

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Towards Dialogical Human Computer Explanation: A Case Study in Qualitative Simulation

Mohamed El Habib LARABA-Zaidi Sahnoun

Department of Computing Science, LIRE Laboratory, Mentouri University of Constantine, Algeria

Abstract: Our concern in this study is viewing explanation as a dialogical interaction between many dialogical agents and the user. Those autonomous software components are designed to cooperatively collaborate with the end user, to provide him with the best explanation. Many tasks are thus achieved. Resulting system should be implemented through a qualitative simulation algorithm.

Key words: Explanation, dialogical agent, qualitative simulation

INTRODUCTION

An agent should have the capability of explanation. That is, when applied in qualitative simulation, the capability to provide the user with some knowledge about the execution of a qualitative simulation algorithm.

Artificial intelligence has been used in simulation since the end of the seventies. Qualitative reasoning (De Kleer, 1977) and particularly qualitative simulation (Kuipers, 1984, 1986) were the principle demonstrations. Later, explanation, another field of artificial intelligence, was used in simulation (Forbus and Talkenhainer, 1990; Gauthier and Gruber, 1993; Gruber and Gauthier, 1993). We found that interesting and proposed to integrate an explanatory module to an environment of simulation (Belatar and Laraba, 1997) (Laraba, 1999). We then proposed to extend and adapt this module to an environment of qualitative simulation (Laraba and Sahnoun, 2002). In Laraba and Sahnoun (2004a), we focused on modeling it as a multi-agent system, appropriate to carry out its distributed tasks: explanatory agents are autonomous. They carry out many explanatory goals and interact for working together to complete the explanatory task. The Companion vision (Forbus and Hirrichs, 2004a) comforted our idea and in order to improve interaction with the human user, we proposed by (Laraba and Sahnoun, 2004b) to carry out explanatory dialogue between the system and human user using dialogical agents (Noriega and Sierra, 1996; Sansonnet *et al.*, 2002). Agent-agent communication was also described using KQML language (Labrou and Finin, 1997). In our current effort, we tempt to reinforce distribution inside our system. Some tasks are thus decomposed and new agents are designed to achieve new subtasks. The resulting explanatory system is tested on a well known example in qualitative simulation literature.

To our knowledge, our system represents the first attempt concerning this step of a qualitative simulation algorithm. Other research (Bredeweg, 2002) also perceived difficulties in communicating simulation results to human users.

Which explanatory approach is to be used for that purpose? What knowledge are necessary to produce satisfactory explanation? How are the explanatory knowledge represented? How is the explanatory module implemented? These are our concerns in this study.

RELATED WORK

Many of previous research focused on generating explanations of system behaviours from qualitative simulation of physical models (Forbus and Falkenhainer, 1990; Far, 1992; Ironi and Stefanelli, 1992; Salles *et al.*, 1997). Some others pointed up the integration of available techniques in qualitative reasoning and intelligent tutoring systems to produce better explanation facilities (Bouwer and Bredeweg, 1999). Other efforts investigated designing explanatory dialogue based on a text planning framework (Moore, 1994), or natural language-based interaction (Carenini *et al.*, 1994). In addition, several works addressed multi-agent interaction including coordination and collaboration in distributed multi-agent systems (Jennings, 1996; Jennings *et al.*, 1998; Lesser, 1998), as well as individual human/agent collaboration needs (Meyers and Morly, 2001), or group interaction among humans and autonomous agents (Shrokenghost *et al.*, 2002). More recently, a new kind of software was developed that could be effectively treated as a collaborator, being capable of high-bandwidth interaction with their human partners (Forbus and Hirrichs, 2004b).

CONCEPTS AND MECHANISMS

Qualitative simulation: Before discussing explanation principles, let's introduce qualitative simulation. Its interests are multiple (Haton, 1991):

- The parameters controlling a system change qualitatively even they are defined quantitatively
- Quantitative data are often lacking in problem solving
- The construction of complete quantitative models is not always possible

In the constraint and propagation approach, qualitative simulation proceeds using the propagation/prediction cycle. Propagation phase allows completion of current state qualitative description by constraint propagation.

Prediction phase determines the state to be inferred using transitions (P-transition from time t_i to time $[t_i, t_{i+1}]$ and I-transition (from time $[t_i, t_{i+1}]$ to time t_i) and external constraints. The result is a successive sequence of qualitative states defining possible behaviors of the system, as shown in Fig. 1.

