Web Based Decision Support System using Artificial Intelligence for Treating Industrial Waste Water

GRADUATE PROJECT TECHNICAL REPORT

Submitted to the Faculty of the Department of Computing Sciences Texas A&M University-Corpus Christi Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

by

Rajitha Peddulla
Fall 2007

Committee Members

Dr. Longzhuang Li
Committee Chairperson

Dr. David Thomas
Committee Member

Dr. Hongyu Guo
Committee Member
Web Based Decision Support System using Artificial Intelligence for Treating Industrial Waste Water

GRADUATE PROJECT TECHNICAL REPORT

Submitted to the Faculty of
the Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Rajitha Peddulla
Fall 2007

Committee Members

Dr. Longzhuang Li
Committee Chairperson

Dr. David Thomas
Committee Member

Dr. Hongyu Guo
Committee Member
ABSTRACT

Proper industrial waste water treatment is essential to meet the stringent environmental regulations in a cost efficient manner. Unfortunately, many of these methods are poorly understood or not utilized by industrialists, often because of the limited number of chemical engineers/environmentalist available. Developing a Web based artificial intelligence system for treating industrial waste water (WAITIW) can effectively replace the limited trained man power. Several inherent disadvantages of a human expert like round the clock availability and easy accessibility can be addressed with this new Web based system. The proposed WAITIW consists of fixed set of rules, representing different types of waste water treatment methods, while the facts, representing the different kinds of industrial waste water conditions (based on composition or content). These rules and facts are supplied to the JESS rule engine that performs pattern matching. When a set of facts matches a pattern the rule corresponding to the facts is fired or executed.
# TABLE OF CONTENTS

Abstract .......................................................... ii

Table of Contents .................................................................. iii

List of Figures ........................................................................ vi

1. Background and Rationale .................................................. 1
   1.1 Introduction .......................................................... 1
   1.2 Importance of Treating Industrial Waste water ............... 2
   1.3 Previous AI techniques ............................................. 3
   1.3 Proposed Web based Expert System Design ................. 3

2. Narrative ............................................................................ 5
   2.1 A simple expert syste ............................................... 5
      2.1.1 User Interface ................................................ 6
      2.1.2 Knowledge Base ............................................ 7
      2.1.3 Knowledge Acquisition: ................................ 7
      2.1.4 Inference Engine ............................................ 8
   2.2 Rule Based Expert System .......................................... 9
      2.2.1 Knowledge base ............................................. 12
      2.2.2 Database ..................................................... 12
      2.2.3 Inference Engine ............................................ 12
      2.2.4 Explanation facilities ..................................... 12
      2.2.5 User Interface ................................................. 12
   2.3 Comparison of Expert system with Conventional Program ........................................................................ 13
   2.4 Domain Background ................................................... 14

iii
2.4.1 Waste water composition .................................................. 14
2.4.2 Treatment of Industrial Waste water ................................. 14
2.4.3 Industries Served .......................................................... 16

2.5 WAITIW as a Rule Based Expert system .................................. 16
  2.5.1 Knowledge Engineering ............................................... 18
  2.5.2 Structuring Data ....................................................... 19
  2.5.3 Testing ................................................................. 19
  2.5.4 Interface Building .................................................... 20
  2.5.5 Writing the Rules .................................................... 20
  2.5.6 Iterative Development .............................................. 21

3. System Design ..................................................................... 22
  3.1 System Requirements ..................................................... 22
    3.1.1 Database Connectivity .............................................. 25
  3.2 WAITIW Architecture .................................................... 26
    3.2.1 Writing Rules ......................................................... 27
    3.2.2 Writing Facts ........................................................ 28
    3.2.3 Matching Rules and Facts ....................................... 30
  3.3 WAITIW Databases ....................................................... 30
    3.3.1 USER Table .......................................................... 29
    3.3.2 FEEDBACK Table ................................................... 30
  3.4 WAITIW Work Flow ....................................................... 32
  3.5 Security Issues of WAITIW Application ............................... 35

4. Evaluation and Results ....................................................... 36
4.1.1 Unit Testing ................................................................. 36
4.1.2 Stress Tests............................................................... 36
4.1.3 Integration Testing ..................................................... 38
4.2 Executing WAITIW Application ..................................... 39
5. Future Work ................................................................. 41
  5.1 Developing a Multi-Media Rich Internet Application ........ 41
  5.2 WAITIW as Web Services ............................................. 41
6. Conclusion .................................................................. 42
7. Bibliography and References ......................................... 43
LIST OF FIGURES

Figure 2.1: Expert System Structure ................................................................. 6
Figure 2.2: Knowledge Acquisition ................................................................. 8
Figure 2.3: Inference Engine .......................................................................... 8
Figure 2.4: Complete structure of a rule-based expert system ....................... 11
Figure 2.5: Development of a Rule Based System .......................................... 18
Figure 2.6: Modeling Conditional Control Flow Logic .................................... 21
Figure 3.1: User Interface of WAITIW ............................................................. 23
Figure 3.2: Architecture of the ADO. NET framework .................................... 26
Figure 3.3: Architecture of WAITIW ............................................................... 27
Figure 3.4: GUI of WAITIW .......................................................................... 29
Figure 3.5: Feedback Table ........................................................................... 31
Figure 3.6: User Interface for feedback form .................................................. 32
Figure 3.7: Login form of WAITIW ................................................................. 32
Figure 3.8: Welcome page of WAITIW ......................................................... 33
Figure 3.9: First step in WAITIW ................................................................. 34
Figure 3.10: Steps in WAITIW ........................................................ .......... 35
1. BACKGROUND AND RATIONALE

