UCRL-CONF-200487

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

iES: An Intelligent Electronic Sales Platform)

William Korbe, Valerie Stanton, and Jerry Gao, Ph.D.

October 23-26, 2003

The Sixth International Conference on Electronic Commerce Research (ICECR-6), Dallas, TX, October 23-26, 2003

Disclaimer

Auspices Statement

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

iES: An Intelligent Electronic Sales Platform

William Korbe, Valerie Stanton, and Jerry Gao, Ph.D.
Computer Engineering Department
San Jose State University, San Jose, CA 95192-0180

Email:
william_korbe3@hotmail.com
stanton1@llnl.gov
jerrygao@email.sjsu.edu

Abstract

Current e-commerce systems support online shopping based on electronic product catalogs. The major issues associated with catalog-based commerce systems are: difficulty in distinguishing one retailer from another, complex navigation with confusing links, and a lack of personalized service. This paper reports an intelligent solution to address these issues. Our solution will provide a more personalized sales experience through the use of a transaction-based knowledge model that includes both the rules used for reasoning as well as the corresponding actions. Based on this solution, we have developed an intelligent electronic sales platform that is supported by a framework which provides the desired personalization as well as extensibility and customization capabilities. This paper reports our design and development of this system and application examples.

Keywords *intelligent agent, web agent, online sales, e -commerce, e-business, artificial intelligence*

1 Introduction

With the explosion of the number of e-commerce sites, the challenges to gain customer trust, retain customers, and distinguish oneself from the competition have become more evident. The design and usage of the numerous e-commerce sites varies greatly, causing confusion and frustration for novice and even experienced users. The number of these online retailers is enormous and the key distinguishers for these companies are not readily apparent. A new way to allow users to shop online is required to meet these challenges. Just as in traditional brick and mortar stores, we need a sales representative to guide the user to the products and services they desire – a virtual sales representative. This virtual sales representative should guide the user through the search and purchasing process by speaking or displaying helpful guidance questions.

We have addressed this need with the development of the Intelligent Electronic System (iES). The iES offers a cost effective solution to provide an equivalent level of customer service and guidance through an intelligence framework.

By designing the application as a framework, we have created a system that is easily adaptable or extendable by any company wishing to sell products or services using their own unique sales model. The iES provides the potential to overcome the remaining challenges regarding e-commerce in a cost effective manner though the efficient adoption, adaptation or extension of our intelligence framework.

A good sales representative should know the expected questions and the answers to those questions beforehand in order to conduct an efficient and helpful sales transaction. Therefore, we provide templates where the questions and answers can be determined at each point in the conversation by the extender or adapter of the iES system. These templates may be adjusted to meet any sales model desired by simply adding or adjusting entries in the database. By using a virtual sales representative, the opportunity for brand recognition increases. The image of the sales representative could be an easy to remember, animated character or a sophisticated and attractive model. In addition, the impression given to the consumer by a company with a virtual representative is likely to be greater for a company as they appear willing to go the extra mile to provide excellent customer service. The usage of a virtual representative provides the potential for increased competition distinction, trust, and consumer retention.

The organization of this paper is structured as follows. The next section reviews the related work. Section 3 discusses an intelligent electronic sales representative system, known as iES, including its system architecture, components, functions, and used technologies. Section 4 presents the knowledge-based intelligent solution used in iES. It consists of a transaction-based knowledge model, reasoning rules, and actions. Section 5 shows an application example and usage. Finally, concluding remarks and future work are given in Section 6.

2 Related Work

In this section, we briefly review existing work in the study and applications of web agents and artificial intelligent techniques in e-commerce applications. Current leading technology groups are increasing their efforts to produce "intelligent agents" to perform a number of web-based activities. Most of them are focused on how to provide more personalized web services using intelligent agents. Leading the pack are online personal assistants. As pointed out by S. Hedberg [5], intelligent agents are becoming an increasingly relevant term in the computing world; therefore, it is important to understand the role of agents in this context. Intelligence refers to the existence of some sort of knowledge model with reasoning and learning algorithms while the term "agent" actually carries many definitions. C. Hayes in [4] pointed out that in developing any kind of agent technology, one must be able to discern exactly what qualifies an entity as an agent, and be aware of the challenges associated with agent systems. It is also important to understand that there are various applications where agent technology can be effectively applied. E-commerce applications are one of them.

As e-commerce technologies and applications advance, people believe that next generation e-commerce systems must provide more intelligent services to customers. According to our observations, there are four types of intelligent services in e-commerce applications. The first type is to provide intelligent assistance for customer services. The major goal is to reduce the costs in customer services by using electronic customer service agents. The basic idea is to use virtual representatives to provide one-to-one direct service to customers to answer most of the questions from customers. These virtual representatives are programs that simulate customer representatives to conduct human-to-human conversations with customers through natural language interactions. They provide a key differentiator to a company's success: online delivery of enhanced customer service. Aberdeen Group Inc. [1] discussed the demand for increased online customer service, the challenges in providing self-customer service, and how automated agents help to meet these challenges.

