(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 September 2002 (19.09.2002)

PCT

(10) International Publication Number
WO 02/073331 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/IB02/01963

(22) International Filing Date: 20 February 2002 (20.02.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/269,995 20 February 2001 (20.02.2001) US

(71) Applicant: **SEMANTIC EDGE GMBH** [DE/DE]; Kaiserin-Augusta-Alle 10-11, 10553 Berlin (DE).

(72) Inventors: **ARETOULAKI, Maria**; Alte Potsdamestr 9, 10785 Berlin (DE). **GREGORY, Brown, Michael**; Marlene-Dietrich-Platz 3, 10785 Berlin (DE). **DABIRI, Gloria**; Prenzlauer Allee 219, 10450 Berlin (DE). **KREUTEL, Jörn**; Kastanienalle 88, 10435 Berlin (DE).

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i)) for the following designations European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR)*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR)*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for the following designations European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR)*

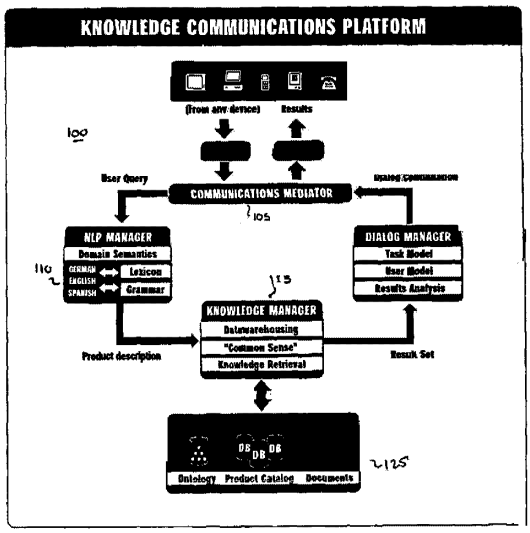
— *as to non-prejudicial disclosures or exceptions to lack of novelty (Rule 4.17(v)) for all designations*

Published:

— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: NATURAL LANGUAGE CONTEXT-SENSITIVE AND KNOWLEDGE-BASED INTERACTION ENVIRONMENT FOR DYNAMIC AND FLEXIBLE PRODUCT, SERVICE AND INFORMATION SEARCH AND PRESENTATION APPLICATIONS



(57) Abstract: A system architecture and a methodology for intuitive natural language (text or speech) dialogue-based electronic commercial transactions and information exchanges are described. The system and methodology allow the user to pose questions over the Internet (via a PC Web or E-mail Browser, a PC microphone, a fixed or mobile phone, or any wireless device such as a Personal Digital Assistant) about products and services of other providers, as well as information in databases, in a natural way, avoiding the constant clicking of links and the selection of keywords that may not have a meaning to the user in the first place. The inventive system is robust towards any type of multimodal input. The user requests are interpreted in the context of the current interaction and the dialogue flow is shaped dynamically by the current status of the application database, i.e. the product and service availability. The system dynamically adopts appropriate repair strategies in the case of misunderstandings and processing difficulties or errors, one of which is the transfer to a human operator with all the data collected from the user up till that point. In another manifestation of the invention, the automated dialogue between the user and the system can constitute the indispensable initial information-gathering phase, before the user can talk directly with a human call-center operator, especially when all the lines are busy. The operator may then pick up the call directly afterwards or process the user requirements later and get back to them on the phone or through e-mail. In yet another embodiment of the invention, the user can carry out an off-line dialogue with the system through e-mail or other kinds of voice- or text-based messaging system. The invented system further allows for the constant updating of the various types of knowledge used and the learning of domain, application, and market-relevant information.

WO 02/073331 A2

— *with a declaration as to non-prejudicial disclosures or exceptions to lack of novelty*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

NATURAL LANGUAGE CONTEXT-SENSITIVE AND KNOWLEDGE-BASED INTERACTION ENVIRONMENT FOR DYNAMIC AND FLEXIBLE PRODUCT, SERVICE AND INFORMATION SEARCH AND PRESENTATION APPLICATIONS

This application claims priority on United States provisional patent application serial no. 60/269,995, filed February 20, 2001, which is incorporated herein by reference.

TECHNICAL FIELD

The present invention relates to the fields of human-machine dialogue, and database retrieval, particularly on the Internet.

BACKGROUND OF THE INVENTION

At present, many electronic transaction applications, e.g. in E-commerce, consist of a visual presentation of available products and services according to predefined categorized schemes defined by the corresponding manufacturer or service provider. Such applications presume that user needs are predictable and uniform, as well as takes for granted that users themselves know what they want when they visit a Web site, which may not be the case. For example, the presentation of package holidays on the Web could include information such as destination, period of travel, number of people traveling, price, and possibly facilities available at the destination. See, for example, the Web Site having the following URL address: http://www.holiday.co.uk/default.asp.

However, when users want to know about a different feature, such as sporting or social events or whether there are any accommodations for small children, the application interface usually cannot either understand the additional requirements, or retrieve the desired information. This is so because the majority of E-commerce sites only allow limited searches by means of keywords that are pre-specified and restricted in number, without any definition of the relationships between individual keywords or possible synonyms used to refer to them in natural language. This limitation leads to the retrieval of irrelevant or inaccurate database information, even after a number of attempts on the part of the user to use more specific keywords.

Even for purely graphics and image-based E-commerce sites, the available information is usually hidden in a maze of hyperlinks, or in a hierarchy of nested hyperlinks which

is difficult to explore in a short amount of time, and definitely impossible to keep track of or have a constant overview of. An example of such shopping gateways may be found at the Web site with the following URL address: http://eshop.msn.com In the latter cases, users have to spend a long time clicking through the site often without the desired result, especially as they often do not know the terminology used for the different options. Users would like, at this point, to be able to query the system about the meaning of available options in order to make their decision. Such clarifications can only take place through natural language and, even more importantly, a dialogue: questions, answers, and counter-questions. Such a need is especially critical in the case of uni-modal language interfaces, such as those provided during a telephone call. Thus, the use of some type of natural language is a prerequisite for any type of customer-centric CRM or electronic transaction application.

Apart from graphics and keyword-based search engines, there are a number of natural language interfaces to databases and WWW content that claim to allow the user to "converse" with the system in an intuitive and unconstrained way. These so-called chatbots process user input on the basis of pre-specified patterns, looking for individual words and phrases. This is straightforward because the user input is usually typed on a keyboard and there is no ambiguity in locating these patterns. The chatbot performs a table lookup to associate the identified patterns with appropriate responses and generates corresponding canned messages. Not only is the context of the user input not taken into consideration in understand its meaning, but also the previous exchanges between the user and the system are ignored as well. A famous example of this technique is the legendary ELIZA system, whose capabilities can be seen in a dialogue with one of the inventors reproduced herein below. See http://www.manifestation.com/neurotoys/eliza.php3 for an interface to this system.

User: Can you help?
ELIZA: Perhaps you would like to be able to help?
User: no, i want you to help
ELIZA: We were discussing you, not me.
User: exactly
ELIZA: OK... "no, you want me to help". Tell me more.
User: i want to end it all
ELIZA: What would it mean to you if you got to end it all?
User: freedom
ELIZA: OK... "i want to end it all". Tell me more.

Recently, there has been an increased activity in the development and marketing of E-commerce virtual agents or

Amazon/VB Assets
Exhibit 1004
Page 4

so-called "virtual shop assistants." Such virtual agents project a certain persona on the WWW for the promotion of specific products, services, and brand names. There are, for instance, virtual agents for the promotion of insurance policies (e.g. Schwäbisch-Hall's *Bausparfuchs* at http://bot.kiwi.de/) or of the Eye Trek™ glasses for private cinema viewing (the English- and German-speaking *Marc* at http://bot.kiwi.de/). These agents sometime combine movement, eye and hand gestures and posture changes to involve the user, e.g. the chatbots marketed by Artificial Life (http://www.artificial-life.com/default luci.asp?pSection=bots). Despite the visual sophistication of these latter systems, the processing of natural language input remains equally simplistic, i.e. dependent on the presence of a limited number of isolated keywords and phrases.

Another problem with many E-commerce sites and related user interfaces is that there is no record kept of the user's previous linguistic input (e.g. options set, preferences, etc.) and what type of decisions they made. As a result, there seems to be no continuation between the latest system message and the user input to earlier sessions. This can be very frustrating for the user who realizes that the interlocutor is "dumb" and cannot really understand what they are saying or what they want. Thus, the user loses trust in the corresponding system and instead would rather speak to a human agent. An additional effect of this lack of a dialogue history is that, once the user has left the original site to visit another suggested WWW link, it is difficult to retrieve sub-pages with search results found earlier, requiring the user to look for the preferred items again.

Furthermore, user requests are interpreted in isolation without taking into consideration partial requirements already specified in a previous interaction step (via keyboard input, for example, or a spoken command). Still another problem relates to how much information the user should provide. The user may find the search too open and navigation too restrictive. Importantly, even in the case of one-click scenarios, mouse-driven or keyword-driven, such approaches do not allow information gathering concerning the user. Thus, invaluable data on user preferences is lost in hyperspace and cannot be taken advantage of by the product manufacturers and service providers to guide future development and marketing campaigns.

SUMMARY OF THE INVENTION

Disclosed is a system architecture and methodology for intuitive natural language dialogue-based electronic commercial transactions and information exchanges. The system allows the user to question the system over the

Amazon/VB Assets
Exhibit 1004
Page 5

Internet about products and services of other providers, as well as information in databases, in a natural way that avoids the constant clicking of links and selection of keywords that may not have any meaning to the user. The interface to the Internet can either be a Personal Computer (PC), a fixed line phone, a mobile phone or any wireless device (such as a Personal Digital Assistant (PDA)).

The system comprises modules for:

(a) the robust processing of natural language text or speech input from the user, unrestrained with regard to vocabulary or syntax or degree of grammaticality. Irrespective of the input modality, the input arrives for processing by the NLP Manager in an ASCII text form;

(b) the context-based discourse and semantic interpretation of the user input in terms of the preceding exchanges between the user and the system and the currently salient concepts and relations in the knowledge base of the system, respectively. The former takes place in the so-called "Dialogue Manager" and the latter in the "Knowledge Manager;"

(c) the dynamic establishment of the dialogue flow depending on the most current results retrieved from the database by the Knowledge Manager , as well as on the confidence levels of the system regarding the degree of understanding of the user's input established by the Dialogue Manager;

(d) the automatic adaptation of the interaction strategies of the system in the case of misunderstandings, erroneous processing, or objections on the part of the user by the Dialogue Manager;

(e) the active look-up of a knowledge base with expert domain knowledge in the course of the dialogue in order to identify and correct early mismatches between the user's beliefs and the stored system data about domain entities and their relationships which is coordinated by the Knowledge Manager;

(f) the continuous learning of new words, domain concepts, and matches from words to concepts in order to identify the user requirements, controlled by the Knowledge Manager;

(g) the continuous learning of domain-specific grammars and lexica for the automatic tuning of the parsers used in analyzing the user's input to new applications, coordinated by the Knowledge Manager ;

(h) the continuous acquisition of market-relevant information on user preferences and desires that can form the basis

Amazon/VB Assets
Exhibit 1004
Page 6

for the development of new products and services in commercial transaction applications.

The invented architecture allows the smooth transfer of the interaction to a human operator for the corresponding provider site, either in the case of repeated misunderstandings between the user and the system, or in the case of targeted Customer Relationship Management (CRM) activities when the human operator needs to intervene to acquire more information on the specific user or to negotiate an offer.

In a different embodiment of the invention, the dialogue with the system always precedes the dialogue with the human agent which can take place later on, after the lines have been freed in a busy call center environment. During the automated dialogue, the user's requirements are collected which are then sent by e-mail to the appropriate operator. The operator can then initiate at his convenience a targeted transaction with the user over the phone or through the World Wide Web on the basis of this data to propose a solution that matches these requirements, to disambiguate certain information, and possibly correct data that was misunderstood or wrongly recognized by the system during the first step.

In another embodiment of the invention, the user carries out an off-line dialogue with the system through e-mail or other text- or voice-based messaging systems.

The invented system accepts any type of input that comes in - directly or indirectly - in a text form, as this is delivered by devices such as a computer keyboard, speech recognizer, a gesture recognizer, a communication prosthesis (for people with various forms of disability), or multimodal input combining one of the above modalities with pointing, mouse clicking, or gestures. The input may also be in any of a number of different languages, e.g. English, German, or Spanish, depending on the native language of the user. The language the input is in is, importantly, independent from the language the stored data is in (the various product, service and information databases), as there is always a standardized translation function from surface form to semantic meaning.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become more readily apparent from the following detailed description of the invention in which like elements are labeled similarly and in which:

Fig. 1 is a block diagram of a high level architecture of the preferred embodiment of the present system;

Amazon/VB Assets
Exhibit 1004
Page 7

Fig. 2 is a diagram illustrating the methodology of the present system;

Fig. 3 is a block diagram of a high level architecture of another embodiment of the present system;

Fig. 4 is a block diagram of the domain knowledge acquisition infrastructure of the present system; and

Fig. 5 is a block diagram of the domain and dialogue knowledge maintenance infrastructure of the present system;

DETAILED DESCRIPTION

The inventive system architecture and methodology for dialogue-based electronic commercial transactions is based on the use of totally unrestricted natural language input on the part of the user to interact with the system to find whatever he is looking for. This freedom of expression is, in part, facilitated through the use of a number of robust language processing techniques, focusing on individual fragments of the user input rather than carrying out a complete analysis of it. In cases of ungrammaticality and incoherence, a partial analysis of the user's input is necessary. This freedom of expression is best illustrated by the following example of a user's input in booking a holiday.

