Natural Language Sales Assistant – A Web-based Dialog System for Online Sales

Joyce Chai, Veronika Horvath, Nicolas Nicolov, Margo Stys-Budzikowska, Nanda Kambhatla, Wlodek Zadrozny

IBM T. J. Watson Research Center
30 Saw Mill River Rd.
Hawthorne, NY 10532
{jchai, veronika, nicolas, sm1, nanda, wlodz}@us.ibm.com

Abstract

This paper describes a web-based dialog system – Natural Language Sales Assistant (NLSA) – that helps users find relevant information about products and services in e-commerce sites. The system leverages technologies in natural language processing and human computer interaction to create a faster and more intuitive way of interacting with websites. By combining traditional AI rule-based technology with taxonomy mapping, the system is able to accommodate both customer and business requirements. Our user studies have demonstrated that, in the context of e-commerce, users preferred the natural language enabled navigation over menu-driven navigation (79% to 21% users). In addition, compared to a menu driven system, the average number of clicks used in the natural language system was reduced by 63.2% and the average time was reduced by 33.3%. The NLSA system is currently deployed by IBM as a live pilot and we are collecting real user feedback. We believe that conversational interfaces like that of NLSA offer the ultimate personalization and can greatly enhance the user experience for websites.

1 Introduction

With the emergence of e-commerce systems (Aggarwal, Wolf and Yu 1998; Muller and Pischel 1999), successful information access on e-commerce websites that accommodates both customer needs and business requirements becomes essential. Menu-driven navigation and keyword search provided by most commercial sites have tremendous limitations, as they tend to overwhelm and frustrate users with lengthy and rigid interactions. User interest in a particular site decreases exponentially with the increase in the number of mouse clicks (Huberman, Pirolli, and Pitkow 1998). Hence shortening the interaction path to provide useful information becomes important. Many e-commerce sites attempt to solve the problem by providing keyword search capabilities. However, keyword search engines usually require that users know domain specific jargon so that the keywords could possibly match indexing terms used in the product catalog or documents. Keyword search does not allow users to precisely describe their intentions, and more importantly, it lacks an understanding of the semantic meaning of the search words and phrases. For example, keyword search systems usually can not understand that "summer dress" should be looked up in women's clothing under "dress", whereas "dress shirt" most likely in men's under "shirt". A search for "shirt" can reveal dozens or even hundreds of items, which are useless for somebody who has a specific style and pattern in mind. Moreover, search engines do not accommodate business rules, e.g. a prohibition against displaying cheap earrings with more expensive ones. The solution to these problems lies, in our opinion, in centering electronic commerce websites on natural language (and multimodal) dialog.

Natural language dialog has been used for call-center/routing applications (Carpenter and Chu-Carroll 1998; Chu-Carroll and Carpenter 1998), email routing (Walker, Fromer, and Narayanan 1998), information retrieval and database access (Androutsopoulos and Ritchie 1995), and for telephony banking (Zadrozny et al. 1998). The integration of natural language dialog with an e-commerce environment is a novel feature of our system. Our work goes beyond the "natural language interface" features of websites such as www.askjeeves.com, www.neuromedia.com, etc. which work in a question-answering mode and do not use dialog. When searching e-commerce sites, users often have target products in mind but do not know where to find information, or how to specify a request. Sometimes users only have vague ideas about the products of interest (Saito and Ohmura 1998). Thus, users need to be able to formulate their requests in their own words as well as revise their request incrementally based on additional information, which can be provided through natural language style of dialog. Our recent studies show that natural language dialog is a very effective medium for negotiating such contexts by understanding users' requests/intentions and by providing help/advice/recommendations to the user.

We have built a Natural Language Sales Assistant (NLSA), a system which allows customers to make requests in natural language and directs them towards appropriate web pages that sell the products. The system applies natural language understanding to interpret user inputs, arranges follow-up dialog to provide explanation and to ask for additional information, and finally makes recommendations. The NLSA prototype system is currently deployed and we are collecting real user feedback.

In this paper, we first give a detailed description of the NLSA system, in particular, the system architecture and components. We then present results from the user studies. Finally, we discuss the lessons learned and outline future work.

