CHAYANI: a shopper's assistant

Sandip Sen, Partha Sarathi Dutta, and Sandip Debnath
Mathematical & Computer Sciences Department
University of Tulsa
E-mail: sandip-sen@utulsa.edu

Abstract

Shopping on the Internet has become a convenient way of purchasing commodities of choice. A key contribution of agent technology will be to develop personal agents that can assist users by generating, filtering, collecting, or transforming information available on the Internet. We believe that knowledgeable agents can increase the productivity and satisfaction on online shoppers by suggesting query reformulations to obtain products and services of particular interest to the user. A consumer interested in a particular product class, e.g., a consumer electronic item, can use such an agent to locate a desired product fitting a specified budget from one of several online stores. Apart from finding good deals on a product, such a "buyer's agent" can also suggest a higher quality alternatives for even a slightly higher price. We identify the types of query reformulation that will be particularly useful for online shoppers. We characterize the information requirements of a domain ontology that enable flexible and effective query reformulations. We also demonstrate the feasibility of this approach by example interactions with a shopper's assistant agent, CHAYANI, that we have developed for the consumer electronics domain.

1 Introduction

The past few years has seen widespread growth of e-commerce and shopping on the Internet. Internet sites like *Amazon.com* provide the buyer with a variety of items at competitive prices. The security of online transactions and the reliability of information provided and the products purchased, however, are of concern to the buyer. Another issue of importance to an online consumer, especially a non-expert, is the abundance of information available at various e-commerce sites. An average user can easily be overwhelmed by the volume of information which is often incoherent, confusing, misleading, and can be more of a hindrance than assistance to the consumer.

The above observation suggests a fruitful application area of agent based systems to enhance the shopping experience of online consumers. The goal of "buyers's agents" would be not just to retrieve information from the Internet based on user queries, but to be able to educate the consumer about alternative or related products and services. The use of agent based technology to address the information processing problem has received the attention of both application developers and system designers (Jennings *et al.*, 1998). Agents are viewed to be a useful metaphor both for developing desktop software designed to assist a particular user (Maes, 1994), as well as for Internet-based server-side software that enables e-commerce (Guarino *et al.*, 1999; Nwana *et al.*, 1998).

A different class of useful agent application has been recognized with the increase in the number of online consumers. Agent based systems allow merchants to tailor the presentation of their merchandise and/or services to the users. This is an effective way for a web-merchant to attract and retain customers (Maes *et al.*, 1999). Most of these agents act as "seller's agents", i.e., their goal is to push merchandise on behalf of the business they represent. We, however, are primarily interested in deploying "buyer's agent"s, whose objective is to serve the buyer's interests. We visualize our "buyer's agent" to be agents "higher up in the food chain" (Etzioni, 1997) which can in turn query current information sources on the Internet.

A "buyer's agent" is viewed to be an intelligent entity with substantial knowledge of the domain of interest to a buyer and is designed to help a buyer make judicious choices in selecting products or services. The main goal of the agent is to identify the buyer's interests, search the information repository based on this information, filter the search results and suggest relevant alternatives based on the search results. Another view of the function provided by the buyer's agent is to help the user rephrase, relax, or reformulate the queries to obtain most satisfactory search results. To be able to provide this functionality, the agent will require a well structured representation of domain knowledge. This is accomplished by building an ontology of the domain that captures semantic and structural relationships between domain features. An ontology in this domain is constructed by a domain expert who has substantial knowledge of the products and their interrelations. The ontology is used extensively by the agent to provide shopping suggestions to the user. Such suggestions can help reduce shopping time or identify products or product categories of particular interest to the user.

Traditional marketing research has identified a *consumer buying behavior (CBB)* model to represent such aspects of electronic commerce like consumer behavior regarding buying and using goods and services (Guttman *et al.*, 1998). With the rapid increase of business-to-consumer exchanges over the Web, the applicability of agent-systems at various level of the CBB model has been studied. Examples of such commercially deployed or prototype agent-based systems are Kasbah, Bargainfinder, Tango, Tete-a-Tete, etc. The technology used encompasses content and constraint based search for an item, online catalog systems that provide hyperlinks to lists of items, virtual shopping stores in 3D (Bryan & Gershman, 1999), etc.

In this paper, we present a particular buyer's agent, CHAYANI, that handles the product brokering stage of the CBB model. CHAYANI is designed to aid consumers in finding appropriate consumer electronics items that are of interest to he user and fit their budget. It accepts as input a query for a consumer electronic item. The query can be a generic one like "portable audio" or a more specific one like "APS camera". The user is also required to specify a target price for the item. Given these two user-selected constraints, our agent searches the product listings of two online stores *amazon.com* (Amazon) and *vstore.com* (Vstore). It can then provide available products, recommend related products of better quality, suggest possible query reformulations if no or few matches are returned, etc. The goal of CHAYANI is to be a knowledgeable well-wisher that can inform and educate while completing user requests.

In the following we first present, in more detail, the motivations for our the development of an online shopper's assistant. We then argue for the kind of knowledge required to develop effective shopper's assistants. This discussion leads to our design of the architecture of CHAYANI. We then present examples of query reformulations with our implemented system. We conclude the paper with a discussion of relevant work and further possibilities of systems like CHAYANI.

2 Enhancing the shopping experience

A recent survey in Business Week magazine identifies the following three relevant problems to be among the top ten pet peeves for on-line shoppers during the last holiday season: lack of choices, lack of gift ideas, lack of relevant information. This suggests the necessity of our proposed agents be able to suggest alternate products as argued in the Introduction section.

The frustration experienced by a user during online shopping can be due to failed search or retrieving too many items. The goal of a "buyer's agent" is to alleviate such user inconveniences. Additionally, the agent can suggest alternative products that might be of interest to the user. We list typical scenarios faced by an average user during online shopping that we would address later in this paper:

No match found for a given query: Let us take the case of a consumer looking for a *CD player* at a price of $50 at an online store. It might be that the online store that the consumer is visiting has *CD payers* with lowest price of \$65. The typical responses to such a query: “No match found” is both uninformative and frustrating for the consumer as it gives no guidance for changing the search. We believe that a shopper’s assistant agent should, in such cases, recommend alternate products and specify the increase in price limit required to obtain a match.