In most of qualitative modeling and simulation packages, simulation process begins with a qualitative description of the behavior of the system and its initial state, using physical parameters and relationships between them. Each parameter is defined as a physical quantity expressed by a real and continuously differentiable function: $f: [a, b] \rightarrow R$ a and b being real numbers. To make qualitative modeling, simulation and design possible, attention was restricted, (Kuipers and Ramamoorthy, 2002), to what said reasonable functions, some continuously differentiable functions with additional constraints making qualitative reasoning possible. Simulation thus produces a description of the system behavior. Each behavior is a series of qualitative states through which the physical system may move over time. Qualitative states are unique descriptions of the physical system. Each state, $QS(f, t)$, defined for a function

f at an instant t , is characterized by a landmark value $qval$ and the sign of its derivative $qdir$ as follows:

$$\forall t \in [a, b] \text{ QS}(f, t) = (qval, qdir) \text{ with}$$

$$qval = \begin{cases} l_j & \text{if } f(t) = l_j \\ (l_j, l_{j+1}) & \text{if } f(t) \text{ in } [l_j, l_{j+1}] \\ l_j & \text{being a member of a totally ordered set} \\ & \text{said quantity set.} \end{cases}$$

$$qdir = \begin{cases} \text{inc} & \text{if } f'(t) > 0 \\ \text{std} & \text{if } f'(t) = 0 \\ \text{dec} & \text{if } f'(t) < 0 \end{cases}$$

Transitions from one state to another are obtained by continuous changes in parameters, creating a behavior tree. We point here at the need of an explanatory module to justify each transition and possibly the absence of an expected behavior in the behavior tree.

We notice, for instance, that the qualitative state $\langle l_j, l_{j+1}, \text{inc} \rangle$ is inferred according to p2 transition from the qualitative state $\langle l_j, \text{std} \rangle$ and is deriving on $\langle l_{j+1}, \text{std} \rangle$ qualitative state according to i2 transition.

Explanation: Let's now discuss explanation. The founder work in this early but still flourishing artificial intelligence field was (Schank and Abelson, 1977). It was then followed by loads of research which almost agree that explanation is a reasoning task having its own knowledge.

Explanation is based on a contextual and cooperative approach.

Explanation targets a human user with his own knowledge, his habits and his doubts (Lemaire and Safar, 1991): that motivates cooperation. Explanations that did not consider the user (Brezillon, 1992), or considered him as novice (Brezillon and Karsenty, 1995), failed reaching their objectives. The final explanation generated, based on a dialogue between the system and the human user (Moore, 1994), results then of a progressive refinement of the first explanation produced, by considering new additional knowledge. That was provided by both of the explanatory module and the user, which have to interact in a common context. Context might be represented (Brezillon and Abuhakima, 1995). Explanatory knowledge, the dialogue, characteristics of the user, those are its main components (Brezillon, 1995).

Dialogical agents: Dialogical agents are software components provided with the capability to interact with human users using natural language requests. They are composed of three parts:

time t_i	P-trans	time $[t_i, t_{i+1}]$	I-trans	time t_{i+1}
$\langle l_j, \text{std} \rangle$	p1	$\langle l_j, \text{std} \rangle$	i1	$\langle l_j, \text{std} \rangle$
$\langle l_j, \text{std} \rangle$	p2	$\langle l_j, l_{j+1}, \text{inc} \rangle$	i2	$\langle l_{j+1}, \text{std} \rangle$
$\langle l_j, \text{std} \rangle$	p3	$\langle l_{j-1}, l_j, \text{dec} \rangle$	i3	$\langle l_{j-1}, \text{std} \rangle$
$\langle l_j, \text{inc} \rangle$	p4	$\langle l_j, l_{j+1}, \text{inc} \rangle$	i4	$\langle l_{j+1}, \text{inc} \rangle$
$\langle l_j, l_{j+1}, \text{inc} \rangle$	p5	$\langle l_j, l_{j+1}, \text{inc} \rangle$	i5	$\langle l_j, l_{j+1}, \text{inc} \rangle$
\rangle	p6	$\langle l_{j-1}, l_j, \text{dec} \rangle$	i6	$\langle l_{j-1}, \text{dec} \rangle$
$\langle l_j, \text{dec} \rangle$	p7	$\langle l_{j-1}, l_j, \text{dec} \rangle$	i7	$\langle l_{j-1}, l_j, \text{dec} \rangle$
$\langle l_{j-1}, l_j, \text{dec} \rangle$		$\langle l_j, l_{j+1}, \text{inc} \rangle$	i8	$\langle l^*, \text{std} \rangle$
\rangle		$\langle l_{j-1}, l_j, \text{dec} \rangle$	i9	$\langle l^*, \text{std} \rangle$

Fig. 1: Successive qualitative states inferred

- The effective component: which supports the actual software and hardware processes
- The mediator: considered as a middleware level, it has to represent the effective component for the user, to handle users requests using this representation, to control and command the effective component and to maintain the coherence of the representation
- The interface: used to display the effective component changes to the user.

VDL and VQL languages: Agent Communication Languages (ACL) are, (Sansonnnet *et al.*, 2002), assumed not satisfying new needs in users/agent interaction. A complementary approach was then proposed consisting of a specific Agent Design and Query language. It would integrate in the same model:

- An Agent Design Language (ADL) dedicated to the representation of the structure and behavior of the agent
- An Agent Query Language (AQL) dedicated to introspection of the structure and behavior of the agent

That drove to VDL (View Design Language) language (Sabouret and Sansonnnet, 2001; Sansonnnet, 2001a) and VQL (View Query Language) language (Sansonnnet, 2001b).