The next type is to provide intelligent assistance for information searching. The primary goal is to provide customers with automatic web search agents that assist them in searching for the right products and relevant information over the Internet. The major function of a web search agent is to search and collect the relevant information by sort through the great number of web pages on the Internet. S. Luke and J. Hendler [6] suggested an addition to the set of standard HTML tags in which authors could describe page content, classify them, and define relationships to other pages. Web agents based on the provided information to offer users with more relevant information.

The third type is to provide intelligent assistance in product purchasing. The basic objective is to assist customers to select and compare products during a purchase process. A. Burns and G. Madey [2] described how web-based agents are used to assist consumers in fashion selection and the online purchasing process. They defined a fashion selection process involving many variables used by the web agent to select fashion apparel. A rule-based intelligence solution is used to support the process. It can be expanded with the addition of rules to the expert system. In addition, they also discussed the problems in the development of their solution and possible suggestions to solve them.

The fourth type is to provide intelligent service in e-commerce trading and e-negotiation. Gao et al. in [3] discussed an electronic system that assists buyers and sellers to conduct electronic negotiations and reach business deals.

This paper reports our research effort on establishing an intelligent sales platform that provides web customers with experience with a virtual sales representative. Unlike the existing work, our focus is to use an electronic sales agent to play as a sales representative who interacts with customers on the web to assist them to find their needs and right products. The platform is developed based on a transaction-based knowledge model that represents the product-oriented objects and attributes, as well as sales-driven business rules and actions. The goal is to provide a model-based virtual sales representative to communicate with online users to enhance their sales-driven shopping experience. The major feature of our intelligent solution is to offer customers with interactive sales-driven communications during online shopping to assist commerce trading. There are two major advantages of our approach. The first is to provide online customers with an intelligent sales-driven service to increase product sales. And the other is to allow merchants to define and configure their marketing rules and policies inside the knowledge models and reasoning rules.

3 System Overview

This section contains an overview of the iES system, including its architecture, functions, components, and used technologies.

An Overview

Since 2002, the Intelligent Electronic System (iES) was developed in the Computer Engineering Department in San Jose State University. The major goal of the system is to not only enhance the online purchasing experience, but to provide a framework that any online store could easily extend. The enhanced online shopping will be accomplished through the use of a virtual agent who will guide the user to the products they are interested in. This guidance will be accomplished through the asking and answering of questions that are easy to understand and are related to the context of the product being searched for.

Additional marketing and business logic will be included in the knowledge model used by the virtual agent to base its reasoning and decision making on. By enhancing the online shopping experience, we hope to decrease abandoned shopping carts and increase overall sales though this lower cost purchasing medium. This is to be accomplished though the increased ease of use and the business differentiation among competitors.

Architecture

The following diagram shows the architecture of the iES system. We have chosen to use the 3-tier client/server architecture. Tier one consists of a standard web-browser such as Microsoft's Internet Explorer. Tier two is made up of the server and the iES intelligence framework, which is built upon this server. The database management server software comprises the third and final tier.

Components

A description of each iES component/subsystem is presented. The complete iES system consists of a client user interface, database management software, a server, and intelligence framework components. The network subsystem is implied by the usage of the 3-tier architecture shown in Figure 3-1.

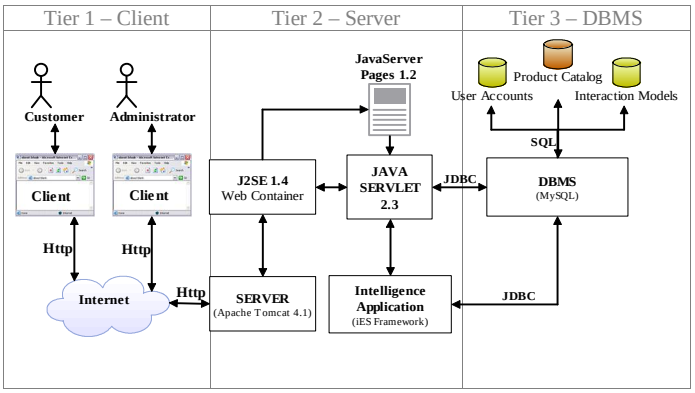

Figure 3-1: iES system architecture

iES client user interface

The client user interface subsystem consists of a collection of components that allow users to access the services and data provided by the iES system. These elements include the web browser and HTML web pages that are dynamically generated or statically delivered to the end user.

The web browser component is responsible for presenting HTML web pages that display text, graphics, buttons, and links to the users. It is also responsible for sending user requests and accepting responses to/from the web application server. The web browser accepts requests from the user from keyboard strokes, pointing device movements, and button clicks. It displays results in the form of buttons, links, images, text, and forms inside the browser window. The web browser sends requests and receives responses from the web application server using the HTTP protocol. Each HTML web page allows the user to receive and request information or services offered by the iES system. Figure 3-2 displays the model for the client user interface subsystem.

iES server

The server subsystem consists of the Apache Tomcat 4.1 web server as well as the Servlet and JSP code supporting the web application. The web server component is produced by a third party software vender and only needs to be installed and configured to meet the requirements of the iES web application. The server accepts incoming requests and delegates them accordingly. If the request requires dynamic data, it is handled by the server's web container which encapsulates the lifecycle, coordination, and handling of the Servlets and JSP pages used to implement the application. If the request is for a static web page, the requested page is delivered back to the client without further processing. The web server accepts requests from clients over the public internet though the use of the HTTP protocol. It also sends responses using this protocol. The standard web browsers are capable of commutating with this protocol and do not require any special configuration to access the web severs interface.