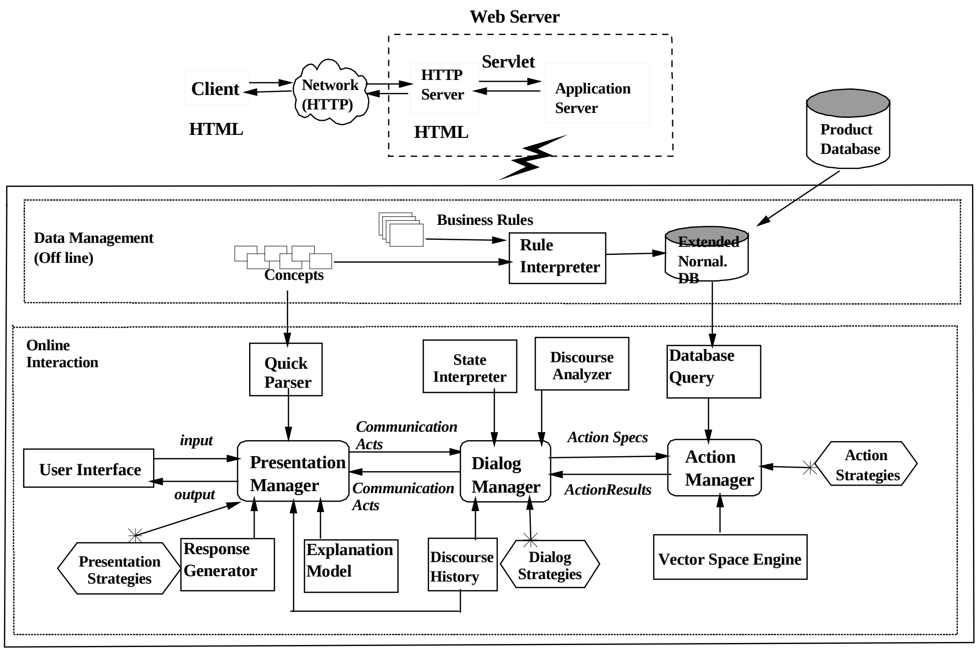

Figure 1: *System Architecture*

2 NLSA System Descriptions

NLSA includes two subsystems: data management subsystem and online interaction subsystem. It is implemented as a client-server system in a three tier architecture using Java servlets and HTTP as communication media between the client and the server.

Our architecture (in Figure 1) supports multimodal dialog in the online interaction subsystem. It is designed to support inputs from different channels and modalities including keyboard, speech input and output over a telephone or microphone, mouse input, etc. The online interaction subsystem consists of three major modules: Presentation Manager (PM), Dialog Manager (DM) and Action Manager (AM). The presentation manager uses a shallow natural language parser (Quick Parser) for noun phrases to transform the user query into a logical form, and sends the logical form to the Dialog Manager. The PM is also responsible for obtaining the system's response from the DM. The PM applies a Response Generator to generate specific presentations based on appropriate modalities such as display tables, natural language output, GUI components, etc. An Explanation Model is also integrated to provide explanations as to what information the system understands from the interaction and why certain products are recommended. After displaying the system’s response, any subsequent user input (e.g. a clarification, a correction or a new request) is again sent to the DM.

The Dialog Manager is responsible for determining the specific action(s) requested by the user and filling the parameters of the identified action (e.g. price range) through a dialog with the user. After filling in the parameters, the DM sends the filled action template (Action Specs) to the AM for execution. The dialog manager uses both short-term conversational discourse history and long-term user history to formulate responses to the user. The Action Specs represents the results of the transformation from user requests to action plans for the action manager to satisfy the requests, for example, converting a “search for products” request into an SQL database query for a particular database engine. The DM is also responsible for receiving the Action Results from the AM; analyzing them and asking follow up questions. A State Interpreter identifies the current dialog state based on both user input and system response. When a particular state is identified, the DM applies state-based dialog strategies to arrange conversation or resolve conflicts.

The Action Manager determines the best mechanism for executing an action, given an Action Spec. Thus, for a product search request, the AM decides which information source should be looked up which information should be extracted. It uses database queries to retrieve exactly matched products or uses a traditional information retrieval technique (Vector Space Engine) to retrieve the best-matched products or information. The AM returns the retrieved information (Action Results) to the DM.