OUR EXPLANATORY SYSTEM

Explanatory system analysis: Two scenarios are identified when a user asks for an explanation: the system replies using either a satisfying explanation, or a non satisfying explanation. In this case, the user asks for another explanation. He then initiates a dialogue with the system. The interaction between the user and the system is represented using the use case notation of OOSE (Jacobson *et al.*, 1992).

The interactions of the use cases are formalized using MSC diagrams. MSC stands for Message Sequence Charts (Grabowski *et al.*, 1993; Rudolph *et al.*, 1996). Obtained diagram is shown in Fig. 2.

Explanatory system conceptualization: At the end of simulation process, explanatory module intervenes to eventually justify any transition or absence of an expected behavior. When responding to a user, the explanatory module associates the question to an explanatory strategy in the explanatory knowledge base. A first explanation is then generated which may not satisfy the user. A dialogue can then take place between

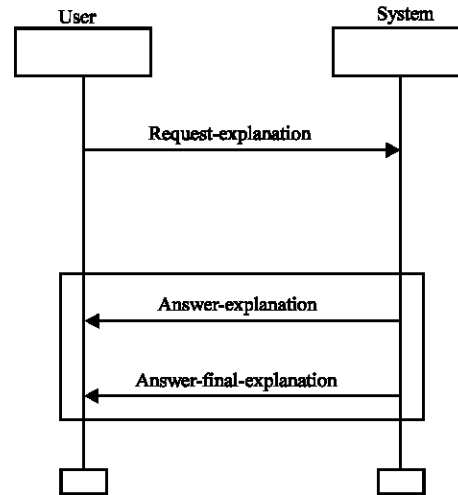


Fig. 2: MSC user request use case diagram

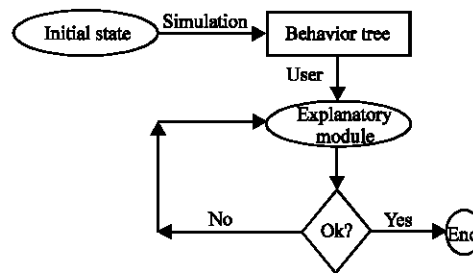


Fig. 3: Explanatory module running principle

the explanatory module and the user. Each actor must consider new knowledge acquired by the other. The process is stopped when explanation provided by explanatory module is finally accepted by the user or terminates.

Figure 3 shows this running principle

An explanatory method for qualitative simulation may have two goals:

- To justify any state transition in the behavior tree
- To justify why an expected behavior is missing.

Therefore, when receiving a question from the user, explanatory module will analyze it to determine what explanatory strategy is to be performed, according to the user request, that either:

- Asking to justify any transition (such a question will begin by why... or how...)
- Asking to understand why a behavior he expected is not in the behavior tree (such a question is thus beginning with why not...).

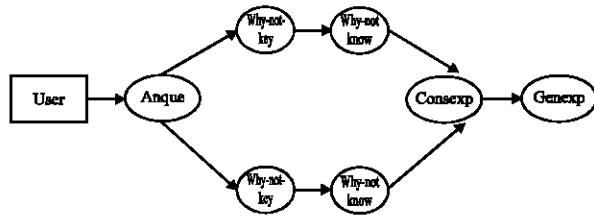


Fig. 4: Explanatory module architecture

It then initiates a dialogue with the user to provide him with a satisfactory explanation. Final explanatory text will then be constructed and generated.

Many tasks are thus performed and the explanatory reasoning conceptual model is represented as shown in Fig. 4.

Figure 4 shows the tasks performed by the explanatory module to provide final explanatory text. Analque task analyses user question. Why-How task is performed to answer why or how questions. In the other hand, Why not task answers why not questions. Consexp task constructs intermediate explanatory texts and Genexp task generates final explanatory text to be provided to the user.

These tasks are now described by giving each task its name, a short text presenting it, input and output ingredients, structure, control and frequency of use.

- Analque task: analyses user request for determining whether it is a why not question, a why question, or a how question
 Input: User question
 Output: Question goal
 Control: If question starts with why not
 Then Why-not task
 ElseWhy-how task
 Frequency of application: Regular

This task is performed by Anque Agent.

- Why-not task: Answers why not user questions
 Input: Analque task output when identifying a why not question
 Output: Resulting response
 Frequency of application: Low

This task is divided into two subtasks: Why-not-key subtask, which is concerned with identifying user question keywords and Why-not-know subtask, which is concerned with retrieving knowledge relating to keywords identified. Why-not-key subtask is performed by

Whot-key agent and Why-not-know subtask is performed by Whot-know agent.

- Why-how task: Answers why and how user questions
 Input: Analque task output when identifying a why or how question
 Output: Resulting response
 Frequency of application: High

This task is divided into two subtasks: Why-how-key subtask, which is concerned with identifying user question keywords and Why-how-know subtask, which is concerned with retrieving knowledge relating to keywords identified. Why-how-key subtask is performed by Why-key agent and Why-how-know subtask is performed by Whow-know agent.

