A KNOWLEDGE-BASED TECHNOLOGY ADVISING SYSTEM
FOR WEB-BASED APPLICATION DEVELOPMENT

A Thesis Presented to
The Faculty of the
Fritz J. and Dolores H. Russ
College of Engineering and Technology
Ohio University

In Partial Fulfillment
Of the Requirement for the Degree
Master of Science

by
Denis Lissitsyn
June, 2001
TABLE OF CONTENTS

LIST OF TABLES v

LIST OF FIGURES vi

1. INTRODUCTION 1
   1.1 Object and Scope of the Thesis 2
   1.2 Organization of the Thesis 3

2. BACKGROUND AND RELATED WORK 4
   2.1 Review of the Technologies/Web Programming Languages 7
      2.1.1 Server Side Programming Languages 7
         2.1.1.1 JSP (Java Server Pages) 8
         2.1.1.2 ASP (Active Server Pages) 9
         2.1.1.3 Java Servlets 10
         2.1.1.4 PHP (PHP: Hypertext Preprocessor) 11
         2.1.1.5 SSJS – Server Side JavaScript 12
         2.1.1.6 ColdFusion (CF) 12
         2.1.1.7 Perl 13
         2.1.1.8 SSI (Server Side Includes) 14
      2.1.2 Programming Languages for both Client and Server Side 15
         2.1.2.1 JavaScript 15
         2.1.2.2 JScript 16
         2.1.2.3 VBScript 16
      2.1.3 Client Side Technologies 18
         2.1.3.1 Java applets 18
         2.1.3.2 DHTML (Dynamic Hyper Text Markup Language) 19
         2.1.3.3 Flash 20
      2.1.4 Other Technologies 21
         2.1.4.1 EJB (Enterprise Java Beans) 21
         2.1.4.2 COM+ (Component Object Model) 22
         2.1.4.3 ActiveX 22
   2.2 Related Work 23

3. KNOWLEDGE-BASED SYSTEMS: GOAL AND STRUCTURE 24
   3.1 Stages of Building an Knowledge-based System 25
      3.1.1 Problem Definition 25
      3.1.2 Knowledge Acquisition 26
      3.1.3 Knowledge Interpretation 26
      3.1.4 Testing 27
   3.2 Exsys Developer™ 8.0 27
3.3 Capabilities of Exsys Developer 8.0 28
3.4 Hardware Requirements 29
3.5 Software Requirements 29

4. TASWAD: DESCRIPTION OF THE DEVELOPED SYSTEM 30
   4.1 System Goals 31
   4.2 System Questions 32
   4.3 Rules of the System 54
   4.4 Tabular Comparison of the Advised Technologies 55

5. SAMPLE TEST 62

6. CONCLUSIONS AND FURTHER RESEARCH 71

References 74

APPENDIX: Rules of the System 76
# LIST OF TABLES

Table 4.1 - Explanations of the notations used in table 4.2  
Table 4.2 - Existing Technologies Comparison – Group 1  
Table 4.3 - Explanations of the notations used in table 4.4  
Table 4.3 - Explanations of the notations used in table 4.4 (continued)  
Table 4.4 - Existing Technologies Comparison – Group 2  
Table 4.5 - Explanation of the notations used in table 4.6  
Table 4.5 - Explanations of the notations used in table 4.6 (continued)  
Table 4.6 - Existing Technologies Comparison – Group 3
LIST OF FIGURES

Figure 3.1 – Example of the Exsys Developer™ window. 28
Figure 4.1 – Structure of TASWAD. 30
Figure 4.2 – Goals of TASWAD. 31
Figure 4.3 – System results window. 32
Figure 4.4 – Sample question of TASWAD. 33
Figure 4.5 – Question 1, Preference of Microsoft products. 33
Figure 4.6 – Question 2, Appearance vs. functionality. 34
Figure 4.7 – Question 3, Familiarity with EJB. 35
Figure 4.8 – Question 4, NAS availability. 35
Figure 4.9 – Question 5, Change of application logic. 36
Figure 4.10 – Question 6, Multiple display formats. 36
Figure 4.11 – Question 7, Time constraints. 37
Figure 4.12 – Question 8, Intention to learn Java. 37
Figure 4.13 – Question 9, Need for attractiveness. 38
Figure 4.14 – Question 10, Speed of communication channels. 38
Figure 4.15 – Question 11, Intention to develop middleware. 39
Figure 4.16 – Question 12, Need for a portable solution. 39
Figure 4.17 – Question 13, Need for an application server independent solution. 40
Figure 4.18 – Question 14, Company size. 40
Figure 4.19 – Question 15, Type of the platform. 41
Figure 4.20 – Question 16, Familiarity with OLE and ActiveX. 41
Figure 4.21 – Question 17, Code reusability.

Figure 4.22 – Question 18, IIS availability.

Figure 4.23 – Question 19, Preference of free software.

Figure 4.24 – Question 20, Access to databases.

Figure 4.25 – Question 21, Application speed.

Figure 4.26 – Question 22, Text processing.

Figure 4.27 – Question 23, Availability of free scripts.

Figure 4.28 – Question 24, Low-level programming.

Figure 4.29 – Question 25, Form processing.

Figure 4.30 – Question 26, Access to a server.

Figure 4.31 – Question 27, Proficiency of the user.

Figure 4.32 – Question 28, Type of browser program.

Figure 4.33 – Question 29, Enhanced functionality provided by ActiveX.

Figure 4.34 – Question 30, ActiveX code reusability.

Figure 4.35 – Question 31, Knowledge of languages.

Figure 4.36 – Question 32, Tool for strings, numbers and dates.

Figure 4.37 – Question 33, Intention to work with COM.

Figure 4.38 – Question 34, Ability to rearrange objects.

Figure 4.39 – Question 35, Experience with JavaScript.

Figure 4.40 – Question 36, Need for advanced animation.

Figure 4.41 – Question 37, Time vs. power.

Figure 4.42 – Question 38, Switching to another application server.

Figure 4.43 – Question 39, Need for different services.
Figure 4.44 – Question 40, Kind of the platform.

Figure 5.1 – Sample run question 1, size of the company.

Figure 5.2 – Sample run question 2, Microsoft orientation.

Figure 5.3 – Sample run question 3, availability of IIS.

Figure 5.4 – Sample run question 4, knowledge of languages.

Figure 5.5 – Sample run question 5, need for attractiveness.

Figure 5.6 – Sample run question 6, kind of browser.

Figure 5.7 – Sample run question 7, intention to learn a sophisticated language.

Figure 5.8 – Sample run question, need for a tool that handles strings, numbers, etc.

Figure 5.9 – Sample run question 9, free software.

Figure 5.10 – Sample run question 10, need for another application server.

Figure 5.11 – Sample run question 11, experience with JavaScript.

Figure 5.12 – Sample run question 12, intention to develop middleware.

Figure 5.13 – Sample run question 13, advanced formatting.

Figure 5.14 – Sample run question 14, programming-centric approach vs. page-centric approach.

Figure 5.15 – Sample run result.
Chapter 1

INTRODUCTION

"The wonderful thing about standards is there are so many different competing ones from which to choose" [1] – this half serious computer science saying can describe exactly the idea behind this master thesis. The number of technologies (and programming languages) available to modern web programmers is very big and it keeps growing. It is very likely that new ones will eventually come into being as the Web expands and its users demand new possibilities and more functionality.

Internet is moving from a giant information storage medium to a universal communication and control instrument. It would be very brave to predict the modern state of the Internet 5 or 6 years ago with cellular phones, and even cars being connected to the network now. It will come at no surprise in a couple of years that you’ll be able to telnet to your home refrigerator from work and see if there is any beverage left.

New functionality and new devices require new software technologies to support them. World Wide Web was not created with mathematical formulas and equations in mind that is why MathML (Mathematical Markup Language) was introduced in 1997 [2]. And it is not obvious yet what other directions the Internet can go in the future and what
new software will be needed to ease that transition. As this is going on a need for a more organized way to choose an appropriate technology is going to increase.

1.1 Object and Scope of the Thesis

This thesis deals with the design and implementation of a knowledge-based Technology Advising System for the Web-based Application Development (TASWAD). The proposed tool uses knowledge from the web programming languages/technologies domain acquired by the developer to perform an advising task. Having significant amount of data encapsulated in it, the system is aimed to a very broad and diverse public ranging from beginners who just make their first effort to create their own web page to experienced programmers who need to develop a large scale enterprise web site. The most suitable instrument for performing Web development can be advised by means of the system. The difficulty and importance of making a right decision about what technology to use for a Web related project should not be underestimated. Such decision is quite a demanding task with the number of existing programming tools being almost 20 and each of them being rather complicated and time consuming to study. Learning all of them is impossible and hardly needed. However, sufficient knowledge and good understanding of the environment is absolutely necessary to make the right choice. Making this choice is the ultimate purpose of the knowledge-based system. It helps making hard decisions by doing all the tough work. All the user needs to do is to have understanding of what he or she is going to achieve. The system will ask a series of
questions and tell the user how to achieve that goal. All the complexity of deciding is hidden within system rules and the inference engine.

