The Role of Semantics in Next-Generation Online Virtual World-Based Retail Store

Geetika Sharma, C. Anantaram, and Hiranmay Ghosh

Tata Consultancy Services, 249 D & E Udyog Vihar, Phase IV, Gurgaon -122015, Haryana, India
geetika.s@tcs.com, c.anantaram@tcs.com, hiranmay.ghosh@tcs.com
http://www.tcsinnovations.com

Abstract. Online virtual environments are increasingly becoming popular for entrepreneurship. While interactions are primarily between avatars, some interactions could occur through intelligent chatbots. Such interactions require connecting to backend business applications to obtain information, carry out real-world transactions etc. In this paper, we focus on integrating business application systems with virtual worlds. We discuss the probable features of a next-generation online virtual world-based retail store and the technologies involved in realizing the features of such a store. In particular, we examine the role of semantics in integrating popular virtual worlds with business applications to provide natural language based interactions.

Keywords: Virtual Environments, Natural Language Processing, Visual Semantics.

1 Introduction

Online web-based retail portals like eBay, Amazon etc., are rather popular for buying and selling used and new items. People shop online to find discounts and savings, save time in comparing product specifications and prices, and the sheer convenience of shopping from their homes or offices. However, online web-based shopping lacks the social aspect of shopping. A user of an online portal is usually shopping individually and without social interaction - a practice far removed from reality. It has been observed that although the majority of consumers still visit real-world malls to shop, they seem to acknowledge that malls serve other purposes than being just a shopping destination, such as watching a movie, shopping with a relative or friend, or attending events [5]. Market research has also shown that the Internet is not typically a place where consumers make impulsive purchases. Consumers utilize the web as a convenient, user-friendly means to browse shopping options, educate themselves on product choices and make informed purchase decisions.

Online virtual worlds are set to change online shopping by providing the necessary social aspect to shopping. Friends and relatives distributed across the globe can go shopping together by logging into the world at the same time and

F. Lehmann-Grube and J. Sablatnig (Eds.): FaVE 2009, LNICST 33, pp. 91–105, 2010.
© Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering 2010

visiting different retail stores. This allows instant feedback from the people whose opinions matter most to the consumer. Virtual worlds can also provide a three-dimensional visual interface to shopping. A consumer can look at a model of the item he wishes to buy from any angle, turn it around and see demonstrations of its features or how to use that item.

In this paper, we explore the technologies to seamlessly integrate a virtual environment with a retail application (such as a shopping application) in such a way that users can interact in a virtual environment, yet have the experience of real-world shopping. Virtual world interactions, at present, have largely been between humans, mostly as conversations in free format text, such as chat sessions, message sessions, and discussion platforms. In such a scenario, we would have to permit free format text interaction with a retail application system so that the user perceives a natural conversation. Moreover, such an interaction could lead to a dynamic, customized experience for the user.

We focus on one part of this problem –*how to handle the semantics to carry out free format text and image based interactions with a retail application in a virtual world, in order to get information from the application system and to carry out real-world transactions.* This problem can be further divided into two subparts- the first part deals with connecting the virtual world to a retail system outside its world and the second deals with the semantics to process and implement tasks requested by the user through natural language conversation and image-based interaction. We describe how to tackle both the subparts in the following sections. We have taken an example of a virtual retail store to illustrate the technologies and their issues.

In section 2, we describe a next-generation virtual world-based retail store. In Section 3, we describe a prototype system called NATAS which is a text-based natural language interface to business applications. In section 4, we show how business systems may be integrated with virtual worlds using NATAS as an interface. In section 5, we discuss the role of visual semantics in retail stores and conclude in section 6.

2 A Next Generation Online Virtual World Retail Store

A significant number of people today physically go to a retail store. Prospective shoppers may interact with the salespersons and may also buy some items from the store. They have a social interaction with the salespersons (such as, they may ask details about the items in the store or get help in finding a suitable item for their need, etc.) and physically see and examine the items in the store. However, every time a shopper visits a store, he spends a significant amount of time and energy to shop. Moreover, the store usually remains the same for each shopper who visits it - there is no personalization of the store for a shopper; that is, based on a shopper's particular need on a day, the store's layout and offerings do not change. This may lead to poor stock location and poor stock promotions from a shopper's point of view.