1.1 Introduction

Our survival as a species on planet Earth has become an open question because of human-produced environmental degradation we have stressed our environmental quality in many ways such as over population, urbanization and industrial growth. Hazardous waste water management remains the primary area of concern for many industries. Regulations, such as the Resource Conservation and Recovery Act (RCRA), the Toxic Substance Control Act (TSCA), and Superfund (CERCLA) as well as regulatory agencies, continue to keep corporate attention and the pressure on. [Nicholas 1996]

The need to make decisions on environmental issues is increasingly vital, yet few people have the tools or knowledge to understand or appreciate them in all their complexity. While decision making is a vital human activity, only recently have artificial intelligence and expert systems emerged as academic fields of inquiry [Zimmermann 1993]. Formal computer based decision support tools have been applied in a number of different problem areas such as medicine, agriculture, and automobile industries. While most decisions require a wide range of knowledge, the complexities of environmental problems are especially noteworthy for the extraordinary range of inputs required to produce an effective decision. For example, environmental problems often involve physical, chemical, biological and technological factors. Omission of any of these factors is likely to oversimplify the problems and render the decision process uncompleted and unrealistic. Taking a right environmental decision requires lots of knowledge and well
trained chemical engineers. The lack of trained man-power to handle the environmental regulatory process has crippled industrial development that otherwise ecologically sound and sustainable. Artificial Intelligence (AI) offers the possibility of revolutionizing the environmental regulatory process as they can emulate human cognitive skills such as problem solving, visual perception and language understanding. Expert system, a research discipline of artificial intelligence, can be distinguished from a more conventional applications program in that it stimulates human reasoning about a problem domain, rather than stimulating the domain itself [Jackson 1990]. This distinguishes expert systems from more familiar programs that involve mathematical modeling or computer animation. This is not to say that the program is a faithful psychological model of an expert, merely that the focus is upon emulating an expert's problem solving abilities, that is performing some portion of relevant tasks as well as, or better than, an expert.

1.2 Importance of Treating Industrial Waste Water:

Industrial waste water must be treated or decontaminated until an adequate level is reachable so that they could be poured to the surrounding hydric medium without causing problems of environmental deterioration. For that, there are waste water treatment plants (WWTPs) with techniques of physical-chemical and biological treatment. The process of waste water treatment is so complex that it is difficult to develop a reliable supervisory technology based only on a classic control approach from chemical engineering [Dale 2002]. Therefore the use of artificial intelligence systems is to be necessary in order to obtain better results in waste water management.
1.3 Previous AI techniques:

In the last decades several researchers have been interested in applying AI techniques to solve problems from the environmental domain. There are very few AI techniques available in today’s market in the field of chemical engineering for treating industrial waste water [Juan 1998]. These techniques are limited to one treatment process such as Reverse Osmosis (RO) or other Membrane technologies. The main disadvantages of existing expert systems are that they are on a single system [Sanchez 1997]. Hence there is a need to develop a web based decision supporting system to identify the most appropriate waste water treatment process, which includes physical, chemical and biological treatment.

1.4 Proposed Web Based Expert System Design:

A Web based expert system resolves many of the issues faced by today’s expert systems which are implemented on stand-alone systems. A Web based expert system provides access to the expert system from anywhere in the world and reduces the overhead of installing it on each stand alone system. This satisfies the real need for the designing of a web based expert system that will be available everywhere at all times. [RODA 2000]

Rule-based expert systems (one of the broadly applied paradigms of AI) are able to cope with some known difficulties that are faced by several WWTP-domain problems, even if they are not definitive solution to the treatment problem as a whole [Luigi 2000]. This project work aimed at making a more exhaustive rule based expert system which can identify most appropriate process for treating industrial waste water and to meet the
stringent environmental regulations in a cost efficient manner as well as to replace trained man power.
2. NARRATIVE

The current focus on Internet-based applications demands new architectures for intelligent systems as well as creating new possibilities for research and development. This project developed a rapid and cost efficient tool for treating industrial waste water to meet the stringent environmental regulations.

2.1 A simple expert system:

Expert systems are computer programs that attempt to solve problems in a way similar to how a human expert would solve them. An expert system may completely fulfill a function that normally requires human expertise, or it may play the role of an assistant to a human decision maker. In order to construct such knowledge-based systems, knowledge is elicited from domain experts or through expert’s publications in literatures. Every expert system consists of three parts: a user interface, a knowledge base, and an inference mechanism as shown in Figure 2.1. [Martin 1988]
2.1.1 User Interface:

User interface software receives information from user and outputs immediate or intermediate results. In some expert systems, input is obtained from additional sources such as databases and sensors. The expert system even accepts and learns new knowledge from the user. [Martin 1998]
2.1.2 Knowledge Base:

Knowledge base of expert systems contains both factual and heuristic knowledge about the domain of the expert system. Factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field.

Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. It is the knowledge that underlies the art of good guessing.

2.1.3 Knowledge Acquisition:

Knowledge acquisition is the transfer of potential problem solving expertise from some knowledge source to a program. The expert system must liaise with people in order to gain knowledge and the people must be specialized in the appropriate area of activity. The knowledge engineer acts as an intermediary between the specialist and the expert system as shown in the Figure 2.2. Typically the information that must be collected is vocabulary or jargon, general concepts and facts, problems that commonly arise, the solutions to the problems that occur and skills for solving particular problems. This process of picking the brain of an expert is a specialized form of data capture and makes use of interview techniques. The knowledge engineer is also responsible for the self consistency of the data loaded. Thus a number of specific tests have to be performed to ensure that the conclusions reached are reasonable. [Martin 1988]
2.1.4 Inference Engine:

The inference engine is the central program which manipulates the rules and facts in the knowledge base to reach conclusions. The structure of the inference engine depends strongly upon the type of the knowledge base which the expert system incorporates. The inference engine may approach the problem from either the top or the bottom, beginning with either facts or conclusions. If a user begins with several hypotheses and wants to determine which, if any, are correct, then the program should examine all the facts using a goal-directed approach. However, if the user begins with a series of facts which are known to be true and wants to determine what conclusions can be reached, the program should use a data-directed approach.
2.2 Rule Based Expert System:

Conventional problem-solving computer programs make use of well-structured algorithms, data structures, and crisp reasoning strategies to find solutions. For the difficult problems with which expert systems are concerned, it may be more useful to employ heuristics strategies that often lead to the correct solution, but that also sometimes fail [Ajith 2005].

Conventional rule-based expert systems, use human expert knowledge to solve real-world problems that normally would require human intelligence. Expert knowledge is often represented in the form of rules or as data within the computer. Depending upon the problem requirement, these rules and data can be recalled to solve problems. Rule-based expert systems have played an important role in modern intelligent systems and
their applications in strategic goal setting, planning, design, scheduling, fault monitoring, diagnosis and so on. With the technological advances made in the last decade, today’s users can choose from dozens of commercial software packages having friendly graphic user interfaces [Ignizio, 1991]. Conventional computer programs perform tasks using a decision-making logic containing very little knowledge other than the basic algorithm for solving that specific problem. The basic knowledge is often embedded as part of the programming code, so that as the knowledge changes, the program has to be rebuilt. Knowledge-based expert systems collect the small fragments of human know-how into a knowledge base, which is used to reason through a problem, using the knowledge that is appropriate. An important advantage here is that within the domain of the knowledge base, a different problem can be solved using the same program without reprogramming efforts. Moreover, expert systems could explain the reasoning process and handle levels of confidence and uncertainty, which conventional algorithms do not handle [Giarratano and Riley, 1989]. Some of the important advantages of expert systems are as follows

- Ability to capture and preserve irreplaceable human experience.
- Ability to develop a system more consistent than human experts.
- Minimize human expertise needed at a number of locations at the same time (especially in a hostile environment that is dangerous to human health).
- Solutions can be developed faster than human experts.

The components of an expert system are illustrated in figure 2.4.
Figure 2.4: Complete structure of a rule-based expert system [UTAS 2003].
2.2.1 Knowledge base:

The knowledge base contains the domain knowledge useful for problem solving. In a rule-based expert system, the knowledge is represented as a set of rules. Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure. When the condition part of a rule is satisfied, the rule is said to fire and the action part is executed. [UTAS 2003]

2.2.2 Database:

The database includes a set of facts used to match against conditional parts of rules stored in the knowledge base.

2.2.3 Inference Engine:

The inference engine carries out the reasoning whereby the expert system reaches a solution. It links the rule given in the knowledge base with the facts provided in the database.

2.2.4 Explanation facilities:

The explanation facilities enable the user to ask the expert system how a particular conclusion is reached and why a specific fact is needed. An expert system must be able to explain its reasoning and justify its advice, analysis or conclusion.

2.2.5 User Interface:

The user interface is the means of communication between a user seeking a solution to the problem and an expert system.
2.3 Comparison of Expert system with Conventional Program:

Comparisons of expert system with conventional systems and human experts are listed in Table 2.1. [UTAS 2003]