Associated with the three major modules are three kinds of strategies. Presentation Strategies select different response content for different user profiles. For example, for experienced users, the display table will include detailed technical specifications such as CPU speed, RAM size etc., while to novice users, the system displays the price and brief market messages to make it easy for the user to understand how the products relate to the user requests[1]. Dialog Strategies define the appropriate kinds of responses at different dialog states, such as the kinds of explanations to be provided to the user and the kind of follow-up questions to be asked. Action Strategies specify the databases to access under specific circumstances and the actions to take if nothing is retrieved, e.g. an approximate match between the user demands and the available products. All these strategies are designed to reflect business goals and requirements.

The Data Management Subsystem contains domain- and application-specific data, information and knowledge. With the ever-evolving business environments and customer needs, tools and processes are needed to maintain related information and knowledge so that updates to these resources can be reflected seamlessly during online interaction. This is especially true in the e-commerce domain where business information changes rapidly. Therefore, the data management subsystem addresses data, knowledge and maintenance (i.e., tools and processes). The subsystem includes the concept base, business rules and the extended database. The concept base provides knowledge about common sense concepts and the user vocabulary. Business rules are used to reflect business goals and decisions by associating common sense concepts with business specifications. The extended database combines both product specifications (directly extracted from the product database) and common sense concepts to provide a unique information repository for information access. In the following sections, we first describe in detail the Data Management Subsystem and then the Online Interaction Subsystem.

3 Data Management Subsystem

In the NLSA, the domain knowledge is represented by concepts and business rules, which address customer needs and business requirements, respectively.

[1] User levels are determined by the user familiarity with notebook computers. (as shown in Figure 3).

3.1 Concepts

The vocabulary set in the NLSA is organized into concepts which represent user intentions and interests in shopping for computers. Three kinds of concepts are maintained. Entity concepts are about things or substances (such as products, accessories, etc.) that a user might be interested in. Attribute concepts represent features users are looking for in products, such as “FAST”, “LIGHT”, “HIGH-PERFORMANCE” etc. Purpose concepts represent the general uses or functions users are looking for in a particular product, such as “SMALL-BUSINESS”, “TRAVEL”, etc. Each concept contains a list of words and/or phrases implying this particular concept.

XML is used to organize and manage concepts (Bray, Paoli, and Speberg-McQueen 1998; Radev et al. 1999). The following is a fragment of the representation and content of the concept <Advanced-Graphic-Capability>:

```
</ENTRY><ENTRY NORMAL_FORM = "AGP", SCALE = 0,
SUPERLATIVE = 0>
    <NOVICE>Is 3-D graphics performance, which
speeds online gaming and graphic design,
important to you?</NOVICE>
    <EXPERT>Are you interested in having an
Advanced Graphic Port for improved 3-D graphics
performance? </EXPERT>
    <NL_FORM>Advanced Graphics Port</NL_FORM>
    <DEF><![CDATA[hd_min >= 6.4 & resolution is
1024x768]]></DEF>
    <WORD>agp</WORD>
    <WORD>graphics</WORD>
    <WORD>game</WORD>
    <WORD>3-D</WORD>
    <WORD>multimedia</WORD>
    <WORD>…</WORD>
</ENTRY>
```

Each concept has different attributes to indicate if the concept is scalable or superlative. Quick Parser (Section 4.2) is provided to identify relevant concepts from the user natural language input (see the example in Section 4.2). Concepts with different attributes lead to different dialog flows during the user interactions.

3.2 Business Rules

Business rules are mappings between concepts and business specifications. Business specifications are values and business terms that are used by the business to describe their merchandises. For example, in the computer sales domain, the specifications could be different models of computers, processor speed, memory size, hard disk size, price, etc. In terms of these specifications, business rules give definitions to the concepts from the business point of view, such as how they position various models and how those models are marketed for different segments of consumers.

Business rules have the following structure:

```
   CONCEPT ::= eval(ATTR) = CONST_VALUE &
               eval(ATTR) in RANGE &
               not(eval(ATTR))
```

For example,

```
<Advance-Graphics-Capability> ::=
eval(hard_disk_min) > 6.4 GB &
eval(resolution) = 1024x768
```

This rule interprets the “Advanced-Graphic-Capability” concept as a combination of minimum hard disk of 6.4 GB and resolution of 1024x768.

During the online interaction, business rules are used to translate identified concepts into business specifications. Thus, the relevant database query can be generated based on business specifications.