> *User: Erm, let me see, I'd like to go to - to a sunny Mediterranean country, say Italy, sometime next month for about 10 days.*

As illustrated above, the user's input can be freely formulated, that is it can contain hesitation (*Erm*), hedge words (*let me see*), false starts and repetition (*to go to - to*). Such an input is acceptable since it is interpreted in the context of the current interaction, i.e. against what was said between the user and the system up till then. In addition, information about a specific user or user group is also employed, firstly in establishing the user's need by using defaults and predictions, and secondly in planning the subsequent system responses. Thus, a new user will be treated differently than a returning user, the former receiving more guidance and longer explanations about the service provided, while the latter being allowed to skip steps and jump directly to the stage they want.

An aspect of the present invention is the employment of knowledge processing and inference methods to find out what users are looking for to satisfy their specific needs. That is, the user is not required to fully understand the application or have the expertise associated with a specific

Amazon/VB Assets
Exhibit 1004
Page 8

product, service or database (e.g. computer memory or skiing equipment). Thus, information and services become accessible to a wider (browsing) public, which can turn into a buying public too. This is due to the fact that the present system comprises modules for the detailed representation of the target domain in terms of higher-level concepts and relationships among them (ontologies), which allow for the interpretation of the user input in this wider context, and its disambiguation, when the need arises.

Another feature of the present invention is that the system is constantly monitoring its performance regarding (a) the recognition and correct interpretation of the user input; (b) the information and the current status of the application database (e.g., product catalog) and the knowledge bases (e.g., ontology) in order to identify conflicts in the user's requirements or make recommendations to the user; and (c) the user behavior, in order to locate communication or understanding problems or modified wishes. This monitoring serves to dynamically establish the next step in the dialogue, i.e. whether the dialogue should proceed as planned, be temporarily interrupted to solve a problem, or be terminated by transferring the user request to a human operator. In other words, the invented system allows for *human-machine*, but also *machine-machine* (for internal repair strategies), and *human-human* interaction (when all else fails). Of course, human-human interaction is allowed at every point, if that is what the user wants. This is also the default in an embodiment of the invention where the users specify their requirements in a preparatory dialogue with the system, which then passes them on to an operator in the form of an e-mail. On this basis, the operator will initiate a targeted conversation with the user. The same is true in the case of an off-line e-mail dialogue between the system and the user.

As a result of the above, the present system and methodology affords the flexibility of the above dialogue between the user and the system. The user is free to change their minds, to correct the system, or to interrupt it in the case of a misunderstanding. Thus, the interaction is natural and human-like, because it is constantly evolving in the context of the current user, the current status of the databases, and the history of the specific dialogue session between the current user and the system. In other words, the present system is both context-sensitive and personalized. Naturally, the user can always directly contact a human operator, if they so wish.

Shown in Fig. 1 is a high level architecture 100 for a system and methodology of providing a natural language dialogue-based electronic commercial transaction in accordance with the principles of the present invention, including the processing modules and the flow of the

Amazon/VB Assets
Exhibit 1004
Page 9

input/output data. There are four basic modules in the preferred embodiment of the invention, including:

1. Communications Mediator, 105
2. NLP Manager, 110
3. Knowledge Manager, 115; and
4. Dialogue Manager, 120

Communications Mediator 105 is the system module which handles input and output of different modalities, e.g. typed text (via a Web browser or E-mail), speech, handwriting, and even gestures. It receives user queries from all available and relevant channels (PC, mobile or normal phone, or devices such as a PDA, etc.) and routes them to NLP Manager 110 for further processing of the resulting input text string. All types of user input are translated into an ASCII string, independent of the communication mode. Nevertheless, information about the actual mode chosen by the user will still be kept for consultation purposes in a dialogue memory. Communications Mediator 105 is also the component which coordinates the presentation of the subsequent system output (determined by Dialogue Manager 120), be it a text message on the screen, a spoken prompt over the phone, images and graphics, or a combination of all.

NLP Manager 110 is another component of the system which processes the user queries that arrived in the form of a text string from Communications Mediator 105. Natural Language Processing (NLP) involves the lexical, syntactic (grammatical), and domain semantic analysis of the user input using both statistical observations of the various surface forms and a deeper interpretation of the relationships and dependencies among words, phrases, and concepts. The coordination of surface and deep processing is performed by an arbitrator sub-module that weighs the significance and certainty of the results of the two separate processes and selectively promotes a number of these results for further validation and interpretation by Dialogue Manager 120. These results, i.e. the output of NLP Manager 110, have the form of frames with embedded structures holding the recognized words from the user input, their syntactic and semantic function and possible semantic relationships among them. This makes up the so-called product (or service) description which will be used for the dynamic database look-up. Depending on the language employed by the user, appropriate grammars and lexica are dynamically loaded, e.g. for English, German, or Spanish.

Knowledge Manager 115 maintains and manipulates information on both the world in general and the specific domain and application under consideration. It controls a generic and extensible ontology of concepts and relationships among those concepts, which represent objects

Amazon/VB Assets
Exhibit 1004
Page 10

and processes that are relevant irrespective of the domain or application ("common sense" information). At the same time, this ontology shows the interdependencies between these generic concepts and the application-specific ones, in terms of classes and instances of these classes, as the latter are contained in the application databases. By controlling such an ontology, Knowledge Manager 115 is able to carry out inferences given some data from Dialogue Manager 120 and to locate inconsistencies, incompatibilities, and contradictions in the evolving specification of the user's requirements (the updated product (or service) description). Knowledge Manager 115 is also the only component in the system architecture with direct access to the most current data from resource agent 125 (i.e., application databases, a product catalogue, ontology, etc.). The retrieved data is communicated on-line to Dialogue Manager 120, whenever the latter asks for it. Dialogue Manager 120 then decides accordingly on the next system action, be it an additional question to the user or the presentation of the result set.

Dialogue Manager 120 is the central controller in the system architecture. It is, firstly, the mediator between NLP Manager 110 and Knowledge Manager 115, passing on the user query from the former module after it has been analyzed lexically, syntactically, and semantically as a product (or service) description to the latter module ready to be submitted to the databases. It is this component which makes calls to Knowledge Manager 115 to access application databases and retrieve the most current information, or to just check the compatibility of individual constraints that the user may have set. Furthermore, it is the module which controls Communications Mediator 105 in determining the goal and the content of the next system action and message. Thus, its output is a semantic representation of the dialogue continuation; i.e. of the next system action in the form of a frame with embedded structures if need be, in order to express interdependencies and ordering information. Dialogue Manager 120 employs a series of models that help it interpret the user input and decide on the next system action. The task model describes the types of application-specific information that are most likely to be talked about with the user, some in a pre-specified order. This is something like the default general plan that the system has, which can be overridden later due to changing user or context requirements. The user model contains data about the general preferences and assumptions of individual users, as well as whole user groups. These preferences refer to both application requirements, such as favoring Mediterranean destinations, and presentation preferences, e.g. images only. User models can be both fixed, as in the case of user classes such as senior citizens, singles, or students, and evolving, as in the case of users whose

Amazon/VB Assets
Exhibit 1004
Page 11

behavior is being monitored by the system in the course of the interaction. The information held in the former type of model may be updated, of course, but not on-line, whereas data in the latter type of user model is collected in real time. Dialogue Manager 120 also has access to and can update a so-called "discourse history," that is a record of everything that has taken place up till then in the dialogue, system messages and user input, their semantic representation in terms of actions and domain parameters, and their ordering in time.

The uniqueness of the present system relies on a) the employment of sophisticated mixed-methodology NLP techniques for the robust analysis and interpretation of the user input, b) the close integration of knowledge about the application domain and the world in the interpretation process, c) the way the two are coordinated by Dialogue Manager 120, and d) the flexible adaptation of the dialogue strategies of the system depending on the current status of the processing, the occurrence of communication problems, and the aggregating history of the interaction.

Figure 2 shows the general flow of the processing in the preferred system architecture. The user expresses their wishes 205 (queries) in their preferred modality and format, for example, by typing on a computer keyboard or writing with a pen on a PDA (Personal Digital Assistant) screen, either in a specified input box or in an e-mail; by speaking over a fixed-line phone, a normal or a WAP-enabled mobile phone or over a head microphone attached to a personal computer. In short, the user can employ any input device, as long as this input acquires some basic textual semantic representation. This includes the employment of a mouse or other pointing device (such as a finger or a pen) when surfing the Internet, and even the use of gestures captured by a camera. The translation of the various input modes and modalities takes place in *Communications Mediator 105 at block 210 to produce an ASCII text representation 215 of user input 205.* Module 105 is responsible for the rendering of non-linguistic input, such as a mouse-selected menu item, into a text showing its correspondence to a domain concept or relationship (e.g. hotel, room price), or even to an interpretation of the user intention in the case of the employment of gestures (e.g. disapproval). For instance, the user may point to the image of a specific hotel on a World Wide Web page, and Communications Mediator 105 will then forward the name and ID of this hotel to other modules of the system for further processing. Thus, *Dialogue Manager 120* may decide to ask the user for a confirmation that they want to book this specific hotel. Communications Mediator 105 will also merge information coming in from more than a single mode in order to interpret the different types of data in the context of one another (Multimodality). For example, the spoken utterance

Amazon/VB Assets
Exhibit 1004
Page 12

"I would like to buy this scanner"

accompanied or immediately followed by clicking on the image of a featured scanner will be translated into a template showing the desired type of product to be purchased and the specific identification for the selected product itself, e.g.

[type:scanner[scannerID: 'Pagis Pro Millennium']]

Importantly, users can choose their preferred language, be it their native tongue or the language in which their target material is in, if they are comfortable with it. This choice does not need to be explicit, e.g. by selecting a menu item or specifying the language preferred at the beginning of the interaction. The user can speak their query in *English*, for instance, and have the system access a product or service database in *Spanish* and *German*, resulting in a database match presentation in English to suit the user's inferred language preference. Preferably, *NLP Manager 110* decides dynamically which language is employed by the user and how to process it. *Dialogue Manager 120* is the module where the planning of the next system message takes place, including the choice of language for it. The Discourse History employed by Dialogue Manager 120 (discussed herein below) contains, among other things, a record of the language preferred by the user, which influences such future choices.

The ASCII text representation 215 of the user input 205 is processed lexically, syntactically, and semantically by NLP Manager 110 at block 220 in order to acquire a representation of the user's requirements as a domain semantic representation 225 (or some of these requirements) about the desired product, service or information. The goal is to obtain a description of the product, service or information which is as detailed as possible in order to be matched against the descriptions of existing products, services and information in the application databases. Lexical and semantic analyses are closely coupled, so that individual words can already point to domain specifications. This is achieved by means of a mapping lexicon that translates application-specific words into the corresponding relevant concepts in the domain ontology.

NLP processing employs statistical but also deep language processing techniques. This is in order to take advantage of both obvious features of the resulting text, such as domain vocabulary (i.e. keywords), but also the chosen structure of language through which the user expresses their intention, and the global context of each word, i.e. syntactic patterns and word collocation. Both types of linguistic analysis are based on application-specific data that has been

Amazon/VB Assets
Exhibit 1004
Page 13

semantically annotated, discussed in more detail herein below.

An *Arbitrator* may be employed to decide on the relative importance and relevance of the results delivered by the statistical and the deep language processing components, respectively. Such an arbitrator checks at every point the confidence levels of the corresponding components about the achieved results and allocates a preference weighting to each. This weighting will be taken into consideration later on in the course of the processing along with other types of information on the specific interaction (such as previous user input and domain restrictions).

The initial interpretation of the user input (Domain Semantic Representation) 225 will be augmented by means of the discourse analysis in Block 230 with user dialogue acts by *Dialogue Manager 120, yielding a domain and dialogue act semantic representation 235*. Dialogue acts refer to the reason why the user expressed the specific information at the specific point in time in the course of the interaction. They have been influenced by the philosophical theory of *speech acts*, as well as conversational and textual discourse analysis. They reflect, for example, the user´s (positive or negative) answer to a system question (*reply y/n*), their positive or negative reaction towards a system suggestion (positive interest, negative interest), or even more importantly the user´s objection regarding a domain parameter and the correction of the corresponding values (*correct*). Dialogue Manager 120 decides on an interpretation of the user intention in saying, writing, gesturing, or pointing to something, based on the information exchanged up till then during the interaction, as well as on the next step in the general plan to be taken in order to fulfill the user wishes.

An exemplary list of the dialogue acts used in the framework of the present invention is provided in Appendices I and II, for the system and the user, respectively. As such, the user input is now represented in terms of both application domain parameters and dialogue acts 235.

At block 240, this dual meaning representation of the user input is further interpreted in terms of (a) the context of the interaction, i.e. the previous exchanges between system and user, including data on the specific user or the user group they represent, and (b) the knowledge base of the system, which holds information about domain-specific objects and more general concepts and their interrelations. Such a contextual analysis and knowledge processing yields validated and disambiguated domain and Dialogue act semantic representation 245. Effectively, this means that at this point in the processing the system holds an internal machine-to-machine dialogue with the discourse history, the

Amazon/VB Assets
Exhibit 1004
Page 14

user model(s), the application database and the ontology of the system.