Online shopping, on the other hand, offers mechanisms through which shoppers browse and shop for items on the Internet without visiting a retail store. However, in such an interaction, which appears rather impersonal, a number of "desirable" features are missing. From a shopper's perspective, the first and foremost is the lack of social interaction. There is no salesperson / store-assistant to help the shopper. Second, there is a lack of personalized services (such as guiding the shopper to the appropriate product shelves), or guided shopping facilities such as "*ask a pharmacist*". Further, there is no mechanism for the retail shop to dynamically alter itself depending on the conversation between the shopper and the salesperson. In the context of these shortcomings, we discuss some of the features of a futuristic online virtual world-based retail store.

In an online three-dimensional virtual world, shoppers can visit a store by logging in with their avatars, going to the store's location in the virtual world. A shopper could converse with an *avatar* (a logged in salesperson) or a *virtual store agent* (a natural-language enabled *chatbot*). The conversation can be in the form of text-based input or through speech. Moreover, the store items could be altered for a particular shopper based on the conversation between the shopper and the avatar or agent. New schemes and promotions deemed suitable for the shopper can be displayed. Virtual environments, unlike real settings, take seconds to alter, so the time and cost benefits of this approach, when compared with traditional mechanism is enormous. This will also enable store planners to provide a rapid means of presenting their latest store and item concepts to potential shoppers catering to their taste.

An online virtual world store is a combination of graphical models of objects such as doors, windows, walls, lights etc. and objects that are on sale inside the store. These objects can be scripted to behave in a particular way when an event occurs. For example, walls could be scripted to change their wallpaper depending on what the customer is interested in. A virtual store also allows the store planners to adjust the heights of displays, the widths of aisles and the design of the background. There can be promotional videos playing on different screens in the store which change depending on the product that a user is interested in.

Retail planning in such a store allows the store to customize product placement in order to encourage shoppers to move to the hotspots along a predetermined path suitable to their tastes. This facility helps retailers explore store design, merchandising and product concepts based on consumer insights without ever having to change a thing in the real-world store. Shoppers would be able to present images of items that they would like to buy, and items that match such images (either fully or partially) can be immediately displayed.

Automated chatbots play an important role to interact with the wide variety of customers who would "*walk-in*" into the store. Natural language based interaction would be some of the important means of interaction for customers with chatbots. In the next sections, we examine some of the technologies to perform the above tasks. In particular we focus on two core technologies - text based natural language interaction, and role of visual images in retail stores and discuss the semantics in such a scenario.

3 Framework for Text Based Natural Language Interaction

A number of attempts have been made to build natural language interfaces to business applications; some of them are reviewed here. Sybase Inc. has built a system called "Answers Anywhere" [6], to provide a natural language interface to a business application through a wireless phone, a handheld PDA, a customized console, or a desktop computer. The method is based on agents and networks. While the system shows promise, their approach does not involve ontology based querying or retrieval. Further, they do not handle semantic description of web resources, or traversal of the ontology graph. PRECISE NLI system [7] is designed for a broad class of semantically tractable natural language questions, and guarantees to map each question to the corresponding SQL. The problem of finding a mapping from a complete tokenization of a question to a set of database elements such that the semantic constraints are satisfied is reduced to graph matching problem. PRECISE uses the max-flow algorithm to solve the problem. While their work seems quite interesting, they restrict each question to start with "wh" token. NaLIX system [8] discusses the construction of a generic Natural Language query interface to an XML database. On the other hand TRIPS [9] enforces strict turn taking between the user and the system and processes each utterance sequentially through three stages – interpretation, dialogue management and generation. These restrictions make the interaction unnatural [10].

We describe a natural language interface system called NATAS [1] that allows end-users to interact with a business application by posing questions and invoking tasks in a natural language such as English. It is based on a framework that uses an explicit domain ontology described using semantic web technology like Resource Description Framework (RDF), Web Ontology Language (OWL) and SPARQL query language for RDF. NATAS parses an input natural language sentence and depending on the context, either a SPARQL query is formulated on the application data, or the relevant APIs of the application are invoked. The result thus generated is phrased into an English language sentence and displayed to the user.