User query returns too many matches: Now we consider the case when a user submits a very general query or specifies a high price limit. Let us take the example when a customer is looking for a *camcorder* with a budget of \$1000. This is a very general query since *camcorder* is a broad category that includes several sub-categories like *8mm, digital, Hi-8, VHS* and *VHS-C*. The price limit set is also quite high. So, the user query will return most of the items under all the above sub-categories. The user when presented with the large list of items finds it difficult to choose one that fits his/her requirements without any additional guidance. The shopping assistant can inform the user of the different sub-categories with their relative quality orderings, and ask the user to resubmit the query for a sub-category.

Recommending alternatives: Even if the user query returns a fair number of results, the user may benefit from suggestions about alternate products or product categories. If we consider the situation when a user is looking for a *19” TV* and has a budget of \$200. The search returns a fair number of *19” TVs* that match the user budget. The agent in such a case can, for example, inform the user that there are also *19” TV VCR combinations* available within the specified budget. This information may allow the user to obtain a product that he/she was either unaware of or was interested in but had thought would be much more costly.

3 Domain Categories and their Relationships

From the discussion above, it is evident that the kinds of reformulation that an agent can perform is critically dependent on the types of information encoded in the domain ontology available to it. In particular, the category information is crucial to specializing or generalizing a query.

Another key aspect of query reformulation requires the availability of relationship information between sub-categories of a given category. Consider the *query reformulation on failed search* example above. The query reformulation or alternative suggestion strategy outlined assumes that the sub-categories of portable audio products are ordered by quality. This enables the agent to quickly suggest a lower quality substitute that fits the user budget.

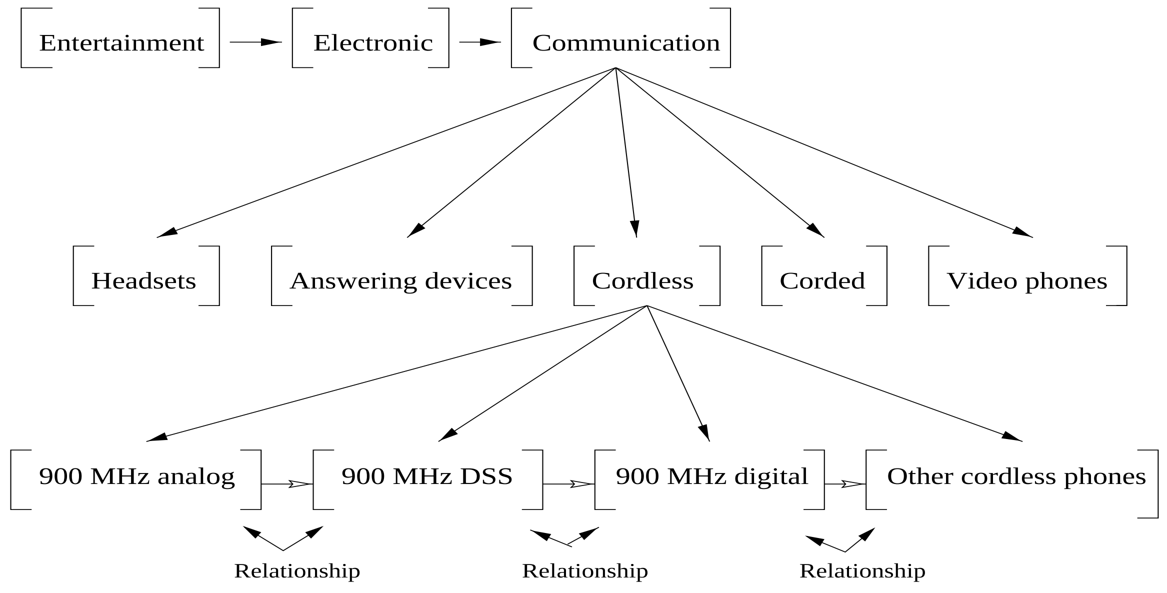


Figure 1: Section of ontology on consumer electronics products.

We now present a categorization of domains based on the availability of such ordering information between sub-categories of a given category. This ordering is essentially partial in nature.

We define *ordered ontologies* to be those ontologies which specifies partial order relations on the sub-categories of a given category. The partial order relation will be defined on a particular attribute or feature of the products, e.g., quality. It is also possible that multiple partial order relations be defined on the same ontology based on different attributes, e.g., quality, price, popularity, etc. This means product sub-categories of a given category can be compared. This enables substitution of products from one sub-category with that of a higher/lower/equivalent product from another sub-category. For example, instead of a portable CD player, an agent might recommend a higher quality MP3 player for a slightly higher price or a lower quality walkman that fits the budget of the user. Ontologies that do not contain such ordering information will be referred to as *unordered ontologies*.

We contend that as the number of partial order relations defined in a domain grows, the buyer's agent will be able to perform more useful query reformulations. As such, the capability of a buyer's agent will be directly proportional to the amount of order information inherent in the underlying domain ontology. In the base case of *unordered ontologies*, i.e., with no ordering information, only certain kinds of reformulations, e.g., specializing an overly general query, can be effectively performed.

