# A Dialogical Agent-based Framework for Explaining the Results of QSIM Algorithm to the End-User

Mohamed El Habib LARABA- Zaidi Sahnoun

LIRE Laboratory-Computing Science Department

Mentouri University of Constantine-Algeria

Tel &Fax: 0021331639010

habib@wanadoo.dz

sahnounz@yahoo.fr

**Abstract**

Our paper addresses the problem of explaining the results of QSIM, the most popular qualitative simulation algorithm, to the end-user. At the end of simulation process, end-user can view those results by refering to the behavior tree or any individual behaviors predicted. However, no justification is given for any state transition in the behavior tree. Moreover, end-user may unfortunately fail looking for understanding the absence of an expected individual behavior! We propose to palliate that lack by providing the end-user with some explanation. We do so by allowing him to converse with an explanatory module, executed at the end of the simulation. When planning this module, we consider explanation, as a cooperative task, between the explainer, and the end-user. For that purpose, we design a five dialogical agents-based framework, which integrates communication procedures.

## 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) (Kuipers, 1986) were the principle demonstrations. Later, explanation, another field of artificial intelligence, was used in simulation (Forbus et al, 1990) (Gauthier et al, 1993) (Gruber et al, 1993). Afterwards**,** it was proposed to integrate an explanatory module to an environment of simulation (Belatar et al, 1997) (Laraba, 1999). Subsequently, it was proposed to extend and adapt this module to an environment of qualitative simulation (Laraba et al, 2002). Other research (Bredeweg, 2002) also perceived difficulties in communicating simulation results to human users

In (Laraba et al, 2004a), it was focused on modeling that explanatory module 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 et al, 2004a) comforted that idea, and in order to improve interaction with the human user, it has been proposed in (Laraba et al, 2004b) to carry out explanatory dialogue between the system and human user using dialogical agents (Noriega et al, 1996; Sansonnet et al, 2002). Agent-agent communication was also described using KQML language (Labrou et al, 1997). In our current effort, we tempt to reinforce distribution inside that explanatory system. Some tasks are thus decomposed and new agents are designed to achieve new subtasks.

The paper is structured as follows: after this introduction, we explore the historic of integrating explanation in qualitative simulation. Therefore, the explanatory system specification is detailed, and finally, a conclusion terminates the paper.

## Related Work

Many of the previous researches focused on generating explanations of system behaviours from qualitative simulation of physical models (Forbus et al, 1990; Far, 1992; Ironi. et al, 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, et al, 1999). Other efforts investigated designing explanatory dialogue based on a text planning framework (Moore J.D, 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 K.L et al, 2001), or group interaction among humans and autonomous agents (Shrokenghost et al, 2002). More recently, a new kind of software that could be effectively treated as a collaborator, was developed, being capable of high-bandwidth interaction with their human partners (Forbus et al, 2004b).

## Our Explanatory System

### Explanatory System Analysis

Two scenarios are identified when a user asks for an explanation: the system replies with 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). The Obtained diagram is shown below:

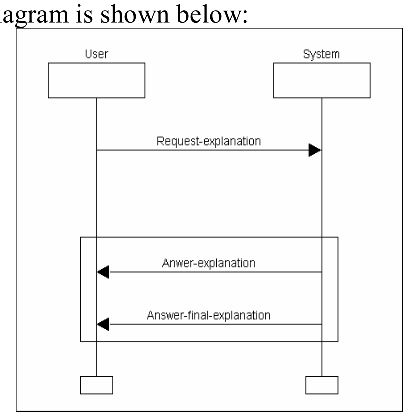


**Figure 1.** MSC user request use case diagram

### Explanatory System Conceptualization

At the end of the 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 which may not satisfy the user is then generated. A dialogue can then take place between the explanatory module and the user. Each actor must consider the new knowledge acquired by the other. The process is stopped when the explanation provided by the explanatory module is finally accepted by the user or terminates. Figure 2 illustrates this running principle.

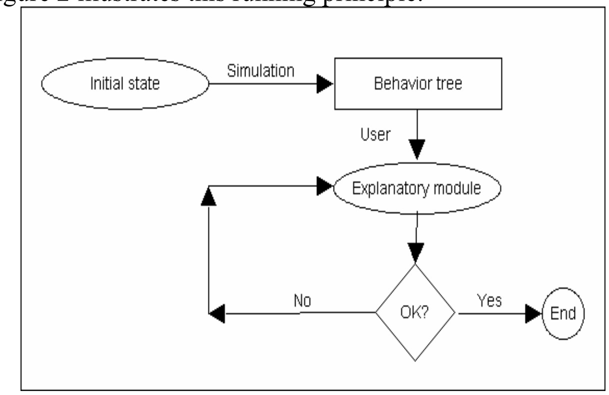


**Figure 2.** Explanatory module running principle

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, the explanatory module will analyze it to determine what explanatory strategy is to be performed, according to the user's question request either:

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

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 Figure 3 below:

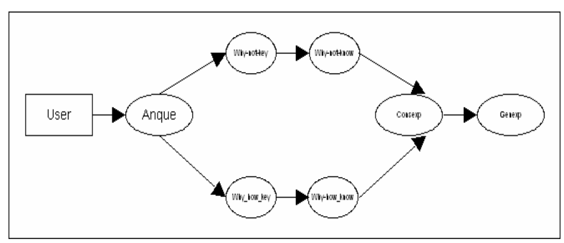


**Figure 3.** Explanatory module architecture

This figure shows the tasks performed by the explanatory module to provide final explanatory text. Analque task analyses the user's question. Why-How task is performed to answer why or how questions. On 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 the user's 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,
Else Why-how task.
Frequency of application: regular.
This task is performed by Anque Agent.