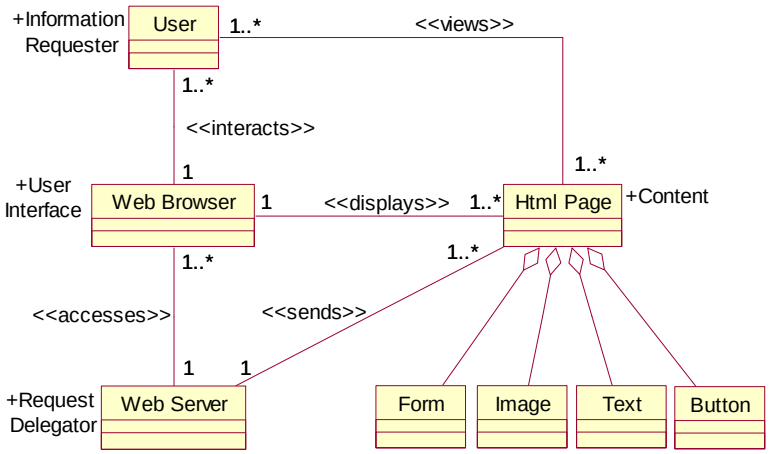

Figure 3-2: Model of iES client user interface

The Java Server Pages are used to display dynamic data to the end user. These pages consist of HTML, JSP, and java code that are arranged in an organized and consistent manner. The JSP pages are accessible to the end user in the same manner a HTML web page is. The extensions of these pages are .jsp as opposed to .html. For example the home page will be index.jsp instead of index.html. The web server accesses these JSP pages though its web container. Behind the scenes the JSP pages are compiled to Servlets and are eventually translated into plain html before it is sent back to the client. The Servlets contain a significant amount of the application logic and coordination needed by the web application. Based on the current request the Servlets may coordinate with the DBMS, and/or the iES intelligence application to gather any required data, update any appropriate java bean's state, and place them into the session where the JSP pages can access them. Implemented in this manner we have used the model-view-controller design pattern where the Servlets are the controller, the JSP pages are the views, and the java beans, which also gather data from the database are the model. Using the appropriate web address, the Servlets are accessed from the end user's web browser. When the web server receives a request for a Servlet, the requested Servlet is then invoked by the server. The Servlet will then have access to session and form data, as well as other HTTP information. A model for server subsystem is shown in Figure 3-3.

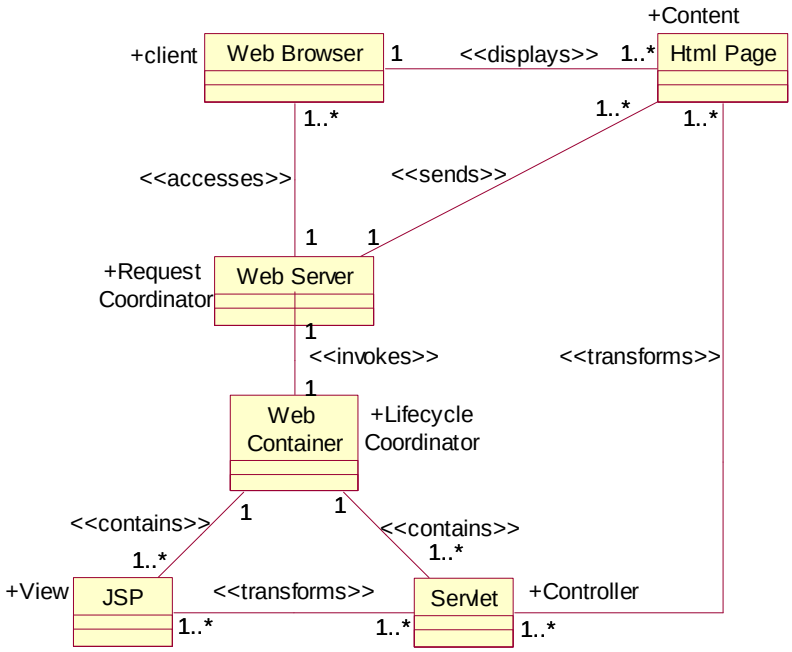

Figure 3-3: Model for iES server

iES database

The database subsystem scope consists of the management, access, organization and storage of data required by the iES application. The database management software component was developed by a third-party software vendor. Since we are using Java's JDBC mechanism, we have a large variety of DBMS options to choose from. For this project we used the MySQL DBMS. This component was installed and configured. The DBMS interface is accessed using java's database connectivity. The DBMS vendor supplies a driver which allows our java classes to communicate with the database. This includes communication from the Servlets and iES intelligence application we have developed. The DBMS will manage the storage of our data. This is organized in the form of tables. Our system consists of three databases, which consist of user accounts, the interaction model, and the product catalog data. Queries and updates are sent to the DBMS from our application programs using JDBC. SQL commands defined by our application are issued on our behalf to the database tables managed by the DBMS. Figure 3-4 shows the model for the database subsystem.