3.3 Maintenance

Concepts and business rules are not static. When new products or features are introduced or when new words or phrases are discovered during deployment, concepts and/or business rules need to be updated accordingly. Currently, we have designed and implemented a semi-automated tool for this purpose. The tool automatically extracts significant keywords from logs of user queries and allows manual updates of business rules through an editing interface. The automatic extraction phase applies unigrams, bigrams and trigrams on a collection of user input and uses a parts-of-speech filter and a simple noun phrase grammar to extract the significant keywords. Figure 2 shows a snapshot of the interface where automatically identified keywords (based on bigrams) can be added to existing concepts.

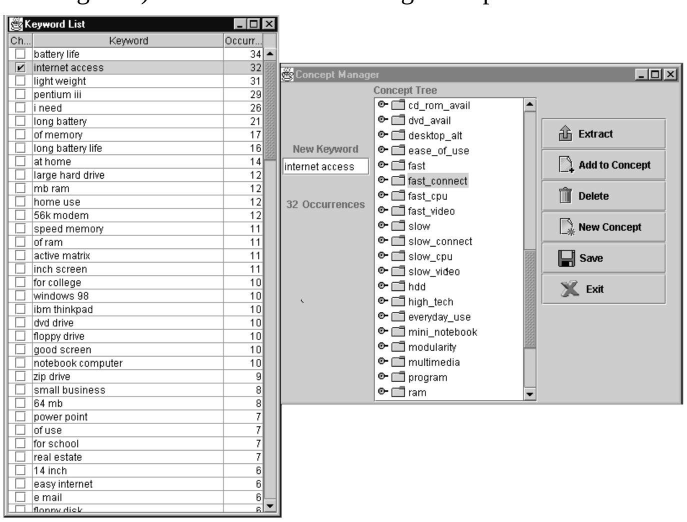

Figure 2: *Editing Interface for Concept Management*

4 Online Interaction Subsystem

The online interaction subsystem addresses the end-to-end interactions between the user and the system. It consists of the Dialog Manager arranging the content of interactions, the Presentation Manager separating content from presentation in the front end and the Action Manager performing database access and business transactions in the back end. The online interaction subsystem is designed to be domain independent. Thus the change of the application domain would only require customization of the data management subsystem.

4.1 Presentation Manager

The Presentation Manager is responsible for interpreting user input and displaying system responses. The Presentation Manager contains user interfaces, a noun phrase parser (i.e., the Quick Parser), a response generator and an explanation model.

One of the major advantages of web-based dialog systems is the addition of a new dimension for information presentation. Through a combination of UI components (such as radio buttons, forms, links, etc.) and the natural language dialog, much more information can be communicated between the user and the system in comparison to a traditional spoken dialog system. Thus, with the reduced number of turns (interactions), the user can get instructions, examples, explanations, and ideas of limitations of the system. A snapshot of a user interface (for user initial request) is shown in Figure 3.

