

Interface agent development in MASA for human integration in multi-agents systems

Ammar LAHLOUHI^{1,2}, Zaidi SAHNOUN^{3,4}, Med Lamine BENBRAHIM^{1,4} and Abdelouahab BOUSSAHA^{1,4}

¹MASA Group, Department of computer science, University of BISKRA, BISKRA, DZ-07000, ALGERIA

²ammarlahlouhi@yahoo.fr

³Lire laboratory, Department of computer science, University of CONSTANTINE, CONSTANTINE, DZ-25 000, ALGERIA

⁴tbd@later.time

Abstract. The base of research works, on interface agents, is the client-server concept, where they regarded the interface agent as humans' assistants in the use of software. This vision does not permit the foundation, in a coherent way, of the human-software cooperation. In this paper, we propose a cooperation-based approach; of interface agents' development for multi-agents systems. We consider the human as an agent H, to which we associate an interaction software F that holds its objective. F is an agent, qualified of interface agent, but it has neither the human's intellectual abilities nor those of its resources' exploitation. We supplement this lack by a human-interface agent interaction. The agent H will then be consisted as a combination of the agent F and the human agent. The agent H cooperates with other software agents and the development of F will follow a multi-agents methodology.

1. Introduction

The base of the research works undertaken on interface agents (see [16], for research works survey on interface agents) is the traditional concept of client-server. They see the human as a client and qualify him as a user whereas they see the interface agent as a human's server, for the use of software systems (tools). Consequently, the interface agent must have some characteristics of intelligence, to better serve the human. This is what makes dominate aspects of traditional artificial intelligence (natural language interface, intelligent assistance, comprehension, learning...).

In this paper, we propose a different viewpoint, co-operation based approach. We try to clarify the usual approach and its disadvantages, in section 2. We then introduce the co-operation based approach and its advantages, in section 3. In the remaining sections, we show how we can concretize this approach and how we materialized it in MASA project.

2. Usual approach

Usually, the human uses software systems through tool invocation. For that, he determines its objective and plans the actions enabling him to reach it. We can subdivide these actions in two classes: those must do by the human and those must do by the software system. Since the human is intelligent, its plan is fuzzy. It revises it according to its requirements by remaking actions, adding new actions...

2.1 Human detention of actions plan

In this situation, it is the human, which holds its plan, and ensures its execution. The latter consists of: (1) sequencing plan actions, (2) execution of the actions intended for him and (3) Demand to software system, through an interface, to execute the actions, that he reserved for it.

2.2 Disadvantages

The disadvantages of this approach are numerous. We can cite:

1. Interface development is based on an interminable research of a general valid model for the use of any software system by any human role,
2. Many inconsistencies in the use and/or sequencing tools,
3. Does not make possible to the designer to exploit the human's intellectual abilities to serve the objective of the system to be developed,
4. The plan detention and its execution by the human is embarrassed, unproductive, causes errors in plan execution and its foundation...

3. Co-operation based approach

In the co-operation based approach, we view the human as an agent H that ensures one or more roles. We associate, to the agent H, an interaction software system. The software system F is a software agent, qualified of interface agent, but it has neither human intellectual abilities nor those of exploitation of its resources. We supplement this lack by an interaction with the human. The agent H then will be a combination of two agents: agent F and human agent. We will show, in the rest of this paper, that the development of F will follow a multi-agents methodology.

3.1 Actions plan detention by interface agent

In this approach, actions plan, qualified of individual task, must be explicit. Its base is organization's roles that the agent H must assume in the MAS. It is then determined during MAS design so that MAS functioning was coherent.

It is the responsibility of the interface agent to hold the individual task, and to ensure its execution. This execution consists of: (1) sequencing actions plan, (2) Demand to the human to execute actions intended for him and (3) Demand to software agents to execute actions which are intended for them.

3.2 Sharing tasks between software agents and the human

In this approach, we can share the tasks, in a coherent way, between software agents and human agent. We can charge the human agent by:

1. Execution of tasks concerning the data processing automation: calculations, storage, research...
2. Making certain decisions.

Whereas, we can charge the software agents by:

1. Achieving treatments of intelligent tasks,
2. Making certain decisions,
3. Achieving actions on the environment, external to software system, decided by itself or by the software system.

3.3 Advantages and disadvantages

The advantages of this approach are numerous. We can cite those, which allow us to:

1. Use multi-agents systems techniques of co-operation, for the derivation of the necessary characteristics of the agent H and, consequently, those of the agent F,
2. Profit the MAS from the intellectual abilities of the humans,
3. Manage human co-operation through a network, for instance...

Its disadvantages lie in the need of preliminary planning. However, we can consider this disadvantage also as an advantage, since it ensures a rigorous development of software systems.

4. Interfaces and Agents

We can classified the solutions proposed for interfacing agent-based systems as follows:

1. Interface agent for traditional applications: The human interacts with the software system as it does it usually. The interface agent observes all its actions while advising it and proposing him solutions for possible difficulties, which it will encounter.
2. Traditional interface for multi-agents systems: The human interacts directly with the agents of multi-agents system through a traditional interface.
3. Interface agents for multi-agents systems: It is an integration of the human in the multi-agents system. We consider the human as a system's component (agent) whose it plays one or more roles in a given organization.