<table>
<thead>
<tr>
<th>Human Experts</th>
<th>Expert Systems</th>
<th>Conventional Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use knowledge in the form of rules of thumb or heuristic to solve problem in a narrow domain.</td>
<td>Process knowledge expressed in the form of rules and use symbolic reasoning to solve in a narrow domain.</td>
<td>Process data and use algorithms, a series of well-defined operations, to solve general numerical problems.</td>
</tr>
<tr>
<td>In a human brain knowledge exists in a compiled form.</td>
<td>Provide a clear separation of knowledge from its processing.</td>
<td>Do not separate knowledge from the control structure to process this knowledge.</td>
</tr>
<tr>
<td>Capable of explaining a line of reasoning and providing the details.</td>
<td>Trace the rules fired during a problem-solving session and explain how a particular conclusion was reached and why specific data was needed.</td>
<td>Do not explain how a particular result was obtained and why input data was needed.</td>
</tr>
<tr>
<td>Use inexact reasoning and can deal with incomplete, uncertain and fuzzy information.</td>
<td>Permit inexact reasoning and can deal with incomplete, uncertain and fuzzy data.</td>
<td>Work only on problems where data is complete and exact.</td>
</tr>
<tr>
<td>Can make mistake when information is incomplete or fuzzy.</td>
<td>Can make mistake when data is incomplete or fuzzy.</td>
<td>Provide no solution at all, or working one, when data is incomplete or fuzzy.</td>
</tr>
<tr>
<td>Enhance the quality of problem solving via years of learning and practical training. This process is slow, inefficient and expensive.</td>
<td>Enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. When new knowledge is acquired, changes are easy to accomplish.</td>
<td>Enhance the quality of problem solving by changing the program code, which affects both the knowledge and its processing, making changes difficult.</td>
</tr>
</tbody>
</table>

Table 2.1: Comparisons of expert system with conventional system and human expert [UTAS 2003].
2.4 Domain Background:

The generation and disposal of large quantities of waste water without adequate treatment results in significant environmental pollution. Degradation of waste in the environmental can lead to the release of greenhouse gases (GHGS) such as methane and carbon dioxide. The environmental damage cost to the society works out to be more than the financial costs to the industry:

2.4.1 Waste water composition:

Waste water contains solids, organic compounds, inorganic compounds, chemicals, pathogens (disease organisms), nutrients (nitrogen, phosphorus, etc.). Waste water must be cleaned up before it is returned to the environment to be recycled for future generations.

2.4.2 Treatment of Industrial Waste water:

Most treatment plants have primary treatment (physical removal of floatable and settleable solids) and secondary treatment (the biological removal of dissolved solids).

Primary treatment involves:

1. Screening: To remove large objects, such as stones or sticks that could plug lines or block tank inlets.

2. Grit chamber: Slows down the flow to allow grit to fall out

3. Sedimentation tank (settling tank or clarifier): Settleable solids settle out and are pumped away, while oils float to the top and are skimmed off.
Secondary treatment typically utilizes biological treatment processes, in which microorganisms convert non-settleable solids to settleable solids. Sedimentation typically follows, allowing the settleable solids to settle out. Three options include:

1. Activated Sludge: The most common option uses microorganisms in the treatment process to break down organic material with aeration and agitation, and then allows solids to settle out. Bacteria-containing activated sludge is continually re-circulated back to the aeration basin to increase the rate of organic decomposition.

2. Trickling Filters: These are beds of coarse media (often stones or plastic) 3-10 ft. deep. Waste water will be sprayed into the air (aeration) and then allowed to trickle through the media. Microorganisms attached to and growing on the media, break down organic material in the waste water. The waste water is collected and then undergoes sedimentation.

3. Lagoons: These are slow, cheap, and relatively inefficient, but can be used for various types of waste water. They rely on the interaction of sunlight, algae, microorganisms, and oxygen (sometimes aerated).

After primary and secondary treatment, waste water is treated with chemicals depending on their composition. An increasing number of waste water facilities also employ tertiary treatment, often using advanced treatment methods.

Tertiary treatment may include processes to remove nutrients such as nitrogen and phosphorus, and exploit carbon adsorption to remove chemicals. These processes can be physical, biological, or chemical. Settled solids (sludge) from primary treatment and secondary treatment settling tanks are given further treatment and undergo several options for disposal.
2.4.3 Applications of WAITIW in Industries:

WAITIW is designed to handle a wide range of industrial waste water. Following are the list of few industries that can handle by WAITIW.

- Textiles: The textile industries use a number of dyes, chemicals and other materials to impart desired quality to the fabrics. These units generates a substantial quantity of effluents, the quality of which in most of the cases is unsuitable for further use and can cause environmental problems, if disposed of without proper treatment.

- Pharmaceutical: Concentration of antibiotics, pharmaceutical intermediates, and peptides in effluent contains organic solvents such as methanol, trichloroethylene, methylenechloride, phenol, toluene and other organic solvents.

2.5 WAITIW as a Rule Based Expert system:

Rule-based programming is one of the most commonly used techniques for developing expert systems. In this programming paradigm, rules are used to represent heuristics, or rules of thumb, which specify a set of actions to be performed for a given situation. A rule is composed of an if portion and a then portion. The if portion of a rule is a series of patterns which specify the facts (or data) which cause the rule to be applicable. The process of matching facts to patterns is called pattern matching. The expert system tool provides a mechanism, called the inference engine, which automatically matches facts against patterns and determines which rules are applicable. The if portion of a rule can actually be thought of as the whenever portion of a rule since pattern matching always occurs whenever changes are made to facts. The then portion of a rule is the set of actions to be executed when the rule is applicable. The actions of applicable rules are
executed when the inference engine is instructed to begin execution. The inference engine selects a rule and then the actions of the selected rule are executed (which may affect the list of applicable rules by adding or removing facts). The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain.