We now discuss three problem domains as instances of highly ordered, simple order, and unordered domains:

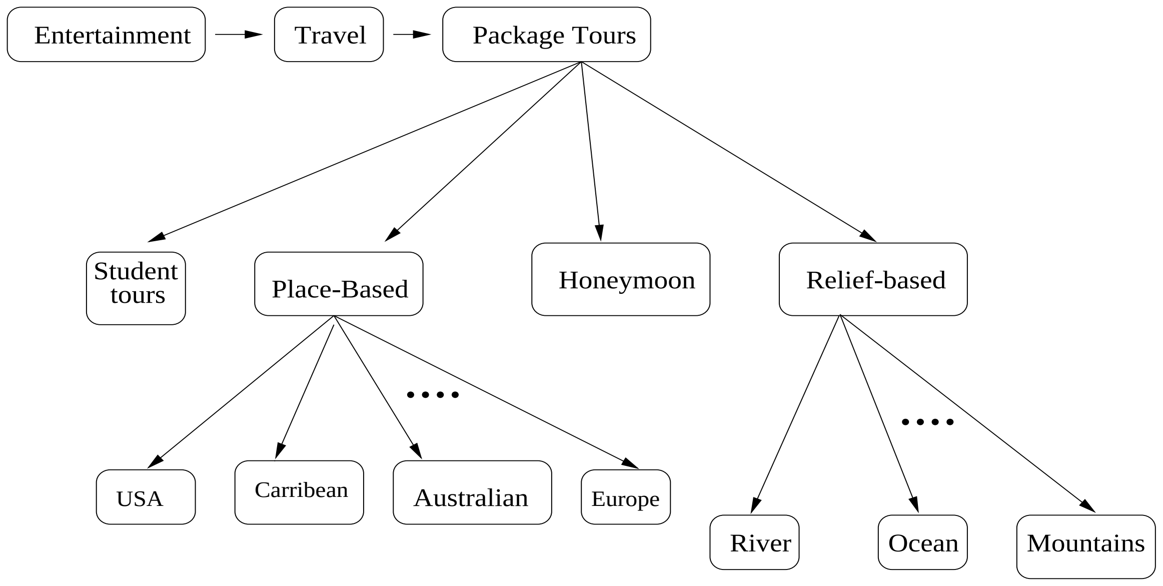


Figure 2: Section of ontology on package tours.

Highly-ordered domain: Such a domain will enable the development of ontologies with partial order relations defined over multiple attributes. The relations are independent of user classes. Consider the domain of consumer electronics. We present a section of the ontology in Figure 1. In this domain, partial ordered relations can be defined for each of the following attributes: quality, price. Moreover, these classifications are user-independent, i.e., the partial order relations are not dependent on the type of user querying the system. We know that a 2.4 GHz phone performs better than 900MHz phone, 25 channel is better than 10 channel and so on. Hence, if the agent finds a similar product with better features like 25 channel for same or slightly higher price, it can suggest this as an alternative.

Simple-ordered domain: Such a domain will enable the development of ontologies with one or few partial order relations. The relations may be dependent on user classes. Consider the domain of packaged vacation tours. We present a section of the ontology in Figure 2. In this domain, partial ordered relation can be defined over attributes like popularity, price, etc. However, the relations may be specific to user classes. For example, for cruise packages, Alaskan cruises may be preferred over Bahama cruises by older users and the reverse is true for young adults. To use such an ontology effectively for reformulation would require the identification of the user class to which this user belongs. This step itself can introduce error, and, coupled with the limited ordering information available, makes such domains less effective for ontology based query reformulation.

Partial orders over multiple attribute domains raises an interesting issue. Two products

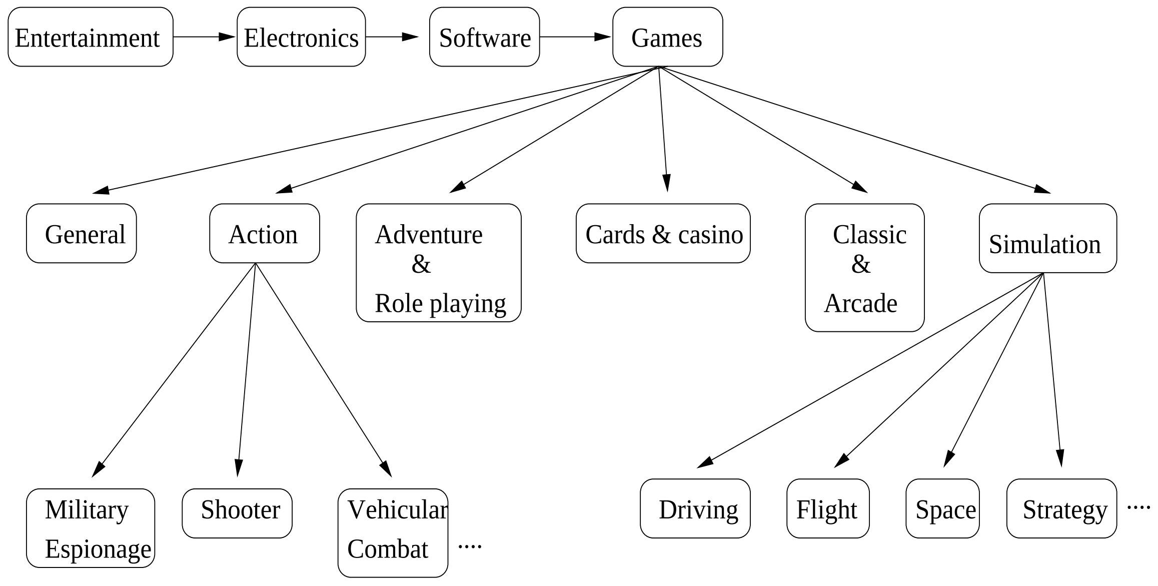


Figure 3: Section of ontology on video games.

may be ranked differently by two or more attributes. For example tour package A may be preferred over tour package B if popularity is concerned, but the opposite is the case if we consider the price attribute. While recommending alternatives to a particular product, CHAYANI can be made to select ordering along each dimension independently, or use a combination of all dimension. In that case, CHAYANI would first form the set of all products that are on the pareto frontier of the product under consideration, i.e., products which are not better or worse on all dimensions. Alternative suggestions will be made to the user from this pareto set.

Unordered domain: Such a domain will have ontologies without any meaningful, consistent, user-independent partial order relations. Consider the domain of software video games. We present a section of the ontology in Figure 3. In this domain, it is not clear how to effectively order sub-categories of a given category. For example, if a user query for flight simulation software at a particular price returns no result, it is not clear which other simulation software should be suggested as there is no well-understood price or quality relationships between these sub-categories. The more general category of "simulation" can be suggested, but this amounts to a different type of reformulation as we have discussed above. As such, we conclude that "unordered" domains are probably least amenable to ontology-based query reformulation.

