NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

19971119 054

THESIS

SUPPORTING DECISION AND NEGOTIATION IN AN INTERNET ENVIRONMENT: AN EXPERIENCE WITH NEGOTIATOR/I

by

Kimberly S. Blood
and
Joseph G. Garcia

March 1997

Principal Advisor: Tung X. Bui

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE **SUPPORTING DECISION AND NEGOTIATION IN AN INTERNET ENVIRONMENT: AN EXPERIENCE WITH NEGOTIATOR/I**	5. FUNDING NUMBERS	
6. AUTHOR(S) Blood, Kimberly S. and Garcia, Joseph G.,		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The purpose of this thesis is to explore implementation of decision support on the Internet. In particular it discusses four traditional decision making models. The information collected from these models will be applied to the creation of an Internet-based DSS. These models are the decision making model, problem solving model, creative thinking model, and the negotiation model. From an implementation point of view, this thesis develops a prototype decision support system for negotiation using Java. Realization of the prototype suggests that a decision support system (DSS) can be implemented using Java provided the DSS meets certain design parameters.

14. SUBJECT TERMS Negotiation in an Internet-based Environment			15. NUMBER OF PAGES 102
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

SUPPORTING DECISION AND NEGOTIATION IN AN INTERNET ENVIRONMENT: AN EXPERIENCE WITH NEGOTIATOR/I

Kimberly S. Blood
Lieutenant, United States Navy
B.A., Xavier University, 1986

Joseph G. Garcia
Captain, United States Army
B.S., College of Santa Fe, 1987

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March 1997

Authors: ______________________
Joseph G. Garcia

____
Kimberly S. Blood

Approved by: ______________________
Tung X. Bui, Principal Advisor

____
Balasubramaniam Ramesh, Associate Advisor

____
Reuben T. Harris, Chairman
Department of Systems Management

ABSTRACT

The purpose of this thesis is to explore implementation of decision support on the Internet. In particular it discusses four traditional decision making models and the information collected from these models will be applied to the creation of an Internet-based DSS. These models are the decision making model, problem solving model, creative thinking model, and the negotiation model. From an implementation point of view, this thesis develops a prototype decision support system for negotiation using Java. Realization of the prototype suggests that a decision support system (DSS) can be implemented using Java provided the DSS meets certain design parameters.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

A. AREA OF RESEARCH

The purpose of this thesis is to analyze and design a Negotiator Support System (NSS) that uses the Internet as a worldwide network backbone. Overwhelming success of the Internet has recently opened opportunities for Decision Support System (DSS) researchers to use that technology for deploying cost effective and user friendly DSSs beyond traditional geographic boundaries. Deployed as such, ownership of DSS resources is preserved and distributed data can be better accessed by all involved parties. Primary to this research is identifying the requirements of an Internet-based DSS by using various existing DSS models. As a proof-of-concept, this research focuses on building a Java-based prototype of a bilateral, multi-attribute NSS deployable on the World Wide Web (WWW).

B. RESEARCH QUESTIONS

- What are the underlying Decision Making Models that can be used to model a DSS?
- How does Internet technology affect the design and deployment of DSS?
- What are the current technologies available for migration of DSS to the WWW?

- What are the design considerations for building a DSS for NSS on the Internet?

- Can a NSS be built using JAVA?

C. OBJECTIVE SCOPE AND BENEFITS OF STUDY

The first objective of this study is to outline a modeling framework for building an Internet-based DSS. We do not however address issues related to the deployment or maintenance of DSS components on the Internet.

The second objective is to analyze, design, implement and test a DSS prototype Negotiator/I, using Java.

D. METHODOLOGY

To explore the potentiality of Internet-based DSS, we simulate the use and benefits of DSS under four theoretical decision models, i.e., the decision making model, the problem solving model, the creativity model, and the negotiation model. To demonstrate the feasibility of using a DSS on the Internet, we developed a NSS prototype in Java using the software developmental tool, Microsoft J++. Designed to execute within any Java enhanced Browser, the prototype, once accessed via the Internet, uses a simple menu-driven user interface which can be easily followed.

Realizing the dynamic environment of Internet technology, considerable effort has been devoted to designing source code that can be easily read, updated, and maintained. An object-oriented approach has been adopted throughout the design of NSS to ensure that a set of logically organized data structures and functions are provided.

F. ORGANIZATION OF THE STUDY

This chapter is followed by the following:

Chapter II discusses and analyzes different underlying decision making models which can be used to model a DSS.

Deployment of DSS technology is discussed in Chapter III. The different software tools available to the webmaster, as well as strengths and weaknesses for each are also analyzed.

Chapter IV offers an example application of Negotiator/I. Throughout this discussion, tables, charts, and screen prompts provide a detailed description of the systems functionality.

Chapter V concludes the study by discussing lessons learned and recommendations for future development and continued study.

II. FRAMEWORK FOR DISTRIBUTED DECISION MAKING ON THE INTERNET

A. DECISION MAKING AND THE WORLD WIDE WEB

Decision making on the Internet is receiving increased attention in management literature as well as management of information systems literature. This chapter examines four models of decision making and discusses the extent to which they can be supported by Internet technology. We describe how decision making, problem solving, creative learning, and negotiation take place under three decision environments. These models of decision making will be examined from the individual's perspective to learn how decision making occurs under three scenarios: without the support of a computer, with the support of a computer, and in a futuristic Internet-based decision support environment.

The phenomenal growth of global information networks like the World Wide Web has created new opportunities for the development and deployment of decision support technologies. In the past ten years, modifications and improvements in the development of decision support systems (DSS) have been continuously reported in academic literature. During this same time, the use of decision support systems beyond the originating site has been small. This limited deployment is attributable to two factors.

> First, potentially interested DSS users are simply not aware of the existence and availability of DSS. Second, DSS are often application-specific and their installation is confined to their operating environment (e.g., database access, hardware and software configuration, technical support), limiting their deployment (Jeusfeld & Bui, 1996).

To date, the WWW is used primarily for electronic commerce, advertising and information distribution. All three describe examples of non-interactive information passed from server to user. Non-interactive denotes that no dialogue occurs between a Web server and a user. Some typical examples of non-interactive uses on the WWW are: making airline reservation, checking stock prices, and browsing for information. Although making an airline reservations could be considered two-way communication, it is nothing more than a user accessing a web site, locating the desired destination and booking the flight. Yes, there may be confirmation of flight information, transaction amount, and cancellation policy. Yet, there is no dynamic processing on either side occurs to detect cheaper or more cost effective flights.

As the use of scripting languages, such as Java, ActiveX, SSI, API, Cobra, and Windows CGI, become more prevalent, implementation and use of DSS on the WWW becomes more readily available. Interactive information can help businesses and consumers in decision making. The best way to illustrate the benefits of DSS on the WWW is through an example. We evaluate four previously stated models using a stock market investment problem under three different circumstances: 1) decision making without a computer; 2) decision making with a computer and Internet access, and 3) decision making with a futuristic Internet-based decision support environment.

B. SIMON'S MODEL OF DECISION MAKING

Simon's classical framework for decision making (Simon, 1960). His framework provides a basis of how users make decisions, the steps or processes involved in analyzing a problem, making a choice, and implementing that choice. According to Simon, decision-making processes fall along a continuum that ranges from highly structured to highly unstructured decisions. Structured processes describe routine or repetitive problems where a standard solution exist. Unstructured processes apply to those problems for which no clear solutions exists. Decision support systems deal mainly with unstructured problems where there are many variables and choices (Turban, 1988). Simon divides the decision-making process into three phases:

- **Intelligence** - searching for conditions that call for decisions.
- **Design** - inventing, developing, and analyzing possible courses of action.
- **Choice** - selecting a course of action from those available.

Table 1 depicts the decision making model applied to investment problems.

Table 1. Model of Decision Making
For Stock Selection Example

<table>
<tr><th></th><th colspan="3">Decision Environment</th></tr>
<tr><th>Phase</th><th>No computer</th><th>Computer W/Internet</th><th>Computer W/Internet Future</th></tr>
<tr><td>Intelligence:
- Objectives
- Search and Scanning
- Problem ID
- Problem Classification
- Problem Statement</td><td>- Determine financial goals and objectives
- Assess if problem exists
- Gather Information: newspaper, library, magazine, broker, company reports
* Information gathered at a minimum 12 hours to several weeks old*</td><td>- Determine financial goals and objectives
- Assess if problem exists
- Evaluate stock performance and ranking
- Gather information: use web sites such as Lombards or WSJ to collect data (20+min delay)</td><td rowspan="2">- Determine financial goals and objectives
- Verbally ask computer to search WWW for investment server
- Narrow search to stocks with variable: ROI, Risk factor and liquidity.
- A DSS located on the WWW will upload user entered information analyze it, formulate a model, and narrow the search by asking the user for more info. i.e., time frame, $ amt, type of stock NASDAQ vs. NYSE
- Results of WWW search and alternatives</td></tr>
<tr><td>Design:
- Formulate a Model
- Set criteria for choice
- Search for alternatives
- Outcomes</td><td>- Understand problem thoroughly
- Evaluate possible courses of action
- Set criteria (ROI, risk, liquidity)
- Formulate a model</td><td>- Formulate a model, and use the Internet to maximize ROI
- Set Criteria and evaluate using Stock quote programs
- Reduce uncontrollable variables w/Internet</td></tr>
</table>

Table 1 (Continued)

	Decision Environment		
Phase	**No computer**	**Computer W/Internet**	**Computer W/Internet Future**
Choice: **- Solution to the Model** **- Sensitivity Analysis** **- Selection of Alternative(s)** **- Plan for implementation**	- Evaluation of solutions among alternatives - Which solution provides ROI required - Select solution	- Evaluation of alternatives using Internet (20 + min delay on information) - Determine which solution provides for greatest ROI - Select solution	- A list of current alternatives will be displayed on the CRT, showing possible ROI and risk involved (virtually no lag time). * May want to do another search for other alternatives*

1. Decision Making Without a Computer

a. The Intelligence Phase

This phase consists of gathering all relevant information with the purpose of gaining a thorough understanding of the problem at hand. It entails scanning the environment, intermittently or continuously. It includes all activities geared towards identifying a problem or an untapped opportunity. In our example of an individual deciding what stock to buy, the intelligence phase begins with the determination of an individual's financial goals. An individual attempts to find out if a problem exists in the current way stocks are purchased/sold, find the symptom of the problem, determine its magnitude, and define the problem. Often, what is described as a problem (e.g., low ROI) may be only a symptom of an underlying problem (e.g., improper stock selection). Due to the volatile nature of the stock market, distinguishing between a symptom and a problem may be difficult.

Additionally, an individual collects data for use in the next phase. Information on the stock market can be gathered from business sections of newspapers, libraries, magazines, and directly from the issuing company. To fully understand failure to receive an expected ROI, an individual may need to enroll in a class, check out books on stock purchasing, or ask friends who are making money trading stocks. The next step is to sort and sanitize the information gathered.

b. The Design Phase

The design phase involves generating alternative solutions and evaluating possible courses of action. With a thorough understanding of the problem, a model of the situation is constructed and validated. This model generates and tests possible solutions for feasibility.

> Modeling involves the conceptualization of the problem and its abstraction to a mathematical-numerical model and/or other symbolic forms. In case of a mathematical model, the dependent and independent variables are identified and the equations describing their relationships are established. Simplifications are made, whenever necessary, through a set of assumptions. For example, a relationship between two variables may be assumed to be linear. It is necessary to find a proper balance between the level of simplification of the model and the

> representation of reality. A simpler model leads to easier manipulation and solutions, but is also less representative of the real problem. (Turban, 1988)

To create a model decision variables, result variables, and uncontrollable variables must be identified. Decision variables are those aspects of a problem for which the decision maker must supply a value. All the possible combinations of decision variables represent the set of alternative courses of action. For the stock market problem, the amount of money invested in each alternative is a decision variable. Other decision variables include duration of investment and timing of investment, i.e., deciding when and at what price to buy and sell.