1.2 Organization of the Thesis

The main purpose of this thesis is the design and development of the prototype. Chapter 1 walks you through a brief introduction of the problem domain. It also introduces the layout and the structure of the document. The second chapter is dedicated to the background of the thesis, the reasons why the system was developed and to the related work. Chapter 3 explains the concept of knowledge-based systems, their purpose, structure, advantages and limitations, and how they can serve in accomplishing the goal of this particular thesis. Chapter 4 introduces the developed knowledge-based system and reveals its details. It discusses why the particular knowledge-based system shell was chosen and how it fits the requirements of the system. The underlying structure of the developed program is also introduced. The components that build up the system (questions, variable, rules, etc.) are presented and their purpose is explained. Chapter 5 is entitled “Conclusions and further research”. It logically brings the discussion to the end summarizing the work done and specifying what else can be improved to make the system more functional and useful. Appendix A consists of the rules that are the main driving force of the system. Rules are written using native Exsys Developer syntax, which resembles natural language.
Chapter 2

BACKGROUND AND RELATED WORK

A modern Web developer cannot complain about lack of appropriate tools for creating Web application. Here is a list of the most popular ones:

- Java applets
- Java servlets
- JSP (Java Server Pages)
- EJB (Enterprise Java Beans)
- ASP (Active Server Pages)
- PHP (PHP: Hypertext Preprocessor)
- SSJS (Server Side JavaScript)
- Perl (Practical Extraction and Report Language)
- JavaScript
- JScript
- VBScript
- DHTML/XHTML (Dynamic Hyper Text Markup Language/extensible Hyper Text Markup Language)
• ActiveX
• ColdFusion
• SSI (Server Side Includes)
• HTML (Hyper Text Markup Language)

And this list goes on. A logical question arises – why are there so many languages? It is known from experience that computer languages (both full functional and scripting ones) differ not so much in what they make possible, but in what they make easy. It is possible to write a CGI program in C but this is not the easiest and the most elegant way to do it. As the World Wide Web is evolving more and more different needs require their ways of implementation. Internet users (not very numerous and picky at the beginning) were satisfied with the static content of the Internet 5 or 6 years ago. There was the good old HTML that was simple to learn but it was not a programming language – it was a markup language that only served a purpose of specifying the way your information should look like in a browser. However, this period of simplistic web applications development did not last long, users of the web were no longer satisfied with it and this is where JavaScript [23] and similar languages came in. They offered a great deal of control over such things as browser windows, bars, buttons, mouse clicks, cursor movements, etc. Hence, the client side of the Web was taken care of. However, scripting languages were not complete and could not provide all the necessary functionality like database access, session control, file manipulation, multithreading etc. This is why server side programming languages were introduced. These are Perl[22], SSJS (Server Side
Pages[8]) to name a few. Programming on the server side lets the developer not to worry about what platform or operating system or browser the user on the other end has, because these programming languages produced plain HTML code as their output after being processed by a web server.

Programming languages are meant to simplify things. Some programming languages are extremely good at crunching numbers, other simplify your life by allowing an elegant approach to text processing and regular expressions manipulation, yet other ones give you a handy solution to low level socket or multithread programming problems.

Imagine you are a Web developer (maybe just a beginner) confronted with some application to develop. With so many different programming languages and technologies out there a logical question arises once you start a new project – what is the best language for me to use? Which tool is the most appropriate for my task? What programming language will save my time, effort and, of course, my money? Unfortunately there isn’t ‘one size fits all’ answer and until recently there was no service, or web site, or program to answer such question. Until this thesis had been written. The knowledge-based system that is the object of this master thesis (TASWAD – Technology Advising System for the Web-based application development) provides a unique capability to determine what implementation tool a web developer (no matter how experienced she or he is) should use for a particular task.
2.1 Review of the Technologies/Web Programming Languages

To simplify the review of all technologies manipulated by TASWAD they were organized in groups. By putting similar tools next to each other their evaluation is carried out easier.

2.1.1 Server Side Programming Languages

First, let us consider server side programming languages. These are

- JSP (Java Server Pages)
- ASP (Active Server Pages)
- Java Servlets
- PHP (PHP: Hypertext Preprocessor)
- SSJS (Server Side JavaScript)
- ColdFusion
- Perl

As the name implies programs or scripts written using these technologies run on a server. Since the programs are executed on the server-side, viewing the source code of the resulting HTML document doesn't reveal the use of any programming what so ever. All that the end user receives is a document containing plain HTML code (unless some client side programming language as JavaScript is used also). You can use server-side scripts to make dynamic Web sites, with different clients seeing different pages.
depending on any number of conditions. Another nifty implementation is to return
different Web pages to users with different browsers. The server approach also offers
security and efficiency. You enhance security by restricting access to your code and
implementation details and you increase efficiency by utilizing database connectivity and
Internet services such as email.

Following are short overviews of the server side implementation tools listed at the
very beginning of this chapter.

2.1.1.1 JSP (Java Server Pages)

JSP is an extension of servlets technology developed by SUN Microsystems. Anything
that is done in JSP can be done with servlets, however, JSP allows you to easily
mix static HTML with your code. Typically, it is also easier to read the code and
visualize the page that will ultimately be generated. A document created with JSP is an
HTML document with Java code imbedded in it. In a nutshell, the JSP page is being
converted to a normal servlet, with static data being written to an output stream. There
are ways to reduce the actual amount of code written in the servlet, but even though they
both generate the same output, JSP is easier to read and easier to write. Clearly, JSP and
servlets have their own distinct roles and uses, which allow the developer freedom and
ease of use. One of the advantages of using JSP over servlets is that the HTML code that
is responsible for the look of a page better separated from Java code that deals with the
logic of an application[14]. Hence, if there are two different Web designers supporting or
maintaining the same site, one being a programmer and the other being an artist there is no need to ask the programmer to change his or her code every time a slight modification of the web page is needed. Java Server Pages are platform independent supporting the concept of WORA – “Write Once Run Anywhere”.

2.1.1.2 ASP (Active Server Pages)

Active Serve Pages technology is developed by Microsoft and it is one of the biggest competitors of JSP. Microsoft ASP is a server-side scripting technology that can be used to create dynamic Web applications. An ASP page is an HTML page that contains server-side scripts that are processed by the Web server before being sent to the user’s browser. You can combine ASP with Extensible Markup Language (XML) and Hypertext Markup Language (HTML) to create powerful interactive Web sites.

Server-side scripts run when a browser requests an .asp file from the Web server. ASP is called by the Web server, which processes the requested file from top to bottom and executes any script commands. It then formats a standard Web page and sends it to the browser[4].

It is possible to extend your ASP scripts using COM+[19] components and XML. COM+ extends your scripting capabilities by providing compact, reusable, and secure means of gaining access to information. ASP allows developers to code custom tags in JavaScript (JScript) or VBScript. VBScript is the default scripting language for ASP, but
if you like you can use JScript, Perl or any other scripting language for server side scripting in an ASP page.

2.1.1.3 Java Servlets

Servlets are the Java technology’s answer to CGI programming. They are programs, which run on the server side and generate dynamic content. Advantages of using servlets are:

Efficiency – while using CGI programs, each time an HTTP request is received a new process is started, which can result in poor performance and scalability issues. Using servlets, the Java VM (Virtual Machine) is always running, therefore, starting a Servlet creates a Java thread as opposed to a system process.

Power – Servlets allow you to easily share data and maintain information by providing access to the entire family of Java APIs (Application Program Interface).

Security – Servlets can run by the Servlet engine in a restrictive sandbox, similar to web browser’s sandbox for applets. This helps protect against malicious Servlets, which represent a threat.

Portability – The Servlet API takes advantage of the Java platform. It is a fairly simple API that is supported by nearly all web servers so that Servlets may be moved from platform to platform, usually without any modification whatsoever.

A Servlet is a Java class and thus needs to be executed by the Java Virtual Machine, called a Servlet engine. Servlets are loaded by the engine when they are called
and remain running until the Servlet is explicitly unloaded or the engine is shut down [14]. An easy way to visualize a Servlet is to reverse the concept of JSP. In JSP Java code is embedded in HTML whereas in Servlets HTML code is contained within Java code.

2.1.1.4 PHP (PHP: Hypertext Preprocessor)

PHP (officially "PHP: Hypertext Preprocessor") is a server-side HTML-embedded scripting language. If you understand the concept used in ASP and JSP (embedding code in some scripting language within HTML) it won't be hard to get familiar with PHP. The idea is the same. However, details (like the language itself) differ. Much of PHP is a combination of Perl, Java, and C concepts. The syntax structure borrows heavily from C, making it an easy language to learn for even the novice programmer. PHP performs sophisticated mathematical calculations, provides network information, offers mail and regular expression capabilities, and much more. PHP's strongest feature is its database interfacing capability. Connecting a database to the Internet is very easy. What's more, it supports many of the most popular database servers on the market, including MySQL, Oracle, Sybase, mSQL, Generic ODBC, and PostgreSQL, to name a few [20]. PHP is free and it is supported by major web servers (such as Apache).
2.1.1.5 SSJS – Server Side JavaScript

Server-side JavaScript is composed of core JavaScript and additional objects and functions for accessing databases and file systems, sending e-mail, and so on. SSJS enables developers to quickly and easily create database-driven web applications by leveraging their existing knowledge of JavaScript. It's used to create and/or customize server-based applications by scripting the interaction between objects.

SSJS is included with the Netscape Enterprise Server and is used for creating web applications that can be run on any platform and on any browser.

2.1.1.6 ColdFusion (CF)

Allaire's ColdFusion is a product that acts as "middleware between any ODBC database (Access, Oracle, etc.) and a Web server for the delivery of "live" data to Web pages. It allows you to develop applications that dynamically input data to and output data from a database via the Internet and the Web.

ColdFusion, which is made up several components, is a powerful application development tool that enables developers to create interactive, dynamic Web sites. As a CF developer, you create CF files by simply extending your existing HTML files with CFML (ColdFusion Mark-up Language - tag-based server-side scripting language) and SQL (Structured Query Language). These pages are processed by the CF server and sent back to the Web server as an HTML page, which is then passed to the browser [24].
ColdFusion also handles e-mail, FTP and other services as well as the Web.

ColdFusion does not require coding in traditional programming languages, such as Perl and Java, though it does incorporate constructs and techniques of programming languages.

More Specifically, ColdFusion allows you to:

- Query existing database applications for data.
- Create dynamic queries facilitating more flexible data.
- Execute stored procedures in databases that support them.
- Enhance the standard HTML form capabilities with data validation functions.
- Dynamically populate form.
- Customize the display of dates, times and currency values with formatting functions.

2.1.1.7 Perl

Perl is a server side scripting language: powerful and adaptable. It was first developed by Larry Wall, a linguist working as a systems administrator for NASA in the late 1980s, as a way to make report processing easier. Since then, it has moved into a large number of roles: automating system administration, acting as glue between different computer systems; and, of course, being one of the most popular languages for CGI programming on the Web [22].
Why did Perl become so popular when the Web came along? Two reasons: first, most of what is being done on the Web happens with text, and is best done with a language that's designed for text processing. More importantly, Perl was appreciably better than the alternatives at the time when people needed something to use.

It also didn't hurt that Perl is a friendly language. It plays well with the programmer's personal programming style. The Perl slogan is "There's more than one way to do it," and that lends itself well to large and small problems alike.

2.1.1.8 SSI (Server Side Includes)

Server Side Includes or SSI is a technique that allows a Web developer to execute commands or scripts from within HTML files. It is very handy for small routine tasks as displaying of current time and date display, last date of file modification or file size. Another good implementation for SSI is insertion of a file inside another file. For example, if the Web developer has a copyright message or a header that needs to be repeated on every page of a Web site, that message or header could be stored as a separate file and then included wherever needed. Not only HTML files that are already created can be inserted in a file, a result of a CGI script (usually it is a Perl script) can be used as well.