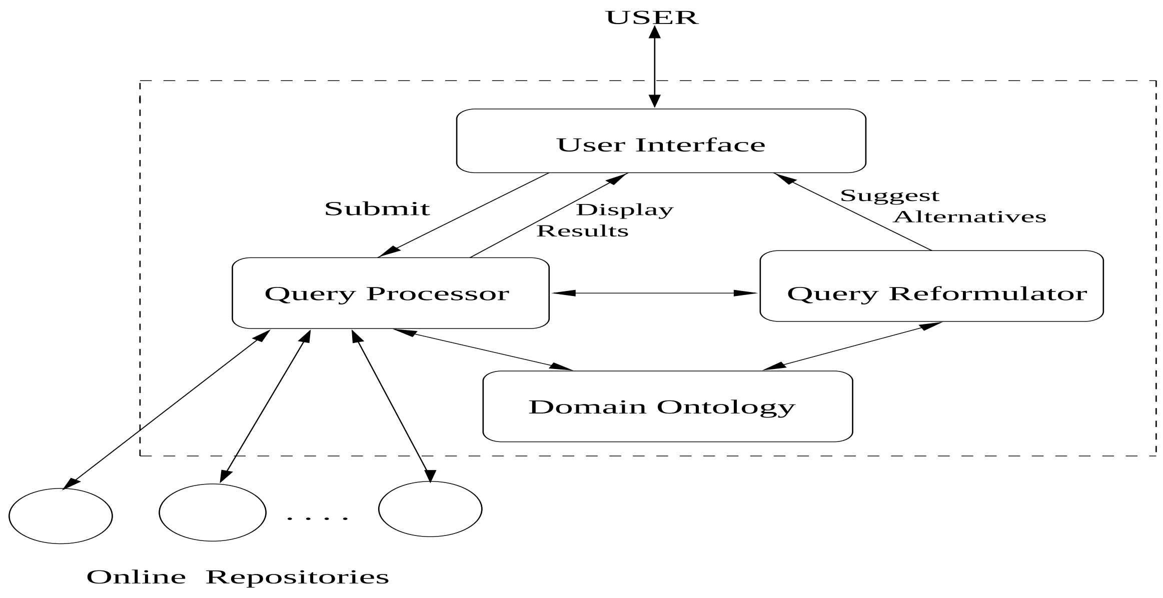


Figure 4: Architecture of CHAYANI.

4 The shopping agent architecture

The architecture that we have used for the agent system we have developed consists of four core modules. Figure 4 shows the architecture along with the online repositories that are accessed by our system to process and reformulate queries. The four modules are described below:

User Interface: The user interface is a form based Web application through which the user submits queries to our agent and receives response from the agent. The query for purchasing an item can be provided by a generic description of the item or can be selected from a list of several consumer electronic items, that our agent provides in advance. In addition to the item name, the user is allowed to provide the price preferred for the searched item. The user interface forwards the query to the *query processor* module and displays the corresponding results to the user. The results include list of retrieved items together with the links to their respective sites, list of related items that might match user preferences, a hierarchy of items that might help the user select a more specific product if he/she had made a very general query and suggestions to the user for a possible better quality product at a different price range. It should be noted that the results actually displayed, given a query, depends upon the nature of the query (general/specific), availability of the item searched for and other related items at the online repositories that the agent searches and, price information available for these products. These results help the user to begin a new search or modify the original query using the suggestions.

Query Processor: The query processor is responsible for analyzing the naive query submitted by the user to construct an effective search. It has to identify the proper item name if the user had entered a generic item description and relate that with the existing items in the system. The processor accesses the *Domain Ontology* module to extract information like the sub-categories subsumed (if any) by the search item and items that might be related to the search item. Having these information, the processor accesses online repositories for the sites where the search item(s) are located. In the system that we have developed, we access two online repositories *amazon.com* and *vstore.com.* The item information and their links that are retrieved from these repositories are transfered to the *query reformulator* module which makes query reformulation decisions. CHAYANI accesses the online repositories using the `java.net` package. The input stream containing HTML pages of product information is subsequently parsed using our own custom made Java classes to extract key product details (e.g., price range) that would be useful to the query reformulator to generate recommendations. An alternative (and, more standardized) way of implementing this functionality would be to have an XML document structure defined for the online sites. Having such a representation, the task of parsing HTML pages could be simplified using a standard XML parser such as JDOM. Nevertheless, it is difficult to have a generic representation for web sites that have completely arbitrary construction patterns.

Query Reformulator: This module provides the key functionality of the system. The *raw* information provided by the query processor about the item being searched is analyzed by this module for suggesting alternate queries in the following manner:

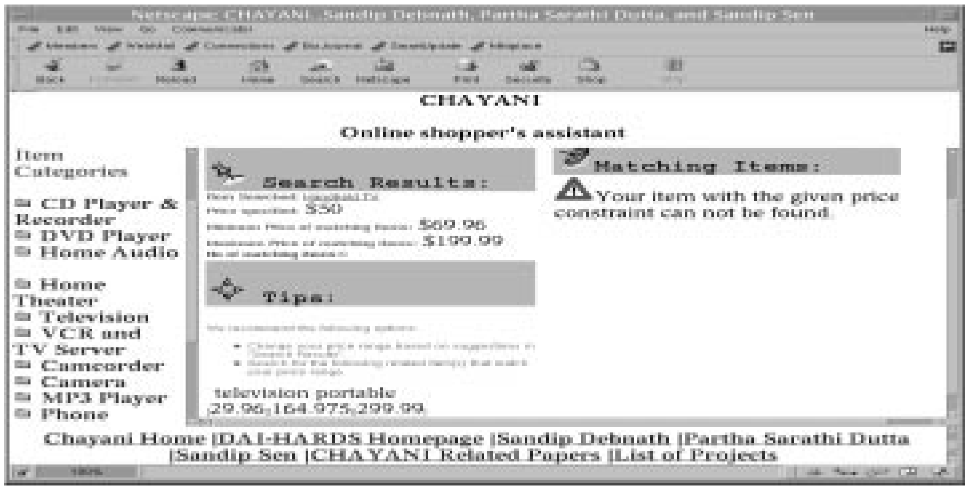
- One case is to deal with situations where a search has not returned any items, or has returned too few items. In this situation the reformulator accesses the *domain ontology* to gather information about items related to the search item. It also retrieves information about the price range of the item searched. The reformulator then generates suggestions like requesting the user to raise the price to get the cheapest available item in the store that matches the given query and also suggesting items that might be considered in place of the specified item and which is within the user's budget.