Result variables indicate the level of effectiveness in attaining desired outcomes among alternative courses of action. Result variables are dependent upon the occurrence of some prior event. In this model, decision variables and uncontrollable variables affect result variables. Success is achieved when result variables respond to other variables in ways consistent with the desired outcome. Some measures of effectiveness for choosing stock are: Total Profit, Rate of Return (ROI), Earnings per Share, and Liquidity of Stocks.

In any decision situation, factors exist which affect the value of result variables but are not under the control of the decision maker. These factors are referred to as uncontrollable variables. Table 2 identifies the variables for the stock market investment problem.

Table 2. Model of Problem Solving

Components of the Decision Making Model			
AREA	**Decision Variables**	**Result Variables**	**Uncontrollable Variables and Parameters**
Financial Stock investing	- Investment Amount - Duration of Investment - Timing of Investment	- Total Profits - Rate of Return - Earnings/Share - Liquidity - Market Share - Investor Satisfaction	- Inflation Rate - Prime Rate - Success of company - Tax Rate

c. *The Choice Phase*

The boundary separating the design and choice phase is not well-defined. Certain activities may be accomplished both during the design and the choice phases. The choice phase includes searching for, evaluating, and selecting an appropriate solution to the model (Turban, 1988).

A solution to a model derives from a set of values for the decision variables which will produce the desired results. In this phase, decision makers state his/her preferences, defines choice criteria, and select solution accordingly. Solving the model is different from finding or solving a problem that the model presents. The investor may find what he thinks is the optimum solution, which may in fact not solve the problem and achieve the desired results. In our example, a solution could be to invest in a diversified portfolio. When this portfolio achieves the targeted ROI, the problem is solved.

2. Decision Making with Computer Support and Internet Access

a. The Intelligence Phase

The Internet provide prospective stock investors with quick access to a wealth of information that would not have been possible otherwise. The only requirements for the investor is to have access to the Internet, a browser, and time to search through massive amounts of information available on the Web. The WWW uses the Internet to transmit hypertext documents between Internet servers and users located throughout the world. Web browsers serve two main purposes. First, as a navigation tool, a browser locates and retrieves requested documents by means of a Uniform Resources Locator (URL). Second, it interprets, formats, and displays HTML documents in a way suitable for the user's console. The stock investor in our example needs to know where the relevant information is stored on the WWW. To get a list of related URLs, a web search engine is required. Several of which are available free of charge, e.g., Yahoo!, Web Crawler, Lycos, and Infoseek. To utilize these search engines, a user enters a keyword(s) about the desired topic. The search engine matches this keyword against a huge index of URLs cross-referenced by topic keywords. Results are displayed according to the degree of consistency between the supplied keywords and the keywords of the matching documents.

A prospective investor can access the WWW via a browser, locate a search engine, such as Yahoo at URL http://www.yahoo.com, type in a search for "Stock Quotes" and "Investor's Information." The search engine traverses the WWW looking at titles, word frequency counts, and subjects that match the user's search. The search results display the first ten matches with the option of viewing the others (e.g., Wall Street Journal, Lombards, Stock quote, Galt Financial Service, Morning-star, Business Week and others). With the first ten, an investor has all the tools necessary to purchase stocks and receive tips from professional investors. Some information providers or low-cost brokerage firms can take bids on stock transactions on-line for a small fee.

With Internet access and websites such as the Wall Street Journal, Lombards, and Galt Financial Services, tasks related to the intelligence phase become easy. An investor can retrieve stock quotes, stock performance several years back (in graph format), hot stock tips, and company information. Not only does this mode of information gathering save time, but also saves money. An investor does not have to purchase various magazines, newspapers and books which may prove not to contain pertinent information. The WWW can also provide information on buying stocks, tips from stockbrokers, and stock rankings by organizations such as Lipper Analytical

Services, Inc. and Morningstar, Inc. An investor can compare stock performance against market benchmarks like the Dow Jones Industrial Average or the Standard & Poor's 500 Stock Index. All of this information is available without ever leaving the computer desk.

b. The Design Phase

In creating a model for our stock market problem, we must again determine the decision variables, result variables, and uncontrollable variables. All the variables described in Table 2 remain unchanged when using a computer. If the various variables do not change, then how does the computer impact levels of effectiveness or result variables?

The result variables of total profit, ROI, and liquidity of stocks depend upon uncontrollable variables of inflation rate, prime rate, and success of a company. Internet access does not change uncontrollable variables, but does provide an investor with current information that can play a critical role in stock purchases. Using the WWW, the investor can judge when is the best time to sell stock by checking the highs and lows for that day/week/month/year. The web provides up-to-date inform-t on stock-splits, mergers, takeovers, company research breakthroughs, and buyouts, all of which affect the price of stocks. A prospective invêstor with information on a twenty-minute delay has far more current data than the investor without Internet access whose information is at least a day old. Timely information reduces the effects of uncontrollable circumstances, thereby increasing the success rate of the result variables.

c. *The Choice Phase*

The Internet increases the user's ability to analyze the merits of the alternatives. Using the information gained via the Internet, a prospective investor can weigh different alternatives before investing dollars. An example is comparing a purchase of Speculative Stocks (high risk, big ROI) versus Blue Chip Stocks (safe, reliable, lower ROI). An investor can also compare diversified port-folios to see what type of ROI can be expected. These comparisons can be conducted as a one time look at the performance statistics or carried out over an extended time period tracking three or four alternatives. The latter portfolio analysis is extremely difficult without Internet access. As an example, portfolio analysis can be accomplished by using Point Cast Network (PCN), choosing ten stocks in each portfolio and following the performance of portfolios. Web tool such as PCN lets the computer do most of the work. PCN will look up the stock display, the current high and low for that day, provide a 30-day graph on each stock in the investor's portfolio, and provide current news articles about a company. After several months of tracking prospective portfolios, a stock holder can select which portfolio is the best investment.

3. Decision Making with Computer Support and Internet Access (Future)

In a sense, computers in the next decade will be similar to those of today. Similar in that they take in data, process the data, and display output. In the future, a user asks the computer to conduct a search for a specific topic and that topic is displayed without ever touching a keyboard or mouse. Voice recognition programs and translation software will be standard equipment on all computers. Video Teleconferencing conducted in two languages without the use of a human translator becomes a reality. The computer translates the information from one language to another and from one form to another (i.e., audio to text). This feature is extremely attractive for nations such as Japan whose alphabet is easier to write by hand than type with a computer. The computer and Internet of the future can best be described through an example. The previous scenario of a stock market investment problem is used to illustrate how the next generation of the WWW helps our prospective stock holder.

Simon's decision making model, used to illustrate the decision making process, consists of intelligence, design and choice as the three steps in and making a decision. The model hypothesis holds true for an investor searching for the greatest ROI using a non-Internet-based decision support systems. In the two previous decision environments, i.e., no computer support and computer support with Internet access, a user methodically went through each step before arriving at a decision. In the scenario of the future, a user has access to an Internet-based decision support systems (DSS). These systems will be available as web sites for use in the decision making process.

a. Intelligence/Design with Computer Support and Access to Internet of the Future

By incorporating DSS on the WWW, Simon's decision making model is implemented in a manner different from the previous decision environments. The interaction between the Intelligence and Design phases is accelerated. Continuing with the same scenario, an investor begins this phase by determining the financial objectives. Once the objectives are determined, a user accesses the WWW, initiates a search with a search agent to locate web sites with a DSS to help investors with stock purchases. Once a financial DSS is found, the user can narrow the search by providing necessary information such as ROI desired, type of risk, and liquidity of stocks. The DSS searches its database and historical data to provide the user with alternatives. Depending on its database access, it can search for historical data and analyze the information and formulate a model. It may be necessary for the user to answer other questions, e.g., NASDAQ vs. NYSE, Blue Chip or Growth. This process continues till the DSS gathers enough information to provide the user with a list of alternatives that meet the criteria entered. A list of alternatives that meet the users financial objectives will be displayed. The alternatives presented are based on historical and current observations and should be used as a guide only.

b. Choice Phase

During the choice phase, the selection of a particular alternative meeting the criteria and objectives of the investor becomes almost effortless. The computer displays several stocks and portfolios which have the ROI desired. The prospective stockholder is responsible for conducting a sensitivity analysis to determine which type of investments meets the objectives. With a sensitivity analysis complete an investor can select which alternative to carry out.

C. NEWELL AND SIMON'S MODEL OF PROBLEM SOLVING

The previous section centered on Simon's model of decision making and how using the WWW can be an integral part of the decision making process. Newell and Simon (1972) define a model of problem solving as an extension to Simon's decision making model. Two additional phases are added to the model: implementation and monitoring. The prospective stock holder example is used to illustrated incorporating the implementation and monitoring phase.

Table 3. Model of Problem Solving
For Stock selection example

<table>
<tr><th></th><th colspan="3">Decision Environment</th></tr>
<tr><th>Phases</th><th>No computer</th><th>Computer avail. W/Internet</th><th>Computer W/Internet Future</th></tr>
<tr><td>Intelligence:
- Objectives
- Search and Scanning
- Problem ID
- Problem

Classification
- Problem Statement</td><td>- Determine financial goals and objectives
- Assess if problem exists
- Gather Information: newspaper, library, magazine, broker, company reports</td><td>- Determine financial goals and objectives
- Assess if problem exists
- Evaluate stock performance and ranking
- Gather information: use web sites such as Lombards or WSJ</td><td rowspan="2">- Determine financial goals and objectives
- Verbally ask computer to search WWW for investment server
- Narrow search to stocks to stocks that meet a specific prerequisite: ROI, Risk factor and liquidity
- A DSS on the investment web server will take the information analyze it, formulate a model, and narrow the search by asking the user for more info. e.g., time frame, $ amt, type of stock NASDAQ vs. NYSE
- Result of WWW search will be displayed</td></tr>
<tr><td>Design:
- Formulate a Model
- Set criteria for choice
- Search for alternatives
- Outcomes</td><td>- Thorough understanding of problem
- Evaluating possible courses of action
- Set criteria (ROI, risk, liquidity)
- Formulate a model</td><td>- Formulate a model, and use the Internet to max. ROI
- Set Criteria and evaluate a using Stock quote program
-Uncontrollable variables are reduced w/Internet</td></tr>
</table>

Table 3 (Continued)

	Decision Environment		
Phases	**No computer**	**Computer avail. W/Internet**	**Computer W/Internet Future**
Choice: **- Solution to the Model** **- Sensitivity Analysis** **- Selection of Alternative(s)** **- Plan for implemen-tation**	- Evaluation of solutions among alternative - Which solution provides ROI required - Select solution	- Evaluation of alter-natives using Internet - Determine which solution provides for greatest ROI - Select solution	- A list of prospective alternatives will show up, showing possible ROI and risk involved. - * May want to do another search for other alternatives*
Implementation: **- Of the chosen solution**	- Call a stock broker to purchase selected solution	- Initiate a stock purchase via a dis-count broker on WWW trade will not go to floor until following day	- Verbally initiate stock purchase on WWW, trade info. goes straight to the trading floor no delay
Monitoring: **- Reviewing to see if need for change exists**	- Look up each stock in business section of paper and manually calculate [price x shares = portfolio amt]	- Use Stock Quote or PCN to track port-folio via the WWW. Manually compute [price x shares= portfolio amt]	- Access a stock provider down load stock prices and trans-action confirmations to a portfolio tracker program - The program updates and provides a detailed list of stocks owned, prices per share and net worth

1. Problem Solving Without a Computer Implementation/Monitoring Phase

Without a computer, the purchase of stock can only be done via a telephone call or letter to a broker. A prospective stockholder tells the broker which stocks to purchase, the quantity, and price to buy. Once the initial transaction takes place, a stockholder is responsible for tracking the portfolio. To do so, a stock-holder must purchase a newspaper and manually look up each stock for ending price and highs and lows for the previous day. This is time consuming and many investors just let their stocks ride and check them on a weekly/monthly basis to determine the net worth of their portfolio. Also, the brokerage firm will send out a quarterly statement indicating stocks owned and the value of a portfolio. The information is typically seven to ten days old.

