siwn.org.uk
International Transactions on Systems Science and Applications
Volume 4 Number 2 May 2008 pp. 122-129
ITSSA

Enhancing Natural Interaction with Circumstantial Knowledge

Dolores Cuadra, Jessica Rivero, David del Valle and Francisco Javier Calle

Computer Science Department, Carlos III University of Madrid - Avda. Universidad 30, 28911 Leganés (Madrid)
Email: { dcuadra; jrivero; dvalle; fcalle }@inf.uc3m.es

Abstract: This work focuses the circumstantial knowledge management for a specific need: the achievement of Natural Interaction (NI). In first place, a cognitive approach to NI is glanced as the framework for such knowledge management. This approach reflects some certain requirements for the whole interaction system, which are met by a multi-agent system implementation. Finally, a Situation Modeling is proposed for a first approach to the interaction circumstances management.
Keywords: natural interaction, situation models, spatio-temporal databases, multi-agent based interaction

1. Introduction

To imitate human interaction is one of most attractive challenges in nowadays HCI. The aim is to implement an interaction paradigm for which no previous technological training or interaction ability is required, since the user is going to behave as he/she would do while interacting with another human being. The so called Natural Interaction (NI) requires modeling any human knowledge and reasoning mechanism regarding interaction, enabling the system to decide how to interact in a human way. However, that knowledge is very diverse and complex. Thus, a knowledge distribution and proper coordination is required to attain a real scalable approach to NI. The diversity suggests a cognitive architecture composed of specialized models, which should be running autonomously all the way, influencing the interaction when they decide to. Hence, a multi-agent system approach seems to fit NI requirements to implement and coordinate those specialized knowledge models.

Present work will focus on the Situation Model, that is, the component managing every aspect and knowledge regarding the interaction circumstances. Its inclusion provides efficiency to other models, enables situation dependant tasks, and enhances interaction *naturality*.

Through next sections, the cognitive approach and its multi-agent implementation are briefly glanced. The rest of the exposition will be dedicated to explain the Situation Model.

2. Cognitive Approach for Natural Interaction

During the whole process of the human interaction, each of the participants uses several skills requiring the handling of complex knowledge of diverse nature. Tackling the problem as a whole poses the problem of properly modeling the influence of each knowledge type anytime during interaction. Furthermore, reasoning based on some of these knowledge types might have influence on processing some other knowledge. As a result, a common trend for the development of systems capable of interacting in a natural way consists in proposals for dividing up the concerned knowledge into several specialized components, which later collaboration and coordination could end up in a seemly natural result (see for example [3]). Such Cognitive Architectures describe the Knowledge Components (or better, Models) covering any knowledge need for NI, along with their interrelations and information flows. Present work is based upon one of these Cognitive Architectures [4] (Fig. 1), which mainly counts on the following knowledge models: Interface Components (IICC), Dialogue Model (DM), Situation Model (SM), User Model, Emotional Model, Ontology and the application (or external agents).

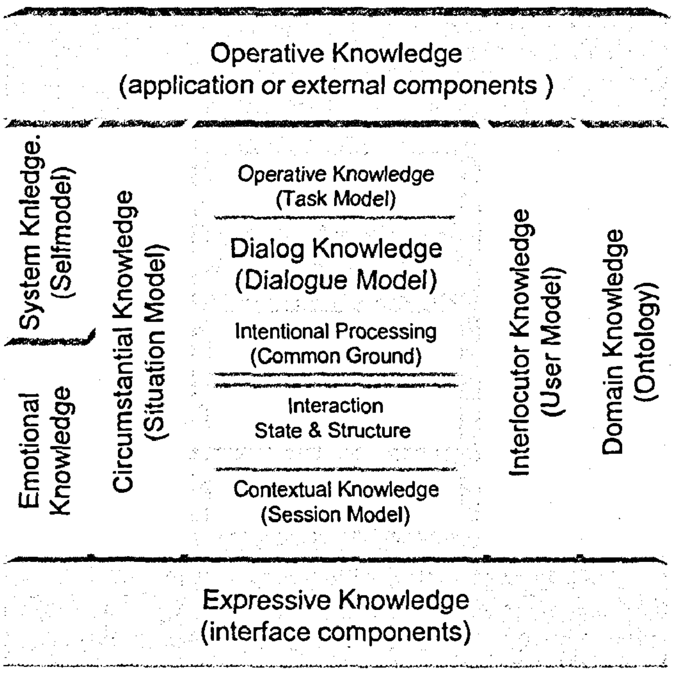

Fig. 1. Cognitive Architecture of the Natural Interaction System.