SSI is easy to use but not every Web server supports it. If it does, files containing SSI calls usually have .shtml extension to tell the Web server that this file requires special kind of parsing before being send to the end user.
2.1.2 Programming Languages for both Client and Server Side

There are some simple tasks (like adding a clock to your web page) that do not need all the functionality that server side programming languages provide. Also, access to a server depends on an ISP (Internet Service Provider) and not all users of the Internet have that access. Therefore, there is a need for client side development tools. These are:

- JavaScript
- JScript
- VBScript

These languages are traditionally considered client side even though some of them can be used for the server side as well.

2.1.2.1 JavaScript

JavaScript is a compact, object-based scripting language for developing client and server Internet applications. JavaScript statements can be embedded directly in an HTML page. These statements can recognize and respond to user events such as mouse clicks, form input, and page navigation. For example, you can write a JavaScript function to verify that users enter valid information into a form. Without any network transmission, an HTML page with embedded JavaScript can interpret the entered text and alert the user with a message dialog if the input is invalid.
Or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page [23].

2.1.2.2 JScript

JScript is Microsoft's response to creation of JavaScript. JScript was designed as a general-purpose scripting language that would appeal to the many programmers who use C, C++, and Java [9]. This means that it "borrows" features from these languages where appropriate, but is a language in its own right and includes many features not found in C or Java (such as conditional compilation and identity checking [16]).

2.1.2.3 VBScript

VBScript is designed as a subset of the full Visual Basic programming system. The guiding principles of VBScript are:

1. Provide the Basic developer with a path to Web development on the client and server.


3. Expand the Visual Basic family of languages to platforms not covered by Visual Basic or Visual Basic for Applications.
Language features are listed below:

1. **Error handling.** VBScript has a subset of the error handling provided by Visual Basic. This includes the error object and on error resume next.

2. **Formatting.** VBScript has the ability to format dates, numbers, and currency built into the language.

3. **Easier COM Interoperability.** Many COM objects return information in the form of a collection. VBScript has built-in support for iterating through collections.

4. **Standard Event-binding Syntax.** Visual Basic developers will immediately recognize the object_event (sub button1_onclick) naming convention for event handlers. VBScript works in exactly the same way, so in any application that supports event binding (such as Internet Explorer or Outlook), you can use this syntax for your event hook-up code. This also means that you can cut and paste existing Visual Basic code into VBScript (assuming you are using features that are in VBScript).
2.1.3 Client Side Technologies

Pure client side technologies are presented in this section. They target different goals and offer yet another possibilities to the Web developers. Technologies that are used on the client side are:

- Java applets
- DHTML (Dynamic Hyper Text Markup Language)
- Flash

2.1.3.1 Java applets

There is a lot to say about Java. Java is powerful, safe, platform independent and notoriously slow. Java is a fine general purpose programming language. It can be used to effectively to generate stand-alone executables, just as other high-level languages can. In a shape of applets Java offers the additional capability of writing code that can live in a Web page and be downloaded and executed when the Web page is browsed. Here is why it is useful to put a program in a Web page and publish it on the Internet:

- Anyone, on any platform, with any version of any operating system can run the program immediately, without installing any additional software. (This is particularly attractive for the client side of e-commerce).
- It is simple to run the program; just browse the Web page it is on.
An applet enables a web page to move beyond a static one-way presentation to one with live content and user interaction capabilities [1].

2.1.3.2 DHTML (Dynamic Hyper Text Markup Language)

DHTML is the technology that enables control over content presentation and behavior, regardless of the user agent (such as a browser), or operating environment. This includes static properties that control the presentation, both visual and verbal, of an element, its position in relation to other elements, and its sensitivity to events, as well as more dynamic properties that can be used to alter the appearance, position, behavior, or display of an element after the contents are fully loaded [7].

Dynamic HTML, or DHTML, is not something that can easily be pointed out, or separated from other technologies. It is a group of technologies, when brought together, enables a Web developer to bring a Web page to life.

The three technologies that make up DHTML are HTML, JavaScript and Cascading Style Sheets (CSS). HTML is used for the basic structure of the document, JavaScript to manipulate the Document Object Model (DOM), and CSS to define the presentation and style of the document. The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented
page [17]. Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to Web documents [18].

In the olden days, you could only change content on a page via CGI. This required a server to perform the changes to the page and re-serve the entire page, modifications and all, back to the client. While workable, this process was quite slow, as it placed a burden on both network traffic and server processing time. With long delays between a user's action and an on-screen response, building effective Web-based applications was quite constricting [7].

2.1.3.3 Flash

Macromedia Flash is a dynamic multimedia program used to produce mainly vector-based presentations, introductions, and full sites. Its popularity stems from its unique ability to deliver crisp sound and colorful, animated, eye-pleasing graphics to viewers around the world. Developers are given a choice of drawing straight into Flash with the many tools provided or importing graphics from external graphics programs such as Adobe Illustrator or Macromedia Freehand. Dealing with Flash everyday, many developers will tell you that it is the definitive tool to break free of the constraints of HTML and basic web design. It provides the viewer with a more unique portrayal of the site and it's contents. Flash is a rather complicated tool to use but it seems to gain more and more popularity due to powerful combination of graphics, audio and video.
2.1.4 Other Technologies

As the name of this section implies it contains descriptions of the Web development technologies that can't be referred to as either server side or client side programming languages. The following subsections provide a brief overview of:

- EJB
- COM+
- ActiveX

2.1.4.1 EJB (Enterprise Java Beans)

Enterprise JavaBeans (EJB) technology defines a model for the development and deployment of reusable Java server components. Components are pre-developed pieces of application code that can be assembled into working application systems. Java technology currently has a component model called JavaBeans, which supports reusable development components. The EJB architecture logically extends the JavaBeans component model to support server components. Server components are reusable, prepackaged pieces of application functionality that are designed to run in an application server. They can be combined with other components to create customized application systems. Server components are similar to development components, but they are generally larger grained and more complete than development components. Enterprise JavaBeans components (enterprise beans) cannot be manipulated by a visual Java IDE in
the same way that JavaBeans components can. Instead, they can be assembled and customized at deployment time using tools provided by an EJB-compliant Java application server [19].

2.1.4.2 COM+ (Component Object Model)

The Component Object Model+ (COM+) is a way for software components to communicate with each other. It's a standard that allows any two components to communicate regardless of what machine they're running on (as long as the machines are connected), what operating systems the machines are running (as long as it supports COM+), and what language the components are written in. COM+ further provides location transparency: it doesn't matter to you when you write your components whether the other components are in-process DLLs, local EXEs, or components located on some other machine [19]. Code reusability is the main issue that is addressed by this component model.

2.1.4.3 ActiveX

ActiveX controls, formerly known as OLE (Object Linking and Embedding) controls or OCX (OLE Custom controls [20]) controls, are components (or objects) you can insert into a Web page or other application to reuse packaged functionality someone else programmed. For example, the ActiveX controls that are included with Microsoft
Internet Explorer version 3.0 or higher allow you to enhance your Web pages with sophisticated formatting features and animation.

A key advantage of ActiveX controls over Java applets and Netscape plug-ins is that ActiveX controls can also be used in applications written in many programming languages, including all of the Microsoft programming and database languages.

There are literally hundreds of ActiveX controls available today with functionality ranging from a timer control (which simply notifies its container at a particular time) to full-featured spreadsheets and word processors.

2.2 Related Work

Usually in this part of thesis a comparison is presented indicating that the developed system or program is better than the existing implementations. However, a search over the literature for a system or program with similar capabilities and functions didn’t produce a result, which could be not an absolute but a good proof of uniqueness of TASWAD.
Chapter 3

KNOWLEDGE-BASED SYSTEMS: GOAL AND STRUCTURE

Knowledge-based systems are a branch of the spread artificial intelligence tree. Knowledge-based expert systems, or simply expert systems, use human knowledge to solve problems that normally would require human intelligence, or experience. These knowledge-based systems represent the expertise knowledge as data or rules within the computer. The rules and data can be called upon when needed to solve problems. Books and manuals have a tremendous amount of knowledge but a human being has to read and interpret them before the knowledge can be used. Computer programs perform tasks using conventional decision-making logic - containing little knowledge other than the basic algorithm for solving that specific problem and the necessary boundary conditions. This program knowledge is often embedded as a part of the program code. A disadvantage of this approach is that if the knowledge changes, the program has to be changed also and then recompiled. Knowledge-based systems collect small fragments of human know-how into a knowledge base, which is used to reason through a problem, using the knowledge that is appropriate. A different problem, within the domain of the knowledge base, can be solved using the same program without reprogramming. The
ability of these systems to explain the reasoning process through back-traces and to handle levels of confidence and uncertainty provides an additional advantage that conventional programming does not offer.

3.1 Stages of Building an Knowledge-based System

Each knowledge-based system goes through a number of steps before it becomes a ready-to-ship product. There are four major stages in building an knowledge-based system:

1. Problem definition.
2. Knowledge acquisition.
4. Testing.

3.1.1 Problem Definition

First, the scope of the problem needs to be defined. It should become clear to the developer what the tasks that need to be achieved are and what the possible ways of doing that are. Different aspects such as acceptance of the use of the knowledge-based system to solve the problem and approximate calculation of resources (time) that will be required are also considered at this time.
3.1.2 Knowledge Acquisition

Sources that can be used to gather the necessary information vary. Most frequently it is a human expert or a group of human experts possessing some unique knowledge and expertise in a particular field being studied. It could also be books, magazines, various articles and publications, and, of course, the Internet. Knowledge acquisition may appear to be the most difficult part contributing to the process of building an knowledge-based system because of the ‘human factor’. The knowledge-based system developer not only needs to have a computer science background or skills to program the system, but also an ability to communicate with people well enough to discover the new domain, which could happen to be absolutely unfamiliar to him or to her.

3.1.3 Knowledge Interpretation

After the required knowledge is acquired time comes for interpreting the data and building rules. The information provided by the expert may need an intermediate step of analysis. This step allows to derive new knowledge that might have been omitted by the interviewee but still be helpful for the ultimate knowledge-based system purpose. When interpreting the data it is important to understand the underlying structure of the domain being studied in order to produce logical and correct dependencies and not to create misleading or incorrect ‘cause - consequence’ connections.
### 3.1.4 Testing

Once the obtained knowledge is interpreted and represented in the form of rules the system testing stage begins. Ideally the system should generate the same results and give the same advise as the human expert (assuming the knowledge was gathered by interviewing a human expert) would produce. It is a common practice, however, to return to the previous step to make necessary revision and apply changes wherever needed. Then testing takes place again. Depending on its results it is decided if another revision is necessary or not.