2. Problem Solving with a Computer and Internet Implementation/Monitoring Phase

Purchasing stocks with a computer and Internet access is a quick process. The user contacts the discount brokers homepage, enters a PIN number, and initiates a stock trade. The information gets processed at the close of business that day and goes to the stock market floor the following business day. Time frame between initiating a stock trade and going to the floor is the same with or without Internet access. The benefit of the Internet is through monitoring stock performance and overall portfolio success. A user can access one of the many stock quote sites, enter the stock trading symbol, and retrieve information on the trading highs and lows for that stock. To actually determine the worth of the investors portfolio, some manual calculations are still necessary. An investor must get the stock price and multiply it by the number of shares owned to determine the dollar value of that particular stock. This process is repeated for each stock owned. Adding all the dollar amounts yields the net worth of the investor's portfolio. By using the WWW to monitor the fluctuation of the market, an investor can gauge the best time to buy/sell stocks and at what price. This information, if used correctly, can help maximize ROI.

3. Problem Solving with an Internet-Based DSS

Implementation/Monitoring Phase

Verbally initiating a stock transaction on the web of tomorrow is expected to be as easy as saying "initiate alternative number one." Once the design phase shows a list of possible alternatives, a user can either choose one or save the list for later implementation or review. Currently, both Netscape's Navigator and Microsoft's Internet Explorer contain tools that let Web sites reach into the hard disk of any PC and run programs. When a user initiates a trade over the Web, a stock broker's applet updates the user's information with the current trade and downloads that data to the user's financial software package, automatically updating it.

> These applets--the best-known are Java and ActiveX--can customize visits to the Web site based on your prior entries.
>
> The applet gathers details about your activities and then resides on our hard disk until you return to the site; when you do, it runs programs that lets you avoid having to re-input your name, interests and other information. Sometimes, the applet even moves you directly to the spot you left off on your last visit (Coffee, 1997).

A typical transaction entails a prospective stockholder verbally initiating a stock transaction. Information is sorted in the brokerage company's database and sent immediately to the stock market floor. If the trade takes place, the user is notified via secure e-mail. The stockholder tracks the net worth of the portfolio by simply entering the trading symbol of the stock owned and the amount of shares owned on his computer. The information is stored on the user's hard disk which can be accessed by the financial services applet each time the user enters that particular web page. To get the net worth of the portfolio, the stockholder accesses the financial services home page, downloads the applet and, gets an update based on personalized information already stored on the hard disk. This applet can also provide "what-if" information to the user. "What if I sold TNCR at forty-six dollars, what would be my net gain after brokerage fees?" A simple algorithm within the applet can compute this information. Most brokerage exchange companies charge a flat rate to a certain dollar amount and a percentage of the trade thereafter.

The Internet-based DSS of the future provides the stockholder with current information on companies and suggestions on how to maximize ROI. By using DSS, databases, and changing some result variables, the stockholder will have all the tools necessary to make informed decisions on what type of stocks to buy and at what price to maximize ROI.

D. WALLAS' MODEL OF CREATIVITY AND DECISION MAKING

Understanding creativity and its effects on decision making requires an understanding of how external factors can influence both groups and individuals and hamper or facilitate creativity processes. Although group problem solving effectiveness is usually higher than the sum of the effectiveness of its members, individual capability is generally a primary determinant of what the group can do (Kelly, 1969). For example, a group is formed to determine the best way to maximize ROI from buying and selling stocks and if none of the group members are knowledgeable in this area, it is unlikely that the group will be as effective as a single individual with some knowledge of the stock market. It is not necessary to have a group present in order for dialogue to take place and assist in creativity thinking. Tools such as the World Wide Web can be used to focus and bring new ideas to mind. This understanding is applied as a foundation for examining how the WWW can influence and assist in fostering the creativity process.

To fully understand how the creative process works and assists in decision making, we use Wallas' model of creative problem solving. Wallas developed a phase model of "creative process" generally regarded as descriptive of how the creative process proceeds (Evaristo, 1993). According to Wallas (Wallas, 1926), creative problem solving comprises four distinct phases: preparation, incubation, illumination and verification. During the preparation phase, the user analyzes the problem and attempts to find a solution for it. This stage ends before finding a suitable solution. The incubation stage is characterized by the individual no longer consciously working on the problem and engaging in another activity. There is no time frame between the phases. An investor can stay in the preparation and incubation phase for several months without ever reaching the illumination phase. While performing other tasks, the individual unexpectedly arrives at a solution to the problem. This is the illumination phase. Last is the verification stage where the user validates and analyzes the solution. If the solution is deemed unacceptable, the user goes through one or more of the previous phases until an acceptable solution is developed. Wallas' model is used to demonstrate the creative problem solving process for the investor attempting to maximize ROI by investing in the stock market.

Table 4. Model of Creativity
For Stock Selection Example

	Decision Environment		
Phases	**No computer**	**Computer avail. W/Internet**	**Computer W/Internet Future**
Preparation: **- Search for solution** **- Develop more complex under standing** **- Break problem down into subgoals** **- Gather information** **- Attempt to solve problem**	- Determine financial goals and objectives - Gather info: newspaper, library magazines, broker, friends - Attempt to solve problem, and place a few chunks of information in long-term memory (ideas)	- Determine financial goals and objectives - Gather info: use WWW and look at WSJ, Morningstar, Lombards, Quotecom - Search WWW for news groups of independent investors - Attempt to solve problem	- Determine financial goals and objectives - Verbally ask computer to search investment DSS on the WWW for information on stock available with the desired requisite - Narrow search variable to stocks that meet a specific prerequisite: ROI, Risk factor and liquidity - A DSS on the investment web server will take the information analyze it, formulate a model, and narrow the search by asking the user for more info. e.g., time frame, $ amt, type of stock etc.
Incubation: **- Formation of new ideas** **- Linking of various Ideas** **- Trimming of subgoals**	- Subgoals worked on are forgotten - Combining of different ideas subconsciously forming new ideas and trimming subgoals - Work on another project	- The Internet can provide new ideas on underrated companies - On-line business journals can provide current data that can be used in modifying goals - Company records and financial statement accessed via WWW - Investment tools such as PCN which can track stocks on a minimum 20 min delay	

Table 4 (Continued)

	Decision Environment		
Phases	**No computer**	**Computer avail. W/Internet**	**Computer W/Internet Future**
Preparation (Continued)			- Result of DSS model will be presented on CRT and can be put in hard copy
Illumination: **- Piecing together of new ideas** **- Arriving at a solution**	- While working on other problems - Talking with friends knowledgeable on investing - Hiring a financial advisor - Flipping a coin	- News groups on investing - E-mail from friends - Daily investor stock tips from investor journals - After conducting a two month sensitivity analysis using free financial tracking tools on the WWW	-The DSS will suggest which stocks meet the desired ROI based on historical data - It is up to the user to pick a solution
Verification: **- Determine if the generated idea is valuable**	- Evaluate the solution - Look up each stock to determine if solution provides desired ROI - If it is not acceptable go through one or more of the phases until a suitable solution is found	- Evaluate the proposed stock investment using PCN to track portfolio - If ROI is not acceptable, surf the Internet and go through one or more of the phases until a suitable solution is found	- Evaluate the solution - Access a stock quote provider and personalize a search on the users portfolio - If portfolio is not acceptable go through one or more of the phases until a suitable solution is found

1. Creative Problem Solving Without a Computer

a. Preparation Phase

The preparation phase requires a creative thinker to search out and evaluate solutions for possible implementation. During this phase, an individual develops one or more chunks of information (ideas) in long-term memory through the process of familiarization. In the example of a prospective investor hoping to maximize ROI in stocks, the investor begins this phase by gathering as much information as possible on stock investing. Typically this type of information can be found in newspapers and magazines or from friends and stock brokers. Again, this type of data collection is not only time consuming, but is meaningless unless all the data is sorted and stored for later retrieval. The best indicator of stocks is history. The prospective investor can get the financial statement of most companies in the local library or from an investment club or stockbroker. Stockbrokers and investment clubs have the resources available to catalog and store pertinent information for later use. This type of manual historical record keeping is quite difficult when done by an individual and further complicated if the portfolio includes several companies.

b. Incubation Phase

The incubation phase is characterized by the investor, after a period of searching, ceasing work on the problem and moving onto another project. This may occur from frustration at not resolving the issue or by other more pressing matters taking precedence. During this time, weak links between the ideas and other pieces of information stored in long-term memory become reinforced and reacquainted. This phase is also characterized by selective forgetting. During the preparation phase, several ideas about what stocks to purchase, types of portfolio, and other types of brain-storming ideas are forgotten. This type of forgetting may take place as the investor does more research and explores new avenues to invest in. As previously mentioned, the prospective investor may stay in the incubation and preparation phase without ever reaching the illumination phase.

c. Illumination Phase

The illumination phase is characterized by the idea suddenly coming to the individual. It has also been referred to as the light bulb being turned on. Simon suggests that the individual enters the illumination phase when choosing to address the same problem a second time. This time, the individual has several ideas generated during the previous phases and can draw upon these to generate a solution or illumination. When an individual pieces together enough information to formulate an idea, illumination is achieved.

In our example, the investor has reviewed literature, done research and learned how to invest in stocks, but does not have a solution that can complete the portfolio. One day, over coffee, a friend suggests a stock to purchase with great return. The prospective investor investigates and finds this is the stock to buy. After attempting to find a solution the investor enters the incubation stage and works on other problems.

d. Verification Phase

Once an investor decides which stocks to buy, it is now time to see if the solution meets with approval. Without a computer, this requires the stockholder to check the financial section of newspapers to see what type of return is being achieved on the portfolio selected. This can be a time-consuming undertaking. Another alternative is to let the stocks ride and wait for the monthly statement. Naturally, with the constant fluctuation of the market and changes in companies' earnings on a daily basis, timing the market is critical to achieving a high ROI. Even checking the newspaper does not guarantee the information is current. The investor can call the brokerage service and check the current price and highs and lows. However, this is also extremely time consuming. Following the advice of all stock investors to buy low and sell high becomes disproportionately difficult without real-time data.

2. Creative Problem Solving with Computer Access and Internet

a. Preparation/Incubation Phase with Computer Access and Internet

Researching stock information during the preparation phase with a computer and Internet access can dramatically reduce the time spent gathering information. The Internet is a one stop library and newspaper stand. The amount of information available can be overwhelming, but with the use of search engines such as Yahoo!! or Web Crawler, searching for and finding required information becomes much easier. (See paragraph B.2a. for details on search engines).

Another source of information on the Internet is news groups and bulletin boards. A prospective investor can access newsgroups whose members invest in the stock market and are current on the latest trends. Newsgroups and bulletin boards afford individuals with the opportunity to post questions for other members to answer. This type of group decision support speeds the entire creative process. In a group decision support arena, the process of selective forgetting and linking normally observed in the incubation phase may be eliminated by this type of data exchange. Thus, preparation and incubation phases are combined in the model of creativity. Searching various journals on the WWW, our investor formulates several ideas on which stocks to purchase that may produce the required ROI. Searching for more information, our investor finds a newsgroup and posts several questions to the group. The idea exchange combines two required processes. The new ideas generated by the newsgroup forces an individual to stop thinking about the current idea (forgetting) and evaluate the new ideas which may connect to other related ideas (linking), thus potentially producing a creative response.

b. Illumination Phase

As described in the preparation phase, the Internet breeds creativity. Tools on the Internet such as Morningstar, Wall Street Journal, newsgroups, and e-mail help a prospective investor reach this stage more quickly by linking new and innovative ideas generated by the Internet. The WWW creates an environment in which the prospective investor has access to hundreds, possibly thousands, of creative thinkers. With such an environment, an investor is illuminated to several solutions. These solutions are scrutinized using tools, such as Point Cast Network (PCN), to determine which solution achieves the desired ROI. (Previously described in paragraph B.2.c.).

c. Verification Phase with Computer Access and Internet

Once an investor actually makes a stock purchase and starts a portfolio, the WWW can be the primary data source to monitor different stock performances and overall portfolio success. The methods to track stock performance in the verification phase is identical to the methods previously under Problem Solving with a Computer and Internet, Implementation/Monitoring Phase (paragraph C.2.a). The information gets processed at the close of business that day, and will go to the stock market floor the following business day. The time frame between initiating a stock trade and going to the floor is the same regardless of Internet access. The benefits of the Internet through monitoring stock performance and overall portfolio success. A user can access one of the many stock quote sites, enter the stock trading symbol and retrieve information on the trading high and lows for that stock. To actually determine the worth of the investors portfolio, some manual calculations are still necessary. The investor must get the stock price and multiply it by the number of shares owned to determine the dollar amount of that particular stock. This process is repeated for each stock owned. Adding all the dollar amounts yields the net worth of the investors portfolio. By using the WWW to monitor the fluctuation of the market, the investor can gauge the best time to buy/sell stocks and at what price. This information, if used correctly, can help maximize ROI.