- Another scenario is the reverse of that discussed above, when a query returns too many items which makes it difficult for the user to make a good selection. Typically this happens when a general query is presented that matches many similar items. In this situation the reformulator again takes the help of the *domain ontology* and generates suggestions like providing the user the hierarchy of items that are subsumed by the search item together with some ordering information on these categories. The user can select one of the subsumed categories that matches his/her preferred group of items better and retrieve items that are closer to his/her preference.

- In the case when a search for an item returns a fair number of the products and they match the user preference well, the reformulator would still provide the user information about the item categories that are close to his/her preference and that satisfy his/her budget limit. This achieves a user friendly environment for online shopping, and attracts the user toward items that he/she might not have noticed but that satisfy his/her preferences.
- Another useful functionality of CHAYANI is to suggest accessories and other associated gadgets that can be used in conjunction with the item searched for. This information is stored in the domain ontology and can be recommended if the additional cost does not exceed the stated budget significantly.

Domain Ontology: The *query reformulation module* performs the key function in our system. But its functionality is depended on the *domain ontology*. This ontology is the representation of the interrelationships among the items present in the domain of consumer electronic goods. It is both a hierarchy that represents how a major category subsumes several sub-categories on the ground of similarity of their properties as well as a representation about the partial ordering among the items, that are not under the immediate major category, on the basis of attributes like price and quality.

In the current implementation of CHAYANI, we have constructed the domain ontology manually after analyzing the categories and the interrelationships among different consumer electronics products at various online sites. In future, we aim to use one of the standard ontology editing tools — KAON (kaon), OilEd (oiled), and Prot*égé* (protege), among others — to partially automate the construction process along with inconsistency checks. Note that full automation of ontology editing (addition and/or deletion of classes and relationships) is a separate research issue in its own right. Thus, in CHAYANI, we use manual editing. We could, however, implement a proactive notification service that alerts the ontology editor whenever a product category, not in CHAYANI's current catalogue, becomes available at one of the stores. Upon receiving an alert, it would be the editor's responsibility to apply the suitable changes in CHAYANI's ontology if necessary.


CHAYANI

Online shopper's assistant

Item Categories

- CD Player & Recorder
- DVD Player
- Home Audio
- Home Theater
- Television
- VCR and TV Server
- Camcorder
- Camera
- MP3 Player
- Phone

Search Results:

[illegible] $50

[illegible] $69.96

[illegible] $199.99

Tips:

[illegible]

television portable
29.96,164.975,299.99

Matching Items:

Your item with the given price constraint can not be found.

Chayani Home |DAI-HARDS Homepage |Sandip Debnath |Partha Sarathi Dutta |Sandip Sen |CHAYANI Related Papers |List of Projects

Figure 5: Example of query reformulation on failed search.

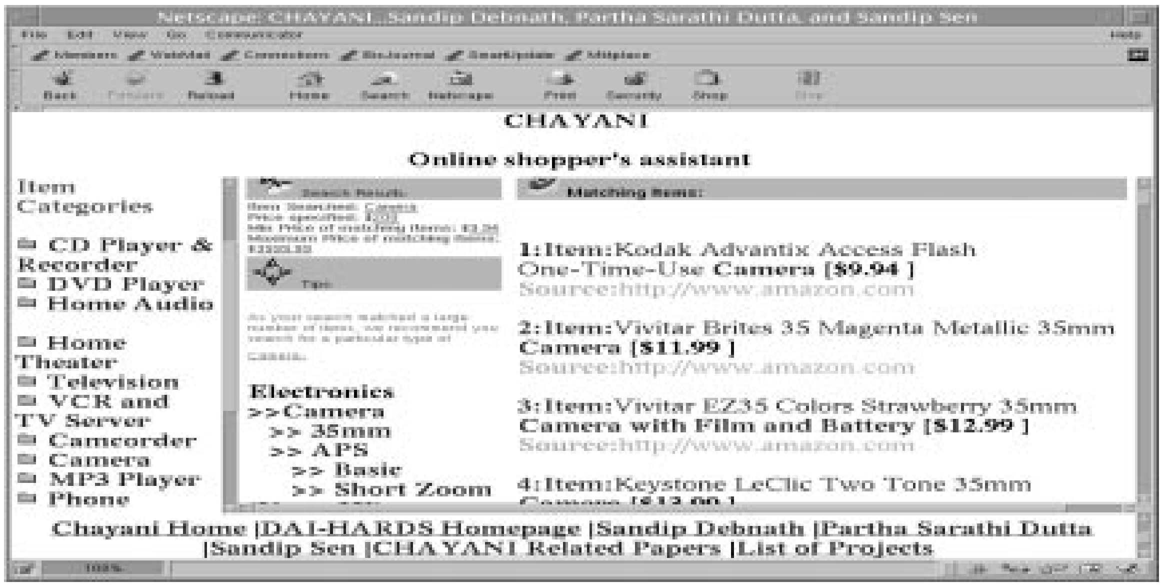
5 Experiment with shopper's assistant

We have developed an agent based system, CHAYANI, that implements the reformulation techniques discussed above. We have restricted our work to a *highly ordered domain* of consumer electronics. This feature makes it attractive to be cast into a complete ontology structure that can be effectively used for query reformulation. This structure is created after gathering information about the consumer electronic products from the online stores *amazon.com* and *vstore.com*.