Each of these components deals with a specific type of knowledge and its related accomplishments. In first place, the IICC are in charge of the expressive abilities in two senses: they have to acquire user expressions, interpret their semantic content and represent it by means of Communicative Acts (CCAA) [11]; and they also have to generate system's

ISSN 1751-1461 print / ISSN 1751-147X cd-rom
sai: itssa.2008.05.031

expressions from the CCAA produced by the DM and synthesize them in system's utterances. For such aims, the IICC should be helped by the Presentation Model, which organizes their intervention in order to have complete and coherent results. On second place, the DM receives the semantic direct interpretation (literal) of user's interventions and updates the dialogue state observing several aspects: intentional, structural, contextual, etc. Apart from this, when the system has got the turn, the DM will decide how to behave and then generate an intervention (again expressed in form of CCAA), updating the dialogue state again as its intervention is uttered. Their connection is found in that dialogue state, which could be from a simple state in an automaton (*Dialogue Games*) up to the more complex concept of 'common ground' (*Join Action theories*) for handling system's beliefs on shared knowledge (between user and system) through the whole interaction.

Through this process, the DM could require the partaking of one or more external applications which will be invoked for a task, then having its results properly interpreted. Knowledge regarding tasks (which, when, invocation, how to interpret outputs, etc) is to be gathered into a Task Model. Moreover, the DM usually needs to apply and feed information for updating the state regarding other types of interaction-related human knowledge. For handling these other types of knowledge, the architecture should include the following components: an Ontology (in charge of fixing all the concepts referred by participants), the User Model (performing the interlocutor characterization), the Emotional Model (managing the emotions that affects the conversation), the Self Model (containing system's own goals and general beliefs), and , last but not least, the Situation Model for dealing with the circumstantial aspects that envelopes the interaction. Dispensing with a particular knowledge model will produce a more mechanical interactive behavior with regard on that kind of knowledge (more patent with shorter corpus).

Among all these components, the DM has an especial relevance because an important part of the naturalness of the interaction progress relies on him. Some interaction domains, particularly those involving the processing of circumstantial aspects, require an interaction especially flexible and versatile: it should be possible to keep up some sub-dialogues opened and progress them in a consistent and committed way, to allow either the user and the system (in a clear pro-active manner) to introduce new goals in the conversation at any time, and to allow the reinforcement of discourse lines when their commitment have decrease. Although there are some different approaches to manage the dialogue [6], either conversational or discursive (such as dialogue grammars and plain based models), the intentional dialogue models based on the Joint Action Theories [5] seem to be the most adequate for achieving all these needs. The system presented in this paper implements its DM based on an particular intentional dialogue model, the Threads Model [4], which has been already applied on several international projects (EU project IST 1999-11305 and IST 2001-32440) and national projects (FIT 350301), and it is currently being extended along another two projects, proving to be quite versatile and natural in all these aspects.

On the other hand, human behavior is clearly aware of circumstantial situation and evolution. Thus, managing its different aspects is key for attaining real NI, apart from other benefits for the whole system (as will be described later in section IV). Besides, SM and DM services integration brings appealing challenges, such as the situation event execution (for the SM), and some advanced turn-taking management for uttering urgent discourses (producing scenarios where the system might take the floor while user is making an utterance, or even enabling the system to interrupt itself). In sum, such joint research involves a quality step forward NI.

3. Multi-Agent Implementation

Implementations of Natural Interaction (NI) systems must be borne on architectures where each component is capable of performing autonomous processing, but working with the rest in a coordinated way. Actually, the NI is not a pure sequential process where each of the problems involved is tackled in a concrete point of the execution ([7]). During the NI, several processes advance concurrently and feedback each other at any time to obtain together the final result. Several components are sometimes qualified to resolve the same problem, each of them following different strategies. Therefore, situations where different components work simultaneously to solve the same problem in a competitive manner may also occur. Produced responses could be different (due to the influence of diverse knowledge and the subjective nature of the problems in NI), and it will be required to select the most appropriate of them for each particular situation, taking advantage of the multiplicity of the solutions. In the same way, each of the components involved in the interaction could also be capable of solving a problem by applying different algorithms.