3. Creative Problem Solving with Computer Access and Internet Future

a. Preparation/Incubation/Illumination Phase

As demonstrated in the previous section (D.2.a-b), Internet access enabled investors to quickly and easily retrieve timely stock information thus collapsing the incubation phase into the preparation phase. The introduction of Internet-based decision support systems on the Internet of the future causes yet another phase to collapse into the previous phases.

During traditional preparation/incubation/illumination phases an investor would define criteria, research stocks, consider the data and finally link various ideas together to form a solution. An Internet-based DSS would perform all these tasks. An investor would need only define the desired values for the result variables. The DSS would present an investor with a list of possible solutions. The investor selects the investment portfolio which feels the most comfortable. Upon purchase of the selected portfolio, an investor moves to the verification phase.

b. Verification Phase

A portfolio is created at the time the prospective investor completes a stock trade. The performance of the portfolio is tracked in the verification phase. Very little changes in this phase from the Internet of today to the Internet of the future. The twenty minute delay for stock updates is reduced bringing updates to near real-time.

D. EVOLUTIONARY SYSTEMS DESIGN (ESD) FRAMEWORK FOR NEGOTIATIONS

Negotiations can be characterized as opportunistic interaction by which two or more parties, with some apparent conflict, seek to do better through jointly decided actions than they could otherwise accomplish alone. Negotiations involve both cooperation and conflict-cooperation to create value (increase the size of the pie) and conflict to claim it (take as big a slice of the pie as possible) (Bui, 1996).

Negotiations have always been an integral part of business, organizational management, and international affairs. With ever increasing competition, negotiations require greater sophistication and faster resolution. When DSS are readily available on a medium such as the WWW, faster problem resolution becomes a reality. Today, the information and knowledge of the parties involved are more technologically complex, making it more difficult to crisply define positions which may lead to agreement. Often, each party to the negotiation knows conceptually the multiple issues of the problem in good detail, but this is not sufficient to define each other's preference/utility functions in a deterministic and interactive fashion. Current DSS systems, however, handle only deterministic information. In reality, utility functions are not deterministic and negotiators are willing to budge their positions in small variants during actual negotiations. Table 5 depicts how a DSS used in conjunction with a mediator and access to the WWW speeds up the mediation process.

1. Negotiation Support Without a Computer

Typically, a negotiation process begins with a difference of opinion between two parties in either the value, goals, or solution phase. The conflict takes place because the two negotiating parties seek to gain the upper hand in bargaining, if necessary, at the expense of the other party. This can best be illustrated by using a real-life labor negotiation. The labor union of a middle-size factory which produces electronic components is seeking new terms and conditions in the labor contract with management. As a result of multiple meetings between the labor union committee and its members, the labor union (Party A) has initiated a request to company management. Three salient aspects have been identified: salary increase (5% increase), duration of labor contract (maintain the existing two-year length), and duration of vacation (maintain the four-week condition).

The problem triggered by the labor union has forced management (Party B) to take a position. They have studied the three issues addressed by the union and have informed the latter that they are willing to engage in negotiation if the union is willing to consider productivity as part of the negotiation. In fact, management has recently discovered that increasing the quality of the products while reducing some production costs is the only solution to surviving fierce competition in a global market. The four issues then form the first collective goal space.

Table 5. Model of Negotiation
For Two Parties Negotiating a Pay Raise/Benefits

	Decision Environment		
Phases	**No computer**	**Computer avail. W/Internet**	**Computer W/Internet Future**
Values: **- Beliefs regarding modes of conduct** **- Maslow's (1954) hierarchy involving safety, love, self-esteem, etc.**	- Television and friends shape the beliefs of union members - The desires to provide a better life for our loved ones can cause business dispute - Media presents an image of corporations being the adversary	- Union members can read news group articles, lookup company's financial statement to derive their own beliefs regarding a strike or walkout - Search WWW for articles about the possible strike, this will provide an unbiased prospective - The WWW can provide the union members with a clearer image of the companies financial situation by looking up the companies financial reports.	- Union members can search the WWW for new group articles, pertaining to the strike and also check both the unions and companies homepage. - The home page will provide union members with current information to assist in shaping individual values - On line newspaper and magazines can assist in giving union members unbiased information concerning the strike

Table 5 (Continued)

	Decision Environment		
Phases	**No computer**	**Computer avail. W/Internet**	**Computer W/Internet Future**
Goals: **- Broadly stated desires** **- Criteria for evaluating the effectiveness of solution** **- Performance measures**	- Union goals are derived by what is perceived - Manually take a vote to determine what members want out of negotiations - During meetings discuss individual preferences/desires and alternatives	- Union members can vote on proposed solutions and have current information concerning the strike - Employee union members can update desires or grievances via the unions' homepage - By actually polling all factory employees, it may become apparent that a strike is not in the best interest of all employees but a select few	- Union members can vote on grievances online and view up to date computed information from all members who have voted - The analogy two heads are better than one come to mind as all members can enter opinions or own solutions via the net - Management will be able to view the employees desires and opinions

Table 5 (Continued)

	Decision Environment		
Phases	**No computer**	**Computer avail. W/Internet**	**Computer W/Internet Future**
Solutions: **- Decisions, actions or measures taken to achieve stated desires** **- Achieving common solution**	- Utilize neutrality of a third party, a mediator that can negotiate on behalf of both parties and establish a consensual compromise that both parties can accept - If unsuccessful redefine arbitrating parties value/goals and renegotiate	- NSS can be used throughout give and take mediation and can present a clearer analytical representation to the parties present at negotiations - NSS can provide solutions and alternatives - The issue in content can be viewed by union members and management via computer screens in an auditorium for feedback	- An NSS on the WWW can be used by negotiating parties as a tool to assist mediation via the net from non threatening environment - Union members can log onto the NSS and view real time negotiations taking place - When management makes a counter offer, members can immediately vote on the issue to speed up in achieving a common solution

Management proposes a freeze in pay, a six-month labor contract with a three-week annual vacation, and a productivity increase of at least 8%. Reacting to the proposal, the union revises its starting position. Based on these starting positions, the two parties begin to analyze the problem.

By entering this bargaining session with the perception of diametrical opposed goals, it creates the feeling of bias. This bias has been shown to generate more hostility and mistrust between parties and diminish the number of suitable solutions generated (Bazerman, 1983). Both sides feel betrayed by the other side or feel that the opposing viewpoint is incorrect. In the case of the employees' union, union members feel betrayed and angry at management for not giving in to the demands. The labor union seeks stability for its members by bringing about a two-year labor contract. By doing so, the employees might feel more secure and create a less adversarial environment between employees and management. The company sympathizes with the employees but the bottom line is profit. The company is barely operating out of the red. Negotiating parties who feel wronged experience feelings of disapproval, blame, anger, and resentment. This can escalate into sanctions to enforce conformity or punish the other party, easily resulting in suboptimal agreements or deadlock (Thomas and Pondy, 1977). In our example, the negotiations have reach a deadlock.

Negotiating a settlement in this type of situation is extremely sensitive and difficult. A neutral third party must be brought in to mediate the dispute. This person must be tactful and skilled at public relations. To be successful, the parties must perceive the mediator as impartial and fair. The mediator attempts to resolve the parties' relations by changing or evolving their values and goals. If one side refuses to negotiate, the mediator attempts to refine the negotiation process until both parties accept the outcome or break off negotiation.

2. Negotiation Support with Computer Access and Internet

Negotiation Support Systems (NSS) are computer-based programs that can function as or assist mediators throughout the negotiation process.

> Research on NSS has primarily focused on two key technological aspects: (1) group decision and/or conflict resolution models to help negotiator reduce discord and increase the chance of reaching consensus, and (2) providing rich communications media to enhance communication exchange between antagonists (Bui and Shakun, 1996).

Disciplines such as operations research, management, artificial intelligence and economics all contribute in the derivation and formulation of NSS models. These models and algorithms can be replicated for use on a computer, and assist the mediators and negotiators by providing interactive information processed in a systematic way. By imposing an orderly level of structure in defining the problem, it may help the negotiating parties better appreciate the other party's reasons for their position on specific points. It is not unusual that negotiating parties define the wrong problem (Shakun, 1992).

In the example of the employees union negotiation with the company, the example ended in a stalemate between labor and management. A computer-based NSS may have averted a walkout by assisting the parties in identifying the most contentious points and working through each issue by assigning weights and values. As long as and the two parties can see progress, the chances of discontinuation are less likely. During negotiations, moving from one point to the next without a hiatus is crucial. During a hiatus, motivation for a quick settlement diminishes and problems and issues that were once close to settlement are forgotten.

Using the same scenario as before, management, this time suggests the use of NSS. Labor agrees. The issues are defined and initial offers are made. Both parties now assign relative weights to each issue and define the ranges of values for all the issues identified. The computer then processes the information entered and suggests alternatives and may even suggest restructuring for noncooperative issues. The NSS can provide both parties with simultaneous displays and printouts of utility graphs, negotiation results, and spreadsheets to help both parties achieve an equitable negotiation.

3. Negotiation Support with Computer Access and Internet Future

In the future, negotiations take place on the WWW using NSS modeled to meet the arbitrators needs. The concept for migrating NSS to the WWW is simple. Negotiation Support Systems (NSS) are interactive, computer-based tools for use by negotiating parties in reaching an agreement. Negotiating tools should be user friendly applications designed to assist the decision makers or negotiators in the process of problem evaluation and resolution. For a negotiation software to be effective, the NSS should be customized to accurately reflect the individual needs. This customization should include behavioral characteristics, cognitive perspective of negotiators, determination of each party's real interests, and generation of options for mutual gain. This information will be taken into consideration in the final analysis of alternatives and solutions.

An NSS is composed of a database, model base, search array and a user friendly interface. To make the negotiation effective, the process of evaluating weighted alternatives must utilize a means by which several issues of contention can be considered at the same time.

In the fast paced ever-changing marketplace of today, a decision support process capable of using the expertise within a company and determining the best course of action is critical. By accessing and executing an NSS session on the World Wide Web, both negotiating parties will have a multi-platform means by which the broad-breadth of expertise within the company can be used to help find a quick solution and end the dispute.

The employees' union and company management using a NSS and the Internet of the future could negotiate from anywhere in the world. Once both parties are online, the multimedia video-teleconferencing capabilities can be turned on. Information entered is displayed instantly via networks to the constituents of both parties. The NSS navigates negotiating parties through a question and answer session where the NSS customizes the session's model for the negotiating parties. During the question and answer session, the NSS will focus both parties on asymmetries or lack of balance in achieving a final resolution that yields to each party those issues that are most important to it.

In Chapter IV, Negotiator/I will be used to demonstrate how NSS on the WWW can enhance the negotiation process. Negotiator/I can be used as a basis for evolutive exploration of new, and hopefully, better solutions. The numbers of issues, issue weights, and utility values can be refined or modified until new and more satisfactory solutions can be found.