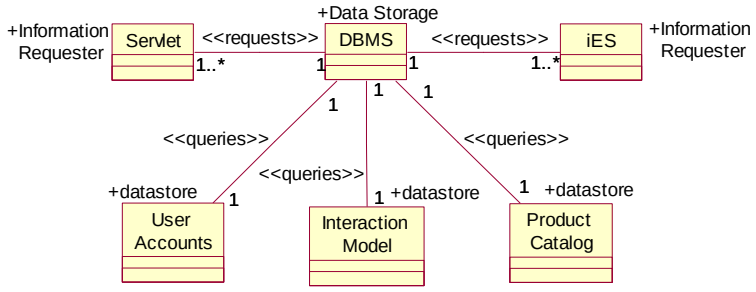

Figure 3-4: Model for iES database

iES Intelligence

The iES intelligence subsystem handles all the reasoning and decision-making based on the knowledge model that is stored in the DBMS and encoded in rules. This subsystem is accessed by the iES web application Servlets. The Servlets present the iES intelligence subsystem with the decision-making criteria. Based on the criteria, the intelligence subsystem conducts reasoning and makes a decision indicating to the system which question, answer, or product details to show next to the user. The iES intelligence subsystem contains a knowledge model where reasoning and decisions can be based upon. The knowledge model's data is stored in the database and the logic to process this knowledge model is embedded into the iES intelligence subsystem. Conversation templates are a part of the intelligence subsystem. These templates define what questions and answers are to be presented to the end users based on the current state of the system. The conversation templates and the product catalog component are also stored in the database. Entries in the product tables are of actual physical products for sale by this e-commerce web site. Various product details that describe each product are required in the decision making and reasoning conducted by this subsystem. These product details are a part of the product catalog database. Tokens are instantiated by the intelligence subsystem and are passed around as parameters or return types in this application. They are used to determine the state of the conversation. Figure 3-5 provides the model for the iES intelligence subsystem. For detailed information regarding the artificial intelligence model, please refer to Section 4, Intelligence Solution.

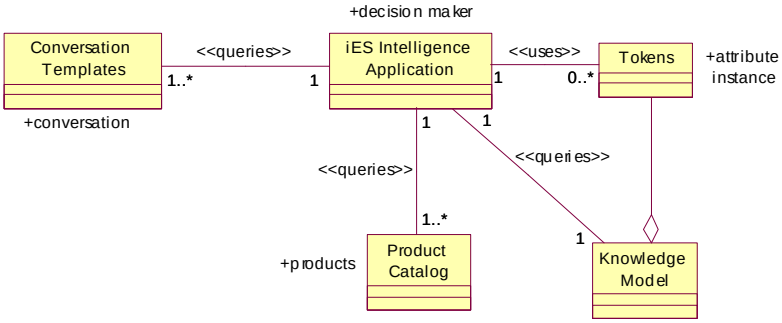

Figure 3-5: Model for the iES intelligence

Technology

This system has been designed general enough that it may be used with any products an ecommerce site wishes to offer or to present any type of sales model wished. Therefore it is a design for a framework, which may be used by any company to sell any products, and may be implemented by any technology capable of establishing and tracking an online session with access to a database. For example, it may be implemented using Java, .Net, PHP, or similar technologies. Our implemented solution was built using a combination of JAVA, JSP, SERVLETS, HTML, XHTML, JavaScript, JDBC, and SQL. We built our system upon the Apache Tomcat server and MySQL database, both being open-source systems.

4 Intelligence Solution

Overview

In order for an application to be considered intelligent it must contain a knowledge base and be able to conduct reasoning. As a result of the knowledge it contains and its reasoning abilities, it must then be able to make logical decisions. In our research for an appropriate artificial intelligence model, we found no general solution that would fit our unique problem domain exactly. We therefore combined the various artificial intelligence knowledge we gained from our research with our domain specific knowledge to come up with a unique AI model to simulate the conversation between a consumer and a sales representative. We have named this model the Token Graph Rule System (TGRS).

The TGRS is a forward chaining rule-based artificial intelligence system that is defined as a graph structure with a set of information. This information includes a Token set T_t, CurrentNode CN_t, Ruleset RS_i, and Conversation Templates CT_i. The subscript 'i' indicates the key of the node that entity belongs to and the subscript 't' indicates a point in time from the beginning of the conversation to the end. This set of information collectively describes a point in a conversation. It also contains the knowledge and reasoning encoded in rules and in a knowledge base to decide where to take a conversation next. The conversation represents communication between a customer and a sales representative, i.e. an exchange of questions and answers to simulate a sales interaction. Each entity in the TGRS is described next.