Besides, the production of responses for some of the complex problems involved in the NI consumes a computational time much too long to keep the petitioner waiting for it. Then, the petitioner could prefer to obtain a first quick response, but incomplete or vague, to be able of progress in the resolution of its own problem while the component that served that response works on its refinement, until the deadline for the service arrives (or it is cancelled by the petitioner). Each time a new solution is provided, the petitioner should decide whether it is useful to discard previous results (and its processing) and apply the newly obtained or not. This could even end up in a self interruption of the system: either by interrupting the currently uttered intervention (discarding it), or just bringing forward newly generated discourse. Invoked knowledge components should anticipate if they are able of providing a response in time or not, and then they should establish if they can refine it in time or not.

Consequently, the petitioner has to characterize its service requests by a set of parameters: a *criticism* value, in order to make possible to decide whenever the service is worth the cost of a particular strategy or not; a certain *quality threshold*, above which any solution will be considered satisfactory; an *expiration timestamp*, that defines the maximum time the petitioner will wait for responses; and the *preferred timestamp* for the first solution.

This set of requirements draw the NI process away from any sequential execution approach and leads to multi-agent one, with autonomous agents collaborating for a coordinated, coherent and complete result.

3.1. The Ecosystem Platform Description

The presented approach of Natural Interaction System (Interactor) is supported on a custom-made multi-agent platform named *Ecosystem*. In this platform agents are created, destroyed and, during their life, can request for the resolution of services to other agents and attend the requests of services of other agents, exchanging messages with this purpose.

Each agent is subscribed to an agency that defines the maximal and minimal number of their agents that can inhabit in the environment in each moment. Ecosystem contains specific agents responsible of creating and destroying agents, with the aim to keep the population of agents of each agency controlled (between the defined bounds), creating more of its agents when the load of the existing ones is too high or destroying some of them when it is too low. The agency also defines the services their agents provide to the rest. Every of these services are characterized by the operation it offers and the field where this operation can be solved. All the possible fields where operation can be requested are organized in a tree (where more generic agents are parents of the most specific ones). Thus, when an agency defines services for an operation in more generic fields than the field where a service is requested, any of its agents can attend this requests (even when she does not support that operation in the specific field).

The next important aspect of the platform is that an agent does not receive directly the request of services from other agents, the platform contains a specific agent (mediator or Broker Agent), in charge of receiving demands of services from the agents of the environment and deciding which agent or agents should attend those demands. Broker Agents have their own strategy to select the agent or agents that will receive each request. Therefore, the Broker can consider parameters like the work-load of each agent, or how close is the field of application of their functions to the requested one. Eventually, several agents could attend simultaneously the same service, offering different results or solutions.

In order to apply this multi-agent approach on the implementation of a Natural Interaction system, a set of specific agents have been developed building the system named *Interactor*. Services in Interactor extend the services of Ecosystem distinguishing inputs and outputs, and adding the session identifier (for storing the state of interaction within each model) and all the needed parameters to represent the conditions of the service: the expiry-date, need-before, criticism and quality. These agents provide the main services described previously in the models of knowledge involved in the Cognitive Architecture.

The characteristics of each knowledge model determine if it should be implemented through several agents (when it is appropriate to arrange several collaborating agents to assure the knowledge management) or not. For instance, the Threads Model [4] current implementation is composed of four autonomous, independent, collaborative and coordinated agents: the Interpretation Agent (including the User Thread), the Generation Agent (hosting the System Thread), the Common Ground Agent (comprising the Thread Joint) and the Presentation Manager Agent. The first provides services of interpretation of user expressions; the second mainly provides services of generation of user expressions and system initiatives introduction; the third represents the interaction state and the common ground management; and the last connects the interaction process with the acquisition of user's interventions and the expression of system's ones.

All this features makes possible to accomplish such kind of interaction system: parallel execution of the different knowledge components; iterative resolution of the services, where requested; competition of several strategies to obtain several solutions for the same problem; selection of the best strategy to solve a problem in function of the provided parameters; improvement of the system pro-activity due to the diversity of source (each knowledge component could request to initiate a new sub-dialogue anytime); and the foundation for a dialogue management able to deal with overlapped turns and multi-user interaction.

Finally, all these features of the multi-agent platform make possible to face the distribution of the system over a LAN (enabling the independent development of each component and attaining more complex and scalable systems). The platform developed allows registering as many agencies as needed, and also to include, delete, or suspend the services they offer during the interaction (useful for developing and evaluating purposes, to compare the pros and cons of different strategies).