- Consexp task: Constructs intermediate explanatory text
 Input: Why-how task output
 Output: Intermediate explanatory text
 Frequency of application: Regular

This task is performed by Conex agent.

- Genexp task: Generates ultimate explanatory text
 Input: Consexp task output when satisfying user
 Output: Final explanatory text
 Frequency of application: High

This task is performed by Genex agent.

Agent modeling

Agent identification: As a Hive agent (Minar *et al.*, 1999), an explanatory agent in our explanatory module is more than a distributed object with an execution thread. To perform the five tasks identified above, many agents are identified:

- An analyse-question agent: Anque
 Name: Anque
 Type: Reactive
 Role: Detecting user’s question type
 Goals: Rooting user’s question to Whow-key or Whot-key agents
 Reasoning capabilities: when receiving user’s question, it determines its type according to its starting word

That is the first dialogical agent.

- A Why-How agent: Whow-key

Name: Whow-key
Type: Reactive
Role: Answering why or how user's questions
Goals: Preparing explanation construct
Reasoning capabilities: Identifying user question keywords

- A Why-How agent: Whow-know

Name: Whow-know
Type: Reactive
Role: Answering why or how user's questions
Goals: Preparing explanation construct
Reasoning capabilities: Retrieving all knowledge necessary to explanation construct

- A Why-Not agent: Whot-key

Name: Whot-key
Type: Reactive
Role: Answering why not user's questions
Goals: Preparing explanation construct
Reasoning capabilities: Identifying user question keywords

- A Why-Not agent: Whot-know

Name: Whot-know
Type: Reactive
Role: Answering why not user's questions
Goals: Preparing explanation construct
Reasoning capabilities: Retrieving all knowledge necessary to explanation construct

- A Construct-Explanation agent: Conex

Name: Conex
Type: Reactive
Role: Constructs preliminary explanation
Goals: Giving user a minimal explanation answering his question
Reasoning capabilities: Constructs explanation using knowledge retrieved by Whow-know or Whot-know agents

Preliminary explanation elaborated may not satisfy human user. Conex agent has to collaborate with him to produce the best explanation. Therefore, Conex agent is the second dialogical agent.

- A Generate-Explanation agent: Genex

Name: Genex
Type: Reactive
Role: Generates final text explanation
Goals: Providing the user with the best explanation in its best form
Reasoning capabilities: Generates the ultimate explanation

Ultimate explanation obtained is to be communicated to human user by Genex agent, the third dialogical agent.

Dialogical agents' architecture: Anque, Conex and Genex dialogical agents are composed of three entities, the effective component, the mediator and the interface, according to general architecture of dialogical agent as described in subsection 3.3. They are described using VDL language.

The user interacts with the effective component by sending requests to the mediator, expressed in natural language, or VQL language, as associated AQL language to VDL language. To avoid the important difficulty of developing natural processing tools, the approach described by (Rich and Sidner, 1997), might be used.

The mediator has the charge of sending actual commands to effective component. Resulting runtime of the dialogical agent is re-displayed to the user through the interface.

Communication between agents: Diverse agents in our system must work together to accomplish the explanatory task of the whole system. They have thus to communicate with each other.

Prototypal scenarios: We first describe the prototypical scenarios of agent/agent interaction using MSC notation. Conversation at this stage is considered consisting of just one single interaction and the probable answer. The diagram (Fig. 5) is obtained:

Interchanged messages representation: We then represent the interchanged messages between agents, called events, using an event flow diagram, which collects the relationships between agents via services. The Fig. 6 shows the following diagram.

Interchanged knowledge modeling: We then model knowledge interchanged in each interaction. Some of them are shown in event flow diagram (Fig. 6) between square brackets.

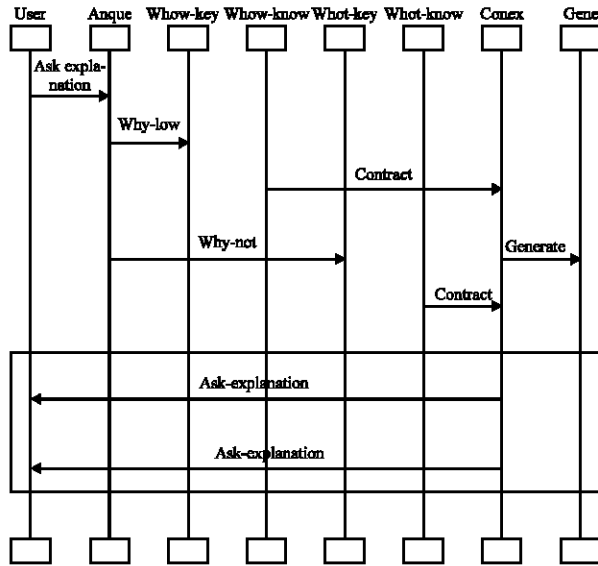


Fig. 5: MSC internal use case diagram

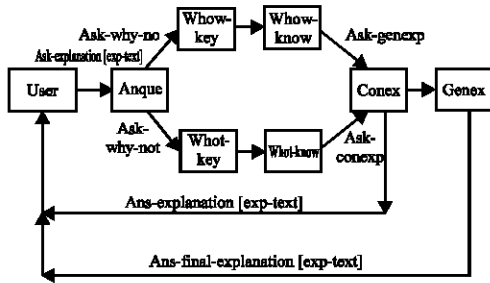


Fig. 6: Event flow diagram

Interaction description: We finally describe each interchanged interaction using KQML language:

- Describing interaction between Anque and Whow agents

Agent Anque sends this performative, identified by id00, to agent Whow:

```
(standby
:sender      Anque
:receiver    Whow
:reply-with  id01
:language    internal form
:ontology    internal form ontology
:content     (ask-one
              :sender      Anque
              :receiver    Whow
              :in-reply-to not used
              :reply-with  not used
              :language    internal form
              :ontology    internal form ontology
              :content     user question))
```

Agent Whow replies with the following performative, known as id01:

```
(ready
:sender      Whow
:receiver    Anque
:in-reply-to id00
:reply-with  id01)
```

- describing interaction between Whow and Conex agents

Agent Whow sends this performative, identified by id2, to agent Conex:

```
(standby
:sender      Whow
:receiver    Conex
:reply-with  id03
:language    KQML
:ontology    KQML ontology
:content     (ask-one
```

```
:sender      Conex
:receiver    Whow
:in-reply-to not used
:reply-with  not used
:language    KQML
:ontology    KQML ontology
:content     Final explanation))
```

Therefore agent Genex replies with the following performative, known as id03:

```
(ready
:sender      Genex
:receiver    Conex
:in-reply-to id02
:reply-with  id03)
```

- Describing interaction between Conex and Genex agents:

Agent Conex sends this performative, identified by id04, to agent Genex:

```
(standby
:sender      Conex
:receiver    Genex
:reply-with  id05
:language    KQML
:ontology    KQML ontology
:content     (ask-one
```

```
:sender      Conex
:receiver    Genex
:in-reply-to not used
:reply-with  not used
:language    KQML
:ontology    KQML ontology
:content     Final explanation))
```

Therefore agent Genex replies with the following performative, known as id05:

```
(ready
:sender      Genex
:receiver    Conex
:in-reply-to id04
reply-with   id05)
```

Explanatory knowledge modeling: Explanatory knowledge may be of different types:

- Explanatory strategies that represent methods of implementation resolution during the construction of the explanation,
- Explanatory principles that represent heuristic knowledge that contribute to the improvement of the explanation proposed by explanatory strategies,
- Knowledge of the simulation area those are useful to the explanation,
- Knowledge elaborated during the explanatory reasoning such as explanatory reasoning trace and the historical of the dialogue between the explanatory module and the user,
- Cooperative knowledge that allows considering specificities of both explanatory module and the user,
- Control knowledge composed of constraints and evaluating knowledge serving to choose between different explanatory strategies or different explanatory principles.
- Linguistic knowledge necessary for the generation of the explanatory text.

These different types may be regrouped in many classes according to their roles in elaborating the explanatory discourse. We can then distinguish:

- Contextual knowledge which improve other classes knowledge efficiency even though not directly involved in explanation elaboration. These are knowledge elaborated during explanatory reasoning
- Constructive knowledge which participate actively in the explanation building by using contextual knowledge. This class includes explanatory strategies, explanatory principles and control knowledge
- Generative knowledge that generates constructed explanatory text. This class includes linguistic knowledge and the content of the first explanatory text
- Contextualized knowledge that participated prior to the explanation elaborating process. These are object reasoning trace and knowledge of simulation area
- Cooperative knowledge