### 3.2 Exsys Developer™ 8.0

There are many commercial (and free) development environments (also called shells) that allow one to build an knowledge-based system no matter what the subject of study is. An knowledge-based system development environment called Exsys Developer™ 8.0 [5] was chosen for the TASWAD. Many other competing tools (Acquire SDK™, Jess™, etc.) were considered though, but being free, very powerful and user friendly Exsys Developer™ 8.0 proved to be the most suitable instrument for this project. An example screenshot of how the developer’s environment looks like is pictured on figure 3.1.
3.3 Capabilities of Exsys Developer 8.0

Exsys Developer is written in "C. It is a basic, versatile rule-based knowledge-based system development tool. The features that Exsys Developer 8.0 possess are:

- Enables application development in tree diagrams or edit individual rules
- Easy to understand and interpret English and algebraic syntax
- Forward and backward chaining
- 6 Confidence Modes - from YES/NO to Fuzzy Logic
- Command Language provides control over system procedural operations
- Gateway to many types of external programs
- Configuration options provide extensive customization
- Report Generator
3.4 Hardware Requirements

Knowledge-based systems developed using Exsys Developer™ are standalone applications that run on a local machine that does not need a network connection. The machine needs to be at least Pentium II with 32 Mbytes of RAM and 10 Mbytes of disk space.

The developed system does not require much space on the disk because it is represented using textual data mostly. Size of TAWAD is around 600Kb.

3.5 Software Requirements

In order to install Exsys Developer™ on your computer it needs to be running Microsoft Windows™ 95/98/2000 or NT. The software also requires Internet Explorer™ version 4 or higher because it uses HTML to display the contents (questions and results).

In order to deploy an knowledge-based system as a Java applet additional software package needs to be used – Exsys Web Runtime, which could be obtained from the Exsys, Inc.(www.exsys.com) – the developer of Exsys Developer.
Chapter 4

TASWAD: DESCRIPTION OF THE DEVELOPED SYSTEM

The major components of the created knowledge-based system are goals, questions, and rules. Figure 4.1 gives the idea of the internal structure of the system as well as the way it communicates with the user and generates results.

Figure 4.1 – Structure of TASWAD.
4.1 System Goals

Goals are basically the results that the system ultimately achieves. In other words, goals are conclusions that the knowledge-based system comes to after analyzing knowledge derived from answering questions. As a result of the analysis, a single goal or a group of goals from the list of existing ones is chosen. There are 18 goals in TASWAD. Shown on figure 4.2 is a screen-shot visualizing the existing system goals.

![Goals of TASWAD](image.png)

Figure 4.2 – Goals of TASWAD.

There is one-to-one correspondence between the displayed goals and the technologies/programming language that is advised if that goal is achieved by the knowledge-based system, except for goal #15, which is “None of the technologies/languages fits you”. This last goal means that the user has been answering
questions in a thoughtless, or inaccurate, or random manner and no result that would agree with the replies is generated. The achieved goal is displayed at the end of a session. Figure 4.3 gives an example of a resulting window with the derived goal (being a server side programming language Perl – Practical extraction and report language) shown.

![System Results Window](image)

Figure 4.3 – System results window.

4.2 System Questions

Another important component of the knowledge-based system is questions. To keep the program user friendly the questions have to be logical and understandable. Questions are based on key features of a particular technology or programming language. By answering a series of questions the user of the system causes a corresponding rule (or
number of rules) to fire and a particular goal to be chosen. Shown on figure 4.4 is a sample screen shot with a question and a list of possible answers to choose from.

![Sample question of TASWAD](image)

Figure 4.4 – Sample question of TASWAD.

Questions are a very important part of the system. The questions contain concentrated knowledge of the problem domain because they (and the rules that are built using questions) are a result of examining of the knowledge. The key features of the Web development technologies are extracted during the process of knowledge interpretation and then the questions are constructed based on those features.

Listed below are all the 40 questions used in TASWAD. Short comments that are placed under the screenshots are not a part of the program; they rather serve a clarifying and justifying purpose for the reader of this manuscript.
Figure 4.5 is a justification for question 1 – Knowing if the developer is Microsoft oriented allows dividing the flow of further questions. There are many technologies like ASP, ActiveX, etc. that need special support by Microsoft products. Some technologies, for example JSP and ASP, are very much alike, so, questions like this allow detecting a more preferable one.

Figure 4.5 – Question 1, Preference of Microsoft products.

Figure 4.6 is a justification for question 2 - Here, an attempt is made to find out if the page-centric or rather the programming-centric approach should be taken. Languages like JSP are more suitable if you have far more HTML than Java [1] code, or at least very little decision making about what document to display based on user input. On the other hand, Java servlets are good for sites that have a lot of processing and decision making, but not much HTML output.

Figure 4.6 – Question 2, Appearance vs. functionality.
Figure 4.7 is a justification for question 3 - EJB often goes along with JSP. EJB allows Web developers to factor out any business logic. Beans can be manipulated from inside JSP files. So, if the developer is planning to concentrate on the functional part of your Web site, JSP can be a good recommendation.

![Figure 4.7 - Question 3, Familiarity with EJB.](image)

Figure 4.7 – Question 3, Familiarity with EJB.

Figure 4.8 is a justification for question 4 - Some server tools, JSP for example, are supported by NAS or third party equivalents. So, having such a server already running will ease a transition to a new technology.

![Figure 4.8 - Question 4, NAS availability.](image)

Figure 4.8 – Question 4, NAS availability.
Figure 4.9 is a justification for question 5 - If frequent change of the application logic is expected, Java servlets might suit the programmer’s purposes better. As it was already mentioned, Java servlets address programming-centric approach, as opposed to JSP, which follows the page-centric approach.

Figure 4.9 – Question 5, Change of application logic.

Figure 4.10 is a justification for question 6 – Support of multiple display formats (as HTML and XML) is one of the advantages of Java servlets – ease of multiple display support.

Figure 4.10 – Question 6, Multiple display formats.
Figure 4.11 is a justification for question 7 - Different scripting and programming languages have different learning curves. JavaScript is a fairly easy language to learn for those having some programming experience. It should not be very time consuming to master. And it can be used for both client and server side development.

![Figure 4.11 - Question 7, Time constraints.](image)

Figure 4.11 – Question 7, Time constraints.

Figure 4.12 is a justification for question 8 – Difficulty of the language can be a real obstacle for some beginning developers. It is important not to underestimate ones abilities to get acquainted with a particular technical aspect.

![Figure 4.12 - Question 8, Intention to learn Java.](image)

Figure 4.12 – Question 8, Intention to learn Java.
Figure 4.13 is a justification for question 9 - Some languages and technologies provide an easy way of adding attractiveness to a Web site. Limitations of HTML can be overcome by using Java applets, DHTML and Flash. These tools can be utilized if the developer wants to catch a visitor's eye.

![Figure 4.13 - Question 9, Need for attractiveness.](image)

Figure 4.13 – Question 9, Need for attractiveness.

Figure 4.14 is a justification for question 10 - Java applets can grow big and take some time to download.

It is especially important for Web sites serving broad range of geographical regions to avoid that.

![Figure 4.14 - Question 10, Speed of communication channels.](image)

Figure 4.14 – Question 10, Speed of communication channels.
Figure 4.15 is a justification for question 11 - Application logic is handled especially easily if EJB, or COM+ components are used. ColdFusion environment is a solution to consider for large-scale businesses as well.

Figure 4.15 – Question 11, Intention to develop middleware.

Figure 4.16 is a justification for question 12 - One of the reasons why Java is so popular is because of portability of Java code. If the application is going to be migrated from one platform to a different one, EJB can handle that with less effort.

Figure 4.16 – Question 12, Need for a portable solution.
Figure 4.17 is a justification for question 13 - Again, when mobility (EJB is supported by most of the application servers) is an issue for a project, the best way to solve the problem if to use a Java-based technology like EJB.

Figure 4.17 – Question 13, Need for an application server independent solution.

Figure 4.18 is a justification for question 14 - Answers to this question differ from small to large. It could be enough to use just PHP or DHTML for a small-scale application whereas serious businesses should probably consider JSP and EJB. For individual developers there is also an option ‘I don’t work for a business’; in which case a lighter tool is recommended.

Figure 4.18 – Question 14, Company size.
Figure 4.19 is a justification for question 15 – Windows is the only platform supported by COM+ (Component Object Model).

![Figure 4.19 - Question 15, Type of the platform.]

Figure 4.20 is a justification for question 16 - If the user is familiar with OLE or ActiveX he or she shouldn’t have much trouble advancing to the next logical level, which is use of COM+ components.

![Figure 4.20 - Question 16, Familiarity with OLE and ActiveX.]

Figure 4.21 is a justification for question 17 - Code reusability is often confused with the process of copying a part of code from one program and pasting it to another. Professional Web programmers should understand that the easiest way to maintain a large-scale project is to factor out application logic and use a component manipulator technique such as EJB or COM+.

![Figure 4.21 - Question 17, Code reusability.](image1)

Figure 4.21 – Question 17, Code reusability.

Figure 4.22 is a justification for question 18 - The only way to run ASP on a server is to have IIS (Internet Information Server) or a third party equivalent installed.

![Figure 4.22 - Question 18, IIS availability.](image2)

Figure 4.22 – Question 18, IIS availability.
Figure 4.23 is a justification for question 19 - If you are coming from UNIX world you should prefer PHP (PHP: Hypertext Preprocessor), which is freely available for installation on Apache or an equivalent Web server.

![Figure 4.23 - Question 19, Preference of free software.](image)

Figure 4.23 – Question 19, Preference of free software.

Figure 4.24 is a justification for question 20 - If access to databases is needed, a server side programming language needs to be chosen. PHP provides an outstanding support for all existing database drivers.

![Figure 4.24 - Question 20, Access to databases.](image)

Figure 4.24 – Question 20, Access to databases.
Figure 4.25 is a justification for question 21 - Performance tests showed that applications written in PHP are faster than ones written using JSP.

Figure 4.25 – Question 21, Application speed.

Figure 4.26 is a justification for question 22 – Perl is very good at processing text (textual data, reports, news, emails, log files, etc.).