III. COMPUTING ON THE INTERNET

A. TECHNOLOGIES FOR CREATING DYNAMIC WEB-SITES

The emergence and popularity of the World Wide Web, in both Internet and Intranet environments, has led to new developments for extending the capabilities of Web servers. Users flocked to the web in droves connecting through workplaces, universities, and commercial Internet access providers. Software has proliferated: browsers, email packages, chat programs, HTML editors, multimedia tools. Entirely new positions have been created: webpage designers and webmasters. Ideas about connectivity and information flow have expanded. Standard protocols and languages (TCP/IP, HTTP, HTML) allow any user anywhere in the world to make static information available to any other user anywhere in the world without regard for hardware, software, or operating system. Demands for static information have transformed into demands for interactive information and an open standard to provide that information. This demand for information creates a market for DSS technology on the WWW. By migrating to the WWW, decision support technology becomes available to everyone. Of particular interest is decision support technology in the form of an NSS on the WWW. Currently three software tools available to webmasters that allow the conversion of a static webpage to a dynamic webpage containing interactive information.

These tools are: Common Gateway Interface (CGI), Sun Microsystems Java, and Microsofts Active-X. Both Java and Active-X provides webmasters with tools to integrate and create a dynamic data interface that is executed away from the Web server. CGI's execution domain is the server where it resides. These programs give webmasters the capability to create dynamic data exchange as well as provide static text and graphics on websites.

B. SERVER SIDE COMPUTING USING COMMON GATEWAY INTERFACE (CGI)

Common Gateway Interface is a standard specification that allows Web servers to run outside applications and to send the external information back to the browser. CGI programs can interact with the user's browser by accepting data and transferring this information to other resources such as a database or a decision support system. The easiest way to visualize a CGI application is to think if it as a simple client/server standard for setting up communications between a web browser and server. In order for CGI to work, a CGI program must be written to execute on the Web server. Once the CGI program executes, it becomes possible for a user via a browser to initiate and pass information to a server or server-based program. Results are parsed into HTML and sent back to the client.

A common example: A prospective stock investor wishes to be added to Morningstar's select few mailing list. All an investor needs is a computer, Web browser and access to the Internet. An investor would access Morningstar's webpage and fill out the HTML form. Once the form is completed, the investor hits the submit button and sends the HTML form to a server-based CGI program as a parameter list. The CGI programs processes the parameter list and provides feedback to the user in the form of a dynamically created HTML page containing a message stating that the request has been processed successfully. Figure 1 graphically displays the CGI process.

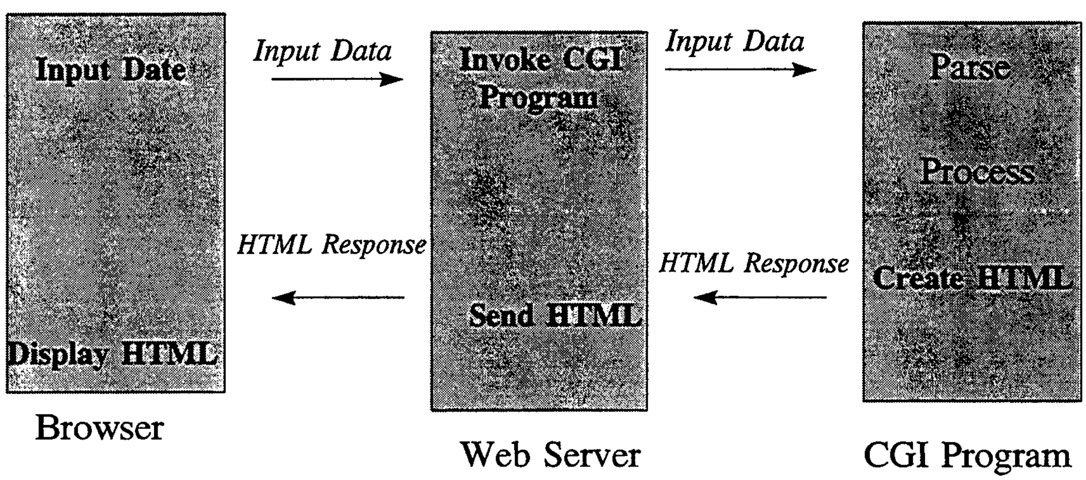

Figure 1. The CGI Process

The highly flexible CGI standard is supported by virtually every Web server and browser. This type of interactive power makes it possible to bring platform-independent interactive applications to just about anyone with a browser. CGI script is appealing to webmasters since it can be written in almost any language native to the Web server's operating system. Some languages that work on a web server are: Visual Basic, PERL, C, C++, Pascal and Fortran. All these features make CGI very appealing to the webmaster, but there are some drawbacks. First, CGI applications requiring complex computations and data exchanges are notorious for poor performance. The reason for its poor performance is that all CGI program processing takes place on the server. In other words, to give the user any type of HTML feedback, a round trip must be made between the server and the client. Also, since CGI programs are executed on the server, it is equivalent to letting anyone run a program on your system. Using the previous example, suppose the investor who inadvertently enters his phone number in the e-mail address field. Before the data can be checked for validity, the information collected by the form must be transmitted to the server-based CGI program. This is the only time information can be checked for validity. Once an error is found, the CGI program transmits an HTML page containing the appropriate message back to the investor. When comparing this type of client/server relationship to more efficient architectures such as Java and ActiveX, the round-trip nature of CGI makes it slower than disconnected type applications. Additionally, since all processing takes place on the server, the server's CPU cycles are consumed while the client CPU cycles are idle.

C. CLIENT SIDE COMPUTING JAVA VERSUS ACTIVEX

One of the most significant developments on the Internet has been the introduction of Sun's Java programming language. Java technology brings cross-platform information sharing and new forms of motion and interactivity to the Internet. Sparked by tremendous media hype in 1995, the Java movement spread and caused Microsoft to offer a competitive technology called ActiveX. It also spawned the idea of the network computer, or "thin client" that would enable users to pull applications as needed off the Internet. Before comparing and contrasting the two technologies for developing interactive Web site, some terminology must be introduced.

Java is a programming language while ActiveX is a component-level technology, a tool for building applications from reusable parts. ActiveX controls are similar to plug-ins and Java applets, small programs that can be sent over the Internet to extend the capability of your browser. "Controls" are ActiveX objects that provide interactive or user-controllable function from within another program or container. ActiveX "documents" provide the ability to view documents, such Excel or Word files, from within a Web browser or other container. Finally, "ActiveX Scripting" can be used to integrate and control the behavior of several ActiveX controls and Java applets on a Web page.

Java was created from the start to be a platform-independent language. Java code will run on any 32-bit operating system (including Windows, Macintosh and Unix systems) that has a Java enhanced browser. The Java browser interprets Java code and acts on its commands. Because Java is an interpreted language, it runs slower than ActiveX code. ActiveX code is pre-compiled, and as a result runs faster than Java. For this reason, ActiveX programs must be compiled for the specific operating system on which it is intended to run. If you want to run your ActiveX program under Windows, Macintosh and Unix, you will have to customize and compile your code three times. Java's interpreted code can be written once and trusted to work on any platform. In the middle of this, is the developer who must decide today how to develop the products of tomorrow. ActiveX does provide some advantages in software development. Although when viewed from the broader perspective of the entire software life cycle, Java emerges as a better choice. The issues a developer must address before proceeding with a project can demonstrate Java's superiority. These issues include users' platforms, execution speed, maintenance, security, and speed of development.

1. Users' Platform(s)

This question is irrelevant when Java is under consideration. Java's primary strength derives from its platform-independence. Applets will run in any Java-capable browser including Netscape's Navigator and Microsoft's Internet Explorer. Java-capable browsers exist for Macintosh, Windows, Solaris, and UNIX operating systems. IBM, Hewlett-Packard, Apple, Silicon Graphics, and even, Microsoft has licensed Java and announced intentions to incorporate Java into the future versions of their respective operating systems. This will provide a wide base for Java standalone applications as well as for the applets within browsers.

Unfortunately, this question becomes all too relevant in regard to ActiveX. ActiveX is compiled code. Compilation must be done for a specific environment's binary code. All portability is lost upon compilation. A further drawback is that currently ActiveX can only be compiled for the 32 bit Windows operating system. Microsoft has announced efforts to port ActiveX to UNIX, Macintosh, and the 16 bit Windows systems. Which version of UNIX will be the target has not been specified. An additional concern is that Netscape Navigator, the dominant browser on the market, does not nor plans to support ActiveX natively. A plug-in must be used with Navigator.

ActiveX controls, once downloaded, remain resident on the client's hard drive until specifically deleted. This requirement for permanent storage makes ActiveX an inappropriate choice if a project is being designed for thin clients or the new network PC's (NC) which will have no permanent memory. Java applets are not cached and, as such, are ideal for the NC environment.

> Instead of connecting to a process or an API from the Web server, users down load Java applets embedded in HTML documents for local execution inside of Java-enabled browsers. The running applet is able to link back to a database server existing anywhere on the Internet or intranet (Linthicum, 1996).

If ActiveX is successfully ported to other platforms, ActiveX controls must become platform-aware since functionality differs between operating systems. As an example, True Type fonts available on Windows machines are not available on UNIX machines. The other option is to write customized source code for each operating system. Either way, the maintenance has grown in complexity.

2. Execution Speed of Final Product

Java was designed for the Internet. As such, the language was constructed to minimize overhead needed by applets. Because the code is not fully compiled when downloaded, Java takes a performance hit relative to ActiveX. Java is translated from source code to byte codes at compile time. At execution, the byte codes finish the compilation process for a specific machine. This causes the performance hit as well as the fact that the applet must be downloaded on each subsequent visit since no caching is done on applets.

ActiveX rightly claims to have a performance advantage over Java since it is compiled in native code. As long as the users will be operating in the Windows environment, ActiveX offers distinct improvements in speed as well as the ability to leverage the desktop functionality of Windows. Whether this execution speed advantage holds in other environments remains to be seen.

3. Resources to Maintain Product

The platform independence of Java lends itself well to lowering maintenance costs. Because the source code needs only to be written and compiled once, maintaining the code becomes significantly easier compared to traditional source code. Additionally, since applets are not cached on client machines, version control is easier. Replacing the original applet with the revised applet makes the most current version instantly available. No problems with outdated versions floating around the Internet occur since the originals exist only at the server.

Even if Microsoft is successful in porting ActiveX to other platforms, different versions of the same program will need to be produced and maintained for each operating system targeted by the software. Additional costs could be incurred if Microsoft individually charges for each ActiveX compiler ported to other operating environments. Version control becomes much more difficult as separate versions of the product need to be maintained for each operating system. Further complicating version control is the fact that a control is downloaded and cached to a client machine.

Any ActiveX control downloaded to a client machine remains resident on that machine and can be called by other ActiveX control. A single ActiveX control may exist in a thousand different computers. Replacing these copies with a revised control would be a hit-or-miss affair, completely reliant upon the users revisiting the homepage from where the control was originally placed. The end result is that a development team would be faced with the dismal task of maintaining several versions of the same product.

4. Security

Security features are built into Java applets. Because the code is transferred over a network to the client machine, the code is assumed to be untrusted. As such, client systems must be protected. Increased security for the user is achieved by limiting the functions an applet may invoke. As an example, applets may not read, write or otherwise access the local file system protecting the client system from viruses and Trojan horses. Another security feature is implemented in the byte-code verification process. When the class is loaded, the byte-code verification process checks the code to ensure pointers and stacks are not manipulated to gain access to the underlying machine.

ActiveX does not implement any built-in security features to protect users. Instead, ActiveX implements security externally to the code through the use of digital signatures on components. When a new ActiveX component is encountered, a certificate is presented to the user who may accept it, permitting the download to continue. A digital signature guarantees that the code is from the person signing that particular certificate and it has not been altered. ActiveX controls, once downloaded, operate with no restrictions on the client's machine. Any functionality of the operating system may be accessed. This accessibility leaves the door open for Trojan horses, viruses, or general snooping.

Several drawbacks with this approach are seen. First, with recent and ongoing debates concerning privacy on the Internet, opening yet another avenue by which outsiders can gain access to potentially private information does not appear to be prudent. Second, this approach assumes that the user has not disabled the security feature of Microsoft's Internet Explorer. No such security feature exists in Netscape's Navigator. Being a graphic image, the certificates can be slow to load on the lower-end machines. Lastly, digital signatures are costly, i.e., $400 annually. Besides increasing costs for maintenance of the product, the fee restricts small-time developers and experimenters from making code available to the general public.