3.2. Performance Evaluation of the Multi-Agent Platform

The Ecosystem multi-agent platform has been subjected to a performance evaluation in a similar test environment as the one supporting the Interactor Natural Interaction System . This environment consists of a server computer (of 2x2.21 GHz of CPU frequency and 4 GB of RAM memory) and four client computers (of 2GHz of CPU frequency and 512 MB of RAM memory). These computers are interconnected through a 100 MB Ethernet Local Area Network. The server contained an Oracle ™ 10g Database Manager System that stores the database instance of the blackboard (where agents are inscribed for exchanging their messages) and each one of the clients contained at least one test agent. Two kinds of test agents were used in the experiment: Server Agent, disposed to attend petitions of one test service named Echo Service; and Petitioner Agent, requesting for Echo Services to the system.In the Echo Service, the server agent replies to the petitioner agent with the same message she receives trough the service.

During the experiment, the Server Agent was set on one of the four client computers, while the others have instances of the Petitioner Agent. All Petitioner Agents request for Echo Services simultaneously to the system, and all these services were served by the only Server Agent of the system. Each Petitioner Agent requested for 1500 services to the system and sent/received up to 100 messages through each one (50 messages from client to server and 50 from server to client). This experiment was repeated 20 times and the averaged results show that Ecosystem platform is able to: (a) Attend at least 56 requests of services by second for each agent, and (b) Send more than 68 messages by second between a petitioner agent and her server agent through a service.

Yet these results are good enough for supporting the Interactor System requirements, they could be improved by tuning the Database (size and use of the cache, auxiliary structures, and base organizations) , or by using some other features of the Database Manager System such as the distribution of database instances. These improvements will make possible to speed up the access to the memory and, consequently, the efficiency of the platform, which would be of interest to host a greater number of concurrent users interacting with the Interactor System.

4. Situation Model Description and Its Influence in the Natural Interaction System

The Situation Model is the component of the Interaction Model which deals with the circumstantial aspects. According to Gee [8] taxonomy, the circumstantial aspects are divided into five categories: semiotic (the language or signs used), politic (the roles assigned to each interlocutor), operative (task underlying interaction), material (spatio-temporal situations) and socio-cultural (the social environment).

Four of the five aspects described before (semiotic, politic, socio-cultural, operative) could seem overlapped by other components of the Interaction Model. However, their functionally is different and the situation knowledge should be processed jointly for obtaining useful results. Hence, the operational range of the SM is well defined and dissociated: to fix and handle the circumstances of any produced interaction.

The inclusion of the SM provides more knowledge influence in the produced interactions, which will lead to decisions that will improve the naturality of the system. This is achieved by circumstances management around the interlocutors and the interaction and integration with the others models. For example, spatio-temporal conditions could force changes in the execution plan on the Task Model (TM), as shown next.

Example 1: The interaction system user wants to buy a book at six o'clock. Then, the TM is invoked and the actions to be achieved are the following:

(i) Check money
(ii) Buy the book.

Eventually, the lack of cash could lead to another task: to get the money. For achieving it, spatial conditions have to be fulfilled (to be in the cash point), and since the TM is not able to perform guiding task, it will have to ask the Situation Agents, which are the proper to solve such problem. Once achieved, the TM will be enabled to perform the second part, for which it is necessary again to fulfil certain conditions. Now, it is not only spatial, but also temporal conditions (since the bookshop is not open but on certain timetable). The solicitor (in this case, the TM) invokes the SM with the situation requirements, and this component will be in charge of performing a plan to fulfil them. When achieved, the solicitor will be invoked back to perform the rest of the task.

However, the need for circumstances knowledge and reasoning is often left apart or assumed by other components. Then we could find systems with several set of signs to interact with them, or several 'user profiles' (even 'system profiles') with a different range of feasible circumstances arranged for each pair of roles; systems which deal with different tasks and consider the situation as the point in which the task is [7]; systems with control of time or spatial-dependant tasks using a positioning component, such as the 'Navigator' in Cyberguide system [1]; or systems which have interaction influenced by multimodal input [7]. However, interaction systems rarely show different interactive behaviour depending on reasoning mechanisms based on situation knowledge and past, current, and predictions on future positions of interlocutor. Finally, hardly any interactive system directly tackles the problem of diversity in the socio-cultural aspect of interaction.

Circumstance modelling has been long researched and developed for context-aware and pervasive systems [9]. However, there exists some focus differences between their approach and SM's. Context-aware systems seek to adapt their operations to the current context, based on observations of current circumstances, processing and consequent behaviour, either learned through the using or specified by the user (profiled). For contextual (circumstantial) information, they consider any element within the environment which location and state could influence system behaviour. In such systems, most of the interaction with the system is implicit, seeking a pervasive system able of understand users (and any element within the environment) behaviour and behave itself consequently. Situation Models, on the other hand, just focus circumstance aspects of the interaction; that is, the contextual parameters representing the interaction situation and influencing its development.