3.1 Domain Ontology and Its Creation

NATAS relies on a domain ontology created using OWL, N3 an, to provide a natural language interface to a business application through a wireless phone, a handheld PDA, a customized console, or a desktop computer. The method is based on agents and networks. While the system shows promise, their approach does not involve ontology based querying or retrieval. Further, they do not handle semantic description of web resources, or traversal of the ontology graph. PRECISE NLI system [7] is designed for a broad class of semantically tractable natural language questions, and guarantees to map each question to the corresponding SQL. The problem of finding a mapping from a complete tokenization of a question to a set of database elements such that the semantic

constraints are satisfied is reduced to graph matching problem. PRECISE uses the max-flow algorithm to solve the problem. While their work seems quite interesting, they restrict each question to start with "wh" token. NaLIX system [8] discusses about construction of generic Natural Language query interface to an XML database. On the other hand TRIPS [9] nforces strict turn taking between the user and the system and processes each utterance sequentially through three stages – interpretation, dialogue management and generation. These restrictions make the interaction unnatural [10].

The framework works on an explicit ontology of the application domain. The ontology of the domain describes the domain terms and their relationships. The data in the business application system forms a part of the domain terms and their relationships in the ontology. This helps forms the main concepts of the domain and their relationships with a ¡subject-predicate-object¿ structure for each of the concepts.

We define three levels of the ontology - **Seed Ontology**, Application Ontology and Domain Ontology. The Seed Ontology describes the basic relations between domain terms that are present in the domain. For example in the Retail domain where details about all the items and their sales promotions are handled; facts like "item has discount", "item has modelno.", etc populate the Seed Ontology.

The application data (also termed as *static facts*) provides the actual data that is present in the system. The Ontology Generator takes in the Seed Ontology and application data and creates an instance of the Seed Ontology populated by the application data. This is called the **Application Ontology**. Next, the rules of the application domain are then evaluated together with the Application Ontology by a Rule Engine, such as Closed World Machine (CWM), to create the **Domain Ontology**. This Domain Ontology is used by the NATAS system to answer questions on / carry out the tasks of the domain.

We assume that all the instances of the objects in the domain are stored in a database associated with the application system in a relational form (such as [a R c]), for example [iPod price 20,000]. A relational database will store it as a set of tables with rows that have attributes (for example, say, we have table `Price` with `ItemId` and `PriceID` as fields and rows with values 160 and 212). We treat the data in the database as static facts of the domain. This data can be used for answering queries posed by a user.

3.2 Concepts of the Domain

The RDF file is read and a <subject-predicate-object> graph structure in created in memory. Once we have the domain ontology in memory, we can traverse it using the graph traversal functions to get the subject, predicate or object (or a combination of these). The set of class objects created in memory to represent each subject, predicate and object of the <subject-predicate-object> structure, form the concepts of the domain. This helps identify the concepts in the natural language sentence that the user inputs.

3.3 Parsing the Input to Identify Concepts

The input sentence in natural language is parsed by the Domain Parser for identifying the parts-of-speech in the sentence. This process identifies the proper nouns, common nouns, verbs, adjectives and adverbs in the sentence, and is called tagging. Further, the root words for each of the tagged words are determined. Once this is done, the tagged words with their root-words are then passed onto the Concept Manager to match against the domain concepts loaded in memory. The matching is carried out as an approximate match with a threshold greater than 75% between the words in the tagged input sentence and the concept and their synonyms. The concepts that match are flagged to indicate that they are referred by the user in the sentence (this is called *referred concepts*). The *referred concepts* are then used to identify that part of the ontology, which needs to be traversed. From the <subject-object-predicate> tuples in memory (treated as facts), the system tries to generate an answer for the referred concepts, or execute the relevant API of the application.

3.4 Handling Queries on the Domain

The query posed by the user is parsed and executed using SPARQL. Since the domain ontology is in RDF format, the general structure of the query is (<subject, predicate, object>). We identify seven types of queries for the subject-predicate-object (hence forth referred to as <s-p-o>) structure of our ontology; these are: s (only subject); p (only predicate); o (only object); s-p (subject and predicate); s-o (subject and object); p-o (predicate and object); s-p-o (subject, predicate and object specified). The actual query is formulated by binding the value of the referred concepts in the input sentence to the generic SPARQL query of one of the above seven types to formulate the precise query and retrieve the answer. For example, let the referred concepts be, say, "models, Canon cameras", then the answer extracted out from the ontology would be "A630". Let the input sentence be "Could you please give me a list of camcorders that have a rebate?". For this query {camcorders, rebate} are the referred concepts respectively and form <s-o> of the query. The exact query fired is as shown below:

Select = ("?f")
where m=GraphPattern([("?a",ds[prd1],ds[val1]), ("?b",ds[prd2],"?f"), ("?a", "?c", "?a") , ("?b", "?c" ,"?a")])
result = sparqlGr.query (select, where) where val1= camcorders and prd1= item_name and prd2 = rebate.