5. Speed of Development

Given that Java is less than two years old, the cadre of experienced Java programmers is small. For most organizations considering Java, expertise will have to be developed in-house. Programmers familiar with C++ will have a shorter learning curve than other programmers since Java was based on C++. The learning curve is being dramatically reduced as numerous Java development tools are being marketed. Symantec's Cafe and Visual Cafe, Sun's JavaWorkshop, as well as Microsoft's own Visual J++, and many others too numerous to mention simplify much of the development process by incorporating drag-and-drop GUI builders and advanced debuggers. Repositories of Java classes available for free grow daily.

The learning curve of using ActiveX is much shorter than Java for a variety of reasons. ActiveX is based upon the Windows OLE technology which has been around for eight years. With this longevity comes a large pool of programmers familiar with Windows programming. Additionally, ActiveX controls are not written in a new language but are written in other languages such as Visual Basic, C++, or Delphi. There are well over 1,000 ActiveX controls for Windows currently available to developers.

There are some concerns, however, with regard to ActiveX controls. First, some of the controls available are OCX controls which are the predecessor of ActiveX technology. OCX controls can run in the ActiveX framework but are less efficient causing a performance hit. Second, OCX controls were not developed for a distributed object-oriented environment but for a stand-alone PC with functional programming standard, e.g., goto's are used. It is questionable whether it is wise to continue using such code.

Returning to the developer who must decide how to proceed, the choices are ActiveX or Java. If the project is for a local Windows network, ActiveX offers distinct advantages over Java. On the other hand, if the project is for a multiplatform environment, then Java is the better choice. When considering Internet application development, it is best to remember a quotation from JavaSoft CEO, Alan Baratz, "ActiveX = (Java + viruses + memory leaks) - Win32."

IV. NEGOTIATOR/I: CREATION OF AN INTERACTIVE NEGOTIATION

A. FRAMEWORK FOR NEGOTIATOR/I

Computer support can be used to assist negotiators in interactive information elicitation and process the information in a timely manner. The framework for designing Negotiator/I is based on the Type IV model of GDSS defined by Bui (1987). All parties can have their own DSS that contains models customized to their needs that individually describe the issues. This arrangement permits the negotiators to engage in a joint and open modeling effort. In practice, technical experts and advisors usually supply the bulk of information to the negotiators either before or during the negotiation process. Even if such information is accurate and complete, there is no reason why the negotiators themselves could not exercise their freedom of choice at the time of negotiations through joint concession and experimentation of simpler models of their own.

Nyhart and Samarasan (1987) contend that this can help negotiators appreciate better the strengths and weaknesses of the other party's position and arguments. A joint and open modeling effort may be to the advantage of all involved parties.

The NSS model-base should provide an interactive process and comprehensive framework which allows parties to concentrate on joint problem-solving rather than on convoluted argument. The objectives of using a NSS are listed below:

- Establish a consensual database as a foundation for negotiation,
- Evaluate the impact of perceived uncertainty,
- Provide communication links for bargaining and discussion,
- Suggest restructuring of non-cooperative issues, and
- Help search for agreements through Pareto-optimization.

Not only can computer support assist the negotiators in interactive information and process it in an orderly manner, but also update data as inputs are entered. This information can then be tabulated and presented in both tabularly and pictorially formats. It should also provide a tool to let negotiators know that their compromise or concessions can be implemented and will produce the desired and agreed-upon results.

It is unreasonable to expect negotiating parties to rely solely on a computer based decision support system for defining an ultimate solution. The NSS is such a decision aid, and its intended purpose is limited to assisting negotiators explore alternatives. By managing data, facilitating information exchange, and applying effective models, the NSS is expected to merely improve the bargaining process - not replace it. It is hoped that through applying the NSS technology, negotiations can move from the fixed-pie scenario of distributive bargaining and closer to the win-win situation achievable through integrative bargaining.

B. NEGOTIATOR/I: AN INTERACTIVE PROCEDURE OF BILATERAL NEGOTIATION

The evolutive approach to designing NSS can be illustrated by the implementation of Negotiator/I, an NSS created in Java, installed on a web-server, and designed to run on the WWW. Using a multi-attribute utility model, Negotiator/I allows negotiating parties to evolve through the three steps in Table 5. The speed at which a client can access Bilateral NSS is critical. Because two parties are accessing the web server concurrently, it is important to provide a near real time feedback. The speed at which information passes from client to server and vice-versa using Java applets, along with its security features, combine to make Java the language of choice for migrating bilateral NSS to the World Wide Web.

Each party can have its own computer support environment that contains models customized to its need. The environment describes the issues in which Negotiator/I allows the negotiators to engage in a joint and open modeling effort.

In practice, technical experts and advisors supply the bulk of the information to the negotiators either before or during the negotiation process. Even if such information is accurate and complete, there is no reason why the negotiators themselves could not exercise their freedom of choice at the time of negotiation through joint concession and experimentation with simpler models of their own. Negotiator/I allows negotiating parties to navigate dynamically through the relations in Value, Goals and Solutions in Table 5. The Evolutionary System Design framework is realized by helping negotiators focus on asymmetries of interests between the parties, so that the terms of the final treaty are better for both (Barclay and Peterson, 1976). A good treaty is one that yields to each party those issues which are more important to it. Thus, the two parties should try to push the negotiation toward the Pareto optimum by capitalizing on asymmetries of interest, and whenever possible, by redefining the situation to reveal more asymmetries. A treaty is Pareto optimum when it is not possible to increase the utility of one party without the decreasing utility of the other (Bui, 1990).

The negotiation of the procedure is described below:

Step 1. Identity values and goal variables in Table 5 which are associated with the major agreements that the parties seek to sign.

Step 2. For each of the agreements being considered, identify a common set of major issues about which the parties may disagree.

Step 3. Each party assigns relative weights to each of the issues.

Step 4. Define the range of values for all the issues as identified by both parties. As the parties enter the negotiations, they offer their initial positions with regard to each of the issues enumerated.

Step 5. For each party, determine individual-issues, weighted-utility curves. The determination is made by taking the product of the utility values and the respective relative weights of the issues.

Step 6. For each issue, compute joint utilities by aggregating the weighted-utility functions of the parties. The aggregation could theoretically take any mathematical form. The simplest form is additive. For each issue, choose the term that corresponds to the highest point of the joint utility curve.

Step 7. Based on the terms of the issues suggested in Step 6, determine the total utility for each party across all the issues.

Step 8. Search for improvements and restructuring. The concept of joint utility allows for the possibility to check for noncoopertive issues and suggests restructuring. A cooperative situation is one in which the highest value of the joint utility curve is higher than the individual maximum utility values of both parties. Conversely, a noncoopertive situations the one in which the highest value of the joint utility curve corresponds to the highest for only on of the parties, leading to unbalanced treaties. In this circumstance, it is recommenced that the single noncooperative issues be split (restructured) into subset of more cooperative (asymmetrical) issues.

As illustrated by the example in Figure 2, Negotiator/I is designed to support the improvement and restructuring process. It provides the user with simultaneous displays and printouts of utility graphs, negotiation results in tabular forms, and a spreadsheet to perform sensitivity analysis on the on the data suggested by Negotiator/I or the modifications requested by different parties. Under a multi-tasking environment, multiple sessions of Negotiator/I can be run, allowing users to conduct parallel bargaining.

C. AN EXAMPLE USING NEGOTIATOR/I

Figure 2 illustrates a two-party bilateral negotiation via the Internet using Negotiator/I. The negotiating parties are conducting negotiations from two different

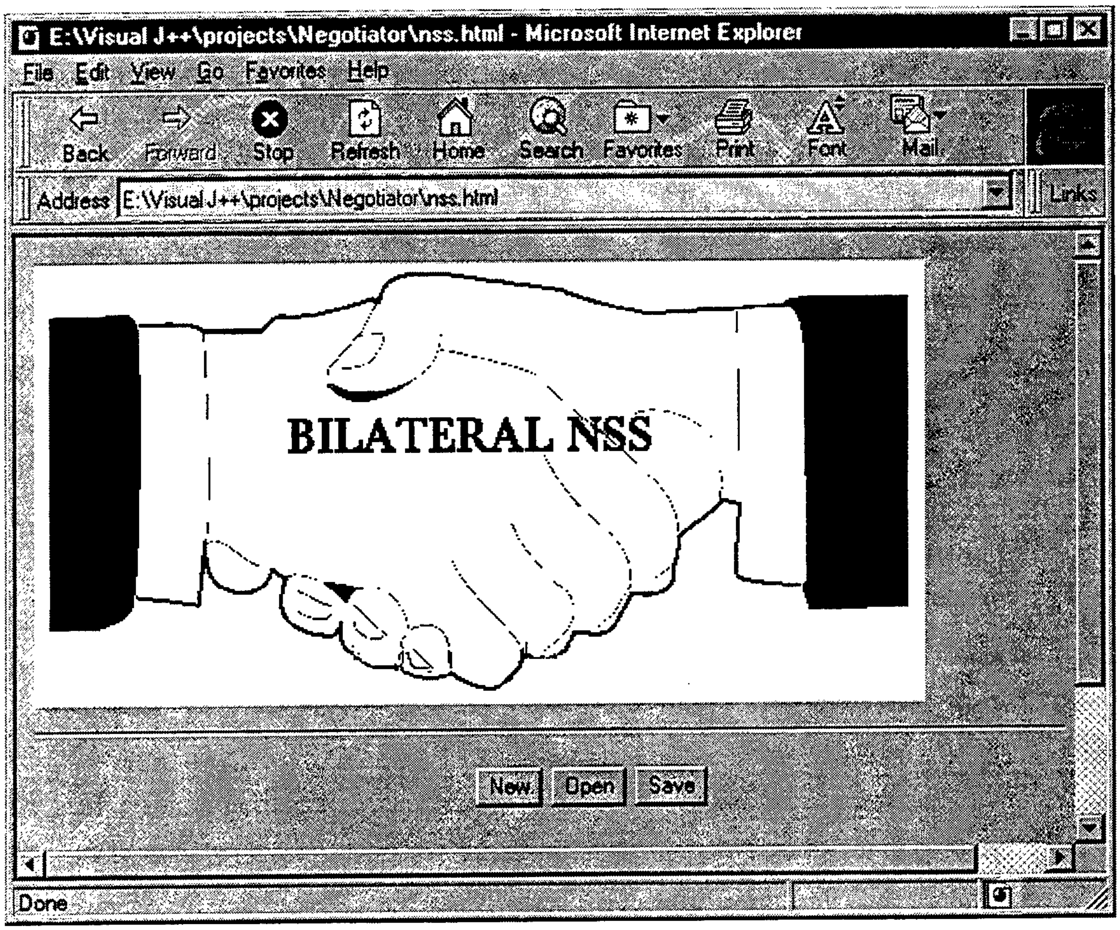

Figure 2. Internet-Based Negotiator/I

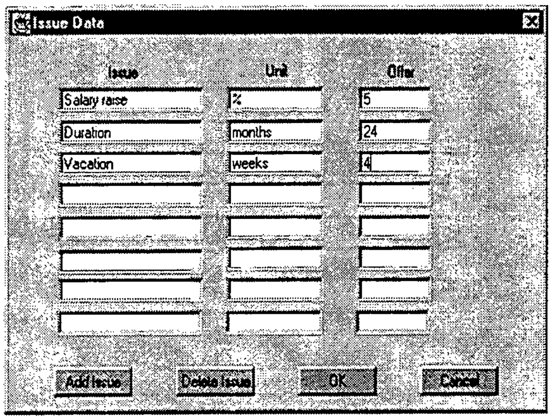

Figure 2a.