The Rule Set RS_i contains general, business, and user sub-rule sets for each node. RSi = (GR_i, BR_i, UR_i). General rules are used for navigation through the graph, which corresponds to knowing which questions and answers to display to the user next. Business rules are used do determine which products are to be displayed to the user based on promotional offerings. For example, if special arrangements have been made for a specific manufacturer, the products from that manufacturer could be displayed before other products that meet the current search criteria of the customer. Business rules are based on these special arrangements made by manufacturers, and also in-store promotions. The User rules are included to help customize the shopping experience for the user. User rules are based on the profiling of the brands and the price range of their previous purchases. For example, if we notice that the customer is purchasing the lowest cost products in most categories, then we would include this information in the sorting of the product search results by placing the lower cost items higher in the list shown to the user. All the rules within these three subsets contain conditions C_i, actions A_i, a rule weighting RW_i, and a rule type RT_i which indicates if this rule is applicable on entering or exiting a node. The conditions are simply states the system is expected to be in when reaching a node. The actions contain which node to go to next, the next node NN_i. The rule weighting is used to help the system reason when it needs to choose one rule from a set of rules that meet the current conditions.

The current node CN_t indicates what node or point in the conversation the customer is currently in. This variable is important in assessing the current state of the system to select a matching rule since each node has its own unique rule set. We made the choice to define a rule set for each node instead of a rule set that applies to the entire system to reduce the overall search time needed to select a rule that corresponds to the current state of the system. This allows us to only compare the current system state to a small subset of system rules rather than the entire set of rules contained in the system.

A token represents an instance of a product or other information stored in our knowledge model or given as dynamic input from the user. Tokens have a type and a value, for example, Sony would be an example of a Brand token – the type being "Brand" and the value being "Sony". At any given point in time the token set, T_t, will contain zero or more of these tokens. This is important in assessing the current state of the system to help in the selection of an appropriate rule to follow and to assess which nodes have been visited in the conversation thus far.

Each node in the system is either described as a static, pre-criteria, or criteria node. A token type exists for every node. The value is either system set in a database or hard-coded, or user selected from the web form interface. A token that is set for a static node is not used to narrow down the list of products shown to a user. Examples of these types of tokens are shopping cart and product tokens. The tokens set by a pre-criteria node or a criteria node would be used in the queries against the database to narrow a list of products down based on user selected values for these tokens. Examples of these are brand, price range, and keyword tokens. The difference between a pre-criteria node and a criteria node is that the criteria node will list buyer statements which allow all other criteria tokens to be set by the user if they have not been set yet in the conversation. The pre-criteria node will not contain this list of buyer statements and its token type will not be listed in any

such list of buyer statements displayed by a criteria node. This gives organizational control to the developers who extend or users who adapt this model to decide what product narrowing questions are appropriate to ask at a given point in the conversation.

A conversation template CT_i is defined for each node. It contains the expression our virtual representative should convey, the questions or answers from our salesperson, the questions or answers the customer may reply with, additional links a user may select, and information to indicate where any promotional ads or product information may be placed with in the web page. The context conversation templates provide the interface, which forms the basis of a sales interaction between a user and our virtual representative. This is accomplished with the exchange of questions and answers defined in these templates for each node that are displayed to the customer.

In addition to the conversational aspects of the conversation template, there are specified areas of the template layout where we apply business rules and user rules to produce a list of products that meet a user's current search criteria. The order of the product listing is determined by showing the products from manufacturers who have made special deals with the online store, followed by products offered as in-store promotions, and finally by products that meet the customer profile of price selection and brand preference if this information is available. These are all additionally influenced by any search criteria the user has indicated so far in the conversation. Rules at each node can be added or removed to adjust this default ordering of products.

Model

Shown in Figure 4-1 is a diagram of the Token Graph Rule System which shows the actions that take place on entering a node, while visiting a node, and on exiting a node. It also indicates all possible links from one node to another for our default sales model.

When entering a node, the template that corresponds to this node is loaded. Tokens are inserted where applicable in this template, and any rules that meet current conditions and are applicable upon entry are executed. Options that depend on what nodes have previously been visited may be included or excluded depending on the template design. The result of these actions is the building and sending of a dynamic web page to be viewed by the customer. While visiting the node the user interacts with our virtual representative by selecting elements from a pull-down menu, and/or checking check boxes and/or inputting text into text boxes which will set token values. When the user clicks a corresponding link the exit actions are then executed. This includes finding the rules which match the current system state, conducting reasoning to find which rule to select if multiple rules meet the current system state, and then executing the action corresponding to the selected rule. The execution of the action will cause the update of the current node and token set, as well as the navigation to the next node.

This system has been designed general enough to reasonably account for all the expected conditions, or states of the system, at each node.

Nodes and Rules

Part of the artificial intelligence model we implemented is described as a set of nodes in which combined form a directed graph. We implemented a default set of nodes to represent a sales model that could be included in every instantiated system. This model could be changed at anytime by anyone using or extending the iES framework.