Figure 4.26 – Question 22, Text processing.
Figure 4.27 is a justification for question 23 – For some tools one can always find supportive examples and ready pieces of code.

![Figure 4.27 - Question 23, Availability of free scripts.](image)

Figure 4.27 – Question 23, Availability of free scripts.

Figure 4.28 is a justification for question 24 – Perl is a powerful instrument to do low-level network programming.

![Figure 4.28 - Question 24, Low-level programming.](image)

Figure 4.28 – Question 24, Low-level programming.
Figure 4.29 is a justification for question 25 - Some languages, especially Perl that was created with regular expressions in mind, handle text data very well. Nowadays many Web sites have forms for gathering information from their visitors.

![Figure 4.29 - Question 25, Form processing.](image)

Figure 4.29 – Question 25, Form processing.

Figure 4.30 is a justification for question 26 - Most beginning Web programmers or people who just want to create their own web page will answer 'yes' to this question. In this case client side programming tools like JavaScript might be advised.

![Figure 4.30 - Question 26, Access to a server.](image)

Figure 4.30 – Question 26, Access to a server.
Figure 4.31 is a justification for question 27 - Novices usually start by exploiting JavaScript or similar scripting languages because they are easy to master and there is plenty of free code examples and documentation.

![Figure 4.31](image1)

Figure 4.31 – Question 27, Proficiency of the user.

Figure 4.32 is a justification for question 28 - Programming languages have their preferences, for example VBScript is not supported by Netscape browser, whereas Jscript is.

![Figure 4.32](image2)

Figure 4.32 – Question 28, Type of browser program.
Figure 4.33 is a justification for question 29 - Advanced formatting, animation, timers, background downloading etc. form a short list of examples of what kind of functions ActiveX components can offer.

![Figure 4.33 - Question 29, Enhanced functionality provided by ActiveX.](image)

Figure 4.34 is a justification for question 30 - Some ActiveX components developed for the Web can be also used in other Microsoft applications.

![Figure 4.34 - Question 30, ActiveX code reusability.](image)
Figure 4.35 is a justification for question 31 - Several options are given to answer the question – What of the following languages (VB, VBA, VBScript, Jscript, Python) do you know? The user input can be useful to determine a suitable Web-programming language. For example, it is much easier to learn VBScript if one knows VB already.

![Figure 4.35 - Question 31, Knowledge of languages.](image)

Figure 4.36 is a justification for question 32 – Dealing with strings, numbers, dates, text files can be effectively performed by programs written in VBScript.

![Figure 4.36 - Question 32, Tool for strings, numbers and dates.](image)
Figure 4.37 is a justification for question 33 - It doesn’t come as a surprise that COM+ components are better supported by another Microsoft products, such as VBScript.

![Figure 4.37 - Question 33, Intention to work with COM.](image)

Figure 4.37 – Question 33, Intention to work with COM.

Figure 4.38 is a justification for question 34 - DHTML offers an outstanding ability to manipulate objects without communication with the server after a Web page has been downloaded. Another tool that can offer different visual effects and interactivity is Java applets.

![Figure 4.38 - Question 34, Ability to rearrange objects.](image)

Figure 4.38 – Question 34, Ability to rearrange objects.
Figure 4.39 is a justification for question 35 - JavaScript is a part of DHTML, so knowing the first one will make learning the second much easier. Of course, DHTML requires the knowledge of HTML (Hyper Text Markup Language) and CSS (Cascade Style Sheets) also.

Figure 4.39 – Question 35, Experience with JavaScript.

Figure 4.40 is a justification for question 36 - There are many advantages that Flash technology can offer. Listed on the figure above is a short description of its functionality. Flash files are also smaller than animated GIFs, hence, it will take them less time to download.

Figure 4.40 – Question 36, Need for advanced animation.
Figure 4.41 is a justification for question 37 - The flipside of the Flash coin is that
the tool is pretty complicated and takes time to learn.

![Figure 4.41 – Question 37, Time vs. power.]

Figure 4.42 is a justification for question 38 - ColdFusion applications need their
own application server. This limits the range of systems that can be used with this
application and, unfortunately, third parties don’t offer support for ColdFusion yet.

![Figure 4.42 – Question 38, Switching to another application server.]

Figure 4.43 is a justification for question 39 – Connection to RDBMS (Relational Data Base Management System), messaging servers, file repositories, directory servers, etc. are some of the services that powerful ColdFusion environment can offer.

Figure 4.43 – Question 39, Need for different services.

Figure 4.44 is a justification for question 40 – Windows NT or Solaris are the only platforms a ColdFusion server can run on.

Figure 4.44 – Question 40, Kind of the platform.
4.3 Rules of the System

A knowledge-based system is managed by the rules. The rules are easier to build if the domain knowledge is somehow structured, divided in groups. By pointing out a feature that is common to a sufficiently big part of the technologies we can divide the original set into two subsets. Each of the subsets could be divided into smaller parts in turn. Use of such a structured approach eliminates unnecessary work (searching through the whole knowledge base) that would have to be done otherwise. For example, it would make sense to divide the existing technologies into client side and server side groups, or another example could be a division of the Web programming languages into those bounded to Microsoft platform (ASP – Active Server Pages, etc.) and those being platform independent (JSP - Java Server Pages).

If all the necessary conditions (the IF part of a rule) are met the rule fires (the THEN part of the rule is executed). Firing of a rule leads to selection of a goal from the goal list (list of all technologies that can be advised). A technology can be advised with a different degree of confidence depending on the user input. For TASWAD a ten-point scale was chosen.

There is an upper limit on number of rules a knowledge-based system developed using an evaluation version of Exsys Developer can use, which is 50. This is sufficient for a system like TASWAD. It proved that 18 rules suffice to serve the task of finding out what programming language/technology a particular project might require. All 18 rules used in TASWAD are listed in the APPENDIX.
4.4 Tabular Comparison of the Advised Technologies

Web development technologies have similarities as well as distinct features. To make the process of choosing a proper tool for a particular task easier these characteristics need to be revealed. The following tables represent an attempt of such systemization that served as a basis for building questions and then rules of TASWAD. The implementation tools were divided in groups according to the schema already used in Chapter 2 – group 1 consists of server side technologies, group 2 – both client and server side, group 3 – pure client side and group 4, which contains all the rest. Tables 4.2, 4.4, 4.6 were present the comparison of existing technologies. For better visual perception supplementary tables 4.1, 4.3 and 4.5 were created. A combination of an upper case letter and a digit within a cell (for example C3 in the table 4.3) denotes that the technology that corresponds to that column has that feature (C3), meaning of which can be looked up in preceding table. A combination of an exclamation sign, upper case letter and a digit means that the feature is not supported by this technology. A feature that does not affect the choice of a technology is presented with a blank cell.
Table 4.1 Explanations of the notations used in table 4.2.

<table>
<thead>
<tr>
<th>Notation used in table 4.2</th>
<th>Meaning of the notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Small applications (big projects become unmanageable, hard to debug).</td>
</tr>
<tr>
<td>S2</td>
<td>Frequent HTML redesign, you have far more HTML than Java code.</td>
</tr>
<tr>
<td>S3</td>
<td>Developer is familiar with Java Beans, or is planning to work with Java Beans.</td>
</tr>
<tr>
<td>S4</td>
<td>Is supported by NAS (Netscape application server) or equivalent.</td>
</tr>
<tr>
<td>S5</td>
<td>Application logic is changed often times (Servlets address programming-centric approach).</td>
</tr>
<tr>
<td>S6</td>
<td>Multiple display formats (HTML, XML) need to be supported.</td>
</tr>
<tr>
<td>S7</td>
<td>Developer is short on time (JavaScript is easier than other server side languages).</td>
</tr>
<tr>
<td>S8</td>
<td>Developer doesn’t want to learn a sophisticated language like Java, has knowledge of JavaScript and thinks it’s enough.</td>
</tr>
<tr>
<td>S9</td>
<td>Developer is rather Netscape than MS oriented developer.</td>
</tr>
<tr>
<td>S10</td>
<td>The business is small to medium size.</td>
</tr>
<tr>
<td>S11</td>
<td>IIS is installed on the server.</td>
</tr>
<tr>
<td>S12</td>
<td>Developer knows how to program in VBScript or Jscript or Python or Perl.</td>
</tr>
<tr>
<td>S13</td>
<td>Developer favors free software.</td>
</tr>
<tr>
<td>S14</td>
<td>Developer needs a cross platform (portable) solution.</td>
</tr>
<tr>
<td>S15</td>
<td>Developer needs access to databases.</td>
</tr>
<tr>
<td>S16</td>
<td>Speed is a concern for the developer (PHP is faster than JSP).</td>
</tr>
<tr>
<td>S17</td>
<td>Developer wants the code to be intellectually protected.</td>
</tr>
<tr>
<td>S18</td>
<td>Developer is starting to build a large scale website from the scratch.</td>
</tr>
<tr>
<td>S19</td>
<td>If the developer has an application server it is possible to switch to CF application server from your existing one.</td>
</tr>
<tr>
<td>S20</td>
<td>The platform is either Windows NT or Solaris</td>
</tr>
</tbody>
</table>
### Table 4.2 Existing Technologies Comparison – Group 1.

<table>
<thead>
<tr>
<th>JSP</th>
<th>Servlets</th>
<th>SSJS</th>
<th>ASP</th>
<th>PHP</th>
<th>ColdFusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>!S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>!S5</td>
<td>S5</td>
<td>!S5</td>
<td>!S5</td>
<td>!S5</td>
</tr>
<tr>
<td>!S5</td>
<td>S6</td>
<td>S7</td>
<td></td>
<td>S7</td>
<td></td>
</tr>
<tr>
<td>!S8</td>
<td>!S8</td>
<td>S8</td>
<td></td>
<td></td>
<td>!S8</td>
</tr>
<tr>
<td>!S9</td>
<td>!S10</td>
<td>S10</td>
<td>S10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!S11</td>
<td>!S11</td>
<td>S11</td>
<td></td>
<td>S12</td>
<td></td>
</tr>
<tr>
<td>S12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S14</td>
<td>S14</td>
<td></td>
<td>S14</td>
<td></td>
<td>!S14</td>
</tr>
<tr>
<td>S15</td>
<td>S15</td>
<td></td>
<td>S15</td>
<td>S15</td>
<td>S15</td>
</tr>
<tr>
<td>!S16</td>
<td>!S16</td>
<td></td>
<td></td>
<td>S16</td>
<td></td>
</tr>
<tr>
<td>!S17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!S18</td>
<td>!S18</td>
<td></td>
<td>!S19</td>
<td>!S19</td>
<td>S19</td>
</tr>
<tr>
<td>!S19</td>
<td>!S19</td>
<td>!S19</td>
<td></td>
<td>!S19</td>
<td>S19</td>
</tr>
<tr>
<td>!S20</td>
<td>!S20</td>
<td></td>
<td></td>
<td></td>
<td>S20</td>
</tr>
</tbody>
</table>
Table 4.3 Explanations of the notations used in table 4.4.