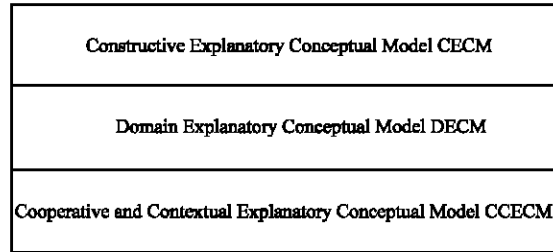


Fig. 7: Explanatory knowledge model

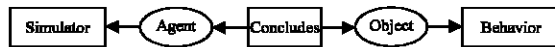


Fig. 8: Conceptual graph corresponding to P1

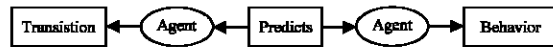


Fig. 9: Conceptual graph corresponding to P2

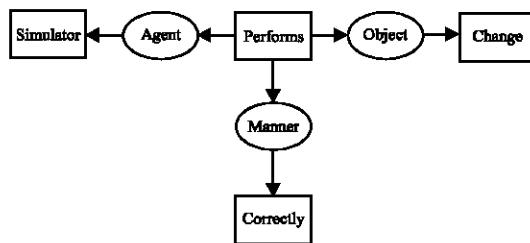


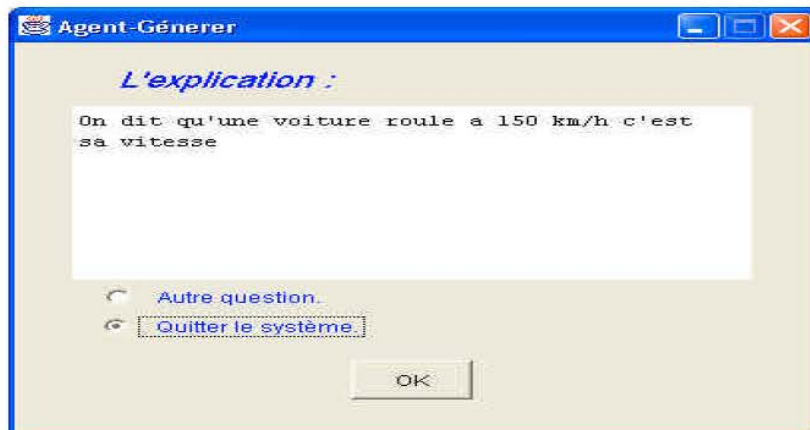
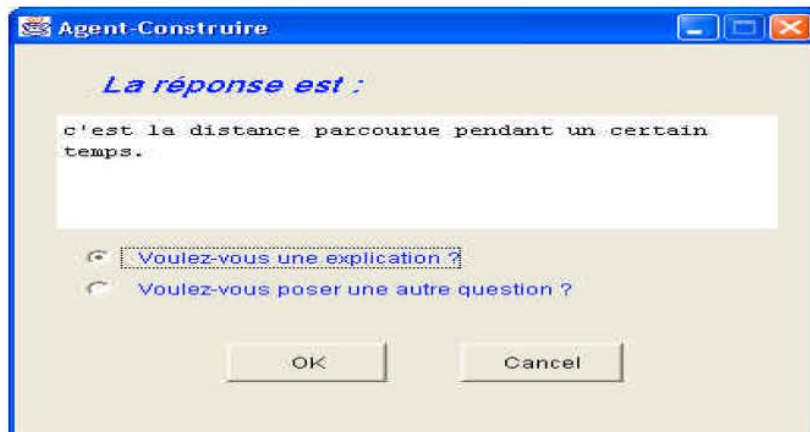
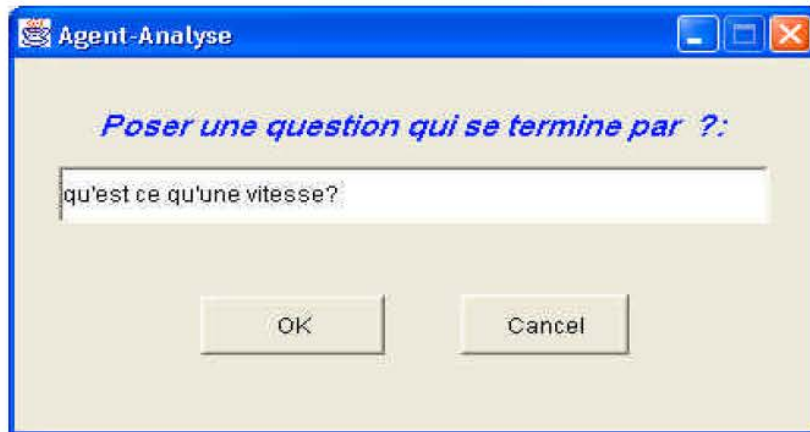
Fig. 10: Conceptual graph corresponding to P3

According to these classes, a three layers conceptual model may be built and is shown in Fig. 7.

The model is shown in Fig. 7 is based on the three layers model of the KADS (Knowledge Acquisition Design Structure) design methodology, developed at Amsterdam University (Wielinga, 1992). Its three layers are:

- Constructive Explanatory Conceptual Model (CECM): that models constructive knowledge
- Domain Explanatory Conceptual Model (DECM): that models generative and contextualized knowledge
- Cooperative and Contextual Explanatory Conceptual model: that models contextual and cooperative knowledge.

Representation of an explanatory text: An interesting form of text representation is presented by Bourcier *et al.* (1994). We naturally propose to adapt it for representing our explanatory text. This may be divided into many propositions linked by argumentative relations. This may be illustrated by the example below, where the lack of any behavior in the behavior tree may be justified as: that



behavior doesn't appear in the behavior tree, despite that I-transition's prediction, because qdir's variation corresponding to the derivative sign variation has not been executed correctly. This explanatory text is divided into three propositions:

- P1: That behavior doesn't appear in the behavior tree (Fig. 8).
- P2: That transition anticipated it (Fig. 9).
- P3: Qdir variation corresponding to the derivative sign hasn't been executed correctly (Fig. 10)

These propositions are then related by argumentative links in spite of and For.