Our default model can be formally described as a set of nodes, N, where

N = {LogIn, Welcome, Category, Type, Brand, Price, Keyword, Product, ShoppingCart, CheckOut}

and as a set of arcs or edges, A, where

A = { (LogIn, Welcome), (LogIn, Login), (Welcome, LogIn), (Welcome, Product), (Welcome, Category), (Welcome, ShoppingCart), (Category, Product), (Category, Brand), (Category, Type), (Type, Category), (Type, Brand), (Type, Price), (Type, Keyword), (Type, Product), (Brand, Category), (Brand, Type), (Brand, Price), (Brand, Keyword), (Brand, Product), (Price, Type), (Price, Brand), (Price, Keyword), (Price, Product), (Product, Greeting), (Product, Type), (Product, Price), (Product, Keyword), (Product, Brand), (Product, Category), (Product, ShoppingCart), (ShoppingCart, Product), (ShoppingCart, CheckOut), (CheckOut, ShoppingCart) }

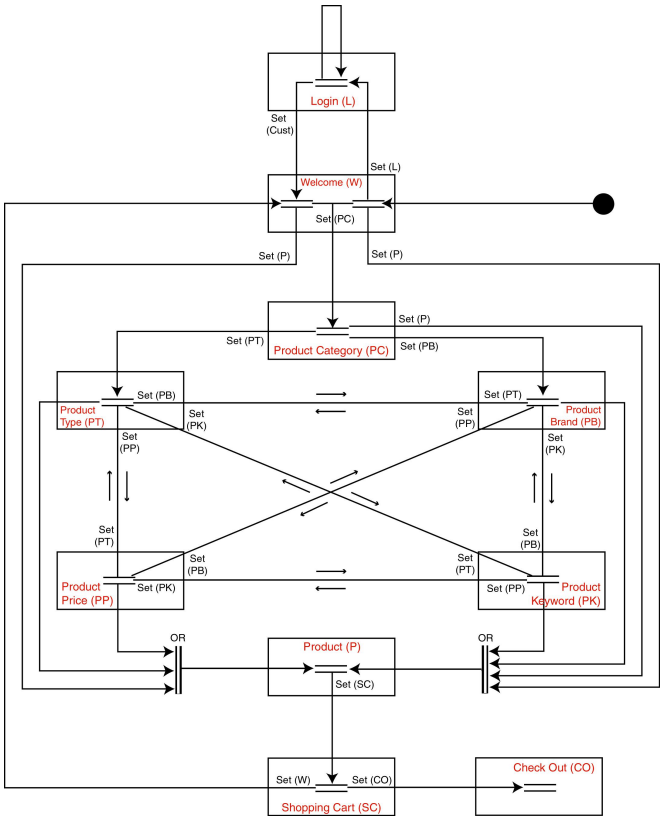

Figure 4-1: Overview of Node Interaction

The set A represents all paths from one node to another as defined by our default sales model. Again, this set of paths can be changed at anytime by anyone extending the iES Framework.

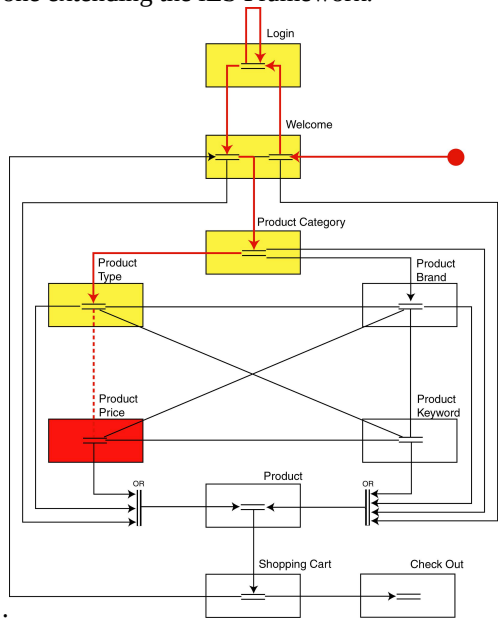

Figure 4-2: Sample Token Graph Rule Scenario

Input (Memory):
Current Node = CN = Type
Token Set = T = {Welcome**:** visited, Login: true, Category**:** Electronics, Type**:** Camera,
Price**:** 0-100)}
Current Token = Price

Input (Database):
RS_{type} = {GR_{type}, BR_{type}, UR_{type}}
GR_{type} = {R_{type1}, R_{type2}, R_{type3}, R_{type4}, R_{type5}}
R_{type1}: C = "Brand ∈ CurrentToken" A = "NextNode → Brand" RW = "1" RT = "Exit"
R_{type2}: C = "Price ∈ CurrentToken" A = "NextNode → Price" RW = "1" RT = "Exit"
R_{type3}: C = "Keyword ∈ CurrentToken" A = "NextNode → Keyword" RW = "1" RT = "Exit"
R_{type4}: C = "Product ∈ CurrentToken" A = "NextNode → Product" RW = "1" RT = "Exit"
R_{type5}: C = "CriteriaNodeSet ∈ NodeSet" A = "NextNode → CurrentToken" RW = "1" RT = "Exit"
BR_{type} = {R_{type6}, R_{type7}}
R_{type6}: C = "ProductList ∈ ManufacturerSpecials"
A = "OrderByWeighting(ManufacturerSpecials) → ProductFrame" RW = "1" RT = "Enter"
R_{type7}: C = "ProductList ∈ StoreSpecials" A = "OrderByWeighting(StoreSpecials) → ProductFrame"
RW = "2" RT = "Enter"
UR_{type8} = {R_{type8}}
R_{type8}: C = "MoreCustomerPurchases ∈ BrandsPurchasedFrom"
A = "OrderByBrand(ProductList) → ProductFrame" RW = "1" RT = "Enter"

Output (Return Value):
NN = Price (Based on R_{type2})

A snap shot of what the system may look like at a given point in time is shown in Figure 4-2. Up until this point the user has logged in and has told our virtual sales representative that he/she would like to find a product in the Electronics category, that the specific products of interest are Cameras, and he/she would like to narrow the search down further by choosing a price range. At the Product Node the token set indicates this situation to be true. Looking at our Rule Set for this node, we notice that Rule type 2 matches the current conditions. This rule's action set indicates that the next node to travel to is the Price Node. This value will be returned by the intelligence subsystem. Upon receiving this value, the appropriate conversation templates can be loaded into memory based on this Next Node value, and the current token set values. This information is then assembled and displayed as a web page where the user may continue to interact with our virtual representative in a conversational manner.