The context of the interaction (or the discourse history) consists of representations in terms of pairs of domain parameters and dialogue acts, as explained above, for both the system and the user. These representations 245 can be any number, from one in the case of a new dialogue where no interaction has taken place yet and the system has just given the user the first opening message, to two or more in the case of an on-going dialogue. These pairs specify the occurrences for these parameters (even the fact that they have been asked about but are as yet unknown), as well as the user´s or the system´s action(s) or reaction(s) to them (e.g. correction or confirmation). Example of such pairs are provided below:

```
system opening (message: "Hello. I´m Printess and I can find
the right printer for you.")
system query_selection (printer: (print_engine:[laser, inkjet, dot-
matrix] ) )
user reply_selection (printer: (print_engine: laser) )
system query_yn (printer: (output: color) )
```

The above semantic representations mirror the fact that, in the beginning, the system greeted the user and asked them whether they would prefer a laser, an inkjet, or a dot-matrix printer. The user replied that they would prefer a laser printer, after which the system asked whether the printer output should be color or not. Multiple representations ordered chronologically, as aggregated in the course of the interaction with the user, make up the history of the dialogue and the interaction (the *discourse history* or *dialogue memory*), against which new input will be interpreted.

The result 245 of this additional interpretation of the user input carried out at Block 240 is partly a representation of its meaning in terms of domain concepts reproduced as domain semantic representation 255. These representations refer to application-specific parameters (e.g. *hotel room price*) and the values set or preferred by the user in specifying their requirements (e.g. *€50*). These parameters are isolated by NLP Manager 110 or inferred by *Knowledge Manager 115* of the system from the user input. Irrespective of whether this input was language- or graphics-based, the knowledge processing is the same. Parameters do not necessarily correspond to individual words, but can be extracted on the basis of syntactic considerations and semantic relations defined in a thesaurus or an ontology, for instance.

At block 250, the original discourse plan may be updated, depending on whether the user has specified their requirements fully or not and without inconsistencies or

Amazon/VB Assets
Exhibit 1004
Page 15

incompatibilities. In updating the discourse history with the latest user input and/or system output, information held in the Task Model and the User Model of the system is consulted by Dialogue Manager 120. The *User Model* contains information on the specific user: permanent data (such as address and previous orders) in the case of a returning customer, and temporary data on the current first-time user collected during the ongoing dialogue with them. Thus, Dialogue Manager 120 can adapt its planning decisions about the next system message from the start of the interaction in the case of the returning customer, or/and dynamically depending on what the user has already said. The User Model also holds data on classes of users against which the current user will be constantly compared, in order to infer information and preferences about them. This is useful in the setting of defaults or the clarification of ambiguities, which speed up the interaction and the search process, avoiding tedious questioning of the user. Thus, the system may assume that all users who do not know the difference between a laser and a dot-matrix printer will also not know what "high resolution" means, in which case the system will ask the user about this parameter using a simplified formulation. At the same time, the user is always given the opportunity to correct or update such defaults, something that will modify the models that the system holds accordingly. This type of user models, which represent whole user classes, is called a *prototype* and is based on statistical data collected by way of marketing analysis and social psychological research. New users are allocated a prototype automatically depending on the characteristics of their language use and, also, application and presentation preferences.

Having identified the relevant domain parameters in the user input, related ontological concepts and possible mappings to real-world entities that these concepts can have (instantiations) are activated in the knowledge base of the system. This is controlled by *Knowledge Manager 115*, which employs an ontology and also has direct access to the application database 125. This ontology holds knowledge about concepts and basic relations among them and is generated semi-automatically on the basis of domain-specific documents (e.g. travel offers) using information extraction and concept clustering techniques with minimal human effort glossing it over afterwards, discussed hereinafter. This ontology is also used for the economic maintenance of generally applicable knowledge and data about features of individual entities and interdependencies among entities. Thus, parts of this ontology can be reused in new applications, e.g. knowledge about financial transactions, or customer-provider relationships. The application database 125 that Knowledge Manager 115 has access to can be, for example, an electronic product catalogue or the list of available holiday offers that the system searches through.

Amazon/VB Assets
Exhibit 1004
Page 16

The processing carried out by Knowledge Manager 115 of the system results in unspecified parameters taking default values, as a temporary solution (until the user specifies otherwise), and ambiguous concepts being interpreted in terms of the *most salient items* in the knowledge base that are currently active (Validated and Disambiguated Domain and Dialogue Act Semantic Representation 245 as well as final Domain Semantic Representation 255 of Fig. 2). This is accomplished on the basis of inference and other knowledge processing methods which take into consideration the context of the current interaction session between the user and the system. Thus, this process has access to the discourse history and the user models, when available.

During the NLP analysis of the user input at Block 220, a number of different interpretations may be generated representing the various alternative meanings in the case of ambiguities. Ambiguity may be caused, for example, when the user employs a keyword that is relevant to more than a single domain or application already covered by the system (e.g. a booking instruction or pricing information request). Information about the context of the exchange with the user (system and user dialogue acts along with parameter instantiations), kept in the discourse history, and about the salient domain concepts in the knowledge base is collectively employed to reassess the relative weighting of these different interpretations and to finally select the one with the highest weight. The modules responsible for this are *Dialogue Manager 120* and *Knowledge Manager 115*. Disambiguation can involve either such an internal machine-to-machine dialogue, or a series of targeted questions to the user. The selected validated and disambiguated semantic meaning representation 245 generated from block 240 will be used in the subsequent processing steps by the system.

Dispreferred interpretations will remain inactive as a part of the discourse history, in order to be employed later on when necessary, in the case where a misunderstanding has occurred and an appropriate repair strategy has to be activated. This related process will be discussed further below.

After the establishment of the salient domain concepts and their values appearing in the user input (product/service description 255 in Fig. 2), *Knowledge Manager 115* dynamically accesses the application database 125 at block 260 and attempts to retrieve the best matches to the user´s requirements as specified up to that point. These matches (Result Set 265) are presented to the user at block 285 as either ACII text, speech and/or images 290.

An application, however, may be employing, for example, a database on printers or on last minute travel offers. If

Amazon/VB Assets
Exhibit 1004
Page 17

the number of matches retrieved is over a pre-specified threshold (e.g. twenty), then an obligation is generated for the system to try and elicit more information from the user on the desired product, service, or information sought, repeating a discourse plan update at 270. This is accomplished in terms of a special system dialogue act (query selection), through which the user is asked to specify the value for a new parameter from a number of available alternatives when appropriate, chosen in such a way as to further restrict the database search (See Appendices I and II for an exemplary list of the system and user dialogue acts, respectively). The choice of which feature(s) to select at this point is a result of a machine-to-machine dialogue between *Dialogue Manager 120* and *Knowledge Manager 115*. It is the latter which decides on the salience of certain application parameters and the temporarily putting aside of others which are irrelevant for the moment. Thus, when the user has inquired for last minute offers about a Spanish seaside resort, the system will choose to ask about the preferred price range or area in the country rather than the desirability of mountain sport opportunities.

The realization of the new system dialogue act (with the corresponding domain parameter) creates expectations regarding the subsequent user input, i.e. the dialogue act for the next user input, as well as the related parameter and its values. Thus, the user is very likely to select one of the proposed parameter values (reply selection) or to request a clarification as to the meaning of the parameter itself (request repair). For example, in the case of a printer application, the system may ask about the desired resolution level, and the user may ask what resolution means. The system is always capable of providing information about the application world and what itself can do for the user (meta-questions).

The generated expectations about the next user input and its meaning in the context of what has taken place before guide the interpretation of the input by the system in two ways: (a) in terms of limiting the size of the salient vocabulary: some words will be more likely to be used in a specific context than others, thus benefiting the language *(NLP) processing modules* and improving their performance and (b) in terms of predicting the user reaction to the latest system action (e.g. a yes or no answer is more likely than not after a yes/no question), thus facilitating processing by *Dialogue Manager 120*.

The dialogue interpretation process is guided by a generic discourse grammar, a type of augmented transition network which specifies such local action-reaction pairs as:

- system question - user answer

Amazon/VB Assets
Exhibit 1004
Page 18

- system confirmation request - user confirmation or correction
- user request for explanation - system explanation, etc.

This state transition grammar also attempts to structure the various action-reaction states hierarchically in order to account for the number of different dialogues possible on a global level. Thus, this grammar can represent the various ways of combining local units to make up a whole dialogue between the user and the system. Despite the fact that this grammar represents a finite number of possibilities, flexibility and novelty are also allowed in the system. This is due to the independent existence of each action-reaction pair, and also because of the dynamic consultation of the application database and the ontology in the course of the interaction with the user. (See Appendices I and II for the list of possible system and user actions and reactions, i.e. the various dialogue acts.)

This discourse grammar probabilistically defines dialogue state transitions in the course of the interaction, with various alternative transitions possible at individual points. The grammar thus also allows for interruptions in the cycle (when the user has got a counterquestion, after being prompted by the system for an answer), e.g.

S: Do you want your printer to be high-resolution?
U: What's high resolution?

recursion (when there are repeated cases of misunderstanding or changes of mind), e.g.

S: Do you need laser quality output?
U: What?
S: Do you need high-quality laser output?
U: Yes, please.

and finally resumption of the topic previously being dealt with.

The discourse grammar is based on statistical data collected over previous user-system interactions and the way these were structured: these can be both dialogues over the phone and uni- or multi-modal dialogues over the WWW, or even off-line dialogues, as in the case of e-mail and voice-mail exchanges between the user and the system. This data is collected in a dialogue archive (shown later as 600 in Fig. 5), where it is annotated partly manually and partly automatically, by a *statistical mark-up module* in the latter case. Thus, alternative transitions from one system/user dialogue act to the next user/system dialogue act are associated with relative probabilities, depending on how likely they are to be applicable at each point. Initially,

Amazon/VB Assets
Exhibit 1004
Page 19

the grammar is based on a manual analysis of example interactions with users, both actual human-to-human dialogues and targeted Wizard-of-Oz dialogues, where a human is simulating the expected behavior of the machine on the basis of the programming behind it. The related learning techniques are discussed herein below.

The existence of such a discourse grammar and the related expectations about what the user will say next helps the system interpret the subsequent user input in its context and also identify understanding problems both on the part of the system itself and on the part of the user. In such a case, the system will adapt its strategies to first deal with the problem and only afterwards continue the normal flow of the dialogue. This is due to the fact that the system always monitors the evolution of the dialogue and how what is being discussed now relates to what has just been talked about, as well as all the topics that were talked about before that in the same interaction session. Breaks in the normal flow, such as requests for explanation or objections on the part of the user or erroneous recognition on the part of the system and the occurrence of ambiguities, can be identified and will trigger a repair mechanism (See Appendix III for an exemplary list of the repair strategies employed.).

Part of the repair strategy is to have the system warn the user that there is a problem and attempt to suggest solutions or request a clarification or confirmation by the user on the contested input. This is effected by means of corresponding system dialogue acts (request repair: warning, suggest, check), listed in Appendix III. An appropriate system message is produced and Communications Mediator 105 will decide on its presentation with or without accompanying images or graphics.

Another aspect of the repair strategies of the system is that, after a specific number of iterations trying to identify and solve the problem that has caused a misunderstanding, the system automatically transfers the interaction to a human operator to better deal with the user and to prevent losing the user as a customer. This is important, because otherwise the user will become frustrated and angry and, hence, feel negative towards the site and the products, services and information presented therein. This functionality of the invented processing environment is illustrated in Fig. 3. The unimodal or multimodal user input is fed through the system comprising Communications Mediator 105, NLP Manager 110, Dialogue Manager 120, Knowledge Manager 115 (Knowledge Base Retrieval 305 in Fig. 3). After the recurrence of an error or a misunderstanding - and despite the activation of clarification and repair subdialogues between the system and the user - Dialogue Manager 120 decides to route the user query, as it stands,

Amazon/VB Assets
Exhibit 1004
Page 20

to a Live Agent 315 who has a Call Center Knowledge Access Client 320 at their disposal. The agent can come back to the user over text chat on the Web, an e-mail or over the phone, taking the currently available user requirements as the starting basis. These requirements can be corrected and complemented with new ones. The invented architecture also allows for the user-controlled routing of the interaction to a live agent 325.

In a different embodiment of the invention, the user holds a dialogue with the system as the preparatory step in having their requirements processed and met. These are then always forwarded as a filled-in form to a human operator, who can then call the user back at the first opportunity. The operator can thus always consult the original dialogue (a recording of speech, for instance) to find out information that was not recognized by the system, as well as to correct data that was wrongly interpreted. The subsequent dialogue between the operator and the user (over the phone, through a Web form, or E-mail) can be used to check on the validity and correctness of the information collected by the system, before an appropriate product or service list, or other information is proposed to the user.

More specifically, the invented system first tries to solve a communication problem by means of internal computations, recruiting *Knowledge Manager 115* and its domain and general world expertise and the *Dialogue Manager* 120 and its discourse history tracing. The result is that the system comes back to the user with targeted clarification questions about ambiguous, incomplete, or nonexistent input. This process can involve two or three repair requests on the part of the system, each formulated and expressed differently from the others to prevent user irritation. If they all fail, then the transfer to live agent 315 takes place. This feature of the inventive architecture provides the solution to a major problem with current man-machine interfaces. Standard interfaces usually involve the constant prompting of the user for a clarification or repetition of what they said, thus sending the user away in frustration; they abandon the WWW site or hang up if they are communicating over a mobile or fixed phone. After this constant prompting, standard systems present a failure message and cut the dialogue off, which is equally frustrating for the user.