The interface allows a user to enter the product category along with the price limit of the item that he/she is looking for. Once the query is submitted, our agent looks for the item in the product lists of two popular online stores, *amazon.com* and *vstore.com*. On the basis of the results of the initial search, the query reformulation function is invoked as follows:

Query reformulation on failed search: Suppose no match is found when a buyer wants to buy a handheld TV at $50 (see Figure 5). CHAYANI suggests that the price-range for hand-held TV is from $69.96 to $199.99. It also recommends to the user to consider the related item of portable TV sets, which are of a lower quality than handheld TVs, but sells in the price range of [$29.96,$299.99] with an average price tag of $164.9.

Reformulating an overly general query: If a consumer searches for a camera within $200 a lengthy list of matches are returned (see Figure 6). Faced with a very large list of options the user is likely to overlook preferred choices. In the ontology for cameras, there are relationships among camera categories. The agent can present the sub-categories of cameras, e.g., 8 mm, APS, SLR, digital cameras, etc. and ask the user to narrow down the search


Netscape: CHAYANI, Sandip Debnath, Partha Sarathi Dutta, and Sandip Sen

CHAYANI

Online shopper's assistant

Item Categories

- CD Player & Recorder
- DVD Player
- Home Audio
- Home Theater
- Television
- VCR and TV Server
- Camcorder
- Camera
- MP3 Player
- Phone

Search Result:

Tips:

[illegible]

Electronics
>>Camera
>> 35mm
>> APS
>> Basic
>> Short Zoom

Matching Items:

1:Item:Kodak Advantix Access Flash One-Time-Use Camera [$9.94]
Source:http://www.amazon.com

2:Item:Vivitar Brites 35 Magenta Metallic 35mm Camera [$11.99]
Source:http://www.amazon.com

3:Item:Vivitar EZ35 Colors Strawberry 35mm Camera with Film and Battery [$12.99]
Source:http://www.amazon.com

4:Item:Keystone LeClic Two Tone 35mm

Chayani Home |DAI-HARDS Homepage |Sandip Debnath |Partha Sarathi Dutta |Sandip Sen |CHAYANI Related Papers |List of Projects

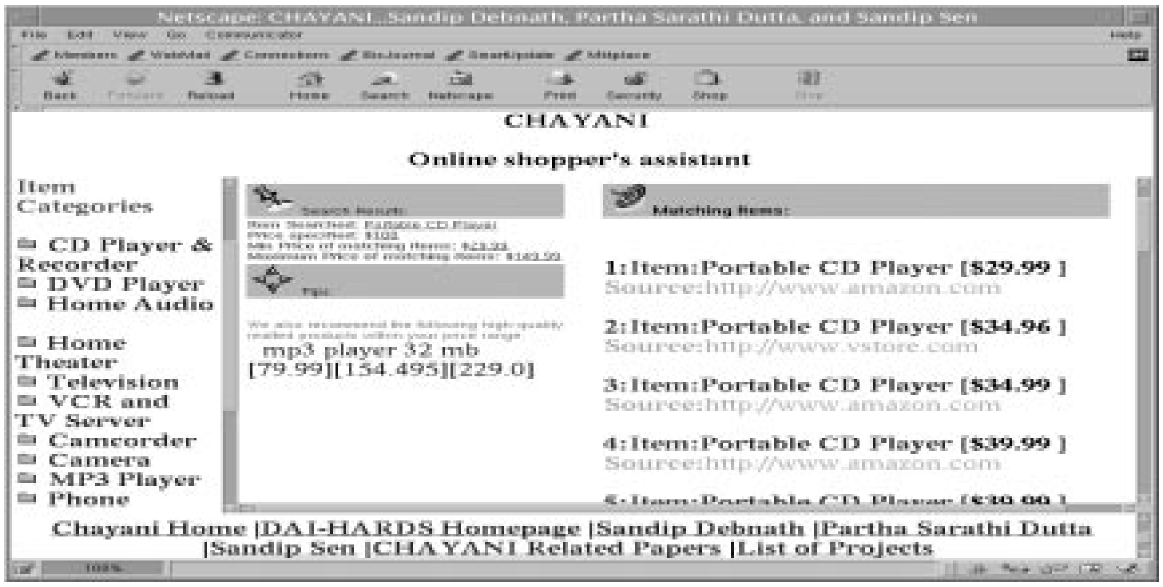
Figure 6: Example of query reformulation on too many results.

to the sub-category of interest as shown in Figure 6. This suggestion may allow the user to sufficiently constraint the search to select products to meet his/her requirements.

Suggesting alternate products: Even when the query of a user returns a reasonable number of matches, alternate product suggestion can be useful. Figure 7 presents the scenario where the user is looking for a portable CD player with a budget of $100. In addition to finding the available products, CHAYANI suggests that the user may want to consider a higher-quality MP3 player that fits the user's budgets. This suggestion can enable the user to purchase a more preferable product. The purpose of the suggestions are to inform about available options without imposing on the user.

Suggesting accessories: CHAYANI can also suggest accessories and gadgets as exemplified in the situation presented in Figure 8. In this scenario, the user is looking for an 8mm or Hi-8 camcorder under $400. In addition to listing the available products in that price range, CHAYANI recommends associated gadgets like charger and battery. Such recommendations can come in handy without requiring the consumer to go through additional searches. At times, this facility will inform the consumer about accessories that they may not be aware of. The consumer can then make a more informed choice about what to purchase.