5. Situation Model Proposal

The Situation Model (SM) proposed observes some important features in comparison with other approaches, both from the Natural Interaction and other confluent research areas..In the first place, humans identify situations where interactions occur for attaining more precise reasoning on interaction events. The SM provides certain information (the situation characterization) to the other components, and this enables them to filter their own knowledge (following a 'relevant for the situation' criterion) for obtaining a 'circumstance relevant knowledge'. This has efficiency benefits (reduces resources requirements) as well improves efficacy (for example, reducing the ambiguity of knowledge).

For example, in the Ontology all the terms, concepts, and relations between terms and concepts are stored. Applying on this component the circumstantial data, only a subset of its knowledge is taken into account each moment, reducing the search space and avoiding some problems, as in the example.

Example 2: A user says: "I need a sheet." Depending on the situation, the Ontology filtering should be:

- If the interaction happens in a classroom, the concept "a piece of paper" is acquired from the term 'sheet' (since the concept 'sheet' should be initially filtered). System's response could be the following: "Sorry, but I do not have any notebook".
- If the interaction happens in a bedding department of a store, the term 'sheet' is matched to the concept 'bed sheet', and final response could be something like this: "OK. What colour do you want?"

Another feature is the SM enables the programming and execution of situation triggering. That is, any component should be able of requesting the SM to perform a certain action (which could be a service request to another component) when a spatial and temporal condition is fulfilled. For example, the Dialogue Model could be invoked to introduce sub-dialogues on a situation event, thus obtaining a more human conversation. For tackling these tasks, the system needs triggering mechanisms, similar as those used for database systems but extended to spatio-temporal dimension (and able of being extended to the other aspects of circumstances).

Last, the SM performs tasks on situation information: for building explanations, provides past positions, predictions about future situations, or strategies for getting from a starting situation (current, for example) to a desired another one (goal). For this, the spatio-temporal information should be stored in a database system. This feature characterizes the SM proposal because databases are not used in general to store the situation knowledge.

5.1. Situation Model Agent Design

In this section, the SM design will be explained focusing only on one of the five aspects proposed by Gee [8]. The chosen aspect is the material because is the most meaningful. The rest of aspects will be left for further work.

To implement the material aspect of circumstances, the spatio-temporal databases seem to be the appropriate technology because it provides the consistence of the data stored and the availability of the data is guaranteed.

The situation material knowledge is designed following the object-relational databases methodology based on [2]. This approach proposes an object-relational model extension, called *STOR*, for representing and supporting spatio-temporal data with different granularities. Abstract data types are defined to support it. The *STOR* model provides a practical vision to implement any domain in a DBMS which defines spatio-temporal data types as extensions of OpenGis specifications to SQL3 (standards of spatial and object-relational data, respectively). The material knowledge associated to SM is designed through this model providing it consistency, portability and easy integration. On the other hand, the spatio-temporal multigranularity handling allows the objects representation with different space and time measures.

For presented work, UML classes diagram is used as a knowledge conceptualization, as its transformation to *STOR* scheme is direct. The classes diagram in the Fig. 2 represents the obtained knowledge model design. The material aspect design is modelled independently of the domain. All the classes represent spatio-temporal objects and could be classified into two groups depending on the kind of spatio-temporal variation:

(i) Classes with topology evolution, that is to say, the temporal evolution represents changes in the object geometry.

(ii) Classes with position evolution, that is to say, the movement of the object (change of position).