The present invention also allows for a user-controlled routing 325 of the call or interaction to live agent 315. The user may feel at a certain point the need for a more personal exchange, for example with regards to credit card payment options or clothing style considerations. They are always given this option, if that's what they prefer, at any stage in the dialogue. The user may then either directly talk to, chat in a separate WWW window with, or e-mail the

Amazon/VB Assets
Exhibit 1004
Page 21

human operator.

Apart from dealing with recognition and understanding problems for the system, the present invention also covers cases of user misunderstanding. The user may have an incorrect or incomplete view of the domain world, as in the case of a naive computer user wanting to buy their first laptop, or of a user who incorrectly assumes that there are printers which cost as low as $50. *Knowledge Manager* 115 will identify such inconsistencies with the system ontology and the application databases and will trigger a modification of the dialogue strategies of the system in this case, too. (Again, see Appendix III for a list of the repair strategies employed).

As a side-effect of this repair strategy, user misunderstandings are also archived and provide the basis for a tool that can identify recurring misunderstandings, i.e. missing items in the product spectrum or confusing system messages. This, in turn, provides invaluable feedback to the product, service or information providers themselves, who can shape their decisions on their future offerings accordingly. It also influences the formulation of the system prompts for the future.

The application of the repair strategies of the system in the case of user misunderstandings and errors includes having the system warn the user about the problem and explain what the reason was. For instance, the system may specify allowable concepts, e.g. that the engine of a printer can be laser, dot matrix, or inkjet; or let the user know the acceptable value range for a domain parameter, e.g. that the lowest price for a CD player is $50. The system will then attempt to offer a realistic alternative (product, service or information item) that is close, or as close as possible, to the user´s original requirements. The user has the option to modify their requirements accordingly, to accept the proposed alternative (positive interest) or even cancel their initial request and leave the site.

Despite the fact that the user always has the option to abandon the site or cancel the initial request, the system will always try to offer attractive alternatives to the user in the case where their requirements cannot be exactly met, so that the user is tempted to remain a little longer. For instance, the user may want to buy a cheap laser printer (less than $200) and the system may suggest a very good color inkjet printer at a lower price. At any rate, the system always has this principle of offering alternatives, even when finding a database match is straightforward. In this context, cross-selling can also be performed, whereby more or less relevant products, services, and information are concurrently presented on the WWW page or the WAP interface, or through speech output.

Amazon/VB Assets
Exhibit 1004
Page 22

Irrespective of whether there have been any misunderstandings on the part of the system or the user, the initial database search in trying to satisfy the user requirements about a product, service, or information item will be repeated during the interaction a number of times. Every time the user has provided values to additional domain parameters, the search will be modified dynamically. For example, in a last minute travel application, the initial user query may have been on the preferred date of travel and the departure airport. Subsequently, the number of people traveling and the resort preferences may also be specified in separate steps or concurrently in a single utterance. In each case, the result of the database retrieval will be different and the result set smaller. It should be noted that asserted "facts" about the values of individual parameters can also be modified at any point during the interaction, following the user´s changes of mind or corrections. Thus, the present system can deal with the case where, for example, the user has first identified their preferred holiday resort as Spain and during the dialogue they decide that they would prefer to travel to Portugal after all.

It is the task of the system to always attempt to pose the right questions to the user that will lead to a fast identification of the best possible database match that will fulfill the user requirements. This means minimizing the number of *questions* asked in order to prevent user frustration and, at the same time, maximizing the *relevance* of the questions posed in order to quickly attain a best match retrieval.

Intelligent search is achieved through *Knowledge Manager* 115, which queries the ontology about the most salient features of single entities and the most salient relationships between entities. This means that the specification of a certain parameter (i.e. of a feature of a product, service or information item) entails the need for the specification of a related parameter and the parallel blocking of a third feature, which need not be asked about at that point in the interaction. Salience is context-dependent, i.e. it cannot be specified in advance or for all possible cases. For this precise reason, dialogue planning cannot be effected in relation to application parameters, although some defaults can be used as a back-up solution. The list of parameters to be asked about changes depending on the inferred or identified profile of the user, as well as on the restrictions identified by Knowledge Manager 115.

The most salient features and entity relationships will remain active throughout the remainder of the interaction with the user, to the extent that they maintain their salience. This is in order for the system to (a) *pose*

Amazon/VB Assets
Exhibit 1004
Page 23

targeted questions about them that will drastically restrict the size of the search space in the application database and (b) successfully predict the features and relationships that will be asked next by the user, which in turn will aid the interpretation of the corresponding user input by NLP Manager 110. At the same time, the selection of a feature or relationship as salient also depends on subsequent assertions by the user, i.e. salience changes dynamically.

After a number of system queries to the user for the further specification of the user requirements, a small number of database matches will have been isolated. These will be presented to the user in certain media combinations, e.g. natural language text and graphics, a decision that is taken by *Communications Mediator 105*. There is the additional possibility for the presence of WWW links to the individual E-commerce sites of each product or service presented, as well as related sites.

The user can select one - or more, in the case of comparison questions - of the database matches proposed by the system and pose additional questions about them, and/or ask to make a purchase (commit). The latter functionality is smoothly integrated in the invented architecture by means of URL links to the respective sites, for the case of Web- or e-mail-based dialogues.

The system always keeps a record of all the exchanges in a specific interaction session with the user (discourse history), both in order to contextually interpret each new user input, and to be able to continue the dialogue from where it was left off after the user has been to one of the suggested sites and decided to examine an alternative product, service or database hit. Thus, the present invention allows for a re-entrance in the dialogue. The user may, for instance, select to view more information on a specific package holiday offer. They click on the corresponding WWW link and view the details. Then, the user decides that the available facilities are not satisfactory for their needs, for example, when the person involved is disabled. The user goes back to the system and picks up the conversation from where it was last left off: having already specified the desired destination and travel dates, the user can now search the database with the new accessibility criterion. As a result of the discourse history record, the system will not ask the user to specify anew the desired destination and travel dates, but will directly accept the new (accessibility) parameter and integrate it with the existing ones.

The advantages of the system architecture and the related methodologies of the present invention can be separated into two different groups, usability-related and technical ones.

Amazon/VB Assets
Exhibit 1004
Page 24

Usability-related

The user does not need to know beforehand or remember during the interaction with the system restrictive keywords about the corresponding domain or application. Consequently, even non-experts and naive users can take advantage of the intelligent and human-friendly search that the invented system offers.

The user may choose their preferred expression medium, whether typed natural language text, speech, mouse clicks, pointing and gestures, or a combination thereof. Thus, the communication channel between the user and the inventive system is also varied: from text- and voice-based messaging systems such as e-mail, to the World Wide Web, to the standard phone, or even internet-enabled (WAP) mobile phones.

The employment of a probabilistic dialogue grammar ensures the robust and efficient interpretation of the next user input in terms of intentions (i.e. user dialogue acts). It also means allowing flexibility in the structuring of the interaction itself, which gives the user more freedom than standard IVR (Interactive Voice Response) and speech recognition systems, or typed natural language interfaces to databases. This means that the user is not obliged to just passively answer the system´s questions, but can also take the initiative to have something explained to them or to sidetrack to a different topic.

The dynamic accessing of the application databases in the course of the interaction with the user ensures efficiency in task completion. Thus, long waiting times and the related user frustration are avoided.

Accessing the application databases dynamically results in posing targeted, pragmatic, and relevant questions to the user in order to collect information on their requirements for the product, service or information sought. Thus, the system exhibits intelligence, by appearing to be coherent and logical.

The user can, at each point, ask the system questions about the application and the domain itself, or the functionality of the system as a whole: what the various parameters mean, what range of values they can take, or what type of queries the system understands. Thus, the user is never at a loss as to what they can do with the system or what they can expect from it. At the same time, the system facilitates product, service and information search for the non-expert user who is not familiar with the special terminology or the domain world. This is achieved by having

Amazon/VB Assets
Exhibit 1004
Page 25

the system resume the initiative, when the user does not.

The invented system environment has an inventory of repair strategies at its disposal, which are adopted dynamically during the interaction with the user, as soon as:

1. a misunderstanding on the part of either the user or the system is identified,
2. no database matches can be found that satisfy the user requirements
3. too few or too many results are identified after the execution of the database query
4. user input is found to be ungrammatical, ambiguous, or conflicting.

Thus, the system can be alerted to any false presumptions of the user and notify them accordingly (*warning, correct, suggest* in Appendix III), so that they can either modify their specifications or cancel the search. Likewise, the system can identify early on instances of erroneous processing of the user input (*check*, request confirmation), rather than continue the interaction for a long time and present irrelevant search results. The repeated occurrence of a misunderstanding between the user and the system a pre-specified number of times automatically forwards the user's query to a human operator. Thus, frustration and a negative image for the corresponding product, service or database can be avoided, and potential customer and site visitor gain and retention accomplished. In other words, the present invention facilitates both human-machine and human-human interaction, such as the employed repair strategies.

The user can themselves initiate the procedure of being routed to a human agent, at any point in the dialogue. There is no predefined series of steps that have to be completed first before this can take place.

In the case of busy call center environments, the user does not even have to wait in long queues in order to speak to an operator, but rather can specify their initial requirements through a voice- or text-based dialogue with the system, which can then pass them on to the agent. The agent can then get back to the user when they are free, already knowing approximately what the user is after and thus appropriately posing targeted questions and proposing a list of matching products, services and database entries.

The present system is capable to learn heuristically from the human operator´s behavior and extend its knowledge base and dialogue strategies accordingly. To this effect, machine-machine interaction is carried out in extending and coordinating the corresponding system components, described

Amazon/VB Assets
Exhibit 1004
Page 26

herein below.

The loss of a prospective customer is also prevented by always attempting to offer alternative products, services and information to the user that largely -- but not totally --match their requirements, even in the case where the database search has failed to generate a result.

The multimodal presentation of the retrieved product, service and other information that best matches the user requirements ensures that the user will be interested in finding out more about it or even make a related purchase by getting directed to the corresponding E-commerce site, when applicable, through a URL link.

The inventive system environment is also multilingual, i.e. it presents the information retrieved from the databases in the native or preferred language of the user, irrespective of the original language this information is stored in. Thus, the user can comfortably express their wishes without having to learn multiple languages or struggle with the ones they speak as foreign. Importantly, the user does not need to explicitly specify the language they are going to formulate their input or query in. Language identification is done automatically by the system in the context of NLP Manager 110 in the preferred embodiment of the invention.

The user can resume the interaction with the system once they have left the main Web page or service and visited one of the suggested sites or services presented by the system.

Importantly, interaction in the inventive environment is personalized, because of the user models and personal profiles maintained and constantly updated by the system with each new session. This entails the user-specific formulation and presentation of system messages, including favoring certain modes and combinations thereof to others. It also means tuning the vocabulary and grammars used for the recognition of the user input to the type of user identified or inferred. Thus, students are expected to speak and write using different expressions to those employed by senior citizens. At the same time, students may like to see videos and animation on a Web site, whereas senior citizens may find them confusing and overwhelming.

Technical Benefits

The present system supports multilinguality, in that there is no need for the user to explicitly choose the language their input is going to be formulated in at the beginning of the interaction. Rather, the initial user query is subjected to automatic language identification procedures, which lead to the dynamic loading of the

Amazon/VB Assets
Exhibit 1004
Page 27

corresponding language-specific grammars and lexica by the NLP modules.

The parallel use of sophisticated language processing techniques and surface statistical methods (based on semantically annotated data) in analyzing the user input means that the user intention can be better understood and inferred on the basis of the structure and the syntax of the user input, as opposed to individual isolated keywords. This is especially important in the case of multiple specifications: for example, when there is a negation combined with a juxtaposition of an alternative value, meaning that the new value should be kept and the old one rejected.

The employment of an arbitrator during natural language processing ensures robustness in the case where sophisticated linguistic analysis fails to deliver a unique result or where statistical analysis has not identified any keyword pattern to use as the basic theme of the conversation.

The weighting of the language processing results ensures that the best will be employed at the subsequent stages in the processing, but also that the results with a lower weighting will be available, when the preferred interpretation is proved wrong or incompatible with the user requirements. Thus, backtracking is allowed in the invented system.

The concurrent use of domain parameters and dialogue acts in the representation of the meaning of the user input assures robustness in its interpretation, in the case where a value for either could not be identified. At every point, the topic of the conversation will be known (the domain parameters) and, or at least, the reason and motivation behind the user providing a specific input at the specific time point (user dialogue act following system dialogue act). Although this type of dual representation is not completely new, the specific list of dialogue acts for the system and the user and the related repair strategies employed in the invented environment, as well as the manner in which the user dialogue acts are automatically identified in the user input are two of the innovations regarding the inventive system and methodology.

The maintenance of a discourse history for each interaction session means that each new user input is interpreted in the context of the preceding ones, including those of the system itself, and not in isolation. This is in contrast to most database interfaces, whether keyword- or natural language-based. As a consequence, even incomplete, ungrammatical, or ambiguous input (e.g. with anaphors) by the user can be interpreted appropriately using this memory

Amazon/VB Assets
Exhibit 1004
Page 28

as the context or the background of the exchanges. This is another facet of the robustness of the invented system and methodology.

The use of a discourse history and a probabilistic dialogue grammar means that the next user input will be processed very efficiently and robustly due to the generated expectations about it. This, in turn, restricts the size of the active vocabulary that needs to be recognized by the system (in the case of a spoken interface, for instance) and the number of the domain concepts and related features and relationships that are likely to be talked about by the user.

The employment of a generic ontology and application-specific knowledge bases by the system entails the activation of inference procedures as part of the processing of the user input, which result in the disambiguation of ambiguous and the completion of under-specified input. This domain knowledge is kept separate from the knowledge about the dialogue structure, which renders dialogue management in the invented environment application-independent and reusable.