The above represents a sampling of some interactions with CHAYANI. So far, we believe CHAYANI does a proficient job of handling the user queries in the consumer electronics domain. Extending the system so that it has a richer domain ontology and access to more



CHAYANI

Online shopper's assistant

Item Categories

- CD Player & Recorder
- DVD Player
- Home Audio
- Home Theater
- Television
- VCR and TV Server
- Camcorder
- Camera
- MP3 Player
- Phone

Search Results:

Item Searched: Portable CD Player
Price specified: [illegible]
Min Price of matching items: [illegible]
Maximum Price of matching items: [illegible]

Tips:

We also recommend the following high-quality related products within your price range

mp3 player 32 mb
[79.99][154.495][229.0]

Matching Items:

1:Item:Portable CD Player [$29.99]
Source:http://www.amazon.com

2:Item:Portable CD Player [$34.96]
Source:http://www.vstore.com

3:Item:Portable CD Player [$34.99]
Source:http://www.amazon.com

4:Item:Portable CD Player [$39.99]
Source:http://www.amazon.com

5:Item:Portable CD Player [$39.99]

Chayani Home |DAI-HARDS Homepage |Sandip Debnath |Partha Sarathi Dutta |Sandip Sen |CHAYANI Related Papers |List of Projects

Figure 7: Example of query reformulation to suggest alternate products.

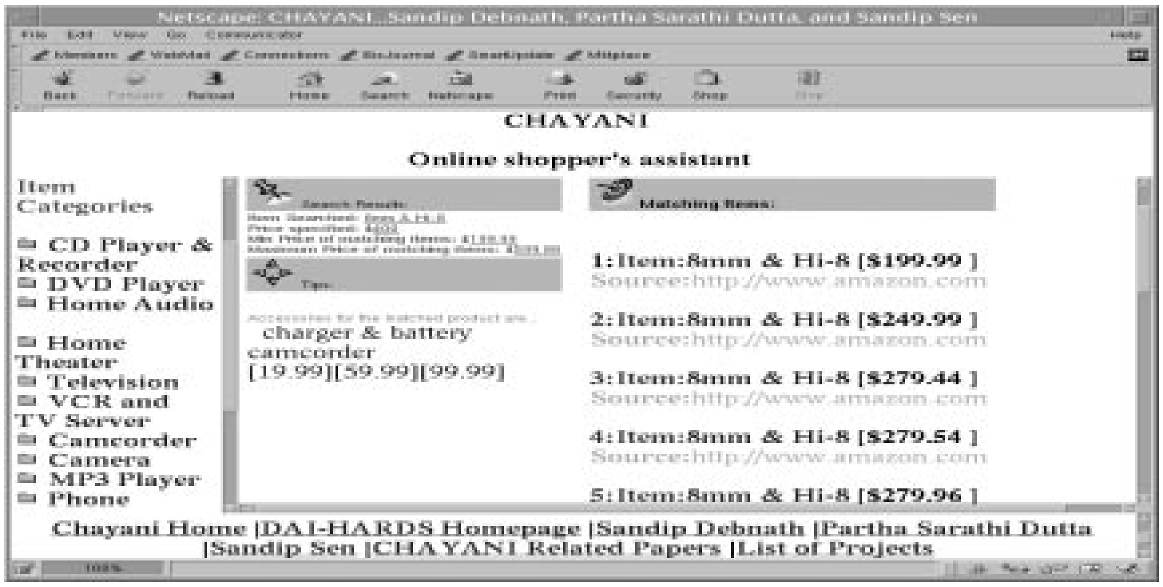
sites will further enhance its usefulness. CHAYANI can also be augmented by a learning component that learns both new product categories as well as new relationships between existing products in the ontology. As more web-site incorporate automated adaptation facilities including agent modules, CHAYANI should equipped to interact and negotiate with such entities to obtain more detailed information about particular products and more choices for product categories of interested to the user.

6 Related work

The increasing volume of online information calls for an intelligent automated entity to manage this information on behalf of the user. The burgeoning interest in agent-based systems and it's applicability has evolved a number of techniques for intelligent information retrieval.

One application of agent based systems is to increase the quality of information access from the Internet which has been growing rapidly to be a huge source of information. Typical search engines return too many links in response to user queries, most of which are irrelevant to the user. Some web-page recommender systems try to alleviate some of this problem by finding pages of interest to a user based on his/her query and usage patterns (Pazzani *et al.*, 1996; Rucker & Polanco, 1997).

Another approach used to facilitate user access to pertinent information is to trace the user profile at a Web store and customize the pages according to that profile (Ardissono



CHAYANI

Online shopper's assistant

Item Categories

- CD Player & Recorder
- DVD Player
- Home Audio
- Home Theater
- Television
- VCR and TV Server
- Camcorder
- Camera
- MP3 Player
- Phone

charger & battery camcorder
[19.99][59.99][99.99]

1: Item:8mm & Hi-8 [$199.99]
Source:http://www.amazon.com
2: Item:8mm & Hi-8 [$249.99]
Source:http://www.amazon.com
3: Item:8mm & Hi-8 [$279.44]
Source:http://www.amazon.com
4: Item:8mm & Hi-8 [$279.54]
Source:http://www.amazon.com
5: Item:8mm & Hi-8 [$279.96]

Chayani Home |DAI-HARDS Homepage |Sandip Debnath |Partha Sarathi Dutta |Sandip Sen |CHAYANI Related Papers |List of Projects

Figure 8: Example of query reformulation to suggest auxiliary products.

& Goy, 1999). In this approach the description of the store catalog is adapted to reflect the preferences of the users. The hypertextual pages are dynamically generated applying personalizing rules to user models developed according to user profiles. A related approach is that of recommender systems. These are used by E-commerce sites to come up with product suggestions. The recommendations that are common are based on the following factors: monitoring and analyzing past buying behavior of customer, customer demographics or similar products being sold by best selected sellers at online stores (Schafer *et al.*, 1999). Such recommendations help to personalize a web site in that they help the site to adapt itself to a customer. A different approach is that of "opportunistic exploration" (Bryan & Gershman, 1999). This concept is based on the premise that users have multiple and overlapping interests, exposure to appropriate items can appeal to a user's latent interests, active interests may be subdued if users are not exposed to their items of interest soon enough, and users prefer a simple navigation of online stores to search for their items. The purpose of the system is to influence the short and long term interests of users shopping at an online store by exposing them to a dynamic environment of items that are updated by interacting and assessing his/her interests on the fly.