The development of WAITIW was based on the basic principles for the design of a typical rule based system expert system. [Friedman 2003] The development of WAITIW comprises of the following steps as shown in figure 2.5

- Knowledge Engineering
- Structuring data
- Testing
- Interface building
- Writing the rules
- Iterative development
2.5.1 Knowledge Engineering:

Knowledge engineering is the process of codifying a human expert's expertise and representing that expertise in the knowledge base. The expert from which the data is to be
acquired for the system can be human experts, books, journals and articles, etc relevant to the domain of the expert system. In Knowledge engineering of WAITIW different water quality conditions like color of the water, hardness of water, levels of chemical oxygen demand (COD), Biological oxygen demand (BOD), amount of total suspend solids (TSS) were collected from various peer review journals and books[Nicholas 1996].

2.5.2 Structuring Data:

Structuring data is the second phase of the rule based WAITIW expert system. In this phase the data collected in the knowledge engineering phase of WAITIW was converted into rules and facts that were used for the development of rule based expert system. This phase includes the structuring of the data which makes the rules and facts to be implemented easily and directly. [Friedman 2003]

2.5.3 Testing:

Software testing is a critical component of the software development cycle and software testing procedures are critical to the success of the testing phase. Software remains in a perpetual state of change which is why software testing, whether manual or automated, is so vital to a software product’s success. Testing was done on the structured rules of the WAITIW rule based expert system by creating test cases for unit testing and regressive testing. Java classes that were developed for rich internet application (RIA) were tested using JUnit testing.

JUnit framework provides an easy way to explicitly test specific areas of a Java program. It is extensible and can be employed to test a hierarchy of program code either singularly or as multiple units. Using a testing framework is valuable because the
framework forces programmers to explicitly declare the expected results of specific program execution routes. When debugging it is possible to write a test case which expresses the result one is trying to achieve and then debug until the test is passed. By having a set of test cases that test all the core components of the project it is possible to modify specific areas of the project and immediately see the effect the modifications have on the other areas. JUnit promotes the idea of first testing then coding, in that it is possible to setup test data for a unit which defines what the expected output is and then code until the tests pass. [JUnit 2007]

2.5.4 Interface Building:

Interface development is one of the major processes in the building of a rule based expert system. The interface building of the WAITIW was basically about the system interacting with the environment on which it is being built. It describes about how to select the process for treating waste water by compares it with the rules and facts of the system. [Friedman 2003]

2.5.5 Writing the Rules:

The rules of WAITIW were developed based on our own rules that we obtain from studying various literatures from books and peer reviewed journals and past professional experience. In the initial stage, the rules were written in plain English using modeling control flow logic (Figure 2.4) which will make it much easier to convert into rules that can be scripted into the expert system language used by WAITIW. The rules of WAITIW were developed as modules so as to prevent unstructured rules. The WAITIW
rules were divided into separate module like physical treatment module, chemical treatment module and suggestion module.

Figure 2.6: Modeling Conditional Control Flow Logic

2.5.6 Iterative Development:

The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what has been learned during the development of earlier, incremental, deliverable versions of the system. A process of continuous feedback from user and experts was implemented using dynamic web page feed back with the help of Active Server Pages (ASP) and code behind classes. This process not only helps the KE to denote the domain knowledge correctly, but also helps the expert to elucidate the reasoning processes. All domain knowledge was analyzed and represented in developing new rules or refining the existing rules for better rules in the rule base.
3. SYSTEM DESIGN

3.1 System Requirements:

WAITIW was developed using Java Expert Shell System (JESS). Jess is a rule engine and scripting language developed at Sandia National Laboratories in Livermore, California in the late 1990s. JESS uses an enhanced version of the Rete algorithm to process rules. Rete is a very efficient mechanism for solving the difficult many-to-many matching problem. JESS has many features including backwards chaining, working memory queries, and the ability to manipulate and directly reason about Java objects. Since JESS was developed from Java 2 Standard Edition (J2SDK) it has several advantages compared to other expert systems. It has access to all the powerful Java APIs for graphics, database access, Java Beans, Networking and Remote Method Invocation (RMI) etc. The pluggable nature of JESS with different platforms enables us to developing industrial robust expert systems using Java 2 Enterprise Edition (J2EE) and in .NET Framework 2.0. This project was developed as Rich Internet Application (RIA) using AJAX (Asynchronous Java scripting and XML) and Microsoft .NET technology. Rich Internet applications (RIA) are Web applications that have the features and functionality of traditional desktop applications. RIAs typically transfer the data processing necessary for the user interface to the Web client but keep the bulk of the data (i.e., maintaining the state of the program, the data etc) back on the application server. A user interface of WAITIW is shown in Figure 3.1.
Figure 3.1: User Interface of WAITIW

AJAX primary contribution to web pages is user-experience improvement. Web pages usually require several applications to function. This can make it seem like a cumbersome operation where users have to wait for the separate applications to refresh before interacting with the complete page. Decreasing user delay, which is a direct result of AJAX techniques, could make the Internet even more popular and pervasive than it already is. Another advantage of AJAX is a decrease in bandwidth use. Bandwidth in
web hosting refers to the amount of data that can be communicated between user and server/website. In AJAX, bandwidth is used only to accomplish specific demands without requiring that the page to be re-loaded which require bandwidth, every time a request is made. Contents are loaded on demand and HTML is produced locally from the browser. AJAX also allows programmers to separate methods and formatting for specific information delivery functions on the Web.