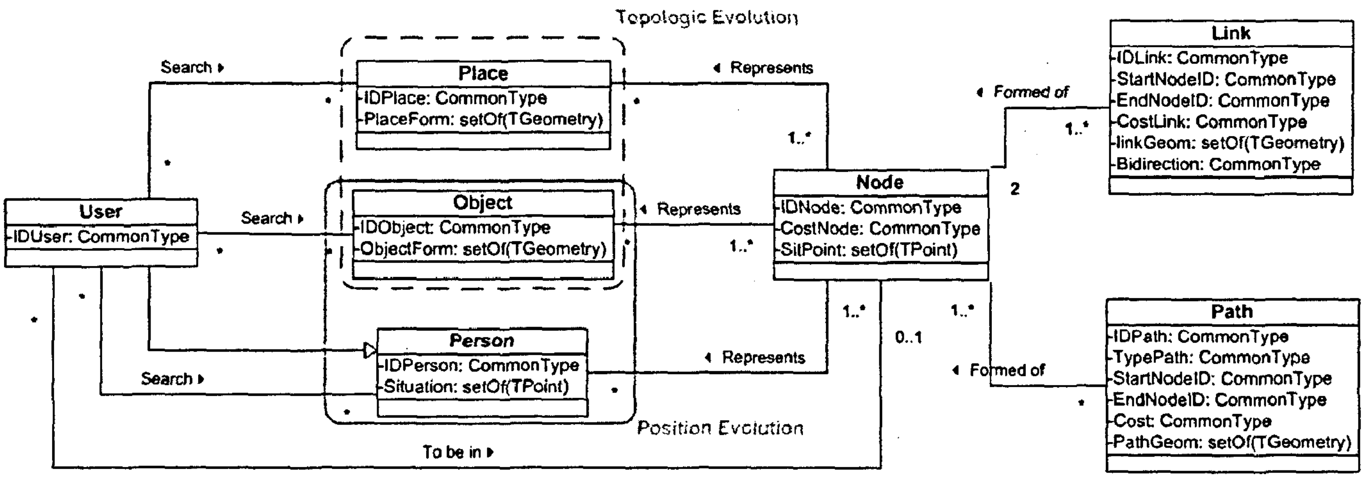

Fig. 2. The Material Knowledge representation.

The spatial and temporal reference system is defined and the measure unit as well as conversion functions to transform to coarser granularity for each spatio-temporal class. In both cases, the spatial granularity (metres, kilometres, millimetres...) and the temporal granularity (moths, seconds, hours...) of each class could be different. Therefore, one class granularity could change with independence of the others.

To generalize, the user could search persons, places and objects (being all these components inside of a spatio-temporal reference system). These components will represent the points of interest in a specific context. To develop the functions of searching a path or finding something, a net graph is used. Then the classes diagram could be divided into classes which belong to the net (on the right of the Fig. 2) and the ones that not (on the left of the Fig. 2):

1) The net part includes the classes 'Node' (representing points in the net), 'Link' (connections between nodes) and 'Path' (the way to reach a node from another using links).

*2)*The elements are represented in the net by a node. Since one of the functions of the net graph is such representation, the topology evolution of 'Object' and 'Place' involves changes in the topology of the net (nodes that disappear or appear to represent the existing elements). Regarding the classes 'Person' and 'Object' (the part of position evolution) they do not have influence over the topology of the net graph (they move over the existing net). To take into account the movement, their position and timestamp are stored. Finally, the 'User' class is a specialization of 'Person', and represents the current user of the system.

This design is implemented over a DBMS with spatio-temporal capabilities, and some triggers are included to attain a more autonomous performance.

5.2. Case Study

To understand the relevance of the SM in the interaction, three examples are presented in this section. The first two examples represent the idea that while the user is talking with the system, if an interesting situation (either programmed in the system or contracted by the user) happens, or there is something which needs the attention of the user, the system can produce an event and interrupts him/her to notify about it. After the interruption, the user takes a decision about the event, and after that the interaction will continue where it was interrupted.

In the third one, the SM ability of knowledge filtering is presented. This example shows how the different components of the cognitive architecture have their knowledge filtered, hence handle only relevant knowledge for attaining coherent interaction with more natural results.

Next there are three specific examples of these cases of use. All of them are in the same context: the environment is a hospital, and the conversation is kept between a worker in the hospital and the system.

(i) A doctor asks for a path to go to the room number 234. On the way to the goal, he talks with the system to obtain the patient history. The system interrupts him to inform about different kind of events during the walking.

...

Doctor – *Could you indicate me the path to go to the room number 234, please?*

System – *Of course. Turn right in the next corridor.*

Doctor – *Ok. Could you give me the patient name?*

System – *The name is John.*

Doctor – *Has he ever been in this hospital before?*

System – *Take care, you have forgotten to turn right in the corridor. Please, turn round and take the first corridor on the left.*

Doctor – *Ok, thanks.*

System – *Regarding the patient, this is the first time that he comes here.*

...

System – *Excuse me, we have arrived. Your patient room is the one on the right.*

...

As shown, while the interaction is going on, the SM interrupts the user to communicate contracted spatial events. Then, a sub-dialogue about the event is developed, and after that the interaction continues where it was interrupted.

The Fig. 3 shows the steps to request the guide service of the SM. In this case the DM calls the Task Model to resolve a request of the user. This component detects that the needed task should be solve by the SM, and builds the proper service request. The SM resolves it and gives the solution to the TM which interprets it and leads the result into the interaction.