<table>
<thead>
<tr>
<th>Notation used in table 4.4</th>
<th>Meaning of the notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>Developer needs to do a lot of text processing (textual data, reports, email, news articles, log files, or just about any kind of text).</td>
</tr>
<tr>
<td>B2</td>
<td>Developer wants a language for which there are many tools and scripts available for free.</td>
</tr>
<tr>
<td>B3</td>
<td>Developer needs a database backend to a website (supports any kind of databases).</td>
</tr>
<tr>
<td>B4</td>
<td>Developer needs to be able to do Internet programming (Mail and News clients, interfaces to IRC and ICQ, right down to lower level Socket programming)</td>
</tr>
<tr>
<td>B5</td>
<td>Developer wants form data processed on the client side before submission to a server.</td>
</tr>
<tr>
<td>B6</td>
<td>Developer wants to add interactivity to the site quickly.</td>
</tr>
<tr>
<td>B7</td>
<td>Developer doesn’t have access to a server to use a server side programming tool.</td>
</tr>
<tr>
<td>B8</td>
<td>Developer doesn’t want to invest a lot of time in learning a scripting language.</td>
</tr>
<tr>
<td>B9</td>
<td>Developer is just beginning to do web programming.</td>
</tr>
<tr>
<td>B10</td>
<td>Developer has knowledge of VB or VBA.</td>
</tr>
<tr>
<td>B11</td>
<td>The servers is IIS (or third party equivalent).</td>
</tr>
<tr>
<td>B12</td>
<td>Developer is thinking about using ASP in future.</td>
</tr>
<tr>
<td>B13</td>
<td>When developing a client side application the developer assumes that the clients use IE as a browser. (It is not supported by Netscape).</td>
</tr>
<tr>
<td>B14</td>
<td>Developer wants to lower the server load (it’s not very powerful) by performing computations (form processing) on the client.</td>
</tr>
<tr>
<td>B15</td>
<td>Developer is going to develop with COM+ objects.</td>
</tr>
<tr>
<td>B16</td>
<td>Developer is considering DHTML</td>
</tr>
</tbody>
</table>
Table 4.4 Existing Technologies Comparison – Group2.

<table>
<thead>
<tr>
<th>Perl</th>
<th>JavaScript</th>
<th>VBScript</th>
<th>JScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td></td>
<td>B1</td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>B2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B5</td>
<td>B5</td>
<td>B5</td>
<td>B5</td>
</tr>
<tr>
<td>B6</td>
<td>B6</td>
<td>B6</td>
<td></td>
</tr>
<tr>
<td>!B7</td>
<td>B7</td>
<td></td>
<td>B7</td>
</tr>
<tr>
<td>B8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!B9</td>
<td>!B10</td>
<td>B10</td>
<td>!B10</td>
</tr>
<tr>
<td>!B11</td>
<td>!B11</td>
<td>B11</td>
<td>B11</td>
</tr>
<tr>
<td>!B12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!B13</td>
<td>B13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!B14</td>
<td>B14</td>
<td>B14</td>
<td>B14</td>
</tr>
<tr>
<td>!B15</td>
<td>!B15</td>
<td>B15</td>
<td>!B15</td>
</tr>
<tr>
<td>B16</td>
<td></td>
<td></td>
<td>B16</td>
</tr>
</tbody>
</table>
Table 4.5 Explanation of the notations used in table 4.6.

<table>
<thead>
<tr>
<th>Notation used in table 4.6</th>
<th>Meaning of the notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Developer wants to add some attractiveness to the site that won’t require communication between client and server.</td>
</tr>
<tr>
<td>C2</td>
<td>Fast communication channel is used.</td>
</tr>
<tr>
<td>C3</td>
<td>Developer wants the users to be able to rearrange objects on the page (ex. Interior design planning, business card editing, simple games, etc.)</td>
</tr>
<tr>
<td>C4</td>
<td>Developer has experience with JavaScript or is willing to learn it.</td>
</tr>
<tr>
<td>C5</td>
<td>Developer wants to add long-form animations, technical illustrations, small and compact navigation menus to your site.</td>
</tr>
<tr>
<td>C6</td>
<td>Developer can afford to spend more time to learn a sophisticated tool that is very powerful.</td>
</tr>
<tr>
<td>C7</td>
<td>Animation loading time is important for the developer (Flash is much faster than Java applets).</td>
</tr>
<tr>
<td>C8</td>
<td>Developer assumes most of the clients use IE as their browser.</td>
</tr>
<tr>
<td>C9</td>
<td>You need to enhance your web page with functionality like advance formatting, animation, timers, background downloading, etc.</td>
</tr>
<tr>
<td>C10</td>
<td>Developer would like to use the components not only on Web but also in other applications written in some MS programming or database language.</td>
</tr>
</tbody>
</table>
Table 4.6 Existing Technologies Comparison – Group3.

<table>
<thead>
<tr>
<th>ActiveX</th>
<th>Flash</th>
<th>Java (applets)</th>
<th>DHTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>C1</td>
<td>C1</td>
<td>C1</td>
</tr>
<tr>
<td>!C2</td>
<td>C2</td>
<td>!C2</td>
<td>!C2</td>
</tr>
<tr>
<td></td>
<td>C3</td>
<td>!C3</td>
<td>!C3</td>
</tr>
<tr>
<td></td>
<td>C4</td>
<td>!C4</td>
<td>!C4</td>
</tr>
<tr>
<td></td>
<td>C5</td>
<td>!C5</td>
<td>!C5</td>
</tr>
<tr>
<td></td>
<td>C6</td>
<td>!C6</td>
<td>!C6</td>
</tr>
<tr>
<td></td>
<td>C7</td>
<td>!C7</td>
<td>!C7</td>
</tr>
<tr>
<td>C8</td>
<td></td>
<td>!C8</td>
<td>!C8</td>
</tr>
<tr>
<td>C9</td>
<td></td>
<td>C9</td>
<td>C9</td>
</tr>
<tr>
<td>C10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5

SAMPLE TEST

To demonstrate the actual results that TASWAD produces a sample run was organized. An imaginary situation was offered to the system to find out if it works correctly. The situation is as follows. A middle size dotcom business that mostly uses Microsoft products is not satisfied with the level of its online customer service and the way the corporation Web site is designed. The company in effort to attract more users decides to make its Web page dynamic and provide the visitors with the most up-to-date information. The company also wishes to modify the appearance of the home page periodically. The manager of the company decides to hire a new person who is not necessarily a computer expert but is a good Web page designer (to maintain the appearance of the corporate image at a high level). In addition the designer should have a moderate background in script programming.

To test the system the scenario described above was presented to TASWAD and the recommendation was recorded.

Listed below is the list of 17 questions in the sequence they were presented to the user and the answers to those questions. The answers are also followed by short explanations to justify them.
First, a question about the company size was asked – figure 5.1. Answer to that question is *medium* (see description of the company at the beginning of the chapter 5).

![Figure 5.1 – Sample run question 1, size of the company.](image)

Then, the system tried to find out if the user prefers Microsoft environment - figure 5.2. The Answer that was given is *yes* (as it was said, the company uses Microsoft products extensively and we need a developer for that platform).

![Figure 5.2 – Sample run question 2, Microsoft orientation.](image)
Next it was asked if the Internet Information Server was installed - figure 5.3. Answer to that question is yes (it is a part of Windows NT that our Web server runs on).

![Figure 5.3](image)

Figure 5.3 – Sample run question 3, availability of IIS.

Language, that the Web designer for the company is familiar with are Visual Basic and Visual Basic for Applications - figure 5.4.

![Figure 5.4](image)

Figure 5.4 – Sample run question 4, knowledge of languages.
Here TASWAD tries to investigate the user’s intentions about the site’s appearance - figure 5.5. Answer is *no* because the web site needs to be attractive to a reasonable extent.

![Image](image1)

Figure 5.5 – Sample run question 5, need for attractiveness.

In this question it is determined what software (browser) is installed on the client side - figure 5.6. An assumption is made that it is Internet Explorer.

![Image](image2)

Figure 5.6 – Sample run question 6, kind of browser.
The company is not intending to do extensive programming on the Web site - figure 5.7. Hence, there's no need for such sophisticated language as Java.

Figure 5.7 – Sample run question 7, intention to learn a sophisticated language.

Usually, there is a need for text processing if the system gathers information from the users - figure 5.8. According to the scenario, the company is rather output than input oriented, which eliminates the necessity of such kind of functionality.

Figure 5.8 – Sample run question, need for a tool that handles strings, numbers, etc.
Cost of the software is not a very important for the company to make a decision that is why this question was answered *no* - figure 5.9.

Figure 5.9 – Sample run question 9, free software.

Since the company already has IIS (Internet Information Server) running, there is no need to install yet another application server - figure 5.10.

Figure 5.10 – Sample run question 10, need for another application server.
Supposedly, the new Web designer has scripting background - figure 5.11.

![TASWAD](image1)

Figure 5.11 – Sample run question 11, experience with JavaScript.

It is not a part of the company’s plan to develop middleware - figure 5.12.

![TASWAD](image2)

Figure 5.12 – Sample run question 12, intention to develop middleware.
Advanced formatting, animation, timers, background downloading are not a part of the company’s Web site improvement plan - figure 5.13.

Figure 5.13 – Sample run question 13, advanced formatting.

Since the company’s Web site needs to be redesigned not reprogrammed, answer to this question would be yes- figure 5.14.

Figure 5.14 – Sample run question 14, programming-centric approach vs. page-centric approach.
At this point the system proceeds with its recommendation being Active Server Pages, which is rather logical for the requirements set for the company's site redesign - figure 5.15. ASP is a Microsoft product; it runs under IIS and it suits Web sites with far more HTML than programming code.

![Exsys Runtime](image)

Figure 5.15 – Sample run result.