In case the query generation does not fetch an answer then the system traverses the RDF graph. Ontology traversal takes in concepts identified from the input sentence and determines which part of the ontology these concepts satisfy. That is, the concepts could be leaf nodes or some intermediate nodes in the ontology graph. Once this is established, the traversal tries to determine the relationship (direct or inherited) between the concepts identified in the graph structure. For example, if a user wants to know "what is common between DXG 3MP Digital Camcorder - DXG-301V and Apple iPod- 80 GB Video", the query generation mechanism is

not going to give an answer as it cannot find out the commonality easily, whereas an ontology traversal would give the answer "Both are on discount".

3.5 A Detailed Example

We consider a Retail Management System for a retail outlet that has a number of products; some promotion offers and that caters to various customer needs. Tables 1, 2, 3 and 4 show a sample data set.

An example of the domain ontology follows:

ds:Item ds:item_id ds:5
ds:5 ds:item_name ds:Aiptek IS-DV2 Digital Camcorder.
ds:5 ds:item_type ds:camcorder.

ds:Item ds:item_id ds:3
ds:3 ds:item_name ds:Canon Digital Camera - SD900.
ds:3 ds:item_type ds:camera.

Table 1. Item_Store

Item_store_ID	Item_ID	Store_ID	Cost_amt	Discount
4	1	201	500	0
6	5	200	750	25
6	7	201	400	15
4	10	201	1150	30
5	12	200	2000	35
5	3	200	200	10
4	2	201	1600	25

Table 2. Item

Item_I	Item_name	Item_type
1	Panasonic Mini DZ Camcorder	Camcorder
2	DXG 3MP Digital Camcorder - DXG-301V	Camcorder
3	Canon Digital Camera - SD900	Camera
5	Aiptek IS-DV2 Digital Camcorder	Camcorder
7	Apple iPod- 80 GB Video	iPod
1	Panasonic Mini DV Camcorder	Camcorder
2	Panasonic 2.8" LCD Camcorder SDR-S150	Camcorder

Table 3. Store

Store_ID	Store_name	Store_addr
200	Nicollete Mall	A123 NYK
201	PoundLand	Udyog Vihar

Table 4. Department

Department_ID	Dept_desc
22	Electronics
23	Apparel

The *table name* and the *primary key* form the subject (Item and 1 are the subjects), the *fieldname* forms the predicate whereas the *values* of the fields form the object in the ontology file. Let us assume that a user asks the question, "Which camcorders have more than 20% discount?". The primary way to answer this question would be query formation and firing one of the seven query templates. In this example it is:

```
Select= ("? f")
    where.addPatterns([("?a","?c","?a"),("? a",ds[prd],"?f"),
    ("?b","?c","?a"),("?b","?d","?e"),("?b",ds[prd2], ds[va l2]),
   ("?e","?d","?e"),("?e",ds[prd1], ds[val1])])
    result= self.sparqlGr.query(select,where)
```

This query when fired fetches the appropriate answer:
"The Camcorders are DXG 3MP Digital Camcorder - DXG-301V, Panasonic Mini DV Camcorder, Aiptek IS-DV2 Digital Camcorder, Panasonic 2.8" LCD Digital Camcorder with 3CCD Technology - Silver (SDR-S150)"
This answer is then shown to the user.

4 Integrating Virtual Worlds and Business Applications

In this section, we describe how virtual worlds and business applications may be integrated. We use NATAS as an example interface. Virtual worlds and enterprise systems may also be integrated using any other similar interface. To the best of our knowledge, the kind of integration does not currently exist in any of the online virtual worlds. However, one related service available in Second Life is Jnana [2], uses voluntary human experts to advice novices on a particular domain. The expert's knowledge is uploaded into an interactive question-answer system. When a novice needs to decide about a particular product or service, the system prompts him with questions based on the expert's knowledge. A series of questions and answers ensues till the novice is able to make an informed decision based on the expert's advice.While Jnana is very stable though, meaning that the user can usually find what he is looking for, this system has the following drawbacks. Firstly, since the expert knowledge is voluntary, it may not be available on all topics of interest to the novice or it may not be complete to the extent required by the novice. Secondly, questions are asked by the expert rather than the novice. So, the novice cannot control the conversation based on what he wants to know as opposed to what the expert wants to tell him.