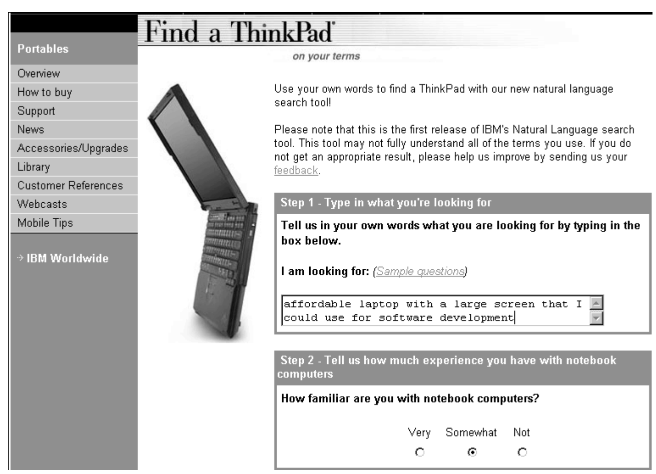

Figure 3: *A Snapshot of Initial User Interface*

4.2 Quick Parser

A shallow semantic parser is applied to identify semantic and syntactic information of interest from the user textual input. More specifically, based on a collection of regular expressions, a noun phrase grammar and concepts, the parser identifies special named entities (such as CPU speed, processor type, etc.), head of the noun phrase and concepts for further processing. The parser is implemented as a finite state cascade. It translates natural language strings into a well-formed XML message called the Logical Form (LF). The Logical Form is then passed on to the Dialog Manager. For example, if the user input is: “nice travel companion for $2000-3000 with at least 400Mhz Pentium processor”, the parser will generate the following logical form in the XML format:

```
<LF>
<NP>
<ATTRIBUTE><SPECS>nice</SPECS></ATTRIBUTE>
<HEAD><SPECS CONCEPT="PORTABILITY"
SCALE="weight">travel companion</SPECS></HEAD>
<PP>
 <PREP VALUE="for"/>
 <NP>
<ATTRIBUTE><SPECS
TYPE="BASE_PRICE">gt2000</SPECS></ATTRIBUTE>
<HEAD><SPECS TYPE="BASE_PRICE">lt3000</SPECS></HEAD>
 </NP>
</PP>
<PP>
 <PREP VALUE="with"/>
 <NP>
<ATTRIBUTE><SPECS
TYPE="CPU_SPEED">gt400</SPECS></ATTRIBUTE>
<ATTRIBUTE><SPECS
TYPE="CPU_BRAND">Pentium</SPECS></ATTRIBUTE>
<HEAD><SPECS>processor</SPECS></HEAD>
 </NP>
</PP>
</NP>
</LF>
```

The shallow parser does not aim at a complete linguistic analysis; it merely extracts some information from the user’s input that is useful for the dialog management. The parser uses several external resources, each configurable to adapt to different domains of application. Adaptation to new domains will only require updating of these resources without modifying the code.

4.3 Dialog Manager

The Dialog Manager maintains the representation of the dialog state and the dialog history, and arranges follow up interaction (such as asking for more information, providing feedback and confirmation, etc) based on those representations. Research in dialog systems presents a variety of ways to model dialog state and dialog history. These typically include a representation of the user’s intentions, beliefs and focus of attention. The models depend on linguistic phenomena that are to be handled by the system such as misunderstandings, interruptions, deictic and anaphoric expressions. A similar representation can be found in LINLIN where the dialog history is a tree with three levels corresponding to the whole dialog, discourse segments and speech acts (Jonsson, 1997).

In NLSA, the dialog state is represented by a combination of the current utterance, the current system response and the interaction history. More precisely, it takes into consideration the following questions: Has the user specified any preferences for computer features? Is there any ambiguity detected in user input? Are there any concepts identified by the Quick Parser from the current utterance? Are there any conflicting constraints detected from a history of utterances. Are there more than three products recommended to the user in the previous interaction? The combination of answers to those questions forms the dialog state for that particular interaction. The dialog history is a tree representation that contains the dialog state at each point of interaction.

The Dialog Manager applies a mixed initiative strategy to arrange different follow-ups. The system first allows the user to take the initiative to specify what they are looking for in their own language. Based on the interpretation of the user input, the system then takes over the initiative to request for specific information.

An example (in British English) is the following:

U_0: I’m looking for a piano.
S_0: I’ll find you a notebook if you tell me what your notebook needs are.
U_1: I’m looking for a laptop less than 500 pounds.
S_1: Display: a table of most popular models
Reply: The price range for Thinkpads is between 799 and 2389 pounds. You may specify a value with the range or continue your search by specifying other features you are looking for.
U_2: I want something easy to carry and inexpensive.
S_2: Display: We have found several notebooks based on your search criteria: Ultraportability: models which weigh less than 3 kilograms, Low price: affordable models which sell for less than 1500 pounds, followed by a table of six products satisfying these criteria).
Reply: Do you need a large hard disk capacity for backup data or storing large files?
U_3: No, I want something top-of-the-line.
S_3: Display: We didn’t find a product that matches your terms perfectly, but we have list the closest match to your request. (Display two tables: one table satisfies the extreme-performance criteria with weights of products sorted from lightest to the heaviest; the other table satisfies the low price criteria with weights of products sorted from lightest to heaviest).
Reply: Battery life is important to frequent travelers. Is long battery life important to you?
….