The outcome of the test proved that the system is capable of accomplishing its goal.
CONCLUSIONS AND FURTHER RESEARCH

Presented in this paper is a knowledge-based system that allows Web developers (no matter how experienced they are) to make an easy decision about what technology/programming language to use for their next project. By taking a user through a series of questions the system’s inference engine determines the most appropriate tool (or a number of tools) for a particular task. The system proved to be user friendly, no special skills or training are required for its use. By recommending to a Web programmer a better instrument for his/her job the system saves significant amount of time and money that the developer would otherwise spend in an effort to find answers to his/her questions.

Future plans for the application include making it available to the public via Internet and including new programming languages and technologies as they come up.

The Web has become the universal communication tool for reaching customers as well as prospects. Deploying software on the Web, such as knowledge-based systems, makes it readily available to users worldwide. Unfortunately, the Web was not created with applications such as knowledge-based systems in mind. Until recently, the manner
in which the Web and Web browsers interfaced made it fairly difficult to perform the actions required by knowledge-based systems.

Availability via Internet can be implemented by using Java applets. As for now not every existing knowledge-based system development environment (such as Jess™, Acquire SDK™) offers an efficient way of making an applet of a small size. Having to downloading several megabytes of binary code from a web site restricts the area of the system availability to high-speed network regions. However, with careful programming this size limitation can be overcome. There are high-powered inference engines built as Java applets of only 100K, which download rapidly even with low bandwidth. These fast running systems, a carefully selected combination of Java classes and smart designing of knowledge-based system functionality, provide only what is required, without excessiveness.

One of the biggest advantages of the Java applet approach is that it is very scalable – the server only sends the applet file to the client. Downloading files is something servers do well. As the number of users increases, all of the processing is done on the client machines. Even highly interactive systems put no additional load on the server. The applet communicates with the server for data, or data is passed in on startup, but interaction with the user is only on the client machine. Applets can also communicate with other applets to obtain data.

Java knowledge-based system applets can be anything from a full window with multiple graphics, to a small area that simply asks a few questions, keeping the look and
the feel of the rest of the site. Since Java gives full access to the entire Web, knowledge-based system recommendations can be linked and integrated into overall Web sites[15].

Java applet based online knowledge-based systems add interactive elements to a page design that are easily integrated and updated. Therefore, future plans for deploying TASWAD as an applet are advantageous.

Another direction of the future system development is real world testing. Giving TASWAD out to Web designers and collecting their feedback can provide significant contribution towards expansion of the system’s functionality.
References

1. Peter van der Linden, *Just Java 2*, Sun Microsystems Inc, 1999


16. What’s new in Jscript 3,  
October 1997


26. Microsoft COM Technologies,  

27. Webopedia definition and links,  
APPENDIX

This appendix contains the complete listing of the rules that are a part of the system. The order in which the rules are listed is not crucial to the system, since inference engine decides what rule to select and how many times to do it. Internal syntax of Exsys Developer is used.

Rules of the System

RULE NUMBER: 1  (JSP)
IF:
[S1] = 1
and Are you more concerned about how you web site is going to look like
rather than about it's functionality? yes
and Are you familiar or are you planning to work with reusable software
components? yes
or: Are you familiar or are you planning to work with reusable software
components? not sure
and Do you have a NAS (Netscape Application Server) installed? yes
THEN:
JSP - Confidence=8/10

NOTE:
JSP uses page centric approach (more HTML code, than Java code), it eases communication with components and if needs Netscape application server installed to run.

RULE NUMBER: 2  (JAVA SERVLETS)
IF:
[S2] = 0
and [S5] = 1
and [B7] = 1
and Do you think you will need to support multiple display formats? (like
XML and HTML) yes
or: Do you think you will need to support multiple display formats? (like XML and HTML) maybe

THEN: JavaServlets - Confidence=8/10

NOTE: Java Servlets could be recommended to those having access to a server \([b_7]\), going to change application logic often \([s_5]\), needing multiple display format support.

RULE NUMBER: 3 (SSJS)

IF: \([s_7] = 1\) and \([s_8] = 0\) and \([M] = 0\) and \([B_7] = 1\)

THEN: SSJS - Confidence=8/10

NOTE: Server side JavaScript doesn't take much time to learn \([s_7]\), it is a good solution for developers who don't prefer Microsoft products \([m]\) and don't want to learn a sophisticated language like Java \([s_8]\). As the name implies, access to a server is needed.

RULE NUMBER: 4 (SSI)

IF: \([B_7] = 1\) and \([B_9] = 0\) and Is code reusability an issue you want to address? yes OR I don't know and Would you like to be able to modify numerous web pages by changing a single file? yes

THEN: SSI - Confidence=9/10

NOTE: SSI are implemented by factoring out some code that is used in several web pages. It requires access to a server \([b_7]\) and it allows to modify a number of html files containing similar information by making changes to a single file. A developer shouldn't be a beginner \([b_9]\), because knowledge of a CGI scripting language like Perl is required.
RULE NUMBER: 5  (JAVA APPLETS)
IF:
   Do you want to add some attractiveness to your site? yes
   and [C2] = 1
   and [M] = 0
   and [S8] = 1
THEN:
   Java (applets) - Confidence=8/10

NOTE:
Java applets are pretty universal, they don't require special browsers or platform or operating system. They can be large though, hence needing fast communication channels [C2]. The look of web site could be improved significantly by using Java applets. Those fond of Microsoft products [M] would rather prefer ActiveX controls.

RULE NUMBER: 6  (EJB)
IF:
   [B9] = 0
   and Are you going to develop middleware? yes
   and Do you need an application server independent solution? yes
or:  Do you need an application server independent solution? not sure
   and Please, characterize the size of your company medium
   and Is code reusability an issue you want to address? yes
   and [S1] = 0
   and [M] = 0
   and [B7] = 1
   and [S8] = 1
   and Do you need a cross platform (portable) solution? yes OR It doesn't matter to me
THEN:
   EJB - Confidence=8/10

NOTE:
EJB requires access to a server [B7], it is perfect for creating platform independent reusable components at medium/large scale web sites [S1]. Unless you prefer Microsoft products [M] and portability is not an issue for you, EJB is a good solution. EJB takes experience with programming [B9], better in Java [S8]. You definitely need an access to a server [B7].

RULE NUMBER: 7  (COM+)
IF:
   [B9] = 0
   and Are you going to develop middleware? yes
   and Are you developing middleware application for Windows (95 or NT) platform? yes
and Is code reusability an issue you want to address? yes
or: Is code reusability an issue you want to address? I don't know
and \([M] = 1\)

THEN:
- COM - Confidence=8/10
- ActiveX - Confidence=6/10

NOTE:
COM and ActiveX are solutions for Microsoft oriented people \([M]\) (not beginners \([B9]\)) wanting to develop reusable software components.

RULE NUMBER: 8  (ASP)
IF:
- \([S1] = 1\)
- \([M] = 0\)
- \([B7] = 1\)
- \([B11] = 1\)
and Which of the following languages do you know? VB
or: Which of the following languages do you know? VBA
or: Which of the following languages do you know? VBScript
or: Which of the following languages do you know? JScript
or: Which of the following languages do you know? Python
or: Which of the following languages do you know? Perl

THEN:
- ASP - Confidence=9/10

NOTE:
ASP definitely requires access to a server \([B7]\), knowledge of a scripting language and Internet Information Server \([B11]\).

RULE NUMBER: 9  (PHP)
IF:
- Do you favor free software? yes
or: Do you favor free software? I don't really care
and Do you need access to databases? yes
or: Do you need access to databases? not sure
and Is speed of your applications a concern for you? yes
and \([S14] = 1\)
and \([B7] = 1\)
and \([S16] = 1\)

THEN:
- PHP - Confidence=7/10

NOTE:
PHP is a server side \([B7]\) programming language, that is faster than JSP \([S16]\), free and also offers portable solutions \([S14]\).
RULE NUMBER: 10  (PERL)
IF:

\[ B_1 = 1 \]
and Is availability of free tools and scripts important for you? yes
or: Is availability of free tools and scripts important for you? not sure
and \[ S_{15} = 1 \]
and \[ B_7 = 1 \]
and Are you going to program such things as mail and news clients, interfaces to IRC and ICQ, or do socket programming? yes
or: Are you going to program such things as mail and news clients, interfaces to IRC and ICQ, or do socket programming? maybe

THEN:

Perl - Confidence=9/10

NOTE:
Perl is a server side \[ B_7 \] programming language that is fabulous at text processing \[ B_1 \]. It also offers low level programming functionality and database connectivity \[ S_{15} \].

RULE NUMBER: 11  (JAVASCRIPT)
IF:

Is your application going to involve form processing? yes
and \[ S_8 = 0 \]
and Do you want to add some attractiveness to your site? yes
and \[ B_9 = 1 \]
and You don't have access to server to use server-side programming language
or you want to lower your server load by moving some computations to the client. yes
and What browser program will your clients mostly use? NN OR other
and \[ B_{14} = 1 \]

THEN:

JavaScript - Confidence=8/10

NOTE:
JavaScript is a scripting language many beginning web developers \[ B_9 \] start with. If allows to perform some data processing on the client side before sending it to the server \[ B_{14} \]. Works best with NN browsers.

RULE NUMBER: 12  (ACTIVEX)
IF:

\[ B_9 = 0 \]
and \[ C_9 = 1 \]
and What browser program will your clients mostly use? IE
and \[ M = 1 \]
and Would you like to reuse the code that you write in some other
applications written in some MS programming or database language? yes

THEN:

ActiveX - Confidence=7/10

NOTE:
ActiveX allows to create reusable software components that can enhance functionality of a web site. It requires some programming experience [B9], Microsoft-orientedness [M]. It could enhance your site with features like advanced formatting, background downloading, etc. [C9].

RULE NUMBER: 13 (VBSCRIPT)
IF:

Which of the following languages do you know? VB OR VBA
and [B11] = 1
and [C8] = 1
and Do you need a tool that works good with strings, numbers, dates, or text-based files? yes OR I don't know yet
and You don't have access to server to use server-side programming language
or you want to lower your server load by moving some computations to the client. yes
and Are you going to develop with COM objects? yes OR not familiar with it

THEN:

VBScript - Confidence=8/10

NOTE:
VBScript only runs in IE browsers [C8], it is good for supporting COM components, processing of strings and numbers. It requires installation of IIS however [B11].