Programmers can use whatever languages that works for their specific goal. For example, raw data, usually obtained in Extensible Markup Language (XML) from a server-side database is separated from the format or structure of the webpage, which is usually structured in Extensible Hyper Text Markup Language (XHTML). This allows for dynamic handling of Document Object Model (DOM). Cascading Style Sheets (CSS) usage allows for the separating of style elements on the page, like fonts and picture placement. AJAX also separates the functionality of web pages by combining different elements in different ways. For example, JavaScript on the client-side browser is combined with XMLHttpRequest to enable communication between client and server browsers. Then any server-side program or scripting language allows the programmer to quickly respond to client requests in a language and format they are familiar with.

Java 2 Runtime environment was used to run the JESS applet. Javascript 1.3, Cascading Style Sheet (CSS) version 2 and XML 1.0 were used for AJAX technology. WAITIW application was implemented using Internet Information Server (IIS) as Web server.
3.1.1 Database Connectivity:

ADO.NET is designed to provide a disconnected architecture. This means that applications connect to the database to retrieve a load of data and store it in memory. They then disconnect from the database and manipulate the in-memory copy of the data. If the database needs to be updated with changes made to the in-memory copy, a new connection is made and the database is updated. Using a disconnected architecture provides many benefits, of which the most important is that it allows application to scale up. This means the database performance just as well supporting hundreds of users as it does supporting ten users. This is made possible since the application connects to the database only long enough to retrieve or update data, thereby freeing up available database connections for other instances of your application or other applications using the same database. ADO.NET is designed to talk to multiple databases, so there are different objects for different database types. To keep the separation, ADO.NET classes are contained within different namespaces. The figure 3.2 shows the architecture of the ADO.NET framework.
3.2 WAITIWIW Architecture:

The Architecture of WAITIWIW consists of the activities like initially requesting the user about the prevailing waste water quality conditions. This user input will be sent to the expert system component of WAITIWIW where they will be converted to facts by constructs provided in JESS. These facts are compared with the rules in WAITIWIW and the matching rules are then fired and the corresponding action is taken.
3.2.1 Writing Rules:

The rules for WAITIW were written using the *defrule* construct of JESS. The content for the rules were obtained by reviewing literature in peer reviewed journals [Groves 1983] [Nicholas 1996] [Rajkumar 2004] [Simpson 1987] [Wang 1987] and from books [John 1999] [Sen 2002] on effluent treatment. A total of 268 rules out of which 138 rules belong to textile industry and remaining 130 rules belongs to pharmaceutical industry were stored in the rule base of WAITIW for treating effluent. A simple rule construct in JESS can be described as follows.
(defrule water3
  (condition 2)
  (?x <- (condition ?number)
   =>
   (printout t " Apply high organic waste water to chemophysical or mechanical pre-treatment " crlf)
   (printout t " Apply effluent to advance aerobic treatment method " crlf)
   (retract ?x))

This rule will not fire just because the function call would evaluate to true. Instead, JESS will try to find a fact in the working memory that looks like (Condition 2). Unless you have previously asserted such a fact, this rule will not be activated and will not fire.

3.2.2 Writing Facts:

The facts of the WAITIW represent different conditions of the effluents that the user will enter through Graphical User Interface (GUI) interface. The GUI interface was developed as Rich Internet Application (RIA) and the user does not need to know the JESS commands to input the data. The user will interact with Jess using buttons.
Figure 3.4: GUI of WAITIW

The input from the buttons as shown in figure 3.4 will in turn invoke JESS commands through Action Listener classes in Java. A simple fact construct in JESS can be described as follows

```
(deffacts
    newComposition
    (the composition is organic)
)
```

Here the fact that the new composition is organic is added to the fact list and the appropriate rules corresponding to this fact will be fired by the inference engine of the expert system. The JESS inference engine automatically matches facts against patterns and determines which rules are applicable. The inference engine selects a rule and then the actions of the selected rule are executed (which may affect the list of applicable rules by adding or removing facts). The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain.
3.2.3 Matching Rules and Facts:

WAITIW has a predefined set of rules, representing different types of waste water treatment methods, while the facts, representing the different kinds of industrial waste water conditions (based on composition or content), that keep changing frequently with each composition. These rules and facts are supplied to the JESS rule engine that performs pattern matching. When a pattern is matched to a set of facts the rule corresponding to the facts is fired or executed. WAITIW can enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. New rules will be added based on the facts that user gives through the feedback process. This feedback will be sent to an expert on the regular bases using an automated email.

3.3 WAITIW Database:

The data about the users and feedback submitted from the users are stored in MS SQL Server 2000 in the form of tables. The tables that were used for WAITIW are described in the following section.

3.3.1 USER Table:

This table is used to store the username and password of the user who want to access this application. Email field of the user table was selected as a primary key. When user enters the username and password on the html form, it sends the request to the server to check whether or not both values match in the user table.