5 GUI Design

This section will describe the various screens and the expected user interaction at each node. The screen is divided into three frames: the buyer/seller dialog, the product information, and the virtual sales representative image. The three frames are displayed in Figure 5-1.

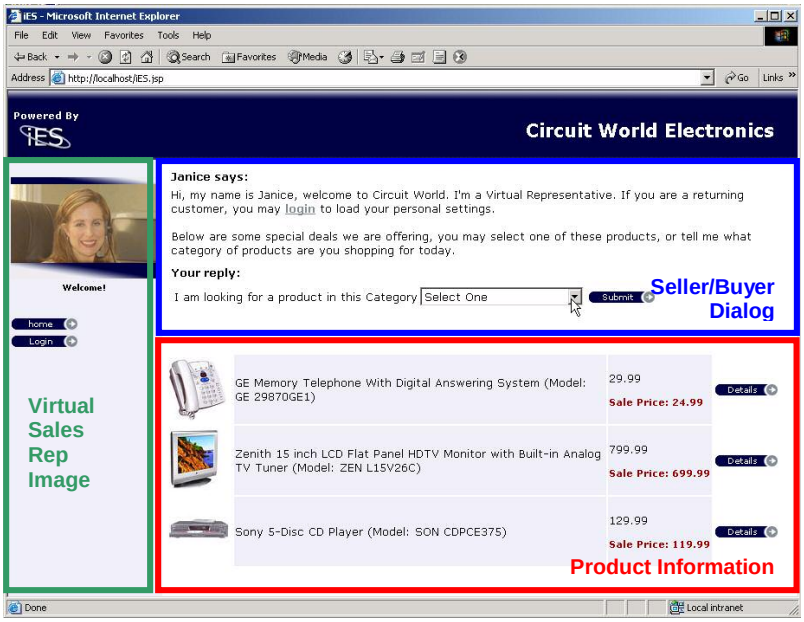

Figure 5-1: Three-frame page layout

Example Use Case

A user is shopping for a portable CD player that costs no more than $60.00. The user has shopped at the site on numerous occasions and usually purchases Sony products when available. The following series of screen shots represents the screens that the user would transition through in completing the above scenario. The order of the products listed in the product list reflects that this shopper prefers Sony. The scenario will begin upon entry into the site and proceed through checkout. Each step in the sequence will display the interface just before an action button is selected. The first screen, shown in Figure 5-2, represents what the user will see upon entering the site. The top three manufacturer's specials will also be presented to the user for consideration.

Figure 5-2: Enter e-commerce site

After the user logs in, the system will check for recent purchases. If the user has made a recent purchase, the system will display corresponding accessories if there any available for that particular product. The user may request to see one of the products listed in the lower frame, or select a category from the dynamically generated list. Figure 5-3 provides a sample of what the user might see.

Figure 5-3: Select 'Portable Electronics' category

The user may select a product based on the current criteria or further narrow the search by choosing a specific type in the selected category or a specific brand. These choices are displayed in Figure 5-4.

Figure 5-4: Select 'Portable CD' type

At this point, the user is offered additional options for narrowing the search criteria – price and keyword. The keyword option is provided as a catch all if there is something specific the user is looking for. If the user chooses to narrow the search by price, the maximum and minimum price for items matching the current criteria are displayed to avoid entering values that have no matching items. Figure 5-5 provides an example of a user entering a price range.

Figure 5-5: Select desired price range

Once the user finds a product that interests them, they may choose to view detailed information about that product. From this screen the user may add the item to their shopping cart, go back to original list, or change any of the currently set criteria. In Figure 5-6, the user has decided to add the item to their shopping cart.