RULE NUMBER: 14 (JSCRIPT)
IF:

What browser program will your clients mostly use? IE OR NN
and [S8] = 0
and [M] = 1
and [B7] = 0
and [B14] = 1

THEN:

JScript - Confidence=8/10

NOTE:
JScript is a light weight scripting language for client side that offers pretty much the same functionality as JavaScript. It is easier to get acquainted with it if you are keen on Microsoft products [M]. No
special access to a server is required \([B7]\). However, if you have one, you could use JScript to move some computational load from the server to a client \([B14]\).

RULE NUMBER: 15  (HTML)

IF:

\[
[B7] = 0 \\
[B9] = 1 \\
[S8] = 0 \\
[B6] = 0 \\
Please, characterize the size of your company I don't work for a business
\]

THEN:

HTML - Confidence=9/10

NOTE:
HTML is the most simplistic recommendation that could be possibly given by the system. People, who are just beginning to do web development \([B9]\) and don't have access to server \([B7]\) side tools, don't need much interactivity \([B6]\) and work on their own are ones that belong to this category.

RULE NUMBER: 16  (DHTML)

IF:

Do you have experience with scripting languages? yes OR I will learn one \\
Do you want your users to be able to rearrange objects on the page (ex. Interior design planning, business cards editing, simple games)? yes \\
Do you want to add some interactivity to your site? yes \\
[C3] = 1

THEN:

DHTML - Confidence=8/10

NOTE:
DHTML allows you to rearrange objects on a web page after is has been downloaded from the server \([C3]\). Previous knowledge of a scripting language is helpful.

RULE NUMBER: 17  (FLASH)

IF:

\[
[B9] = 0 \\
Do you want to add long-form illustrative animations, technical illustrations, small and compact navigation menus to your site?
\]
yes
and Is speed of your applications a concern for you? yes
and [C6] = 1

THEN:
Flash - Confidence=8/10

NOTE:
Flash provides you with powerful ability to create light animations. The tool is difficult to learn though [B9]. It allows you to create illustrative animations, technical illustrations, navigation menus, etc.

RULE NUMBER: 18  (COLDFUSION)
IF:
[B9] = 0
and [S1] = 0
and [S19] = 1
and [S18] = 1
and [S20] = 1
and In the application that you are going to build will you need connection
to such services RDBMS, messaging servers, file repositories, directory servers, or distributed object middleware? yes
and [B9] = 0

THEN:
ColdFusion - Confidence=8/10

NOTE:
ColdFusion is a powerful server side programming tool (not for beginners [B9]) for big companies [S1], [S18], that provides the best among the others (PHP, JSP, ASP) connectivity to the databases. It needs special application server [S19]. ColdFusion runs on Windows NT or SUN Solaris [S20].

RULE NUMBER: 19  (MS ORIENTED)
IF:
Are you a Microsoft products oriented developer? yes OR not sure

THEN:
[M] IS GIVEN THE VALUE 1

ELSE:
[M] IS GIVEN THE VALUE 0

NOTE:
This rule defines if the user is MS oriented or not.
RULE NUMBER: 20  (COMPANY SIZE)
IF:  
Please, characterize the size of your company medium OR small
THEN:  
[S1] IS GIVEN THE VALUE 1
ELSE:  
[S1] IS GIVEN THE VALUE 0
NOTE:
Knowing the size of the company helps choosing appropriate language.

RULE NUMBER: 21  (FREQUENT REDESIGN)
IF:  
Do you think you will redesign you web site frequently? yes OR not sure
THEN:  
[S2] IS GIVEN THE VALUE 1
ELSE:  
[S2] IS GIVEN THE VALUE 0
NOTE:
This rule lets define if the site is going to be redesigned frequently.

RULE NUMBER: 22  (FREQ. CHANGE OF APPLICATION LOGIC)
IF:  
Are you going to change you application logic often times? yes
THEN:  
[S5] IS GIVEN THE VALUE 1
ELSE:  
[S5] IS GIVEN THE VALUE 0
NOTE:
Frequent change of application logic is handled better with JSP.

RULE NUMBER: 23  (SHORT ON TIME)
IF:  
Are you short on time? yes
THEN:  
[S7] IS GIVEN THE VALUE 1
ELSE:

[S7] IS GIVEN THE VALUE 0

NOTE:
An easy to master tool like JavaScript is recommended for those being short on time.

RULE NUMBER: 24  (WANT TO LEARN JAVA.)
IF:
You don't want to learn a sophisticated language like Java. I don't OR I don't know what Java is
THEN:
[S8] IS GIVEN THE VALUE 0
ELSE:
[S8] IS GIVEN THE VALUE 1

NOTE:
If there's no desire to learn a sophisticated language an easy one can be recommended.

RULE NUMBER: 25  (NEED FOR PORTABILITY)
IF:
Do you need a cross platform (portable) solution? yes OR It doesn't matter to me
THEN:
[S14] IS GIVEN THE VALUE 1
ELSE:
[S14] IS GIVEN THE VALUE 0

NOTE:
Here a need for a platform independent solution is determined.

RULE NUMBER: 26  (NEED FOR DATABASES)
IF:
Do you need access to databases? yes OR not sure
THEN:
[S15] IS GIVEN THE VALUE 1
ELSE:
    [S15] IS GIVEN THE VALUE 0

NOTE:
Here a need for database access is determined.

RULE NUMBER: 27 (SPEED IS A CONCERN)
IF:
    Is speed of your applications a concern for you? yes OR don't know yet
THEN:
    [S16] IS GIVEN THE VALUE 1
ELSE:
    [S16] IS GIVEN THE VALUE 0

NOTE:
Here it is determined if the speed of an application is a concern for the developer. If it is PHP is a good choice otherwise JSP (which is somewhat slower) can be recommended.

RULE NUMBER: 28 (LARGE SCALE FROM THE SCRATCH)
IF:
    [S1] = 0
    and Are you building your web site from the scratch? yes
THEN:
    [S18] IS GIVEN THE VALUE 1
ELSE:
    [S18] IS GIVEN THE VALUE 0

NOTE:
An intention to build a large-scale web site from the scratch is determined here.

RULE NUMBER: 29 (ANOTHER APPLICATION SERVER)
IF:
    Can you install another application server and switch to it from your existing one? yes
THEN:
    [S19] IS GIVEN THE VALUE 1
ELSE:
    [S19] IS GIVEN THE VALUE 0

NOTE:
This rule is needed because CF requires a dedicated application server.

RULE NUMBER: 30  (SOLARIS OR NT.)
IF:
    Is your platform is either NT or Solaris? yes
THEN:
    [S20] IS GIVEN THE VALUE 1
ELSE:
    [S20] IS GIVEN THE VALUE 0

NOTE:
One of the conditions to run ColdFusion is to have the web server running either Solaris or Windows NT.

RULE NUMBER: 31  (TEXT PROCESSING)
IF:
    Are you going to do a lot of text processing (textual data, reports, email, news articles, log files, etc.)? yes OR maybe
THEN:
    [B1] IS GIVEN THE VALUE 1
ELSE:
    [B1] IS GIVEN THE VALUE 0

NOTE:
Text processing is best done by Perl.

RULE NUMBER: 32  (INTERACTIVITY)
IF:
    Do you want to add some interactivity to your site? yes OR I don't know
THEN:
    [B6] IS GIVEN THE VALUE 1
ELSE:
    [B6] IS GIVEN THE VALUE 0
NOTE:
Now it is needed to find out if the user wants to add some interactivity to the web site.

RULE NUMBER: 33  (ACCESS TO A SERVER)
IF:
   Do you have access to a server? yes
THEN:
   [B7] IS GIVEN THE VALUE 1
ELSE:
   [B7] IS GIVEN THE VALUE 0

NOTE:
Access to a server is important to know about when deciding between a server side or a client side language.

RULE NUMBER: 34  (BEGINNER)
IF:
   Are you a beginning web-programmer? yes
THEN:
   [B9] IS GIVEN THE VALUE 1
ELSE:
   [B9] IS GIVEN THE VALUE 0

NOTE:
It is important to know confidence of the user to recommend an appropriate language.

RULE NUMBER: 35  (IIS INSTALLED)
IF:
   [B7] = 1
     and Do you have IIS installed on your server? yes OR I will install it
THEN:
   [B11] IS GIVEN THE VALUE 1
ELSE:
   [B11] IS GIVEN THE VALUE 0

NOTE:
A kind of application server is important for the server side programming languages like ASP, JSP, etc.
RULE NUMBER: 36  (MOVE LOAD TO CLIENT)  
IF:

\[ [B7] = 1 \]
and You don't have access to server to use server-side programming language
or you want to lower your server load by moving some computations to the client. yes OR I don't know

THEN:

[B14] IS GIVEN THE VALUE 1

ELSE:

[B14] IS GIVEN THE VALUE 0

NOTE:
Some client side languages can perform data processing before sending it to the server. This approach can help reduce the load on a server.

RULE NUMBER: 37  (FAST NETWORK)  
IF:

Are your clients communication channel (networks) fast? yes

THEN:

[C2] IS GIVEN THE VALUE 1

ELSE:

[C2] IS GIVEN THE VALUE 0

NOTE:
Fast communication channels are needed for Java applets.

RULE NUMBER: 38  (REARRANGE OBJECTS)  
IF:

Do you want your users to be able to rearrange objects on the page (ex. Interior design planning, business cards editing, simple games)? yes OR I don't know

THEN:

[C3] IS GIVEN THE VALUE 1

ELSE:

[C3] IS GIVEN THE VALUE 0

NOTE:
Ability to rearrange object on a web page without communicating with the server is achieved with DHTML.
RULE NUMBER: 39  (TIME TO LEARN A SOPHISTICATED TOOL)
IF:
    Can you afford to spend more time learning a sophisticated tool that is very powerful? sure
THEN:
    [C6] IS GIVEN THE VALUE 1
ELSE:
    [C6] IS GIVEN THE VALUE 0
NOTE:
Macromedia Flash is a powerful tool but it takes time to master.

RULE NUMBER: 40  (IE)
IF:
    What browser program will your clients mostly use? IE
THEN:
    [C8] IS GIVEN THE VALUE 1
ELSE:
    [C8] IS GIVEN THE VALUE 0
NOTE:
Some languages are only supported by certain browsers.

RULE NUMBER: 41  (ADVANCED FORMATTING)
IF:
    Would you like to enhance your web site with functionality like advanced formatting, animation, timers, background downloading, etc.? yes OR I don't know yet
THEN:
    [C9] IS GIVEN THE VALUE 1
ELSE:
    [C9] IS GIVEN THE VALUE 0
NOTE:
ActiveX components provide this functionality.