Depending on the dialog state and the dialog history, the Dialog Manager uses different follow-up responses. For example, if nothing from the domain of interest is interpreted from the user input (U_0), the system will give some suggestions and direct the user to restrict their queries to notebook related questions (S_0). If the user asks a question which is identified as out of the available product range (U_1), the system will first explain to the user the valid range and then will prompt the user to re-formulate his/her request (S_1). If more than three products are recommended in the previous interaction, the system will ask the most discriminating question to narrow down the number of products to recommend (S_2). When the user request causes some conflicts with previous requests in the dialog history (U_3), the system will engage in a sub-dialog with the user to explain the conflicting situation, provide suggestions and help users to reformulate their constraints (S_3). In a user study we conducted, we found that conflicting requirements occurred more than 40% of the time. Therefore, for dialog systems like ours, it is crucial that the source of conflict is identified and that the system can provide sensible feedback in such situations.

4.4 Action Manager

The Action Manager deals with the back end product retrieval and business transactions. The Action Manager uses the Action Strategies that specify different actions based on the Action Specs sent by the Dialog Manager. When a valid SQL query is presented in the Action Specs, an access to the extended database takes place and products matching the query are retrieved. When the user preferences for features are included, a sorting algorithm is applied to yield a ranked list of products.

When a valid SQL retrieves zero products from the database due to conflicting constraints, the Action Manager will notify the Dialog Manager that the constraints cannot be simultaneously met. It then resolves the conflict by using a vector space based on similarity measure between constraints (i.e., specifications and concepts) from the user input and constraints indexed to each product. The top similar products will be retrieved.

5 NLSA Evaluation

To better understand the user language and design system responses, we conducted an online user market survey prior to the deployment of the system. Furthermore, we conducted three user studies to objectively evaluate the usability of the prototype and to better understand the user needs.

5.1 Market Survey

In the market survey, participants were first given three questions: "What kind of notebook computer are you looking for?", "What features are important to you?", and "What do you plan to use this notebook computer for?" By applying statistical techniques (n-gram models) and a noun phrase grammar on a collection of user natural language responses, we extracted significant keywords and phrases that express user intentions and interests in shopping for computers. Then, participants were asked to rank 10 ten randomly selected technical terms (from 90 computer related terms) in terms of familiarities. This study allowed us to group technical terms into different complexity groups and better formulate system responses to different user groups. Over a 30-day period, we received 705 survey responses. From the natural language responses, we learned 195 keywords and phrases and included those in the vocabulary set for the deployed system.

5.2 User Studies

The studies were designed to reveal how successful the prototype system fared at meeting the users' expectations within the following areas: system flow, ease of use, validity of the system response, and user vocabulary.

The first user study was a proof of concept for dialog-based systems in an e-commerce environment. The study compared the first version of the prototype system to a fully developed menu driven system. The study (Chai, et al., 2000) showed that comparing NLSA navigation with the menu driven navigation in finding products, the number of clicks (a click was counted when the user clicked on a submit button, a radio button or a link, etc. to take action) was significantly reduced by 63.5% and the amount of time spent was significantly reduced by 33.3. Less experienced users preferred the NL enabled navigation much more than the experienced users. Overall, respondents preferred the NL dialog based navigation (NLSA) to the menu driven navigation two to one (2:1). Respondents thought the NLSA was extremely easy to use, and they were comfortable and confident with the resulting information it provided. Users liked the fact that they could express their needs in their "own jargon" instead of the foreign "computer jargon". There was also the perception that with the NLSA model, the computer did all the work for them instead of them doing all the work for the computer (as in the menu-driven model).

The second and the third user studies were conducted to evaluate the current version of the prototype with regards to the ease of navigation, clarity of terminology and their level of confidence that they could find the product they were looking for. In both studies, participants commended an interactive site where user's inputs can be interpreted and were very receptive to the natural language, dialog-based search site. The study clearly showed that dialog-based searches were preferred over non-dialog based searches[1] (79% to 21% users). The users liked the fact that the system narrowed down the search as they proceeded, provided that it was responsive and geared to users' specific needs. Participants in both studies shared the opinion that a system that worked for the user was better than a system that made the user work. When the NLSA worked according to design, it left the user with the feeling that the system was easy and the search was narrowed with relatively little effort. Table 1 sums the ratings (1 – least desirable, 10- most desirable) from the two studies for different categories of users.