The NATAS engine, on the other hand, has knowledge about the domain it is being queried on as defined by the domain ontology - the more detailed the

ontology, larger is the question set that NATAS can answer. Further, expert comments or reviews may be included, if available. Since the conversation is initiated by the customer, it can be specific to what the customer wants to know, rather than what the system wants to tell the customer. Further, as the interface is in natural language, the customer can phrase the question in his own style rather than having to figure out if the question posed to him answers what he wants to know.

4.1 Integration Mechanisms

Virtual worlds provide tools for building objects or allow graphical models to be uploaded. Scripts or code may run on the models so that they may have a behavior associated with them. For example, a door may be coded so that it opens and closes, a car maybe coded so that it can be driven around. Depending on the underlying architecture of the virtual world, scripts may be written in standard languages like Python, Java or in specially created languages like the Linden scripting language used in Second Life. We use this functionality to link Second Life with NATAS in the context of a retail store.

The tasks associated with a business application for a retail store include providing information about products like price, features, discounts and availability, completing transactions for purchase and shipping the product to a real-world destination. Scripts on objects in the virtual world also reside and run on the server. It is possible to script an object to connect to an external web service via hyper text transfer protocol (HTTP). NATAS is available as service enabled on a web server to which external applications can connect. Thus, NATAS can easily be integrated with an object, in this case a virtual assistant, in a virtual world. Note that Second Life is just one example world to which NATAS has been linked. In principle, NATAS can be linked to any virtual world using similar or other mechanisms.

We have designed a retail store in Second Life with objects like Cameras, iPods, T-shirts to name a few that can be bought in-world. Further, we have added a robotic sales assistant in the store to answer queries posed by customers on the items in the store. Since the assistant is created using the building tools of Second Life, it is a graphical object within Second Life and does not require a human to be logged on. Thus, assistance is always available to customers who may login across different time zones. Also, the assistant can be programmed so that it moves around with the customer, if he/she so desires, or can stand in one place, answering any questions that the customer might have. The integration of Second Life and NATAS is shown diagrammatically in figure 1. When a user asks a question to the assistant using text chat, the query is extracted and sent as an HTTP request to a web server on which NATAS is running. NATAS connects to the appropriate retail store application, processes the query and formulates the answer. The answer is then sent back as an HTTP response to the virtual world where it is displayed as chat from the virtual assistant. Multiple users can query NATAS at the same time and it maintains context of each conversation.

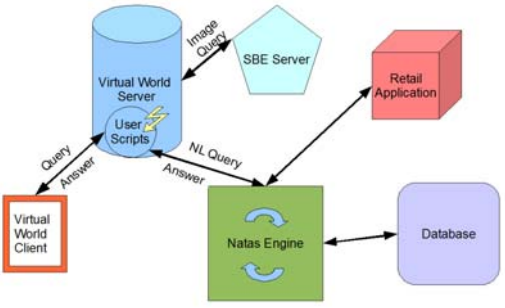

Fig. 1. Broad architecture of Second Life integration with NATAS

(a) (b)

Fig. 2. Broad architecture of Second Life integration with NATAS

Figure 2 (a) and (b) show some screenshots of a possible interaction with NATAS through the sales assistant (or chatbot). Since the querying is done via HTTP transfers, the response comes within a few seconds so that the conversation takes place in real-time.

The interaction with a business system may also be used to create a customized interaction for the user. For example, the look and feel of the retail store may change depending on the profile of the customer. Certain products may be highlighted and others made to disappear completely depending on what the customer is interested in. This information may be extracted from the conversation the user has with the business system. We will discuss this aspect in the dynamic rendering section of this paper.

4.2 Carrying Out Real-World Transactions

The interactions through the virtual world can lead to carrying out concrete tasks and transactions on the business application system, such as "*buy a camera*".

Such a task or transaction can actually lead to generation of an invoice for the customer and billing activities. An order form can be automatically filled and pushed to the customer (either directly onto a window on the virtual world, or via an offline mode such as an email) to confirm the order he or she has placed. Once the order is reconfirmed and the payment mechanism (such as credit card) is confirmed, the shipping of the product can occur. Thus interactions in the virtual world can lead to actual real-world transactions.