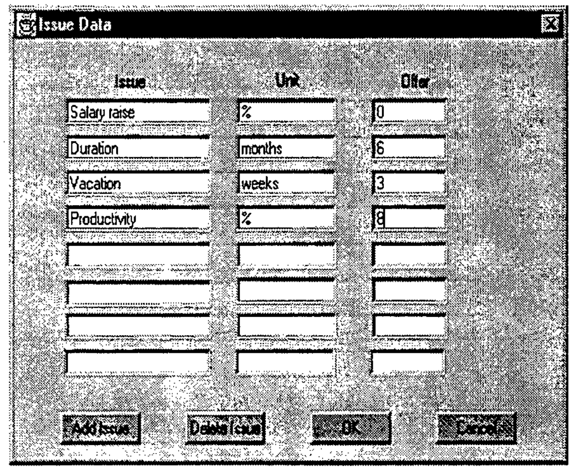

Figure 2b.

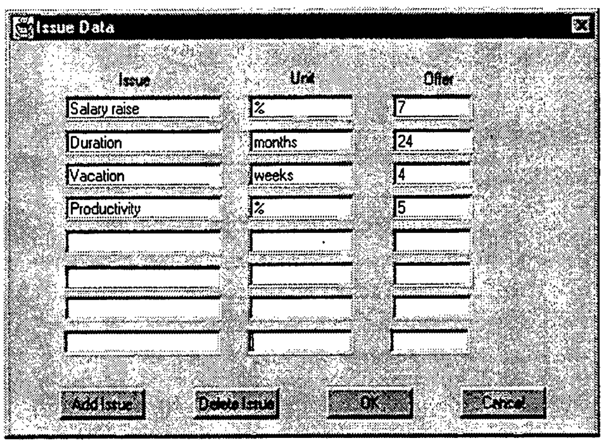

Figure 2c.

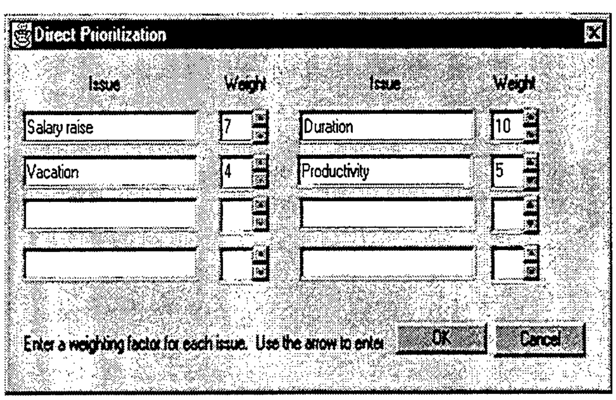

Figure 2d.

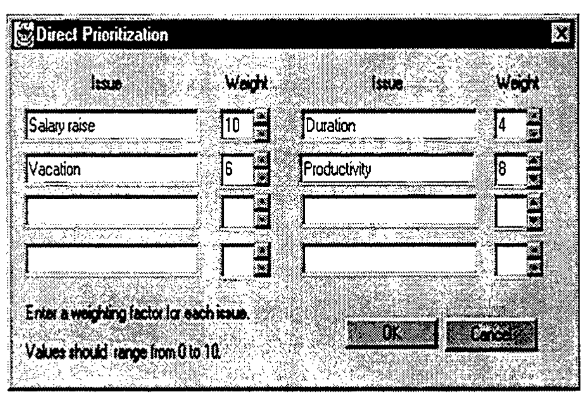

Figure 2e.

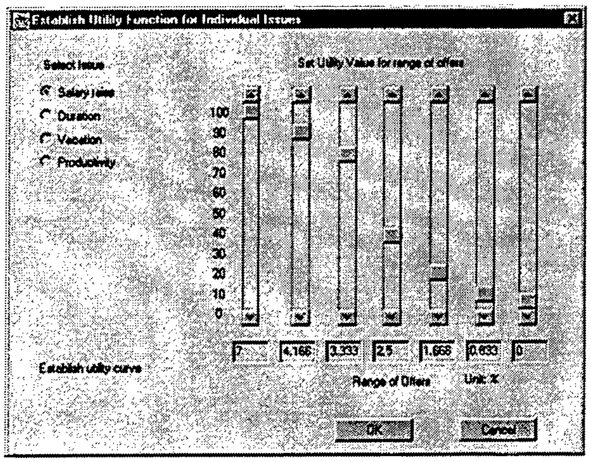

Figure 2f.

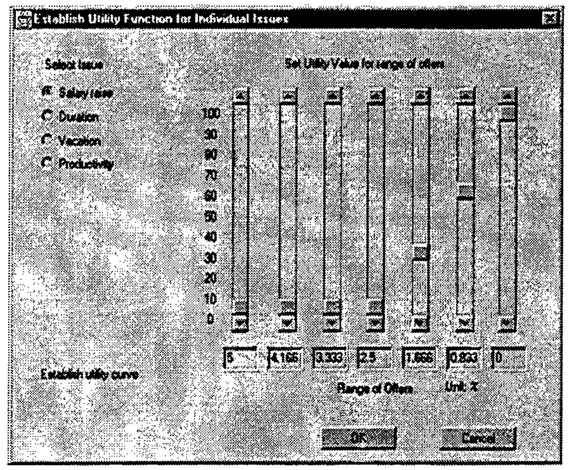

Figure 2g.

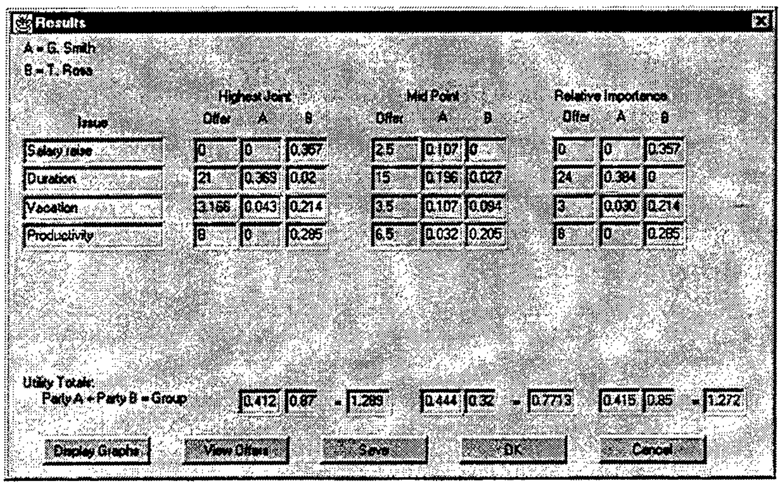

Figure 2h.

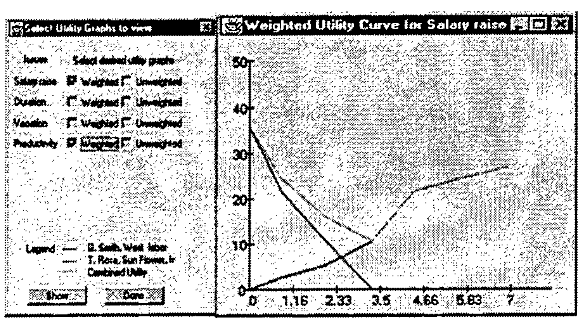

Figure 2i.

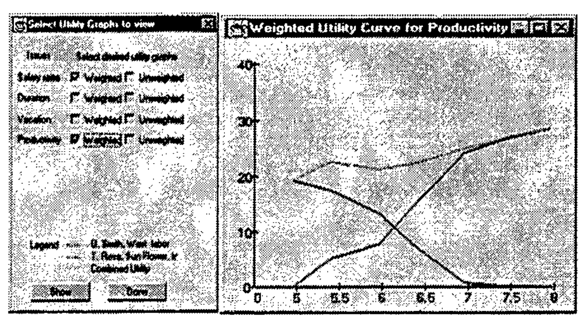

Figure 2j.

locations. Labor union representatives are located in Albuquerque, New Mexico while their management counterparts are situated in San Jose, California. The web server providing Negotiator/I is located in Monterey, California. The labor union of a middle-size factory which produces electronic components is seeking new terms and conditions of their labor contract with the management counterpart. As a result of multiple meetings between the labor union committee and its members, the labor union (Party A) has initiated a request to the company management. Three salient aspects have been identified: salary increase (5% increase), duration of labor contract (maintain the existing two-year length), and duration of vacation (maintain the four-week condition).

The problem triggered by the labor union has forced Management (Party B) to take a position. They have studied the three issues addressed by the union and have informed the latter that they are willing to engage in negotiation if the union is willing to consider productivity as part of the negotiation. In fact, management has recently discovered that increasing the quality of their products while reducing some production costs would be their only approach to surviving fierce competition in a global market (Figure 2b). The four issues then form the first collective goal space. As such, the goal in NEGOTIATOR/I can be viewed as an aggregation of the spaces of the two negotiation parties. Note that in this negotiation, the goal space is also the control solution space.

The management proposes a freeze in pay, a six-month labor contract with a three-week annual vacation, and requests that productivity be increased by at least 8%. Reacting to the proposal, the union has revised its starting position (Figure 2c). Based on these starting positions, the two parties have begun to analyze the problem. Figure 2d to 2g, respectively, show the parties' weights on issues, and a sample of their utility curves for the issues of salary raise.

Three solutions proposed are: (1) highest joint utility, (2) midpoint solution, and (3) relative importance (Figure 2h). The first proposed solution yields the highest possible joint utility, i.e., 1.29 points total for both parties. Another solution is based on the midpoint principle that yields a joint utility of .77. As its name suggests, the midpoint principle is one that finds the solutions by equally splitting the terms requested by the negotiators. For example, the midpoint principle suggests that the term of duration for the contract is 15 months (Figure 2h). Fifteen months is the midpoint of the management's six-month proposition (Figure 2b) and the union's 24-month proposition (Figure 2c). The third solution is based on the concept of relative importance, which gives each party what it wants on those issues for which its relative importance is larger than that of the other party. The relative importance concept suggests a solution whose terms yield a joint utility of 1.27.

The solutions suggested by Negotiator/I in (Figure 2h) are, however, only a basis for evolutive exploration of new, and, hopefully, better solutions. The numbers of issues, issue weights, and utility values can be refined or modified until new and more satisfactory solutions can be found.

In fact, the highest joint utility solution proposed by the NSS at the first round of negotiation was not well received by the union. Although the solution proposed four-week vacation, which the union wanted, no salary raise was recommended. Furthermore, the management seemed to come out winner for a total utility of .87 versus .41. This discontent was further substantiated by a close examination of the issues utilities. While both parties seemed to have found a compromise on productivity (cooperative issues, as shown in (Figure 2j), with a joint utility curve of convex shape), the salary issue (Figure 2i) clearly went in favor of the management. In return, the labor union obtained almost what it wanted for the duration of the contract.

V. SUMMARY AND SUGGESTIONS FOR FUTURE RESEARCH

A. SUMMARY AND FINDINGS

We explore the framework for decision support in an Internet-based environment using four decision making models. The models examine how use of the Internet can enhance the decision making process. The decision making process is examined using three scenarios: non-electronic decision making, computer access with Internet support, and computer access with Internet-based DSS. Our work suggests that having Internet support stimulates creative thinking by providing alternatives, thus, likely enhancing the quality of the decision making process as well as the decision outcome. Our analysis can also be used to define specifications for creating a DSS for implementation on the WWW.

Using the requirements identified above, as well as the program specifications, several methodologies are examined for setting up an Internet-based decision support prototype. CGI was eliminated due to slow processing speed of the scripting, security issues on the server, and the drain on server resources. Active X was eliminated due to the limited browser support and the client resident nature of the components. Due to the platform independence and built-in security features of applets, Java was selected as the methodology to implement the prototype.

B. LESSONS LEARNED

The true impact of the Internet has been the delivery of timely information. As the Internet grows, so does the amount of static data available. Sorting and synthesizing this data is a formidable task that, in the past, has been accomplished by the user. In such an environment, information overload is a real issue. An Internet-based DSS provides a means to more efficiently process data.

Java has proved to be an ideal medium for developing the prototype but, as the language currently exists, it is not the answer for every DSS. NSS/I has small data requirements with all the data used being generated internal to the program. Many larger decision support systems must access large legacy databases and perform complex mathematical computations. Java's capabilities in both of these areas are greatly limited. To do either would require the use of an outside scripting language to access the database or a mathematical solution software package.