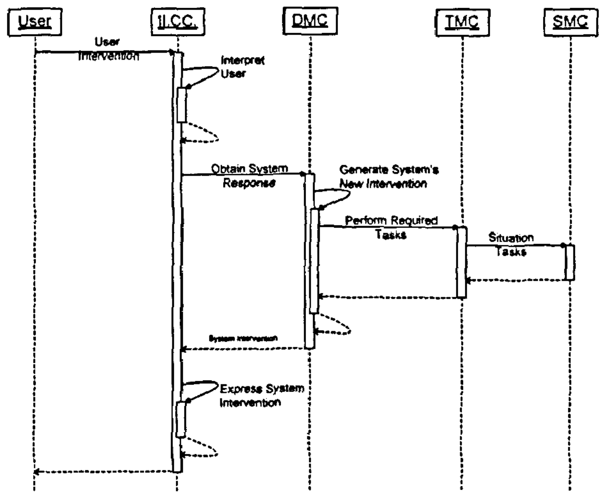

Fig. 3. Execution of a Situation task

This response of the SM only gives the first step to be followed. Next steps are notified to the user when he/she is in a given spatial position. The form in which this notification is produced is by an event to the DM, as shown in Fig. 4.

(ii) In this case, the doctor was interacting with the system (to obtain some information) when the SM detected a programmed spatial event and performed its consequence. In this case, the consequence involved invoking a task (from the task model, within de DMC), which execution result interrupted the dialogue to inform about it. This sequence is depicted in Fig. 4, and exemplified below.

...

Doctor – *Please, give me statistics about patients of cancer.*
System – *Of course. The number of patients of cancer is... Excuse me, you are passing by the reception of patients and I am checking that you have a new one.*
Doctor – *Oh! Thanks!*
System – *Regarding the information you asked for, Would you like us to continue with it?*

...

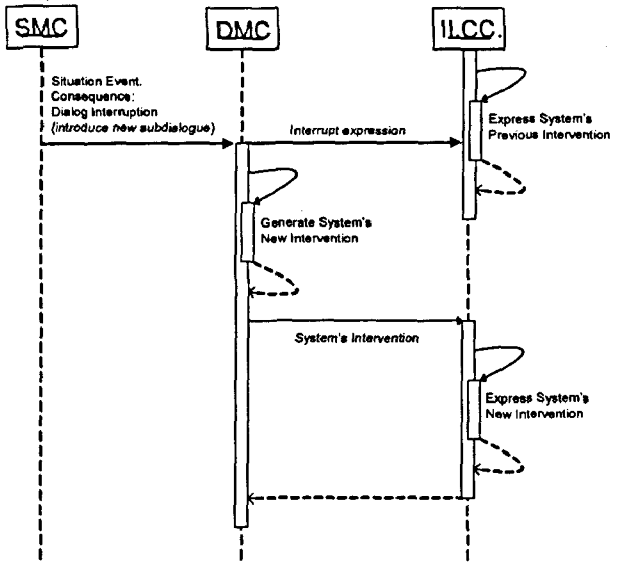

Fig. 4. Self-interruption of the NI system forced by a situation event.

(iii) In the last example, the knowledge filtering ability of other components of the fore presented Cognitive Architecture is shown. In this case, a cleaning lady of the hospital is talking with the system while she is working. She uses the word "sheet" in two different situations. In the first one she is in the laundry taken the clean bed sheets to change the dirty ones of the beds of the patients. In the second case, she is in the point of control in which she has to write her name, the number of the room that she was cleaned and the hour in which she did this.

...

User – Where are the sheets?
System- For examining couches or for beds?
User- For beds.
System- There are clean sheets in the second shelf.

....

User- I need a sheet please.
System- There is a notebook upon the table.
User- Thank you.
System- Do not forget write the hour in which you finish of cleaning the room.

...

To obtain this final procedure, the Ontology component (in this case) should ask the SM to inform it whenever significant changes on the situation occur. This will be done through situations events, as shown in Fig. 5.

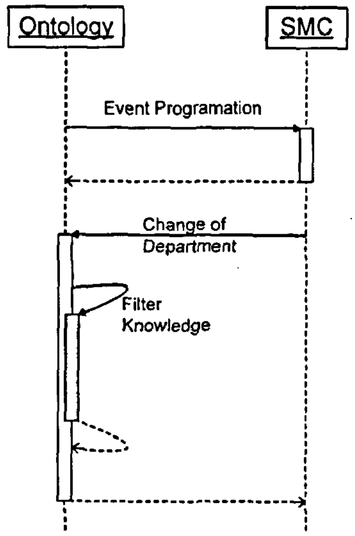

Fig. 5. Filtering of the knowledge of the Ontology.