4.3 Dynamic Rendering

Dynamic rendering can help a virtual space to change on the fly depending on feedback from the user. There are different kinds of changes that can be triggered in a virtual space. For example, external changes such as the entire architecture of the building in the virtual space can be changed from say multi-storied to a single floor. Internally, the space can be made to look different- the colors of the walls, layout or presence of objects etc. can all be changed. The experience inside the space can also be changed - for example the same set of objects can be made to behave differently.

Dynamic rendering has a number of advantages. From the perspective of the owner of the virtual space, the same piece of land can be used for multiple purposes. For example, one can create a retail store that turns into an insurance information centre, a bank or a space for holding virtual conferences. Thus, dynamic rendering can be used to switch between different domains. Even within a domain, dynamic rendering can be used to highlight items or information that the user may be interested in.

From a virtual retail store perspective, for example, users visiting a retail store will be looking for different things - a younger person could be interested in a particular style of clothes or music, while an older person can be interested in another style. Dynamic rendering could help the same store cater to the needs of both customers by rendering it according to customers preference. This has a two-way benefit since the customer sees only what he is interested in, and this cuts down on the time required for him to decide what he wants to buy. The store owner, on the other hand, has more and quicker sales, as a customer does not waste time in identifying what he wants and is aided in quicker decision-making.

5 Role of Visual Semantics in Retail Stores

While natural language based interaction in retail stores provides a powerful shopping paradigm, there are many articles, whose properties cannot be easily articulated and are better illustrated with visual examples. Paintings and ethnic garments are a few examples of such media-rich commodities. Even for many articles of common use, it is often the visual appearance of the package that the buyer tends to remember rather than the detailed product attributes. For example, packages of grocery products, the design of DVD and book jackets help to uniquely identify the products. Thus, there is a need to deal with the

semantics that is hidden in the visual appearance of the products and packages, their color and texture, the distinctive product-marks. In this section, we describe two examples that exploit visual semantics.

5.1 Shopping by Example

With ubiquity of high resolution cameras with mobile phones, it is easy to capture the image of an empty carton of a grocery item or the jacket of a DVD or a book. This motivates example based shopping, where the shopper provides a visual example to request the intended product [3].

The overall operation of the system is depicted in figure 3. The product database of an on-line store includes a few image examples of the product packages from different perspectives. The shopper uses his mobile camera or a webcam to take a snap of the product package, which is submitted over MMS / the Internet. A search algorithm operates on the image database and retrieves the closest matching image. The desired product is so identified and the product details are shown to the buyer to make the final purchase decision. In a retail store in a virtual world environment, the avatar of the buyer shares a snap taken in the real world with a seller agent in the virtual store and requests the desired product. In this application, we identify the desired product by the visual appearance of the distinctive product-mark. The low level image features, such as color and texture are not suitable for this purpose. Difficulties also arise from imperfections in the user supplied images, because of imperfect lighting conditions, surface glare and improper alignment of the hand-held camera as well as wrinkles and damages of the used packages.

PCA-SIFT [4] provides a robust way to compare the product-marks with keypoints derived from the images and can take care of many of the imperfections. The key-points are sharp and distinctive corners in the visual pattern characterizing the product-mark and can somewhat be compared with keywords in a text segment. Each product image contains an arbitrary number of key-points. Each keypoint is represented as a 128-dimensional vector. The similarity between two key-points is measured by the cosine of the angle between the vectors. The similarity between two images is computed as follows

1. Let $K_1 = k_{11}, k_{12} \ldots k_{1m}$ be the set of key-points in the query image Q and $K_2 = k_{21}, k_{22} \ldots k_{2n}$ be the set of key-points in a product P
2. Let $k = 0$ (no. of matching key-points in K_1 and K_2)
3. For each member $k_{1i} \in K1$, do
   a. Let $s_i = 1$ (the largest possible value)
   b. For each member $k_{2j} \in K2$, do
      i. $s_{ij} = similarity(k_{1i}, k_{2j})$
      ii. if $s_{ij} < s_i$, then $s_i = s_{ij}$
   c. If $si > t$ (threshold), $k = k + 1$
4. $Similarity(Q, P) = \frac{k}{|K1|}$

In summary, a key-point in the query image Q is said to match a key-point in the image of a product P, if the similarity between them exceeds a threshold

Fig. 3. Shopping by example

(t). The similarity of the query Q and the product P is established in terms of number of matching key-points and is normalized by dividing the number with the cardinality of the key-point set in the query image.