A large part of the ontology of the system concerns knowledge about general entities and their relationships, which ensures their portability to other domains and applications. Thus, reusability is inherent in the present invention.

Intelligent database search means that the system always poses the next question to the user in a targeted way that is based on the current search results that were retrieved dynamically during the interaction, and on general knowledge about the domain concepts and their interrelationships.

In summary, the present system and methodology allow for the smooth integration of human-machine, human-human, and machine-machine interaction for robustness, efficiency, and user-friendliness.

The deployment of the present multimodal interaction system follows a "boot-strapping" methodology; an initial, functional version of the system is installed and activated, and the system is then successively improved through the monitoring, classification and archiving of actual dialogues. Archived dialogues can be classified with respect to a number of dimensions. These include:

1. Whether the dialogue was *human-machine* or whether it was *human-human* (e.g. after a referral from the machine dialogue system, when it detected sufficient difficulty, or when the user themselves opted to talk to a live agent directly).

Amazon/VB Assets
Exhibit 1004
Page 29

2. Whether the dialogue was known to be *successful* (e.g. led to a purchasing transaction or to the retrieval of a sufficient number of results) or whether it was known to be *unsuccessful;* e.g. the customer ended the dialogue, because their wishes could either not be understood or fulfilled.

Each new dialogue, classified in this way, will be held in a dialogue archive (shown as 600 in Fig. 5, and discussed herein below). This archive is inspected and analyzed on a periodic basis via Performance Evaluator 605 in order to effect the continual improvement of the machine component of the dialogue system. The goals of the improvement are to reduce the number of referrals from the machine dialogue component (i.e. reduce the number of human-human dialogues that were not initiated by the user themselves, except in the case where the interaction with a live agent is the default after a preparatory dialogue with the system), while maintaining or improving the overall number of successful dialogues. Secondary metrics, such as reduced average transaction time, could be used to measure the process of continual improvement.

The infrastructure for the above type of analysis and learning is shown in Fig. 4 for initial domain knowledge acquisition and in the domain and dialogue knowledge maintenance infrastructure of Fig. 5. The archived dialogues 600 in Fig. 5, which would constitute part of corporate knowledge 405 in Fig. 4, first undergo an information extraction procedure 625 and are filtered via a statistical clustering technique 550 (see also processes 410 and 425 in Fig. 4). The goal is to recognize dialogue types or parts of dialogues (e.g. user utterance and machine response) that commonly re-occur (resulting in the Indexed, Normalized and Classified Dialogues 555 of Fig. 5). This conceptual analysis is aided by linguistic (NLP)-based processing (NLP Analyzer 505) similar to that used to analyze the user utterances at run-time, in order to extract other types of information about the user input and the dialogue structure itself. Thus, new rules can be discovered and old ones modified on the correspondence between user intentions (dialogue acts) and the way users express these (their surface lexical and syntactic manifestations). The same principle is used for the training of the NLP modules of the present system: annotated dialogues are employed to automatically develop domain-specific lexica and grammars and to tune the parser of the system. Fig. 4 shows the same procedure specifically for the acquisition of ontological knowledge, i.e. new concepts for entities, their features, and their interrelationships.

It is important to recognize that there is a number of different classes of failure that can occur in the dialogue

Amazon/VB Assets
Exhibit 1004
Page 30

system and these classes need to be handled and repaired in different ways. Therefore, the output of the dialogue cluster analysis 555 is fed into a *Failure Classifier (560,* 565, 570) which routes each individual dialogue fault or each recurrent dialogue fault to one of a number of modules in a *Dialogue Maintenance Component,* in a preferred embodiment of the invention.

The first class of failure is one of missing terminology on the part of the NLP components of the dialogue system (Terminology Failure Recogniser 560). This occurs whenever the user employs a word or phrase to refer to an existing concept within the domain ontology 535 and the mapping for that word or phrase does not yet exist. The repair strategy is to extend the domain-specific lexica 540 used by the NLP component. A candidate set of unknown phrases 510 is fed to a *Lexical Repair Module* 520 (process "1" in Fig. 5*)*. In some cases, the appropriate mapping can be created automatically by analysis of successful human-human dialogues which also involve the unknown terminology (note that "unknown" here is scoped with respect to the automated dialogue system only; the human agent is assumed to understand the terminology). This analysis extracts the interpretation that the human made for the unknown phrase or word.

A second class of failure is in the sub-optimal performance of the knowledge base. Initially, the knowledge base will contain a straightforward structuring of the domain knowledge (e.g. an ontological classification hierarchy for related types of product, Domain Ontology 535). For example, this might represent that "Laser Printer is a type of Printer". Associated with each concept in the ontology is a collection of meta-knowledge extracted from the portion of the database of product instances that the concepts represent. For instance, for each concept an explicit representation of the price range of that concept exists and can be exploited by the dialogue system, e.g. to interpret the meaning of a "cheap Laser Printer". Statistical analysis of the Dialogue Archive 565 will help to identify combinations of product and service features that are frequently asked for by customers, e.g. "a fast color printer". By adding such commonly occurring combinations of features (Frequent Feature Value Assertion Sets 575) as explicit concepts in the Domain Ontology 535 via the Ad Hoc Category Generator 585 (process "2" in Fig. 5), the range of explicit knowledge that Dialogue Manager 120 can access and use in a transaction with a customer is broadened: for example, providing a more precise interpretation of "cheapness", when a customer asks for a "cheap, fast, color laser printer". By deriving such ad hoc concepts as "fast color printer" on the basis of archived dialogues and not via some a priori knowledge engineering, a proliferation in the number of additional concepts in the knowledge base is avoided, while ensuring that the most

Amazon/VB Assets
Exhibit 1004
Page 31

salient concepts for the specific transaction application are covered.

A third class of failure can be referred to as a lack of coverage in the product catalogues and databases themselves (Unavailable Product Recogniser 570). This occurs, for instance, whenever a customer specifies a set of requirements for a product and no one product exists that fulfills all requirements. This can occur even in successful dialogues, assuming that the dialogue system subsequently carries out a successful negotiation to weaken some of the initial requirements of the customer. However, the initial set of requirements posted by a customer should not be discarded. Rather they should be detected by the Dialogue Maintenance component 570 and sent to a *Market Analysis Module* 590 (process "3" in Fig. 5) as failed feature value assertion sets 580. If the same or similar set of unmatched customer requirements re-occurs often enough, this will be recognized by the module and included in an automatically generated *Market Analysis Database* 595. One of the main purposes of such a database is to provide valuable feedback to the product manufacturers and service providers themselves. This type of feedback is only enabled by having a front-end dialogue system that encourages a returning or prospective customer to freely and openly enter their requirements (in contrast to menu- or keyword-driven systems). This type of failure does not require an improvement to the core dialogue system itself.

A fourth class of failure is a specialization of the first class of failure (Terminology Failure Recogniser 560). It involves the customer referring to a concept, such that not only is the terminology not represented in the domain lexical 540 used by the NLP component, but there is no existing representation of the underlying concept within the Domain ontology 535 (process "4" in Fig. 5). A failure of this class is recognized by upgrading a failure of the first class, once it has been shown that the failure cannot be repaired by a simple extension to the terminology represented in the lexica. Usually this type of failure can only be manually corrected by a human knowledge engineer via Ontology Editor 530. This process is however supported by sending the unknown term/concept 525 detected by missing concept recognizer 515 to a *Knowledge Repair Scheduler* that internally stores the failure. The failure is exported to a knowledge engineering team on request or in response to a periodic maintenance timetable, e.g. via automatically generated and electronically sent change requests.

Amazon/VB Assets
Exhibit 1004
Page 32

APPENDIX I: List of System Dialogue Acts

1. **Opening:** This is a Dialogue Act (Dact) for the initial system message, either a

   "Hello, I can give you information on printers and scanners"

   or a short explanation of what the user can expect from or ask the system.

   In this way, the system can guide the user in what they could say and thus avoid the situation of unknown or out-of-domain input.

2. **Query_w:** This is for open-ended questions (What, Which, When, Why, How) and in the case of the system is always followed by a Query selection or a Query yn so that the user is prompted to choose from the alternatives suggested or to answer with a yes or a no, respectively.

3. **Query_selection:** This Dact aims to guide the user in the world of product and service features by offering them specific alternative values to choose from. So, in the case of printer output, the system can ask

   "Do you want to have black &white or color output?"

4. **Query_yn:** is used to restrict the user in their answer by asking for a simple yes or no answer to a question, e.g.

   "Are you interested in a color printer?"

   The user may reply with more than a yes or no (e.g. with an additional specification about the resolution), but at least the system will understand the general preference or attitude of the user to the corresponding product or service feature.

5. **Repetition:** This Dact is used for when the system repeats a request or explanation to the user, because the user themselves asked for it (with a Request repetition Dact) or because the system did not understand what the user's reaction to the initial request or explanation was (erroneous speech recognition or occurrence of misspellings and typos). It should be noted that the system formulates the second or third request always slightly differently from the initial query, so that the user is not irritated by the repetition.

6. **Explain:** This Dact is used to offer more information to

Amazon/VB Assets
Exhibit 1004
Page 33

the user about a product or service feature, or the meaning of a domain parameter, so that the user can provide the desired value for the database search. For example, it will be used when the user is asked about the resolution of the printer they would like to buy and the user does not know what resolution means in the first place. Irrespective of whether the system should go on asking for this parameter value further, the user has to be given an explanation, as the case should be also with a failed database search that fulfills the user's criteria.

7. **Check:** This Dact is used to have data that the user has just given confirmed, especially in the case of ungrammaticality and uncertain speech recognition (below threshold recognition scores). The system is trying to confirm that it has understood correctly, as the case can be when the user changes the value for an already discussed parameter (because they have changed their mind for example, or because the system misrecognized a previous user query). The reaction to this Dact on the part of the user is either an acknowledgment, i.e. positive feedback (Ackn, positive interest) or a correction of the corresponding parameter value (Correct, negative interest).

8. **Suggest:** This Dact is used to present appropriate database entries to the user after a number of their criteria has been collected. It is also used to offer alternatives to the user when an exact match to their requirements cannot be found (through constraint relaxation).

9. **Busy:** This Dact could be employed to indicate to the user that the system is carrying out a search operation or is in the middle of processing (possibly, the user utterance itself!) at the moment. This is an especially useful feature in a spoken dialogue system, where the user needs more frequent feedback on what is going on due to the lack of visual clues (a problem that is obviated in the case of WAP interfaces). E.g.

   "Let me see. I'm looking through what is available."

10. **Warning:** This Dact is used to tell the user that their requirements cannot be met exactly, i.e. no product or service listed has all the features the user has asked for. This can be followed by an explanation (Explain), so that the user can be informed about the reason of this failure. This Dact can also be employed for the case where the system has had difficulties processing or fully interpreting the last user input (unknown words or low recognition scores). In this case, the Dact will be

Amazon/VB Assets
Exhibit 1004
Page 34

followed by a Check in order to have the recognized information confirmed by the user.

11. **Failure**: This Dact will be employed to tell the user that absolutely no product or service available meets the user's wishes. This can be necessary when the user is a domain expert, for example, and knows exactly what they are looking for and cannot be offered just any similar product or service. This does not need to be the end of the interaction, as the system may suggest alternative search sites or products all the same (Suggest). The point is that the system is able to tell the user 'the truth' sometimes rather than trying to sell at all costs. This Dact will probably only be used when customer satisfaction and retention are more important in an application than market segment augmentation or profit making.

12. **Request_repetition**: This Dact is used to ask the user to repeat their request or utterance because of bad speech recognition or internet server problems which resulted in the system not receiving any user input at all or only incomprehensible segments. This will also be useful when the system cannot obtain a semantic representation of the user input, either because of out-of-vocabulary words and phrases, or/and because of an out-of-domain user request. This Dact can be followed either by a Repetition or a Clarify act on the part of the user. In the latter case, the user may have chosen to reformulate their initial query differently, probably with the addition of information on new parameters.

13. **Reply_w**: This Dact carries the search result that answers an open-ended query (Query w) by the user, such as "Show me all HP color laser printers".

14. **Reply_selection**: In contrast to the previous Dact, this answers a user query about a list of elements (usually two), e.g. "Are there only European or also US trips on offer?". The system can reply that both are on offer, but the point is that this type of reply will be differently formulated from a reply to a yes or no question (e.g. "Have you got US offers?"). The "neither - nor" case is also covered here.

15. **Reply_y**: This is the direct positive answer to a yes / no question that was posed by the user (Query yn). It can also be used to accept something that the user has said (e.g. because it is allowed by the knowledge base or covered by the database).

16. **Reply_n**: This Dact is used to give a negative response to the user after the latter has posed a yes / no question (Query yn). It is usually followed by an Explain dialogue

Amazon/VB Assets
Exhibit 1004
Page 35

act, providing a reason for this negation, and possibly alternatives that the user can consider (Suggest).

17. **Meta-statement:** This is a special Dact that is used to convey information about the interface itself. It will be used, for example, to talk about the product images that are shown on the screen or to refer to web links that can be clicked and also to their relevant position in the layout. This is especially critical when the user has decided they want to buy the suggested product or service and want to know how to proceed, e.g.

"You can see what this product looks like from the image at the bottom left of the screen."

18. **Instruct:** This Dact is used to explain to the user how to proceed with a purchase, for example, or with a search (what features they could ask about). It is an important trait of any man-machine interface to tell the user what to do in a step-by-step fashion, when they are not sure how to go about asking or searching for something. E.g.