SOWA conceptual graphs (Sowa, 1984) are well adapted to this kind of propositions. Those are bipartite graphs. Concepts and conceptual links are their two kinds of nodes.

The three propositions above may then be represented as follows:

Current status: A first attempt for implementing our explanatory system has been achieved for a well known example in qualitative reasoning literature: throwing a ball in the air (Kuipers, 1986). Agents have been written using JAVA. A part of the original execution process is shown below:

In the first window, user asks the system to define a speed. Before that, he was invited to ask a question ending by ?. The second window illustrates the system answer. In the third one, the system gives an explanation, which satisfies the user, who then chooses to leave.

DISCUSSION

We presented multiple interests of qualitative simulation and proposed to improve a qualitative simulation algorithm with explanation.

Explanation, viewed as a problem solving process, was then described at a high level of abstraction leading to a best characterization of the explanatory module behavior.

A three layers explanatory knowledge conceptual model has been constructed. Different explanatory knowledge types were identified. These were: explanatory strategies, explanatory principles, knowledge of the simulation area, knowledge elaborated during the explanatory reasoning, cooperative knowledge, control knowledge and linguistic knowledge.

An explanatory reasoning conceptual model was also built. It consisted of many cooperative explanatory tasks.

These were: Analyzing Question task, constructing explanation task, generating explanation task, Why-not-key task, why-not-know task, Why-how-key task and Why-how-know task.

Many agents were designed to achieve these tasks. Dialogical agents receive user questions and provide corresponding explanatory text, which have been cooperatively elaborated by other agents. Agent/agent communication has been described using KQML language.

A first attempt for implementing our explanatory system has been achieved using JAVA language, for a well known example in qualitative reasoning literature. In a future work, dialogical agents and their interaction with human user would be implemented using VDL and VQL languages.

REFERENCES

- Belatar, B. and M.E.H. Laraba, 1997. Contributions of explanation in simulation. In Proceedings of MOSIM Conference, Rouen.
- Bourcier, F., M.H. Greboval, G. Kassel and P. Trigano, 1994. Construction of an Explanation and generation in natural language: A case study. In Proceedings of RFIA94.
- Bouwer, A. and B. Bredeweg, 1999. Explanation and qualitative reasoning. In Proceedings of QR'99 Workshop, Skotland.
- Bredeweg, B., 2002. Communicative interaction and qualitative reasoning. In Proceedings of Qualitative Reasoning Workshop, Sitges.
- Brezillon, P., 1992. Users Intervention in cooperative explanation. In Proceedings of AARI'92 Workshop on Explanation.
- Brezillon, P., 1995. Explanation and context. In Acts of Seminar on Man-machine Explanation and Cooperation, CNAM, Paris.
- Brezillon, P. and S. Abuhakima, 1995. Using knowledge in its context. Report on the IJCAI 93 Workshop, Laforia.
- Brezillon, P. and L. Karsenty, 1995. Cooperative problem solving and explanation. In Acts of International Meeting Expert Systems with Applications.
- Carenini, G., V.O. Mittal and J.D. Moore, 1994. Generating patient-specific interactive natural language explanations. In Proceedings of 18th Symposium on Computer Applications in Medical Care, McGraw Hill inc.
- De Kleer, J., 1977. Multiple representations of knowledge in a mechanics problem-solver. In Proceedings of IJCAI Workshop.