The studies pointed towards improvements in the area of

[1] Where dialog meant either the radial selection or typed inputs

system responsiveness including tuning up of the follow up questions, prompts and explanations to the user's input. To a large extent, the success of a dialog system has been shown to depend on the kind of questions asked and the type of feedback provided. The types and nature of the questions asked throughout the NLSA were based on features and functionality of a computer. The studies conclude that asking user questions about lifestyle and usage of computer to solicit feedback would have been a more user-friendly line of action. Users' confidence in the system decreases if the system responses are not consistent with the user's input. The system feedback and the follow up questions should manage a delicate balance of exposing system limitations to the user but at the same time making sure the user understands the degree of flexibility and advantages of using a dialog system.

	Small Business		**Consumers**	
	Study 2	**Study 3**	**Study 2**	**Study 3**
Overall Rating of the Site	5.8	6.5	7.5	7.7
Level of Confidence	7.8	6.5	7.6	7.6
Ease of Use and Navigation	8.2	7.6	8.1	8.4
Clarity of Terminology	6.0	6.9	7.6	7.1

Table 1: *Comparison of Study 2 & Study 3 Ratings.*

Even though most users preferred the NL dialog based navigation, the study also showed the utility of menu driven searches. Some users definitely liked the ability to select options from a menu, specifying that the multiple-choice method was easy. There were also users who liked having questions asked of them. Typically, such users were either not comfortable with their typing ability or unable to express what they were looking for without additional external cues. More results can be found in a separate paper focusing on user studies (Chai, et al. 2001).

5.3 Deployed Pilot

Prior to the deployment of the system as a pilot, in addition to the usability user studies, a set of rigorous system testing (including express testing and server load balancing testing) was conducted to test the robustness and stability. The pilot is running on an IBM Websphere server with the servlet engine. The backend product data is stored in a DB2 database. Real-time data propagation (from the product database to the extended database used by the NLSA) is supported through staging servers. During the pilot (launched in December 2000), we are collecting information about real user interactions. In particular, we keep logs of user natural language inputs, logical forms generated by the system, system responses, SQL database queries and user feedback (in the form of questionnaire about the pilot). We believe this information will help us further evaluate and improve the system.

6 Conclusions

This paper describes a system that provides natural language dialog capabilities to help users access e-commerce sites and find relevant information about products and services. The system leverages technologies in natural language processing and human computer interaction to create a faster and more intuitive way of interacting with websites. By combining traditional AI rule-based technology with taxonomy mapping, the system is able to accommodate both customer needs and business requirements. Our studies showed that dialog-based navigation is preferred over menu-driven navigation (79% to 21% users). In addition, our studies confirm the efficiency of using natural language dialog in terms of the number of clicks and the amount of time required to obtain the relevant information. Comparing to a menu-driven system, the average number of clicks used in the natural language system was reduced by 63.2% and the average time was reduced by 33.3%.

In these studies we learned that the current internet keyword search engines have created a "search culture" which is widely accepted by most internet users. As a result, many users are accustomed to typing keywords or simple phrases (the average length of a user query was 5.31 words long with a standard deviation of 2.62). Analysis of the user queries reveals the brevity and relative linguistic simplicity of their input; hence, shallow parsing techniques seem adequate to extract the necessary meaning from the user input. Therefore, in such context, sophisticated dialog management is more important than the ability to handle complex natural language sentences.

From a historical perspective, users have experienced different interaction styles ranging from command-driven and form-fill applications to question-answer sequences, menus and natural language dialog interaction. Although naturalness is one of the winning points for natural language dialog, it also faces serious challenges. For novice users, a conversational system by itself may be overwhelming and it may indeed be quicker to use a menu-driven system. For an experienced user, on the other hand, the amount of typing may be a drawback and browsing may be the best and quickest way to navigate. Ultimately, in order to satisfy different user needs, the natural language dialog navigation and the menu-driven navigation should be combined. While the menu-driven approach provides choices for the user to browse around or learn some additional information, the natural language dialog provides the efficiency, flexibility and natural touch to the user's online experience.

Moreover, in designing NL dialog based navigation, it is important to show users that the system does understand his/her requests before giving any recommendation or relevant information. Users remarked in our studies that they appreciated the recommended results because the system offered some explanation. This feature allows the user to "trust the system." Good navigation aids can be provided by summarizing the user's requests by paraphrasing it using context history, or by engaging in meaningful conversations with the user. Our studies found that robust natural dialog had a very big influence on the user satisfaction – almost all of the respondents appreciated the additional questions prompted by their input and the summary of their previous queries.