"You can now click on the desired product and be transferred to the manufacturer´s site where you can further process or confirm your order."

19. **Correct:** The system can tell the user through this Dact that their view of the world (in terms of features and their allowed values) is different from that of the knowledge base and discrepancies have occurred. For example, the user may think that Lisbon is in Greece and they want to take a flight to Athens to this effect. The system has to correct the user before trying to find something appropriate in the database, because the user may change their minds when they realize their mistake. E.g.

"Unfortunately, the cheapest printer available at the moment costs €80, so there is nothing at €40."

20. **Ackn:** This Dact is used to confirm information and data that the user has assumed and asked the system about. For example, the user may first want to make sure there are no printers that are cheaper than €80 before putting forth their price requirements. This is a reaction to the user Dact Check.

21. **Closing:** This Dact represents the final system message before an interaction with the user ends, especially in the case of a spoken interface, where an explicit end to a conversation has to be made due to the lack of visual clues (except in the parallel use of WAP). The exact message formulation can be adapted to different user types (depending on age, sex, expertise). E.g.

Amazon/VB Assets
Exhibit 1004
Page 36

"See you later" for young users
"Thank you for using the SemanticEdge service" for older ones.

Appendix II: List of User Dialogue Acts

1. **Opening:** This Dialogue Act (Dact) can be used to represent an opening or greeting phrase by the user, such as *"Hi <Name of System>"* followed by a specification of their requirements. This can be a reaction to an Opening Dact on the part of the system.

2. **Query_w:** This Dact is for open-ended questions on the part of the user, i.e. What, Who, When, Where, How questions, which do not pose any limitations to the scope or detail of the answer. E.g.

   "I´m interested in a color laser printer."

   This roughly translates into

   "What color laser printers have you got?"

3. **Query_selection:** In contrast to Query_w, this Dact expresses a question about a specific list of items, which can be interpreted either disjunctively (as mutually exclusive) or conjunctively (Don´t mind which), depending on the existence of constraints in the knowledge base about the parallel activation of more than a single value for the same parameter. E.g.

   "Have you got color printers or just black and white?"
   "I´m interested in HP and CANON printers."

4. **Query_yn:** This is an even more restrictive type of question than the previous two, as the user is asking the system for either a "yes" or a "no". E.g.

   "Are HP printers very expensive?"

   Of course, the system can and should complement the answer (Reply_y or Reply_n) with more details from the database (Explain), especially in such a sensitive case for successful selling. For instance:

   "Yes, HP printers are more expensive than Epson or Canon printers, but they have some additional features, such as high speed and high resolution."

   or alternatively

   "The cheapest HP printers cost €170, whereas the top of the range can cost up to €750."

5. **Repetition:** With this Dact the user repeats the same

Amazon/VB Assets
Exhibit 1004
Page 38

information that they have given in a previous dialogue step. This can occur, for example, in the case where the system has not been able to recognize the (spoken) user utterance and has asked the user to repeat it (Request repetition). Alternatively, the user may repeat something after the system has asked the user to confirm the value for a parameter (i.e. after a Check on the part of the system). E.g.

"Yes, at €300."

6. **Clarify:** This Dact is used to convey additional information on the user´s requirements for the desired product or service. This means that the user provides the values for as yet undiscussed parameters. This can also occur while the user is answering a system question about a different parameter. The user could provide an answer and also specify new parameters that the system was going to ask about later, e.g.

"Yes, color with 600x600 resolution."

Sometimes, a clarification act can concern data that is outside the domain of coverage of the system or outside the vocabulary, in which case it is going to be marked as such and dynamically learnt thus extending the lexical and the knowledge base. The user will also be alerted to the problem and its cause. E.g.

"We´ve got just a few employees, so there is not much printing being done."

7. **Check:** This user Dact aims to have something confirmed by the system, either because the user is not certain they know or because they are not certain they have understood (or don´t want to believe!) what the system has previously said. E.g.

"There are only inkjet printers by Epson, isn´t that right?"
"What?! The cheapest HP printer costs €250??"

8. **Request_repetition:** This Dact is used to ask the system to repeat the last prompt, because the user did not hear it well over the phone or because of internet server problems which resulted in the user not seeing the prompt on the screen in a web interface environment. This Dact can be followed by a Repetition and an Explain act on the part of the system. In the latter case, the system tries to clarify its request or the information it has just provided.

9. **Request_repair:** This Dact is employed for the cases where

Amazon/VB Assets
Exhibit 1004
Page 39

the user does not know the meaning of a parameter just asked by the system (with a Query selection or a Query yn Dact). The system has to provide an explanation (Explain) of what this parameter stands for and a listing of the possible instantiations it can take, irrespective of whether it is going to ask once again for this parameter to be specified or just move on to a different parameter. E.g.

"What is high resolution? I don´t know."

10. **Reply_selection:** This Dact represents the user reaction to a system Query selection Dact, i.e. to a prompt by the system for the user to choose among specified alternatives (usually instantiations for a parameter). E.g.

"Colour printers" or
"Not black and white, color of course" (spoken interface)

The "neither - nor" and the "both" cases are also covered here.

11. **Reply_y:** This is the positive reaction to a system Query yn Dact, whereby the user has been asked to reply with a yes or a no to a question.

12. **Reply_n:** This Dact expresses negation or rejection on the part of the user, for example after the system has asked the user a simple yes/no question (Query yn). E.g.

"No, I´m not interested. What about Canon printers?"

13. **(Meta-statement):** This Dact is for the cases where the user asks about the interface itself, for example, the images or the text, the associated web pages or even the different domains covered, if more than a single domain is covered by a system at the same time. E.g.:

"Where does this link take me?"
(pointing to a detailed product description Web link)

14. **Correct:** This is probably the most important Dact, which intends to correct the instantiation for a parameter, either because the user changed their mind in the course of the interaction or because the system has misrecognized or misinterpreted a previous user input in the first place. This is especially relevant for the spoken language interface, where input recognition is much more difficult and uncertain. E.g.:

"Not Epson, Inkjet printer I said" (spoken interface)

Amazon/VB Assets
Exhibit 1004
Page 40

Whichever the case, there is always going to be a system follow-up with a Check Dact, trying to have the new / modified parameter value explicitly confirmed by the user, so that the misunderstanding is cleared up early on.

15. **Ackn**: This Dact is the positive feedback to a Check Dact on the part of the system, i.e. the user acknowledges a piece of information just queried about by the system. This information may be something that the user has already provided or a default parameter value that the system has assumed and wants to have confirmed (knowledge base inferences or hard-coded rules). When the user wants to express disagreement, the Correct Dact will be used instead, probably accompanied by a Reply n (direct rejection) by the user. E.g.:

"Yes, that's right."

16. **Cancel**: This Dact will be used to change the topic or even the domain in the middle of an interaction. The user may want to switch from printers to scanners or from computers to travel offers in the same dialogue session. Cancel clears all the parameter values from the dialogue history and a new dialogue history is set up for the new topic or domain. This Dact will be triggered when a new domain pattern is identified in the user input. In the case of spoken input, the canceling operation has to be more direct, because it is inefficient to have all vocabularies active at all times (which leads to inaccuracies in speech recognition). E.g.

"Forget it. What about last minute offers"

17. **Closing**: This Dact expresses the final utterance of the user before they hang up or leave the site and is mainly relevant for the spoken interface, as the user who interacts with a Web page can walk away any time. E.g. *"Okay, thanks."*

18. **Positive_interest**: This Dact is used to express a positive reaction on the part of the user towards a suggestion that the system has just offered, i.e. about a database result just proposed (a package holiday, a specific flight, a laser printer etc.). It usually follows the system Dact suggest. This Dact differentiates between a simple acknowledgment by the user about the offered results (ackn, which just lets the system know that the user has heard or seen the results of the retrieval) and a strongly positive attitude to the system suggestion. Example positive reactions are:

"That's perfect!"

Amazon/VB Assets
Exhibit 1004
Page 41

"Just the right thing."
"Exactly what I wanted."
"That's interesting. Can I have more details please?"

19. **Negative_interest:** Similarly to the case of positive interest, this Dact is employed to distinguish between a simple reply n, i.e. a negative answer by the user to a system y/n question (query yn), and a downright negative reaction to the database results or other suggestions (suggest) the system has come back with. It shouldn't be confused with cancel either, which is a user Dact used to abandon the whole task (a trip to Majorca in April, for example) and start a new one (a trip to India in the summer) or end the interaction. Example negative reactions could be:

"No, I don't like this type of thing. What else have you got?"
"What? Forget it. What about at a 3-star hotel?"
"Too expensive. What about from Nuremberg?"

20. **Commit:** This Dact indicates that the user is committed to buy or book a product or service just presented by the system. This is important for the system actions to follow, for example whether a cross-selling operation is going to be activated to promote similar or related products and services, or whether the new purchase is going to be integrated in the specific customer's profile (buying behavior). Example manifestations of this Dact could be:

"Great! I'd like to book that."
"Sounds good. How can I pay?"
"Excellent. Will I get a receipt directly sent to my home address?"

Amazon/VB Assets
Exhibit 1004
Page 42

Appendix III: List of Repair Strategies

1. **System Query_yn**: is used to restrict the user in their answer by asking for a simple yes or no answer to a question, e.g.

   "Are you interested in a color printer?"

   The user may reply with more than a yes or no (e.g. with an additional specification about the resolution), but at least the system will understand the general preference or attitude of the user to the corresponding product or service feature.

2. **System Repetition**: This Dact is used for when the system repeats a request or explanation to the user, because the user themselves asked for it (with a Request Repetition Dact) or because the system did not understand what the user´s reaction to the initial request or explanation was (erroneous speech recognition or occurrence of misspellings and typos). It should be noted that the system formulates the second or third request always slightly differently from the initial query, so that the user is not irritated by the repetition.

3. **System Explain**: This Dact is used to offer more information to the user about a product or service feature, or the meaning of a domain parameter, so that the user can provide the desired value for the database search. For example, it will be used when the user is asked about the resolution of the printer they would like to buy and the user does not know what resolution means in the first place. Irrespective of whether the system should go on asking for this parameter value further, the user has to be given an explanation, as the case should be also with a failed database search that fulfills the user´s criteria.

4. **System Check**: This Dact is used to have data that the user has just given confirmed, especially in the case of ungrammaticality and uncertain speech recognition (below threshold recognition scores). The system is trying to confirm that it has understood correctly, as the case can be when the user changes the value for an already discussed parameter (because they have changed their mind for example, or because the system misrecognized a previous user query). The reaction to this Dact on the part of the user is either an acknowledgment, i.e. positive feedback (Ackn) or a

Amazon/VB Assets
Exhibit 1004
Page 43

correction of the corresponding parameter value (Correct).

5. **System Warning**: This Dact is used to tell the user that their requirements cannot be met exactly, i.e. no product or service listed has all the features the user has asked for. This can be followed by an explanation (Explain), so that the user can be informed about the reason of this failure. This Dact can also be employed for the case where the system has had difficulties processing or fully interpreting the last user input (unknown words or low recognition scores). In this case, the Dact will be followed by a Check in order to have the recognized information confirmed by the user.

6. **System Failure**: This Dact will be employed to tell the user that absolutely no product or service available meets the user´s wishes. This can be necessary when the user is a domain expert, for example, and knows exactly what they are looking for and cannot be offered just any similar product or service. This does not need to be the end of the interaction, as the system may suggest alternative search sites or products all the same (Suggest). The point is that the system is able to tell the user 'the truth' sometimes rather than trying to sell at all costs. This Dact will probably only be used when customer satisfaction and retention are more important in an application than market segment augmentation or profit making.

7. **System Request_repetition**: This Dact is used to ask the user to repeat their request or utterance because of bad speech recognition or internet server problems which resulted in the system not receiving any user input at all or only incomprehensible segments. This will also be useful when the system cannot obtain a semantic representation of the user input, either because of out-of-vocabulary words and phrases, or/and because of an out-of-domain user request. This Dact can be followed either by a Repetition or a Clarify act on the part of the user. In the latter case, the user may have chosen to reformulate their initial query differently, probably with the addition of information on new parameters.

8. **System Correct**: The system can tell the user through this Dact that their view of the world (in terms of features and their allowed values) is different from that of the knowledge base and discrepancies have occurred. For example, the user may think that Lisbon is in Greece and they want to take a flight to Athens to this effect. The system has to correct the user

Amazon/VB Assets
Exhibit 1004
Page 44

before trying to find something appropriate in the database, because the user may change their minds when they realize their mistake. E.g.

"Unfortunately, the cheapest printer available at the moment costs €80, so there is nothing at €25."

9. **User Correct:** This is probably the most important Dact, which intends to correct the instantiation for a parameter, either because the user changed their mind in the course of the interaction or because the system has misrecognized or misinterpreted a previous user input in the first place. This is especially relevant for the spoken language interface, where input recognition is much more difficult and uncertain. E.g.:

"Not Epson, Inkjet printer I said" (spoken interface)

Whichever the case, there is always going to be a system follow-up with a Check Dact, trying to have the new / modified parameter value explicitly confirmed by the user, so that the misunderstanding is cleared up early on.

10. **User Cancel:** This Dact will be used to change the topic or even the domain in the middle of an interaction. The user may want to switch from printers to scanners or from computers to travel offers in the same dialogue session. Cancel clears all the parameter values from the dialogue history and a new dialogue history is set up for the new topic or domain. This Dact will be triggered when a new domain pattern is identified in the user input. In the case of spoken input, the canceling operation has to be more direct, because it is inefficient to have all vocabularies active at all times (which leads to inaccuracies in speech recognition). E.g.