- Why-not task: answers why not user's 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 concerned with identifying user's question keywords, and Why-not-know subtask 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 concerned with identifying user's question keywords, and Why-how-know subtask concerned with retrieving knowledge relating to keywords identified. Why-how-key subtask is performed by Whow-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 the 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 the user's question type.
Goals: rooting the user's question to Whow-key or Whot-key agents.
Reasoning capabilities: when receiving the 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 the user's why or how questions.
Goals: preparing the explanation construct.
Reasoning capabilities: identifying the user's question keywords.

- A Why-How agent: Whow-know

Name: Whow-know.
Type: reactive.
Role: answering the user's why or how questions.
Goals: preparing the explanation construct.
Reasoning capabilities: retrieving all the knowledge necessary to the explanation construct.

- A Why-Not agent: Whot-key

Name: Whot-key.
Type: reactive.
Role: answering the user's why not questions.
Goals: preparing the explanation construct.
Reasoning capabilities: identifying the user's question keywords.

- A Why-Not agent: Whot-know

Name: Whot-know.
Type: reactive.
Role: answering the user's why not questions.
Goals: preparing the explanation construct.
Reasoning capabilities: retrieving all knowledge necessary to the explanation construct

- A Construct-Explanation agent: Conex

Name: Conex.
Type: reactive.
Role: constructs the preliminary explanation.
Goals: giving the user a minimal explanation answering his question.
Reasoning capabilities: constructs explanation using the knowledge retrieved by Whow-know or Whot-know agents.

The Preliminary explanation elaborated may not satisfy the 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 the final text explanation.
Goals: providing the user with the best explanation in its best form.
Reasoning capabilities: generates the ultimate explanation.
```

The ultimate explanation obtained is to be communicated to the 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. They would be described using VDL language (Sansonnet, 2001a).

The user interacts with the effective component by sending requests to the mediator, expressed in natural language, or VQL language (Sansonnet, 2001b), as associated AQL language to VDL language. To avoid the important difficulty of developing natural processing tools, the approach described in (Rich et al, 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. This section describes that agent/agent communication.

**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 (Figure 4) is obtained:

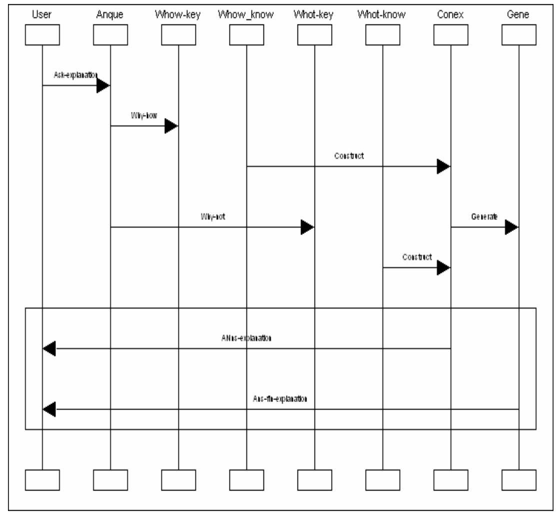


**Figure. 4**. MSC internal use case diagram

***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 following diagram is obtained:

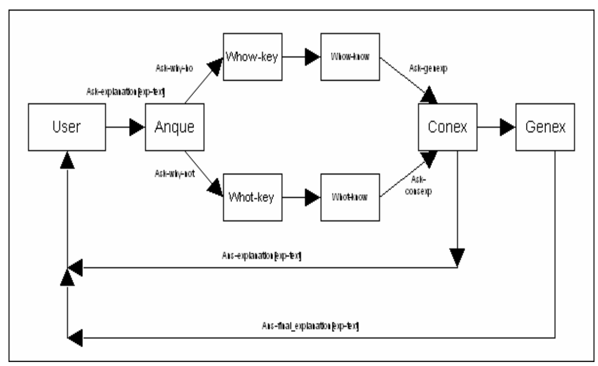


**Figure 5.** Event flow diagram

***Interchanged knowledge modeling.*** We then model knowledge interchanged in each interaction. Some of them are shown in event flow diagram, in Figure5 above, between square brackets.

***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 )
```

## Conclusion and Future Work

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

An explanatory reasoning conceptual model was 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 the user's questions and provide corresponding explanatory text, which has been cooperatively elaborated by other agents. Agent/agent communication has been described using KQML language. A first attempt for implementing our explanatory system is under developing 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 Laraba, M.E.H. 1997. *Apports de l'explication en simulation*. In Proceedings of MOSIM'97, Rouen.