Not all software developmental tools are created equal. Several software developmental tools exist for Java: Java Workshop, Cafe, Cafe-lite, Visual Cafe, Jamba, and J++. Our initial choice was Java Workshop. Only after expending money and a great deal of time did we discover that Java Workshop wraps its own unique classes around the Javas core libraries. These shadow classes as they were denoted by Java Workshop greatly complicated the development because the core libraries, such as java.awt and java.lang, could no longer be directly accessed. All work was scrapped, and the prototype begun again. Our second choice, Microsoft's J++ does not have this problem as it only uses Java's core libraries.

C. RECOMMENDATION FOR FUTURE RESEARCH

At present, Negotiator/I is designed as a bilateral NSS. The next phase is to expand Negotiator/I to support more than two antagonists.

Adding the additional capability of real time communication capability to Negotiator/I could further enhance negotiation process. This capability would have two aspects. The first relates to the simultaneous access by two clients to the data file of a particular negotiation session. A server applet would need to be written which accesses the data, monitors input from both clients, and mirrors any changes from one client to the other.

The second capability would build upon the first by adding an alternate communication channel similar to a chat session. This technology can eventually migrate to a video teleconference capability.

APPENDIX. NSS/I CLASS LIBRARY

public class **Applet_Negotiator** extends Applet {
//public Constructors
public **Applet_Negotiator()**
//Public Instance Variables
static Info_Table **myData**;
//Public Instance Methods
public String **getAppletInfo**()
public void **init**()
public boolean **handleEvent** (Event event)
public void **destroy**()
public void **paint**(Graphics g)
public void **start**()
public void **stop**()
boolean clicked**Open**()
//protected Instance methods
protected void **HandleButtons** (Object object)
}

public class **Components_DirectPriority** {
// Constructor
public **Components_DirectPriority** (Container parent)
//Public Instance Methods
public boolean **CreateControls**()
}

public class **Components_Graphs** {
// Constructor
public **Components_Graphs** (Container parent)
public boolean **CreateControls**(Info_Table data)
}

public class **Components_InitialOffers** {
// Constructor
public **Components_InitialOffers** (Container parent)
//public Instance Methods
public boolean **CreateControls**(Info_Table data)

```
}

public class Components_Issue {
        // Constructor
        public Components_Issue (Container parent)
        //Public Instance Methods
        public boolean CreateControls()
}


public class Components_MenuScreen {
        // Constructor
        public Components_MenuScreen (Container parent)
        //Public Instance Methods
        public boolean CreateControls()
}

public class Components_MessageBox {
        // Constructor
        public Components_MessageBox (Container parent)
        //Public Instance Methods
        public boolean CreateControls()
}

public class Components_Priority {
        // Constructor
        public Components_Priority (Container parent)
        //Public Instance Methods
        public boolean CreateControls()
}

public class Components_Results {
        // Constructor
        public Components_Results (Container parent)
        // Public Instance Methods
        public boolean CreateControls(Info_Table data)
}
```

```
public class Components_UserID {
        // Constructor
        public Components_UserID (Container parent)
        // Public Instance Methods
        public boolean CreateControls()
}

public class Components_Utility2 {
        // Constructor
        public Components_Utility2 (Container parent)
        // Public Instance Methods
        public boolean CreateControls()
}

class EventHandler_AddUser extends Dialog {
        // Constructor
        public EventHandler_AddUser (Frame incoming, Info_Table data, String
        title, int user)
        //Public Instance Methods
        public boolean handleEvent(Event event)
}

class EventHandler_DirectPriority extends Dialog {
        // Constructor
        public EventHandler_DirectPriority (Frame incoming, Info_Table indata)
        // Public Instance Methods
        public boolean handleEvent (Event event)
        // Protected Instance Methods
        protected void HandleButtons (Object label)
        boolean readWeights()
        boolean populateFields()
}

class EventHandler_Graphs extends Dialog {
        // Constructor
        public EventHandler_Graphs (Frame frame,Info_Table data,
                                     EventHandler_Results incall)
        // Public Instance Methods
```

```
        public void paint (Graphics g)
        public boolean handleEvent (Event event)
        private void handleButton(Object label)
}

class EventHandler_InitialOffers extends Dialog {
        // Constructor
        public EventHandler_InitialOffers (Frame incoming, EventHandler_Results
        caller, Info_Table indata)
        // Public Instance Methods
        public boolean handleEvent (Event event)
}

class EventHandler_IssueScreen extends Dialog {
        // Constructor
        public EventHandler_IssueScreen (Frame incoming, Info_Table inData)
        // Public Instance Methods
        public boolean action (Event event, Object object)
        public boolean handleEvent (Event event)
        void customizeScreen()
        int getIssueCount()
}

class EventHandler_MenuScreen extends Dialog {
        // Constructor
        public EventHandler_MenuScreen(Frame incomingFrame, Info_Table
        dataTable)
        // Public Instance Methods
        public boolean handleEvent(Event event)
        // Protected Instance Methods
        protected void HandleButtons (Object object)
}

class EventHandler_MessageBox extends Dialog {
        // Constructor
        public EventHandler_MessageBox (Frame incoming, String title, String
        message)
        // Public Instance Methods
```

```
        public boolean handleEvent(Event event)
}

class EventHandler_PriorityScreen extends Dialog {
        // Constructor
        public EventHandler_PriorityScreen (Frame incoming, Info_Table inData)
        // Public Instance Methods
        public boolean handleEvent (Event event)
        // Protected Instance Methods
        protected void HandleButtons (Object object)
}

public class EventHandler_Results extends Dialog {
        // Constructor
        public EventHandler_Results (Frame incoming, Info_Table inData)
        // Public Instance Methods
        public boolean handleEvent (Event event)
        // Protected Instance Methods
        protected void HandleButtons (Object label)
}

class EventHandler_UserID extends Dialog {
        // Constructor
        public EventHandler_UserID (Frame incoming, Info_Table data)
        // Public Instance Methods
        public boolean handleEvent (Event event)
}

class EventHandler_Utility extends Dialog {
        // Constructor
        public EventHandler_Utility (Frame incoming, Info_Table inData)
        // Public Instance Methods
        public boolean handleEvent(Event event)
        void setCheckLabels()
        void retrieveUtility()
        //Protected Instance Methods
        protected void HandleButtons (Object label)
}
```

class **Graph extends Frame** {

// Constructor

public **Graph** (Info_Table data, String title, int which Issue, Event Handler_Graphs call,
boolean weight)

// Public Instance Methods

public void **paint** (Graphics g)

public boolean **handleEvent**(Event event)

}

public class **Info_Table** {

// No Constructor

//Constants

public final static int **USER1** = 0;

public final static int **USER2** = 1;

public final static int **JOINT** = 0;

public final static int **MID** = 1;

public final static int **REL** = 2;

// Public Instance Methods

public boolean **setUserData**(int user_index, String name, String organization, String password)

String **getUserName**(int whichuser)

String **getOrganization**(int whichUser)

boolean **checkPassword** (String guess, int guesser)

void **setCurrentUser**(int whichUser)

void **setStructure** (int num_Issues)

boolean **isInitialized**()

void **setIssueTitles**(int index,String data)

String **getIssueTitles**(int index)

void **setIssueUnits**(int index, String data)

String **getIssueUnit**(int index)

void **setIssueWeight**(float inValues)

float **getIssueWeight**(int whichUser,int whichIssue)

void **setDirectWeight**(int i, float inValue)

void **setInitialOffer**(int issue, String data)

String **getInitialOffer**(int whichIssue)

String **getInitialOffer**(int whichUser,int whichIssue)

float **getOffer**(int whichIssue, int whichOffer)

void **setUtilityTable**(int whichIssue, int whichOffer, float data)

```
        float getUtility(int whichUser, int whichIssue, int whichOffer)
        float getUtility_Weighted(int whichUser, int whichIssue, int whichOffer)
        int getIndexIssues ()
        void computeResults()
        float getJointUtility(int whichUser, int whichIssue)
        float getJointOffer(int whichIssue)
        float getMaxJointUtility_Weighted(int whichIssue)
        float getMaxJointUtility_Unweighted(int whichIssue)
        float getTotalJointUtility_Weighted (int whichissue,int whichoffer)
        float getTotalJointUtility_Unweighted(int whichissue,int whichoffer)
        float getRelativeUtility(int whichUser, int whichIssue)
        float getRelativeOffer(int whichIssue)
        boolean sendToFile(String filename)
        boolean loadFile(String filename)
}

public class NewSession {
        // Constructor
        public NewSession (Container parent)
        // Public Instance Methods
        public boolean CreateControls()
}

class NewSessionScreen extends Dialog {
        // Constructor
        public NewSessionScreen(Frame incoming, Info_Table table)
        // Public Instance Methods
        public boolean action (Event event, Object object)
        public boolean handleEvent (Event event)
```

LIST OF REFERENCES

Bazerman, M., "Negotiator Judgement: A Critical Look at the Rationality Assumption," American Behavioral Scientist, Vol. 27, No.2, 1983.

Bhargava, H. K., and Krishnan, R., and Muller, R., "Decision Support on Demand: Emerging Electronic Market for Decision Technologies," dnet-dss, February 1996.

Bui, T., and M. Shakun, "Negotiation Processes, Evolutionary Systems Design, and Negotiator,"Journal of Group Decision and Negotiation, June 1996.

Bui, T., and Sivasankaran, T., "Fuzzy Preference in Bilateral Negotiations Support Systems," 24th Annual Hawaii International Conference on System Sciences, IEEE Computer Society Press, January 1991.

Coffee, P., "ActiveX vs. Java: Choose Your Weapon," Windows Magazine, January 1997.

Evaristo, E., and M. Eierman, "Creativity in Idea Generation: An Application to GDSS," IEEE Computer Society Press, 1993.

Flanagan, D., Java in a Nutshell, O'Reilly & Associates, Sebastopol, CA, 1996.

Gilley, Bruce H., "Group Decision Support Systems and Bilateral Negotiations," Master's Thesis, Naval Post Graduate School, Monterey, CA, March 1991.

Hamilton, Marc A., "Java and the Shift to Net-Centric Computing," IEEE Computer Society Press, 1996.

Lax, D. A., and K. Sebenius, The Manager as Negotiator: Bargaining for Cooperation and Competitive Gain, The Free Press, New York, NY, 1986.

Lemay, L., Teach Yourself Web Publishing with HTML in 14 Days, Sams.Net Publishing, Indianapolis, IN, 1996.

Lemay, L., and Perkins, C., Teach Yourself Java in 21 Days, Sams.Net Publishing, Indianapolis, IN, 1996.

Linthicum, David S., "Linking Web Servers with Live Data," PC Magazine, September 1996.

Murray, W., and Pappas, C., The Visual J++ Handbook, Academic Press, Inc., Orlando, FL, 1997.

Naughton, P., The Java Handbook, McGraw-Hill, Berkeley, CA, 1996.

Newell, A., and H. Simon, Human Problem Solving, Prentice-Hall, Englewood Cliff, NY, 1972.

Simon, H., The New Science of Management Decision, Harper & Row, New York, NY, 1960.

Turban, E., Decision Support and Expert Systems, Macmillian Publishing Company, New York, NY, 1988.

Varney, Sarah E., "Datawebs! Link the Web to Your Legacy Data An Apps," Datamation, April 1996.

Walnum, C., Using Java Workshop, Que Corporation, Indianapolis, IN, 1996.

Wehling, J., and Bharat, V., Late Night Advanced Java, Ziff Davis Press, Emeryville, CA, 1996.

Wood, C., Visual J++, Prima Publishing, Rocklin, CA, 1996.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Suite 0944
   Fort Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Prof. Tung X. Bui (Code SM/Bd) 1
   Naval Postgraduate School
   Monterey, CA 93943-5103

4. Prof. Balasubramaniam Ramesh (SM/Ra) 1
   Naval Postgraduate School
   Monterey, CA 93943-5103

5. Joseph G. Garcia 2
   219 Colmar Road
   Seaside, CA 93955

6. Kimberly S. Blood 2
   103 Malloway Lane
   Monterey, CA 93940