6. Conclusions

For achieving Natural Interaction paradigm it is necessary to model any sort of knowledge affecting human interaction. That knowledge is to be managed independently and autonomously for approaching a more natural interaction, hence multi-agent implementations including specialized models for fulfilling each knowledge requirement.

Particularly, the need for circumstance knowledge leads to Situation Models.

These models mission is to define situation, but just regarding the interaction and the interactive agents (human and system). They are not supposed to handle contextual or environmental state and information, as they are left for other models in a global ubiquitous system. Spatio-temporal databases technology support the development of Situation Models as they can be used for covering the material aspect of circumstances, which is one of the most significant, and could be extended to cover the rest of them.

Acknowledgement

The presented work has been developed within the MAVIR project (S-505/TIC/0267) endorsed by the Regional Government of Madrid, and is being extended through the SOPAT project (CIT-410000-2007-12), supported by the Spanish Ministry of Science and Education.

References

[1] Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper,R., Pinkerton, M. Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, 3:421-433, 1997

[2] Bertino E., Cuadra D., Martínez P. An Object-Relational Approach to the Representation of Multi-Granular

Spatio-Temporal Data. Proc. of 17th International Conference, CAiSE 2005, Porto, Portugal, 2005.

[3] Blaylock, N., Allen, J., Ferguson, G. Managing Communicative Intentions with Collaborative Problem Solving. Jan van Kuppevelt and Ronnie W. Smith (eds.), Current and New Directions in Discourse and Dialogue, pp. 63-84.

[4] Calle J., García-Serrano A., Martínez P.. Intentional Processing as a Key for Rational Behaviour through Natural Interaction. Interacting With Computers, Vol. 18/6, pp. 1419-1446.

[5] Clark H. H. Using Language. Cambridge Univ. Press, 1996.

[6] Cohen P. R. Dialogue Modeling. Survey of the state of the art in Human Language Technology. 1997. chap. 6, pp.234--240.

[7] Cohen P. R., Johnston M., McGee D. R., Oviatt S.L., Pittman J., Smith I., Chen L., Clow J.. QuickSet: Multimodal interaction for distributed applications, in the Proceedings of the Fifth International Multimedia Conference (Multimedia'97), pp. 31-40. 1997.

[8] Gee, J.P. Introduction to Discourse Analysis. Routledge, 1999.

[9] Meyer, S., Rakotonirainy, A. A survey of research on context-aware homes. In Procs. of the Australasian information security workshop conf. on ACSW frontiers, 2003.

[10] Paternò, F. Model-Based Design and Evaluation of Interactive Applications. Springer, 2000

[11] Searle J. R. Speech Acts. Cambridge University Press, 1969.

Author Bios

Dolores Cuadra received the M. Sc. in Mathematics from Universidad Complutense of Madrid in 1995. In 1997, she joined the Advanced Databases Group, at the Computer Science Department of Carlos III University of Madrid, where she currently works as lecturer. In 2003, she obtained the Ph.D. degree in Computer Science from Carlos III University of Madrid. Her research interests include advanced database technologies, spatio-temporal databases and their applications to Situation management. She has been working in Computer Science Department at Purdue University of West Lafayette (Indiana) for nearly a year, where she has applied her research in Spatio-Temporal database.

Jessica Rivero finished her degree in Telecommunication Engineering from the Carlos III University of Madrid in 2006, and obtained an official master in Computer Science in 2007. Since 2006, she joint at the Advanced Databases Group in the Computer Science Department of Carlos III University of Madrid, where she is currently teaching Files Organization and Database Design. Her research interests include Spatio-Temporal Databases and the Situation Model inside the Natural Interaction.

David del Valle got the degree in Telecommunication Engineering from Carlos III University of Madrid in 2004 and obtained an official master in Computer Science in 2007. He is working in the Advanced Databases Group in the Computer Science Department of this university since 2005. His research interests include Dialogue Management and Natural Interaction Systems. He is working on several projects about these topics.

Francisco Javier Calle got his degree in Computer Science from the Technical University of Madrid in 1997. In 2000, he joined the Advanced Databases Group in the Computer Science Department at the Carlos III University of Madrid, where he is currently working as lecturer. In 2005, he obtained the Ph.D. degree in Computer Science from Technical University of Madrid, with a thesis in the Natural Interaction research area. Apart from this, his research topics are focused on dialogue modeling, intentional processing and joint action models, and cognitive components of the interaction. He has also been working in several European and National research projects regarding Human-Computer Interaction.