We believe that conversational interfaces offer the ultimate kind of personalization. Personalization can be defined as the process of presenting each user of an automated system with an interface uniquely tailored to his/her preference of content and style of interaction. Thus, mixed initiative conversational interfaces are highly personalized since they allow users to interact with systems using the words they want, to fetch the content they want in the style they want. Users can converse with such systems by phrasing their initial queries at a right level of comfort to them (e.g. "*I am looking for a gift for my wife*" or "*I am looking for a fast computer with DVD under 1500 dollars*"). Based on our results, we conclude that conversational natural language dialog interfaces offer powerful personalized alternatives to traditional menu-driven or search-based interfaces to websites. For such systems, it is especially important to present users with a consistent interface integrating different styles of interaction and to have robust dialog management strategies.

Acknowledgement

We would like to thank Catherine Wolf for the input on user testing and user interface design, Jimmy Lin for the contribution on the first version of the system and first user study, Prem Mellville for implementing tools for data management and survey analysis and our colleagues from the Conversational Dialog Systems group for illuminating discussions.

References

Aggarwal, C.; Wolf, J.; and Yu, P. 1998. A framework for the Optimizing of WWW Advertising, Trends in Distributed Systems for Electronic Commerce. *LNCS 1402*, Lamersdorf and Merz Eds.

Androutsopoulos, I., and Ritchie, G. D. 1995. Natural Language Interfaces to Databases – an Introduction. *Natural Language Engineering* 1(1):29-81, Cambridge University Press.

Bray, T.; Paoli, J.; and Sperberg-McQueen, C. M. 1998. Extensible Markup Language (XML) 1.0. Technical Report, http://www.w3.org/TR/REC-xml-19980210, World Wide Web Consortium Recommendation.

Carpenter, B., and Chu-Carroll, J. 1998. Natural Language Call Routing: A Robust, Self-organizing Approach. In *Proceedings of the Fifth International Conference on Spoken Language Processing*.

Chai, J.; Lin, J.; Zadrozny, W.; Ye, Y.; Budzikowska, M.; Horvath, V.; Kambhatla, N.; and Wolf, C. 2000. Comparative Evaluation of a Natural Language Dialog Based System and a Menu Driven System for Information Access: a Case Study. In *Proceedings of RIAO 2000*, pp 1590-1600, Paris, France.

Chai, J.; Horvath, V.; Kambhatla, N.; Nicolov, N.; and Stys-Budzikowska, M. 2001. A conversational Interface for Online Shopping. To appear in *Proceedings of the First Human Language Technology Conference*.

Chu-Carroll, J., and Carpenter, B. 1998. Dialog Management in Vector-based Call Routing. In *Proceedings of the 36th Annul Meeting of the Association for Computational Linguistics*.

Huberman, B. A.; Pirolli, P. L. T.; Pitkow, J. E.; and Lukose, R. M. 1998. Strong Regularities in World Wide Web Surfing. *Science*, Vol. 280.

Jonsson, A. 1997. A model for habitable and efficient dialog management for natural language interaction. *Natural Language Engineering*, 3(2/3):103-122.

Muller, J., and Pischel, M. 1999. Doing Business in the Information Marketplace. In *Proceedings of the 1999 International Conference on Autonomous Agents*, Seattle, USA.

Radev, D.; Kambhatla, N.; Ye, Y.; Wolf, C.; and Zadrozny, W. 1999. DSML: A Proposal for XML Standards for Messaging Between Components of a Natural Language Dialog System. In *Proceedings of the AISB'99 (Artificial Intelligence and Simulation of Behavior) Workshop on Reference Architecture and Data Standards for NLP*. Edinburgh, England.

Saito, M., and Ohmura, K. 1998. A Cognitive Model for Searching for Ill-defined Targets on the Web – The Relationship between Search Strategies and User Satisfaction. In *Proceedings of 21st International Conference on Research and Development in Information Retrieval*, Australia.

Walker, M.; Fromer, J.; and Narayanan, S. 1998. Learning Optimal Dialogue Strategies: A Case Study of a Spoken Dialogue Agent for Email. In *Proceedings of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada.

Zadrozny, W.; Wolf, C.; Kambhatla, N.; and Ye, Y. 1998. Conversation Machines for Transaction Processing. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI) and Tenth Conference on Innovative Applications of Artificial Intelligence Conference (IAAI)*, Madison, Wisconsin, USA.