"Forget it. What about last minute offers"

Amazon/VB Assets
Exhibit 1004
Page 45

CLAIMS:

1. A method for conducting a commercial transaction and information exchange through an electronic interface between a system and user, the method comprising the steps of:

using natural language, submitting a query about products and services offered by providers thereof to the system;

identifying the language employed by the user;

maintaining a knowledge base about products and services offered by providers thereof, as well as a database model of the user's preferences;

interpreting the query on the basis of preceding dialogue exchanges between the system and user, and on the basis of information contained in the knowledge database;

requesting clarification about the query when the query is not understood or is incompatible with information contained in the knowledge base of the system;

updating the history of the dialogue exchange between the user and the system; and

if clarification regarding the query is not obtained to a desired confidence level after a specific number of attempts, forwarding the question to a human operator for resolution, otherwise

generating a response to the query on the basis of the information contained in the knowledge base, on the basis of the preceding dialogue exchange between the user and the system, and on the basis of the user's preferences .

2. The method of claim 1 further comprising:

upon the user's request, forwarding the query to the human operator to effect a human-human dialogue.

3. The method of claim 1 wherein the step of interpreting includes drawing inferences via machine-machine dialogue about the product or service the query is about.

4. The method of claim 1 further comprising

on the basis of the query made, identifying misbeliefs the user has about the product or service.

5. The method of claim 1 further comprising:

updating the knowledge base on the basis of the preceding dialogue exchanges between the user and system.

6. The method of claim 1 wherein the step of interpreting includes representing the user input as user dialogue acts which characterize the reason why the query was made or the information provided at the specific point in the dialogue exchange.

7. The method of claim 6 further comprising modifying the dialogue acts on the basis of misunderstandings of the query, user misbeliefs, modified queries, or unavailability

Amazon/VB Assets
Exhibit 1004
Page 46

of the products or services desired by the user.

8. The method of claim 1 wherein the step of generating a response includes generating a system dialogue act for eliciting more information about the product or service the query is about.

9. The method of claim 8 further comprising modifying the dialogue acts on the basis of misunderstandings of the query, user misbeliefs, modified queries, or unavailability of the products or services desired by the user.

10. The method of claim 1 wherein the step of requesting clarification includes repair strategies dialogue acts for resolving a misunderstanding or problem in the query .

11. The method of claim 10 further comprising modifying the dialogue acts on the basis of misunderstandings of the query, user misbeliefs, modified queries, or unavailability of the products or services desired by the user.

12. A human-machine communication system comprising:

a communication mediator for receiving user input of different modalities and converting the user input into text form, and for presenting responses to the user queries about product or services;

a natural language processing manager for identifying the language employed or preferred by the user, and interpreting the user input on the basis of preceding dialogue exchanges, as well as the knowledge base;

a knowledge base including an ontology of concepts and relationships therebetween; and

a dialogue manager for generating a response to the user input on the basis of information contained in the knowledge base, on the basis of the preceding dialogue exchanges and on the basis of user preferences, said dialog manager requesting clarification regarding the user input when not understood or incompatible with information contained in the knowledge base, and wherein the dialogue manager forwards the user input to a human operator if the user input is not clarified to a certain level of confidence after a specific number of attempts.

13. The system of claim 12 wherein the response to the user input is in a single modality or combinations thereof depending on explicit or inferred user preferences.

14. The system of claim 12 wherein said knowledge base includes specific mappings between items in the current application database and concepts and features in the ontology.

Amazon/VB Assets
Exhibit 1004
Page 47

15. The system of claim 12 wherein said knowledge base includes a collection of user profiles.

16. The system of claim 12 wherein said dialogue manager generates predictions on the subsequent user input, and validates the interpretation of the user input made by said natural language processing manager on the basis of previous dialogue exchanges and information in the knowledge base.

17. The system claim 12 further comprising:
upon user request, means for forwarding the query to the human operator to effect a human-human dialogue.

18. The system of claim 12 wherein said means natural language processing manager includes means for drawing inferences about the product or service the user input is about.

19. The system of claim 12 further comprising
on the basis of the query made, means for identifying misbeliefs the user has about the product or service.

20. The system of claim 12 further comprising:
means for updating the knowledge base on the basis of the preceding dialogue exchanges between the user and system.

21. The system of claim 12 wherein said natural language processing manager includes means for representing the user input as user dialogue acts which characterize the reason why the query was made or the information provided at the specific point in the dialogue exchange.

22. The system of claim 21 further comprising means for modifying the dialogue acts on the basis of misunderstandings of the user input, user misbeliefs, modified user requests, or unavailability of the products or services desired by the user.

23. The system of claim 12 wherein said dialogue manager includes means for generating a system dialogue act for eliciting more information about the product or service the user input is about.

24. The system of claim 23 further comprising means for modifying the dialogue acts on the basis of misunderstandings of the user input, user misbeliefs, modified user requests, or unavailability of the products or services desired by the user.

25. The system of claim 12 wherein said dialogue manager includes repair strategies dialogue acts for resolving a misunderstanding or problem in the user input.

Amazon/VB Assets
Exhibit 1004
Page 48

26. The system of claim 25 further comprising means for modifying the dialogue acts on the basis of misunderstandings of the user input, user misbeliefs, modified user requests, or unavailability of the products or services desired by the user.

Amazon/VB Assets
Exhibit 1004
Page 49

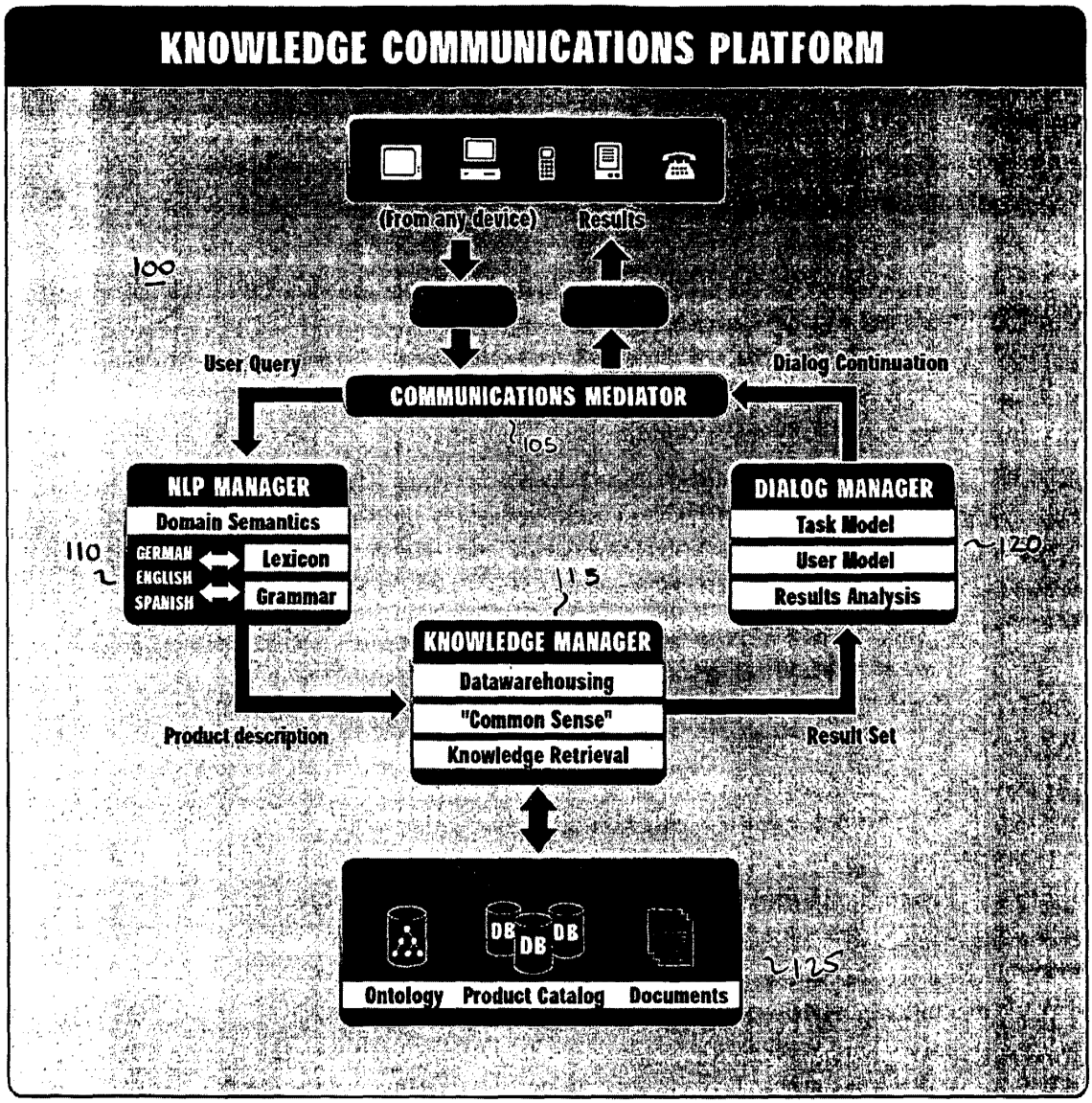

KNOWLEDGE COMMUNICATIONS PLATFORM
(From any device)
Results
100
User Query
Dialog Continuation
COMMUNICATIONS MEDIATOR
105
NLP MANAGER
Domain Semantics
GERMAN
ENGLISH
SPANISH
Lexicon
Grammar
110
DIALOG MANAGER
Task Model
User Model
Results Analysis
120
115
KNOWLEDGE MANAGER
Datawarehousing
"Common Sense"
Knowledge Retrieval
Product description
Result Set
DB
DB
DB
Ontology
Product Catalog
Documents
125

Fig. 1

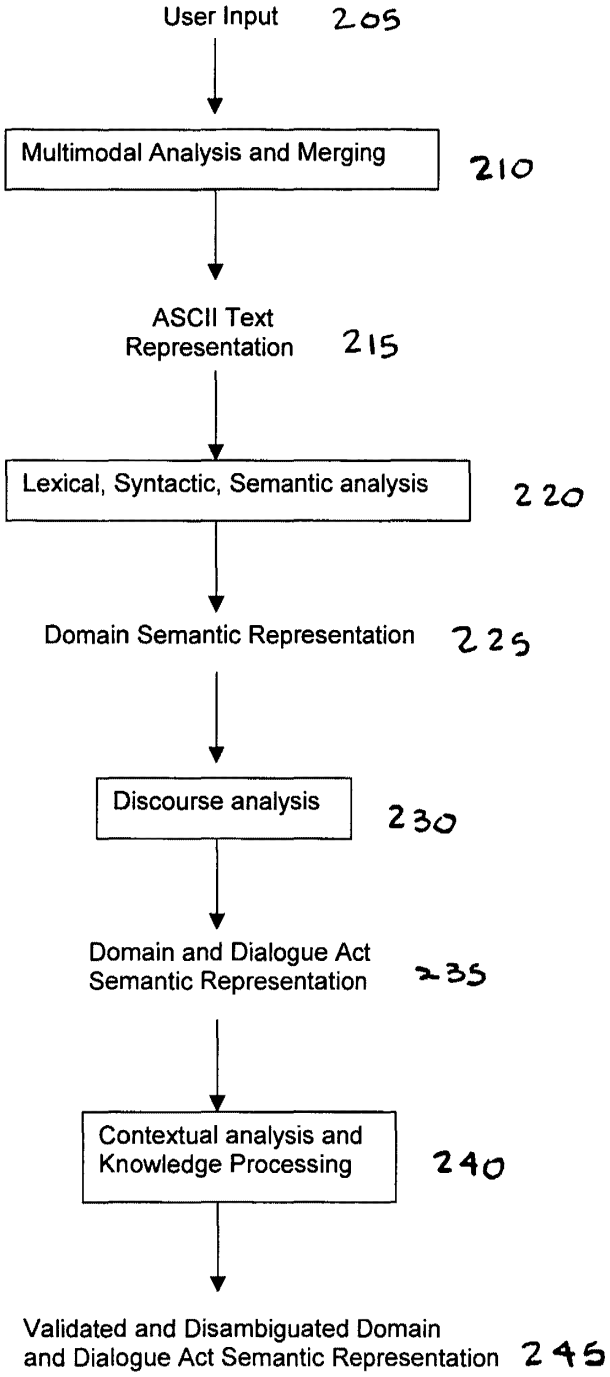

User Input
205
Multimodal Analysis and Merging
210
ASCII Text Representation
215
Lexical, Syntactic, Semantic analysis
220
Domain Semantic Representation
225
Discourse analysis
230
Domain and Dialogue Act Semantic Representation
235
Contextual analysis and Knowledge Processing
240
Validated and Disambiguated Domain and Dialogue Act Semantic Representation
245