3.3.2 FEEDBACK Table:

This table is used to store the feedback that user would like to submit about a water condition that WAITIW is unable to detect. The data from this database will be
automatically submitted to an Expert and developer through an email. This will facilitates the iterative development and thus enhances the rule base of WAITIW application. Figure 3.5 shows feed back table in sql server 2000.

Figure 3.5: Feedback Table.

A user interface for inputting feedback from the user is designed using AJAX technologies as shown in Figure 3.6. An asynchronous xml message that contains the user feedback will be sent to the backend web application. The web application creates an automated email from the feedback message and sends it to the expert. Besides email process the user feed back will also be saved in the database using SQL Sever.
3.4 WAITIW Work flow:

Workflow of WAITIW defines the application processes flow and they are several phases attached with this process flow. User first needs to be authenticated and authorized to access the application. The figure 3.7 shows the application login process.
Upon successful login the user will be take a welcome page that has a hyper link to the WATIW application as show in the figure 3.8.

![Welcome page of WATIW](image)

**Figure 3.8:** Welcome page of WATIW

A user must be ready with waste water composition before they begin using the tool. To reset the WATIW tool use needs to click the ‘Reset Application’ button and upon resting the application the user needs to click the ‘Start Expert System button in order to load the CLIPS file into WATIW application. Upon successful loading of the CLIPS file the user can start using the application by clicking the ‘Run Application’ button as show in the figure 3.9.
Figure 3.9: First step in WAITIW

WAITIW initially prompts the user to provide input with regard to various water quality conditions in a form of multiple choices. Depending on user input the WAITIW tool will use the expert system's inference engine to conclude about the nature of effluent water and suggest the user in treating the effluent water as show in the figure 3.10.
Figure 3.10: Steps in WAITIW

3.5 Security Issues of WAITIW Application:

The security vulnerability that occurs as the result buffer overflow was prevented by developing the WAITIW using C# programming language. All the user data is stored in a secured MS SQL server database. The password that is supplied at login is encrypted and sent to the database by using Secure Hash algorithm-1 (SHA-1) using a code behind class.
4. EVALUATION AND RESULTS

Testing of WAITIW was done in two phases, once during the development phase and once during the post development phase. The process of debugging was an important aspect in the development of the project. During the development of the WAITIW expert system, care was taken to make sure that all the rules were written properly. Later the appropriate facts which fire the rules were added to the facts list.

4.1.1 Unit Testing:

Unit Testing of WAITIW was done using NUnit testing. Microsoft Visual Studio comes with NUnit built into the Workbench. This allowed to create test case and test suite classes rapidly. All the core java classes that were used in this project were tested using JUnit test cases. By having a set of tests that test all the core components of the WAITIW project it became possible to modify specific areas of the project and immediately see the effect the modifications have on the other areas.

4.1.2 Stress Tests

Extensibility and scalability of the application can be known thorough stress test. WAITIW was subjected to stress test by entering different water quality conditions as facts. WAITIW did reasonably well with all the possible water quality conditions and it was able to give appropriate solutions in treating the waste water during this test phase. The test cases were conducted in India with a help of a chemical engineers who works for Indian Institute of Chemical Technology (IICT) and Reddy Labs. The results of those test cases were show in the table 4.1
<table>
<thead>
<tr>
<th>Test Case No</th>
<th>Waste water Composition</th>
<th>Industry</th>
<th>Human Expert Result</th>
<th>WAITIW Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>pH</strong>: 8.09 Conductivity: 16.27 mS/cm Sulphates: 1650 mg/L COD: 1420 mg/L BOD: 259 mg/L TDS: 12500 mg/L Color: 2200 Pt-Co units</td>
<td>Textile</td>
<td>Need to reduce more coloring compounds. Suggestion: Treat with Reverse Osmosis process.</td>
<td>Treat alkaline water with dilute acid until pH reaches to 7. The Coloring and COD levels in water are too high. Need to reduce total dissolved solids and COD level in water. Suggestion: Send the water to Primary treatment process Sedimentation and then to intermediate clarifiers and then to final Clarifiers followed by Reverse Osmosis process.</td>
</tr>
<tr>
<td>2</td>
<td>Phenol: 2.0 mg/L Toluene: 3.0 mg/L Benzene: 10.0 mg/L Trichloroethylene: 9.0 mg/L Methylenechloride: 4.5 mg/L Ethylbenzene: 10.0 mg/L Chloroform: 7.5 mg/L</td>
<td>Pharmaceutical</td>
<td>Contains high content of organic compounds. Suggestion: Send to Aerobic Lagoon followed by Anaerobic fixed film</td>
<td>Send the water to liquid-liquid evaporation to absorb all benzene content in water then Send the water to activated carbon filters and then send water to Aerobic Lagoon and then to Anaerobic fixed film</td>
</tr>
<tr>
<td>3</td>
<td><strong>pH</strong>: 6.03 Conductivity: 16.23 mS/cm Sulphates: 1450 mg/L COD: 950 mg/L BOD: 239 mg/L TDS: 900 mg/L Color: 1900 Pt-Co units</td>
<td>Textile</td>
<td>Need to reduce more coloring compounds. Suggestion: Treat with Ultra</td>
<td>Treat the acid water with dilute calcium hydroxide until its pH reaches to 7. Coloring content are might be due to TDS. Need to reduce TDS,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>filtration.</td>
<td>Sulphate and COD level in water. Suggestion: Send the water to primary treatment for sedimentation process to remove sulphates and then send the water to Ultra filtration.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>pH: 7.09 Conductivity: 15.07 mS/cm Sulphates: 920 mg/L COD: 200 mg/L BOD: 290 mg/L TDS: 350 mg/L Color: 830 Pt-Co units</td>
<td>Textile</td>
<td>Suggestion: Treat the effluent water with sedimentation process</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sulphates contents are high Suggestion: Please send the effluent to primary treatment sedimentation in order to reduce sulphates.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Phenol: 10.0 mg/L Toluene: 12.3 mg/L Benzene: 13.0 mg/L Trichloroethylene: 9.0 mg/L Methylenechloride: 4.5 mg/L Ethylbenzene: 10.0 mg/L Chloroform: 6.5 mg/L</td>
<td>Pharmaceutical</td>
<td>Contains high content of organic compounds: Suggestion: Send to two stage aerobic process.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Send the water to Aerobic Lagoon and then to Anaerobic fixed film.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.1**: Test cases