The list of products to be shown as the candidate solutions are computed as follows

> The products are ranked in the decreasing order of similarity. Let si be the similarity value of the i-th product in the ranked list.
> If $s_j > \lambda * s_{j+1}$ (where λ is an arbitrary number), then j is treated as the cut-off point in the list, i.e. the buyer is shown the products till (and including) the j-th product from the top of the list.
> If $s_1 > \lambda * s2$, only the first result is shown and the product is said to have been identified uniquely.
> If $j > k$ (when k is a pre-decided constant), we conclude that the system has failed to identify the product, either because the product has not been in the database, or because extreme aberrations in the query image.

PCA-SIFT has the ability to distinguish key-points with great accuracy, and in most of the cases, the algorithm produces a unique and correct result. Figure 4 depicts some such query image examples. It may be noted that the images are distorted, have surface glare and out of focus. The system performs well despite these defects in the input images, which are expected in a real application scenario. Since there is a unique correct result for every query image, we use Mean Reciprocal Rank (MRR) as a performance measure of the system. With a database of more than 1000 products, the MRR of the system is found to be 97%.

Shopping by Example has an interesting application in the context of virtual worlds. Suppose while navigating through a virtual environment, a user comes across a real-world or virtual item of interest, for example, a new CD at a friends house. The user may click an image of the item and store it on his hard-disk. Many virtual worlds allow the user to click photos "in-world" using their client software. Even otherwise, a user can use the print-screen facility to take an image of what is being displayed on the screen. This image can be submitted as a query to the SBE system to get more information about the item. Figure 5 shows the usage of the SBE system from Second Life.

Fig. 4. Example query images

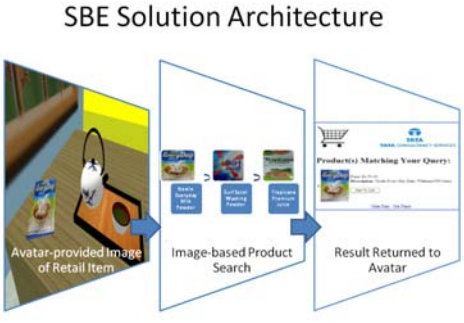

Fig. 5. SBE in Second Life

6 Conclusion

A next-generation virtual world-based retail store is a distinct possibility in the near future. Such a store can provide the potential retail customer with a variety of mechanisms to interact and select the appropriate product suitable to his or her requirements. Natural language based interactions with a chatbot combined with visual image based search can lead to an easy shopping experience for the customer. With the store dynamically changing its layout and offerings, the customer can get a rich and enhanced shopping experience. The role of semantics in such interactions is important to be addressed, and it is also important to have a framework that delivers such an experience. We have described an innovative syst

References

1. Bhat, S., Anantaram, C., Jain, H.: Framework for text-based conversational user-interface for business applications. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 301–312. Springer, Heidelberg (2007)
2. `http://www.jnana.com`
3. Ashish, K., Hiranmay, G., Jagannathan, J.S.: SHOPPING BY EXAMPLE - A New Shopping Paradigm in Next Generation Retail Stores. VISAPP (February 2009)

4. Yan, K., Rahul, S.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2 (2004)
5. 2003 Mall Shopping Patterns, Consumers Spent More Time in the Mall. ICSC Research Quarterly 11(2) (Summer 2004)
6. Answers Anywhere, Sybase Inc. An Application of Agent Technology to Natural Language User Interface
7. Popescu, A.M., Etzioni, O., Kautz, H.: Towards a Theory of Natural Language Interfaces to Databases. In: IUI 2003, Miami, Florida, USA (2003)
8. Yunyao, L., Huahai, Y., Jagadish, H.V.: Constructing a Generic Natural Language Interface for an XML Databases. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 737–754. Springer, Heidelberg (2006)
9. Ferguson, G., Allen, J.F.: TRIPS: An integrated intelligent problem-solving assistant. In: Proceedings of AAAI 1998, pp. 567–573 (1998)
10. Allen, J.F., Ferguson, G., Stent, A.: An Architecture for More Realistic Conversational System. In: IUI 2001, Santa Fe, New Mexico, USA, January 14-17 (2001)