- Far, B.H, 1992. Functional reasoning, explanation and analysis. In Proceedings of 6th Annual Conference JSAI, Tokyo, Japan.
- Forbus, K. and B. Falkenhainer, 1990. Self explanatory simulation: An integration of qualitative and quantitative knowledge. In Proceedings of AAAI Conference.
- Forbus, K. and T. Hinrichs, 2004a. Self modeling in companion cognitive systems: Current plans. Darpa Workshop on Self-aware Systems, Washington DC.
- Forbus, K. and T. Hinrichs, 2004b. Companion cognitive systems, a step towards human-level AI. In Proceedings of AAAI Fall Symposium on Achieving Human-level Intelligence through Integrated Systems and Research, Washington DC.
- Gauthier, P.O. and T.R. Gruber, 1993. Generating explanations of device behavior using compositional modeling and causal ordering. In Proceedings of Eleventh NCAI, Washington.
- Grabowski, J., P. Graubman and E. Rudolph, 1993. The Standardization of Message Sequence Charts. In Proceedings of IEEE Software Engineering Standards Symposium.
- Gruber, T.R. and P.O. Gauthier, 1993. Machine generated explanations of engineering models: A compositional modeling approach. In Proceedings of IJCAI Workshop, Chambery, France.
- Haton, J.P., 1991. Reasoning in Artificial Intelligence. ISBN: 2729603352 Intereditions,
- Ironi, L. and M. Stefanelli, 1996. Generating explanations of pathophysiological system behaviors from qualitative simulation of compartmental models. In Proceedings of Qualitative Reasoning Workshop.
- Jacobson, I., M. Christerson, P. Johnson and G. Overgaard, 1992. Object oriented software engineering: A Use Case Driven Approach, ACM Press.
- Jennings, N.R., 1996. Coordination Techniques for Distributed Artificial Intelligence. Foundations of Distributed Artificial Intelligence, Sixth-generation Computer Technology Series, O'hare, G.M.P. and N.R. Jennings (Eds.), (New York: John Wiley and Sons inc), pp: 187-210.
- Jennings, N.R., K. Sycara, M. Wooldridge, 1998. A roadmap of agent research and development. Autonomous agents and multi-agent systems. J. Autonomous Agents and Multi-agent Syst., 1: 7-38.
- Kuipers, B., 1984. Common sense reasoning about causality: Deriving behavior from structure. Artificial Intelligence, 24: 169-204.
- Kuipers, B., 1986. Qualitative simulation. Artificial Intelligence, 29: 289-338.
- Kuipers, B. and S. Ramamoorthy, 2002. Qualitative Modelling and Heterogeneous Control of Global System Behaviour. Tromlin, C.J. and M.R. Greenstreet, (Eds.), Hybrid Systems: Computation and Control, LNCS, Springer Verlag.
- Labrou, Y. and T. Finin, 1997. A Proposal for a New KQML Specification. Technical Report CS-97-03, University of Maryland, Baltimore, US.
- Laraba, M.E.H., 1999. Simex: A task architecture for an explanatory simulator. In Proceedings of MOSIM'99 Conference. Annecy.
- Laraba, M.E.H. and Z. Sahnoun, 2002. Towards a framework for enhancing qualitative simulation with explanation. In Proceedings of QR'02 Conference, Sitges.
- Laraba, M.E.H. and Z. Sahnoun, 2004. A Multi-Agent System for Explaining Qualitative Simulation. In Proceedings of ISIICT'04 Conference, Amman.
- Laraba, M.E.H. and Z. Sahnoun, 2004. Explanatory dialogical agents for qualitative simulation. In Proceedings of ACIT 2004 Conference, Constantine.
- Lesser, V.R., 1998. Reflections on the nature of multi-agent coordination and its application for an agent architecture. J. Autonomous Agents and Multi-agent Sys., 1: 89-111.
- Lemaire, B. and B. SAFAR, 1991. ESMERALDA, An architecture to build explanation using knowledge cooperation. In Acts of 11th International Conference on Expert Systems Experts and Applications Avignon.
- Meyers, K.L. and D.N. Morly, 2001. Directing agent communities: An initial framework. In Proceedings of IJCAI 2001 Workshop on Autonomous, Delegation and Control Interacting With Autonomous Agents, pp: 81-88.
- Minar, N., M. Gray, O. Roop, R. Krikourian and P. Maes, 1999. Hive: Distributed agents for networking things. In Proceedings of ASA/MA'99. First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents.
- Moore, J.D., 1994. Participating in explanatory dialogues: Interpreting and Responding to Questions in context, MIT Press.
- Noriega, P. and C. Sierra, 1997. Towards layered dialogical agents. In Proceedings of ATAL' 96, Budapest, pp: 69-81.
- Rich, C. and C.L. Sidner, 1997. Collagen: When agents collaborate with people. In Proceedings of the first Conference on Autonomous Agents.

- Rudolph, E., J. Grabowski and P. Graubman, 1996. Tutorial on message sequence charts. In Proceedings of FORTE-PSTV 96 Conference.
- Sabouret, N. and J.P. Sansonnet, 2001. Extracting behaviours from local interactions. In Proceedings of Francophone Days of Distributed Artificial Intelligence and Multi-agent Systems JFIADSMA.
- Salles, P., B. Bredeweg and R. Winkels, 1997. Deriving Explanations from Qualitative Models. B. Boulay and R. Mizoguchi (Eds.), *Artificial Intelligence in Education Knowledge and Media in Learning Systems*, Japan.
- Sansonnet, J.P., 2001. Notes on VQL 0.4. Technical Report, LIMSI-CNRS.
- Sansonnet, J.P., N. Sabouret and G. Pitel, 2002. An agent design and query language dedicated to natural language interaction. In proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy.
- Shank, R.C. and R.P. Abelson, 1977. *Scripts, Plans, goals and understanding: An Inquiry Into Human Knowledge Structures*, Hillsdale, NJ: Eirbaum.
- Shrokenghost, D., C. Martin, P. Bonasso, D. Kortenkamp, T. Milam and C. Thronesbery, 2002. Supporting group interactions among humans and autonomous agents. *Connection Sci.*, 14: 361-369.
- Sowa, J.F., 1984. *Conceptual structures: Information Processing in Mind and Machine*. Addison Wesley, Reading mass.
- Wielinga, B.J., T.H. Shreiber and J.A. Breuker, 1992. KADS: A modeling approach to knowledge engineering, knowledge acquisition special issue, 4: 5-53.