Fig. 2

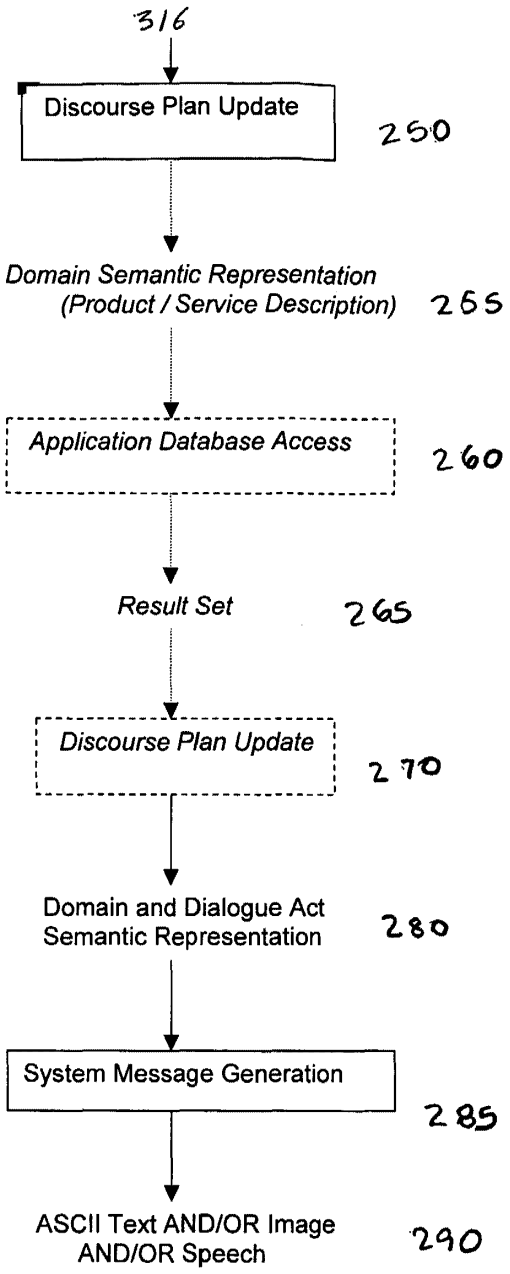

316
Discourse Plan Update
250
Domain Semantic Representation
(Product / Service Description)
255
Application Database Access
260
Result Set
265
Discourse Plan Update
270
Domain and Dialogue Act
Semantic Representation
280
System Message Generation
285
ASCII Text AND/OR Image
AND/OR Speech
290

Fig. 2 (cont)

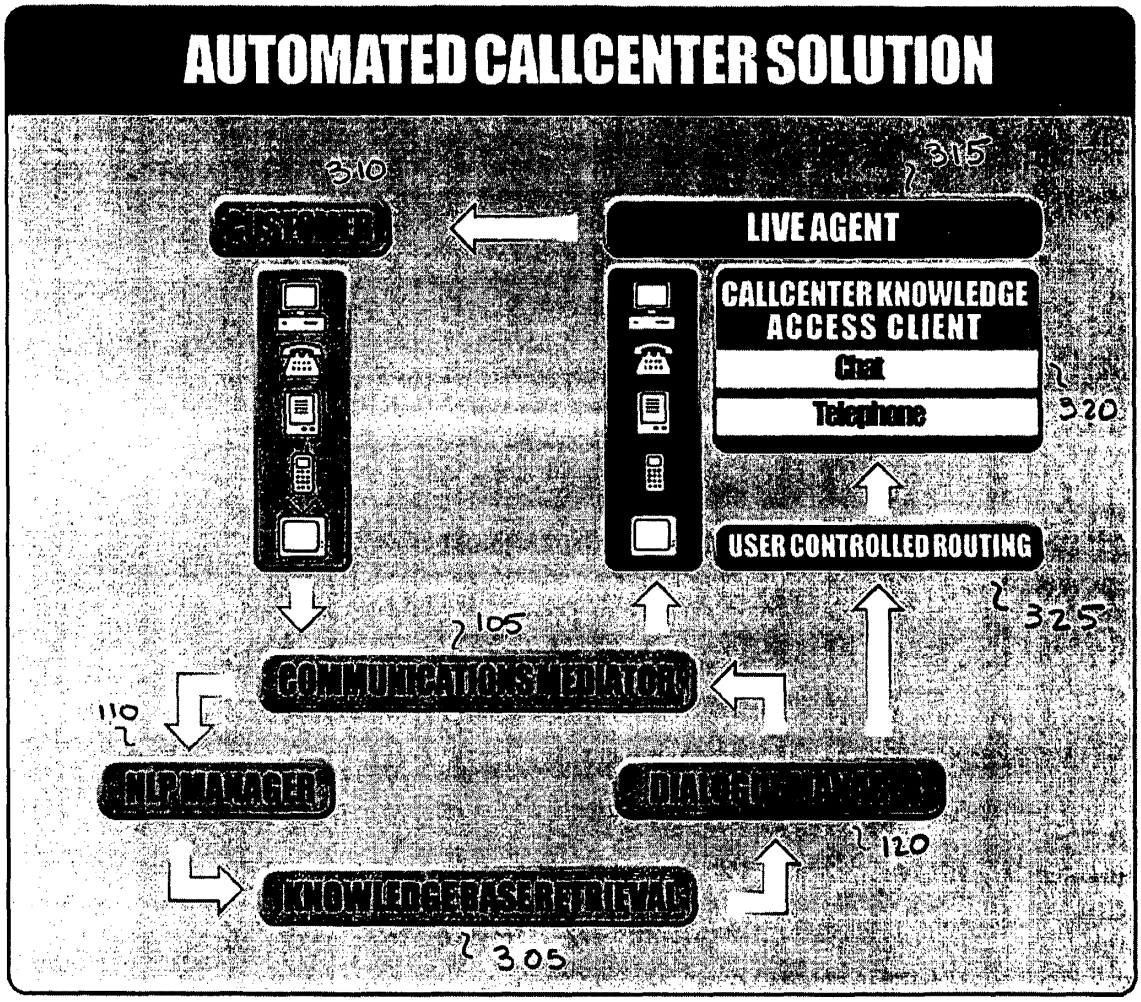

AUTOMATED CALLCENTER SOLUTION
310
315
LIVE AGENT
CALLCENTER KNOWLEDGE ACCESS CLIENT
Chat
Telephone
320
USER CONTROLLED ROUTING
325
105
110
NLP MANAGER
120
305

Fig. 3

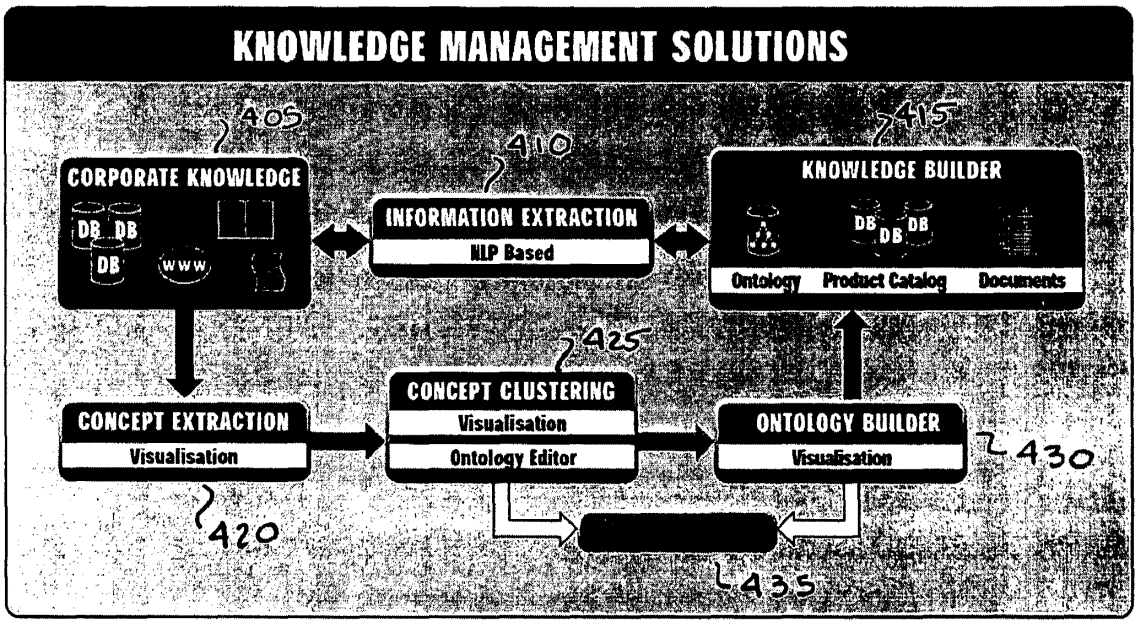

KNOWLEDGE MANAGEMENT SOLUTIONS
405
410
415
CORPORATE KNOWLEDGE
DB
DB
DB
www
INFORMATION EXTRACTION
NLP Based
KNOWLEDGE BUILDER
DB
DB
DB
Ontology
Product Catalog
Documents
425
CONCEPT EXTRACTION
Visualisation
CONCEPT CLUSTERING
Visualisation
Ontology Editor
ONTOLOGY BUILDER
Visualisation
430
420
435

Fig. 4

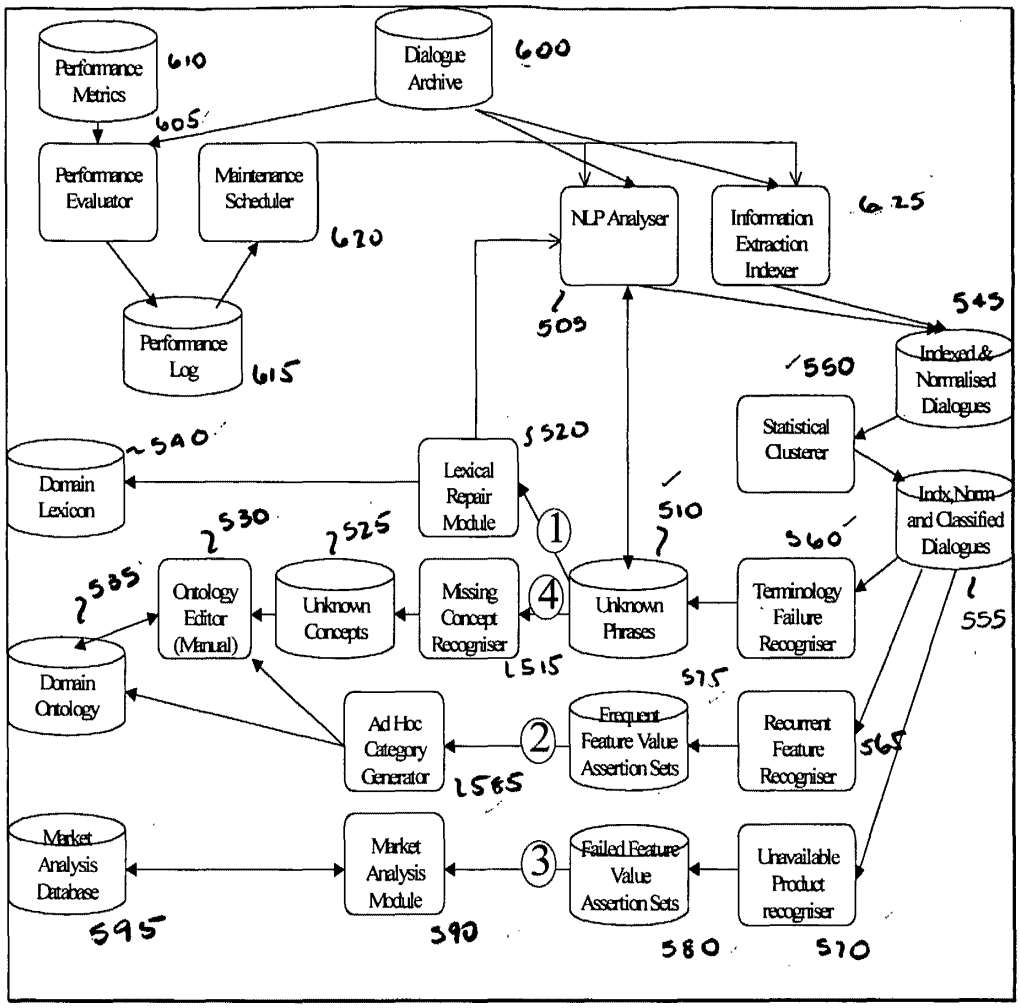

Performance Metrics
610
Dialogue Archive
600
605
Performance Evaluator
Maintenance Scheduler
620
NLP Analyser
Information Extraction Indexer
625
509
545
Performance Log
615
Indexed & Normalised Dialogues
550
Statistical Clusterer
540
Domain Lexicon
520
Lexical Repair Module
1
510
530
525
560
Indx,Norm and Classified Dialogues
585
Ontology Editor (Manual)
Unknown Concepts
Missing Concept Recogniser
4
Unknown Phrases
Terminology Failure Recogniser
555
515
Domain Ontology
575
Ad Hoc Category Generator
2
Frequent Feature Value Assertion Sets
Recurrent Feature Recogniser
565
585
Market Analysis Database
Market Analysis Module
3
Failed Feature Value Assertion Sets
Unavailable Product recogniser
595
590
580
570

Fig. 5

Box No. VIII (v) DECLARATION: NON-PREJUDICIAL DISCLOSURES OR EXCEPTIONS TO LACK OF NOVELTY

The declaration must conform to the standardized wording provided for in Section 215; see Notes to Boxes Nos. VIII, VIII (i) to (v) (in general) and the specific Notes to Box No. VIII (v). If this Box is not used, this sheet should not be included in the request.

Declaration as to non-prejudicial disclosures or exceptions to lack of novelty (Rules 4.17(v) and 51*bis*.1(a)(v)):

In relation to this International Application:

SematicEdge, GmbH declares that the subject matter claimed in this International Application was disclosed as follows:

(i) In US provisional application no. 60/269,995 filed February 20, 2001 with the United States Patent and Trademark Office; and

(ii) This declaration is made for the purposes of all designations.

☐ This declaration is continued on the following sheet, "Continuation of Box No. VIII (v)".