More traditional approaches to online product recommendation have also been explored. The Yoda system combines collaborative filtering and content-based querying for providing product recommendations (Shahabi & Chen, 2003). Fast online recommendation is obtained by training the aggregation function offline using genetic algorithms based on data from, among other things, user navigation patterns. The Helpful Online Purchase Environment

(HOPE) system uses data mining to generate suggestions predicated both on the user and the content of the web page (Sun & Trudel, 2002). It uses tag fields with products to organize them into a hierarchical structure and uses a nearest neighbor algorithm to find related products based on the customers purchase history.

In the work that we present here, the usage of ontology has been shown to be critical for the query reformulation task. We also argue that the effectiveness of ontology-based query reformulation depends upon the partial ordered structure of the domain being used. The satisfaction of the user will be based on how efficiently the query reformulation helps the user in finding items of interest.

There are many different ways that ontologies can be used to build smarter agents. The most straightforward application of ontologies is to map a user query into a more appropriate form by using domain-specific terminology. This will allow, for example, to transform an ineffective query into a more productive one that returns results of interest to the user (Guarino *et al.*, 1999). In this content-based information retrieval from the Web, ontology has been used to increase recall and precision of retrieved pages. A relatively novel application of ontologies can be to aid opportunistic exploration of the product space (Bryan & Gershman, 1999) by interacting with the user. The application of linguistic ontology has been found to be effective in Web-based retrieval of software components (Borgo *et al.*, 1997). The software component is first encoded by an analyst into a "lexical semantic graph" structure of words, which is then converted to a graph of word senses. The query is then encoded into graph by the user and the search process that follows is that of a graph matching procedure, returning the features of the components that match the query.

7 Conclusions

The general concept of a buyer's agent is a very powerful one: the vision is that of a knowledgeable well-wisher helping the user to select the most satisfying product for the user. In this paper we point out the advantages of an agent coupled with a linguistic ontology in making a consumer more informed about available products. A consumer is assumed to have limited knowledge about the current market conditions. In this context, an intelligent "buyer's agent" is capable of educating the customer by suggesting alternatives, providing guidance to selecting a more appropriate query, and using underlying domain structure.

Our goal in this paper was to look beyond developing suitable ontological structures for domain specific search problems. Domain features that are essential to develop ontological constructs that can be used by agents to effectively guide users by reformulating queries have been identified and analyzed.

To demonstrate the feasibility of such ontology based query reformulation by a shopper's assistant agent, we chose the domain of online shopping of consumer electronic goods. We have incorporated the query reformulation functionalities mentioned here in our shopper's agent, CHAYANI, that acts as a smart interface between the user and Internet E-commerce sites *amazon.com* and *vstore.com*. We have used a rich ontology for the consumer electronics and demonstrated sample query reformulations to further user satisfaction.

It is necessary to further augment the usefulness of CHAYANI by enriching its domain ontology and having it retrieve information from more sites. More importantly, we plan to implement a version of CHAYANI for other domains that may have less structure and uniformity than the consumer electronics domain.

A significant enhancement to the effectiveness of buyer's agents or shopper's assistants is to design them as mixed-initiative systems or systems with adjustable autonomy. Such systems can better serve the user by engaging the user in a dialogue and thus obtaining a better understanding of the actual needs and preferences of the user (Rich *et al.*, 2001).

Acknowledgement: This work has been supported in part by an NSF award IIS-0209208.

References

Amazon.com. URL: http://www.amazon.com/.

Ardissono, L. & Goy, A. (1999). Tailoring the interaction with users in electronic shops. In *Proceedings of the 7th International Conference on User Modeling*.

Borgo, S., Guarino, N., Masolo, C. & Vetere, G. (1997). Using a large lingusitic ontology for internet-based retrieval of object-oriented components. In *International Conference on Software Engineering and Knowledge engineering*.

Bryan, D. & Gershman, A. (1999). Opportunistic exploration of large consumer product spaces. In *Proceedings of the First ACM Conference on Electronic Commerce*, 41–47, ACM Press, New York: NY.

Etzioni, O. (1997). Moving up the information food chain: Deploying softbots on the world wide web. *AI Magazine*, **18**, 11–18.

Guarino, N., Masolo, C. & Vetere, G. (1999). Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 70–80.

Guttman, R.H., Moukas, A.G. & Maes, P. (1998). Agent-mediated electronic commerce: A survey. In *Knowledge Engineering Review*.

Jennings, N., Sycara, K. & Wooldridge, M. (1998). A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems*, **1**, 7–38.

The KAarlsruhe ONtology and Semantic Web tool suite. Http://kaon.semanticweb.org/.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, **37**, 31–40.

Maes, P., Guttman, R.H. & Moukas, A.G. (1999). Agents that buy and sell. *Communications of the ACM*, **42**.

Nwana, H.S., Rosenschein, J., Sandholm, T., Sierra, C., Maes, P. & Guttman, R. (1998). Agent-mediated electronic commerce: Issues, challenges and some viewpoints. In *Proceedings of the Second International Conference on Autonomous Agents*, 189–196, ACM Press, New York: NY.

The OilEd Ontology Editor. Http://oiled.man.ac.uk/.

Pazzani, M., Muramatsu, J. & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. In *Proc. of the 13th National Conference on Artificial Intelligence*, 74–79, MIT Press/AAAI Press.

The Prot*égé* Project. Http://protege.stanford.edu/.

Rich, C., Sidner, C.L. & Lesh, N. (2001). COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, **22**, 15–25.

Rucker, J. & Polanco, M.J. (1997). Personalized navigation for the web. *Communications of the ACM*, **40**.

Schafer, J.B., Konstan, J. & Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the First ACM Conference on Electronic Commerce*, ACM Press, New York: NY.

Shahabi, C. & Chen, Y.S. (2003). An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distributed and Parallel Databases*, **14**, 173–192.

Sun, T. & Trudel, A. (2002). An implemented e-commerce shopping system which makes personal recommendations. In *Internet and Multimedia Systems and Applications (IMSA 2002)*, 58–62, IASTED/ACTA Press.

Vstore.com. URL: http://www.vstore.com/.