decomposition is composed of subactivities, which themselves may have a decomposition, and an activity's specializations. Displaying these relationships properly will be key. Showing hierarchies can be purely graphical if an ActiveX control or Java applet is used. However, the disadvantages of using these technologies makes it compelling to find a way to show hierarchies using standard HTML. Below, two techniques are covered. One technique uses definition lists and another technique uses tables.

B.1.1 List Hierarchies

List elements have been a part of HTML since its conception. Specifically, definition lists - the HTML element "<DL>" - provide a good mechanism to emulate hierarchy. Definition lists can be nested within each other, allowing the emulation of hierarchies. Items in a definition list are created by two HTML elements, "<DT>" and "<DD>". The "<DT>" element creates an entry in the definition list, and the "<DD>" element contains the definition of the entry. Figure B.1 gives an example of how definition lists can be used to create a hierarchy. Figure B.1a shows the HTML source, and Figure B.1b displays the results in a web browser. By inherently providing different amounts of indentation, different levels of the hierarchy can be easy recognized, and it will be easy to see the children of a particular parent. Definition lists are easy to generate, but it can only display hierarchies in one way - in a top-down manner analogous to an outline. Because of this, siblings may be greatly separately visually when their descendants are also being displayed.
A) ...

<DL>
  <DT>Parent 1
  <DD>
  <DL>
    <DT>1st Child of Parent 1
    <DD>
    <DL>
      <DT>1st Grandchild of Parent 1
      <DD>
    <DT>2nd Grandchild of Parent 1
    <DD>
  </DL>
  <DT>2nd Child of Parent 1
  <DD>
  </DL>
  <DT>Parent 2
  <DD>
  <DL>
    <DT>1st Child of Parent 2
    <DD>
    <DT>2nd Child of Parent 2
    <DD>
  </DL>
</DL> ...

Figure B.1 - A simple list hierarchy. A) the source code for the hierarchy. B) The display of the hierarchy in a Web browser.

B.1.2 Table Hierarchies

Tables, were introduced in version 3.0 of HTML. If formatted properly, tables can also be used to display hierarchical information. Simple tables consist of cells of information that form rows and columns. Using certain attributes available in HTML, a table cell can be formatted to fitted next to several adjacent cells, thus appearing to encompass these cells and emulating a parent-child relationship. However, reading the HTML code does not give the same immediate indication of a hierarchy that the HTML code for definition lists does. Also, the visual properties of tables emphasizes parent-child relationships differently.

The main advantage that tables have over definition lists is that tables can show hierarchies in a vertical or horizontal fashion. However, tables may be visually confusing for some because there is no space between non-siblings in a table display, and it lacks a strong visual separator for groups of siblings.

The HTML tag for a table is “<TABLE>“. To define a row in the table, use the “<TR>” element. Cells are defined using the “<TD>” element. Figure B.2 shows how tables can be used to create a vertical (top-down) hierarchy. Figure B.2a shows the HTML source, and Figure B.2b displays the results in a web browser. Child cells are clustered underneath the parent cell in the table. The “COLSPAN” attribute of the “<TD>” element plays a key role in proper formatting of this table.
Figure B.3 shows how tables can be used to create a horizontal (left-right) hierarchy. Child cells are clustered to the right of their parent cell in this format. In this table, the "ROWSPAN" attribute is the key to proper formatting of the horizontal hierarchy.

A)

```
<TABLE border=1>
  <TR>
    <TD COLSPAN=3> Parent 1
    <TD COLSPAN=2> Parent 2
  </TR>
  <TR>
    <TD COLSPAN=2> 1st Child of Parent 1
    <TD> 2nd Child of Parent 1
    <TD> 1st Child of Parent 2
    <TD> 2nd Child of Parent 2
  </TR>
  <TR>
    <TD> 1st Grandchild of Parent 1
    <TD> 2nd Grandchild of Parent 1
  </TR>
</TABLE>
```

B)

![Horizontal Table Hierarchy](image)

Figure B.3 - The horizontal table hierarchy. A) shows the source code. B) shows the table in a Web browser.

A)

```
<TABLE BORDER=1>
  <TR>
    <TD ROWSPAN=3> Parent 1
    <TD ROWSPAN=2> 1st Child of Parent 1
    <TD> 1st Grandchild of Parent 1
  </TR>
  <TR>
    <TD> 2nd Grandchild of Parent 1
  </TR>
  <TR>
    <TD> 2nd Child of Parent 1
  </TR>
  <TR>
    <TD ROWSPAN=2> Parent 2
    <TD> 1st Child of Parent 2
  </TR>
  <TR>
    <TD> 2nd Child of Parent 2
  </TR>
</TABLE>
```

B)

![Horizontal Table Hierarchy](image)

Figure B.3 - The horizontal table hierarchy. A) shows the source code. B) shows the table in a Web browser.
Bibliography