Bouwer, A., and Bredeweg, B. 1999. *Explanation and qualitative reasoning*. In Proceedings of QR'99 workshop, Scotland.

Bredeweg, B. 2002. *Communicative interaction and qualitative reasoning*. In Proceedings of Workshop for qualitative reasoning QR2002.

Carenini, G., Mittal, V.O., and Moore J.D. eds 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 77 Conference.

Far, B.H. 1992. *Functional reasoning, explanation and analysis*. In Proceedings JAERI-M 91-225, Tokai, Japan.

Forbus, K., and Falkenhainer, B. 1990. *Self explanatory simulation an integration of qualitative and quantitative knowledge*. In Proceedings of AAAI 90 Conference.

Forbus, K., and Hinrichs, T. 2004. *Self modeling in Companion cognitive systems: current plans*. In the proceedings of Darpa workshop on self-aware systems, Washington DC.

Forbus, K., and Hinrichs T. 2004. *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 Gruber, TR. 1993. *Generating explanations of device behavior using compositional modeling and causal ordering*. In Proceedings of eleventh NCAI Conference, Washington.

Grabowski, J., Graubman, P., and Rudolph, E. 1993. *The standardization of message sequence charts*. In proceedings of IEEE software engineering standards symposium.

Gruber, T.R., and Gauthier, P.O. eds 1993. *Machine generated explanations of engineering models. A compositional modeling approach*. In Proceedings of IJCAI Conference, Chambery, France.

Ironi, L., and Stefanelli, M. 1996. *Generating explanations of pathophysiological system behaviors from qualitative simulation of compartmental models*. In Proceedings of Workshop of Qualitative Reasoning.

Jacobson, I., Christerson, M., Johnson, P., and Overgaard, G. 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, edited by O'hare, G.M.P and Jennings N.R, (New York: John Wiley & Sons inc), pp 187-210.

Jennings, N.R., Sycara, K, and Wooldridge, M. 1998. *A roadmap of agent research and development.* Autonomous agents and multi-agent systems, 1(1), pp 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 Ramamoorthy, S. eds 2002. *Qualitative modelling and heterogeneous control of global system behaviour*. C. J. Tromlin and M. R. Greenstreet (eds.), Hybrid systems: computation and control, LNCS, Springer Verlag.

Labrou, Y., and Finin, T. 1997. *A proposal for a new KQML specification*. Technical Report CS-97-03, University of Maryland.

Lesser, V.R. 1998. *Reflections on the nature of multi-agent coordination and its application for an agent architecture*. Autonomous agents and multi-agent systems, 1(1), pp 89-111.

Laraba, M.E.H. 1999. *SIMEX: une architecture de tâches pour un simulateur explicatif.* In Proceedings of MOSIM'99 Workshop, Annecy.

Laraba, M.E.H., and Sahnoun, Z. 2002. *Towards a framework for enhancing qualitative simulation with explanation*. In Proceedings of QR'02 Workshop, Sitges.

Laraba, M.E.H., and Sahnoun, Z. 2004. *A multi-agent system for explaining qualitative simulation*. In Proceedings of ISIICT'04 conference, Amman.

Meyers, K L., and Morly D.N. 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., Gray, M., Roop, O., Krikourian, R., and Maes P. 1999. *Hive: distributed agents for networking things*. In Proceedings of ASA/MA'99, the 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 Sierra, C. 1996. *Towards layered dialogical agents*. In Proceedings f ATAL' 96, Budapest, pp 69-81.

Rich, C., and Sidner, C.L. 1997. *Collagen: when agents collaborate with people*. In Proceedings on the first conference on autonomous agents.

Rudolph, E., Grabowski, J., and Graubman P. 1996. *Tutorial on message sequence charts*. In Proceedings of FORTE-PSTV 96 conference.

Salles, P, Bredeweg, B, and Winkels, R. 1997. *Deriving explanations from qualitative models*. Artificial Intelligence in Education Knowledge and Media in learning systems, Japan.

Sansonnet, J.P. eds 2001. *Introduction à VDL 0.4.* Technical report.

Sansonnet, J.P. eds 2001. *Notes sur VQl 0.4**. Technical report.

Sansonnet, J.P., Sabouret, N. and Pitel, G. 2002. *An agent design and query language dedicated to natural language interaction*. Poster session at AAMAS 02.

Shrokenghost, D., Martin, C., Bonasso, P., Kortenkamp D;, Milam, T., and Thronesbery, C. 2002. *Supporting group interactions among humans and autonomous agents*, In connection science, Vol 14, No 4, pp 361-369.

Sowa, J.F. eds 1984. *Conceptual Structures Information processing in mind and machine*. Addison Wesley, Reading mass.

Wielinga, B. J., Shreiber, Th., and Breuker, J.A. 1992. *KADS: a modelling approach to knowledge engineering*. Knowledge acquisition special issue, 4, pp5-53.