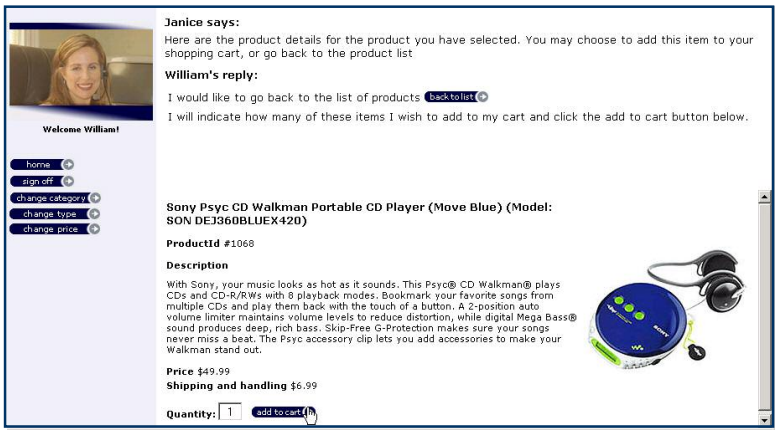

Figure 5-6: Add item to the shopping cart

6 Conclusions

Traditional brick and mortar stores employ human sales representatives that help and guide customers to the products they desire. The Intelligent Electronic Sales Representative was designed and developed to allow the online shopping experience to simulate the one held in a traditional brick and mortar store. In addition, we wanted to provide a cost effective solution to companies wishing to enhance their customers shopping experience. The iES system has accomplished this through its intelligence framework, which can be easily adopted, adapted, or extended by any company, selling any product-base, using any sales model. Table 6-1 compares and contrasts the iES e-commerce system with that of traditional e-commerce.

Table 6-1: iES system versus traditional e-commerce

iES system	Traditional e-commerce
• Increased brand identity	• Blend in with the competition
• Shopping guidance	• Self-discovery
• Targeted at novice users, accommodates experienced users	• Targeted at experienced users, difficult for novice users
• Sales model easily adaptable	• Extensive work to adopt new sales strategy

To make effective changes to the default sales model, programming experience is not required. However, if more advanced customization is desired, the iES system is extremely flexible and will allow further product selecting/sorting algorithms to be added to the default set of algorithms included in the iES library.

Developers who wish to extend this system will have a wealth of information to work with. This information could be used to further enhance the customer profiling that determines which products are shown to the user and in what order. It could also be used to display additional promotional offers to the customer at any point in the sales conversation. The adapters of this system will have the ability to customize and/or change the questions that are asked as well as the answers the user may be offered. The iES system is able to provide these capabilities by using a unique sales domain independent artificial intelligence model.

The development of this model began as a complex sales domain specific system. Its implementation was to be specific to one company and would not be easily modified. But, through abstraction and generalization, we were able to simplify the design. As we simplified the model, the implementation required to support the model decreased and the reliance on the database increased. The effect of this was an efficient, domain-independent, artificial intelligence model that could be modified at any point through simple updates and additions to the database.

Future work and research will include increasing the intelligence of the web agent by: adding abilities to communicate with other web agents, providing additional rules within the system to provide a more customized and targeted sales

experience, and the dialog could be modified to add a voice to the virtual sales representative providing audio feedback to the user.

7 References

[1] Aberdeen Group Inc., "Interactive Customer Care: Enriching the Self-Service Experience with Automated Agents)," *An Executive White Paper*, November 2000.

[2] A. Burns, G. Madey, "Development of a Web-Based Intelligence Agent for the Fashion Selection and Purchasing Process via Electronic Commerce," *Proceeding from Americas Conference on Information Systems*, 1998, pp. 140-141.

[3] J. Gao, et al, "An Intelligent Electronic Negotiation System", Technical Report, Computer Engineering Department, San Jose State University, 2002.

[4] C. Hayes, "Agents in a Nutshell – A Very Brief Introduction," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, January/February 1999.

[5] S. Hedberg, "Agents for Sale: First Wave of Intelligent Agents Go Commercial," IEEE Expert, December 1996.

[6] S. Luke, J. Hendler, "Web Agents That Work," *IEEE Multimedia*, 1997, pp. 76-80.

Authors

William Korbe, Webmaster, ActiveFrameworks Inc. Palo Alto, CA
William received a B.S. in Software and Information Engineering at San Jose State University, May 24th 2003. His current interests in the software industry include web-based development, artificial intelligence, and enterprise application development.

Valerie Stanton, Computer Scientist, Lawrence Livermore National Laboratory, Livermore, CA
Valerie received a B.S. in Software and Information Engineering at San Jose State University, May 24th 2003. She currently works for Lawrence Livermore National Laboratory and is involved with database system design and development.

Jerry Z. Gao, Ph.D., Associate Professor, Computer Engineering Department, San Jose State University, San Jose, CA
Dr. Gao has co-authored two technical books on software testing, and written many technical papers on software engineering and object-oriented technology. His current research interests include component-based software engineering, software testing methodology and test automation, information technology and Internet computing, mobile computing and wireless commerce. Dr. Gao has taught various academic classes in software engineering, including software engineering I & II, object-oriented software design analysis, object-oriented software testing, software testing and quality assurance. Before he joined San Jose State University, he worked for Fujitsu Network Communications System, Inc. as a manager of an R&D group in the software-engineering department. At Fujitsu, he was instrumental in developing an enterprise software production environment over the Internet. His team is the driving force to establish an enterprise-oriented software-engineering environment over the Internet in Fujitsu. He has over 45 technical papers in IEEE/ACM journals and magazines as well as international conferences. He is a co-author for the book on *Object-Oriented Software Testing* published by IEEE Computer Society Press in 1998. Recently, he and his co-authors (Dr. Jacob Tsao, and Dr. Ye Wu) just completed a new book, titled as "Testing and Quality Assurance for Component-Based Software". Artech House Publishers will publish it in September 2003.