### 4.1.3 Integration Testing:

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. Integration testing identifies problems that occur when units are combined. WAITIW system was subjected to integration testing when it was running on IIS server while integrating the JESS jar files with .NET Framework. Different water quality conditions of waste water were manually entered on **expertSystem.aspx** and the
rules that got fired were observed. This test was done several times by restarting the server to ensure that the whole application works well repeatedly without any problem.

### 4.2 Executing WAITIW Application

The execution of the WAITIW application was designed in a user friendly way keeping in mind that the users will be mostly industrial workers who have few or no knowledge about JESS and Java. This application can be executed by just clicking the right HTML buttons. WAITIW application consists of a login screen where the user enters a valid name. An asynchronous call to the server will be submitted through AJAX and the user credentials will be evaluated. In case of unauthorized user the page will show invalid user password without the reloading the page. A dynamic web page is generated with a greeting message and the expert system is ready for the user input. The file `expertSystem.aspx` has an applet embedded in it and invoking the right button will start the application. When the user clicks the ‘waste water quality’ button the system will automatically load the built in `waterTreatment.clp` file. This file contains all the rules that represent different water quality conditions in a well defined data structures. The user input to WAITIW will be in the form of choices through radio buttons. The user input is stored internally and the facts that user gave as inputs will be matched with the rules. Once WAITIW has enough facts to conclude about the right effluent treatment it will display on the applet text area as shown in the figure 4.1, treatment process and preventive measure. Since expert systems can’t solve all the problems in a given domain it would be a nice design consideration to have the user communicating with an expert. WAITIW application has a feed back form in the file `feedback.aspx` where the user can post about the problems of the domain and the performance of the WAITIW application.
This will help in iterative development of the application by adding new rules into the rule base of WAITIW from the user input. This is very important as different water quality conditions will be observed with more urbanization and industrialization.
5. FUTURE WORK

5.1 Developing a Multi-Media Rich Internet Application

The WAITIW was implemented as a RIA using Java Applet and AJAX. These technologies can support partial screen updates, asynchronous communication, modal dialogs and menus. But other RIA goodies may be severely limited in functionality or missing altogether. Typically, the set of rich UI widgets that supports direct manipulation will be poor compared to those we find in toolkits for desktop applications and in multimedia applications like Flash and Adobe Flex. In future this application needs to be modified as a full RIA using open source Lazlo or Adobe Flex 2. This enables to create personalized, multimedia-rich applications that dramatically enhance user experience, increasing customer satisfaction and making users more productive.

5.2 WAITIW as Web Services:

The WAITIW can be implemented as a web service application so that it can be a component of an integrated industrial expert system or as component of some other large software that were aimed at improving the environmental resources.
6. CONCLUSION

WAITIW project was an effort in implementing the first Web based tool for treating industrial effluents using a rule based expert system. It was able to eliminate the requirement of an expert by efficiently treating most of the industrial effluents that were found in textile and pharmaceutical industries. With several tests conducted in South India with the help of industrial workers it was proved to be a reliable tool for treating industrial waste water. Since this application was developed using JESS and .NET technologies scalability and other performance issue will be effectively addressed in future that comes with enhanced rules, database interaction, multimedia user interface, and automated KE mechanism.
7. BIBLIOGRAPHY AND REFERENCES


8. ACKNOWLEDGEMENTS

WAITIW I would like to express my sincere thanks to my project committee chairperson Dr. Longzhuang Li, for his guidance, expertise in the area of Expert Systems, support, and valuable insights all along the project, right from the proposal phase. I also thank the other committee members Dr. David Thomas and Dr. Hongyu Guo for their encouragement and valuable feedback during various stages of the project.
8. ACKNOWLEDGEMENTS

WAITIW I would like to express my sincere thanks to my project committee chairperson Dr. Longzhuang Li, for his guidance, expertise in the area of Expert Systems, support, and valuable insights all along the project, right from the proposal phase. I also thank the other committee members Dr. David Thomas and Dr. Hongyu Guo for their encouragement and valuable feedback during various stages of the project.