4.1 Interface agents for traditional applications

In this case, the interface agent behaves as an assistant who must be intelligent, convivial, understanding, learning, and communicating in natural language.... This approach is that proposed by Maes [4, 5], improved by several interface agents collaboration for learning from several users by Lashkari [6] and by introducing the autonomy by Lieberman [1, 2, 3] and continued by Rich and its colleagues [7, 8, 9, 10, 11].

Several works, of this class, treat the learning of the interface agents from user's actions [4, 5, 11, 13]. Another approach is that of the development of MAS for multi-modal interfaces [14, 15].

In this vision, the software system is heterogeneous, i.e., it comprises components based on agent technology and traditional components. The interface agent's adaptation, to traditional components, makes that they do not satisfy agent's minimal characteristics.

4.2 Traditional interface for multi-agents systems

It is the approach followed by the MAS developers whose interfaces do not constitute their concern. The interface's role is to assist the human, as in the previous situations, in the use of the various agents. The base of this approach is, generally, the object-based interfaces approach.

In this latter, one presents objects, materialized by icons, representing accessible software system objects, to the human. The latter chooses icons and, using a global or contextual menu, he invokes methods of this object that are accessible to him.

In this case, the agents share the communication resources between them and with the human. The agents are then not autonomous, from the resources. Moreover, the software system cannot profit, explicitly and coherently, from human's intellectual abilities and/or human's actions on the environment. It cannot manage, either, the co-operation between the human, since it regards them as users.

4.3 Interface agents for humans' integration in multi-agents systems

We distinguish, in this case, two environments: (1) Virtual or software and (2) Real or physical. The human agent and the interface agent are associated to constitute only one agent. The interface agent wraps the human and represents it in the virtual environment, so that it uses human's capacities to serve software agents. In the other hand, the human wraps the interface agent and represents it in the real environment.

5. MASA: Multi-Agents modeling of Systems with Autonomous and heterogeneous components

MASA Project consists of three poles: MASA-Meta-model, MASA-Method and MASA-Applications.

5.1.1 MASA-Meta-model

The autonomy characterization, in MASA-Meta-model, constitutes its strong point where it makes it possible to support heterogeneous agents (robots, software and human).

The autonomy of an agent concerns of three aspects: Control, functioning and resources. These three aspects stipulate that each agent has, respectively: (1) its own control, (2) its own objective, decision process or individual task indicating what it must make, and (3) all resources, which are necessary for him to achieve its individual task.

5.1.2 MASA-Method

We describe MASA methodology, MASA-Method, in the section 5.2. We are applied it, successfully, in two projects on software processes and in a third on multi-agents simulation of collective robotics. Its application, in network exploration and in multi-agents treatment of images, is in hand.

5.1.3 MASA-Applications

The applications of MASA, MASA-Applications, include several types: software processes, simulation, and images... They are important applications but they also have the role of evaluate, validate, correct, improve and enrich MASA-Meta-model and MASA-Method.

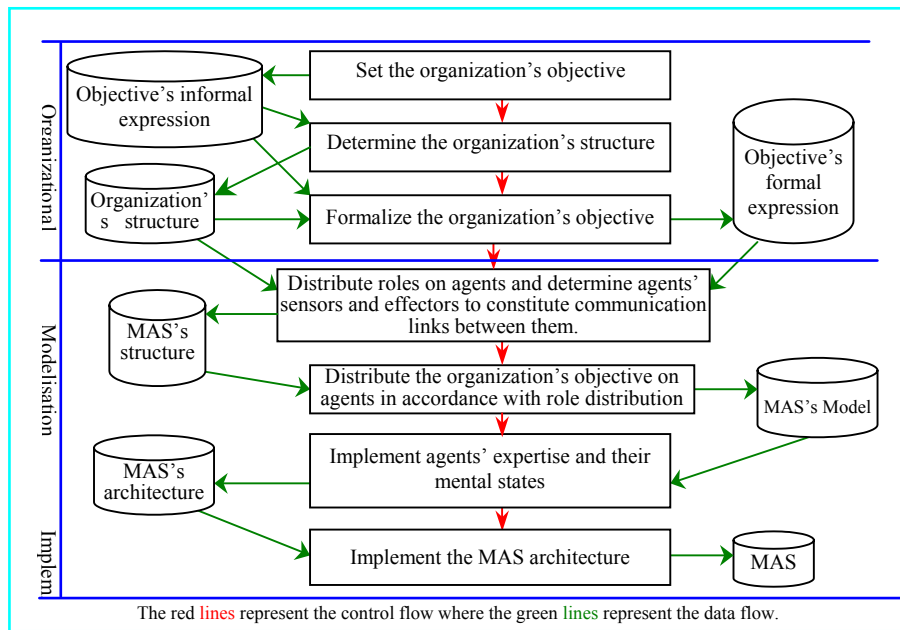


Fig. 1. MASA-Method description

5.2 MASA-Method Methodology

MASA-method, for what we give its diagrammatic description in figure 1, is an organizational methodology. Its specificity is in the characterization of the organization as being a couple (Structure, Objective) rather than as an only structure, as other multi-agents methodologies consider it. That is what clarifies the development of agents' co-operation.

We adopted the procedural description of the objective and it we qualified it of global task. We then approached the co-operation as follows:

1. Global task description in CPN, Colored Petri Net, which ensures actions synchronization,
2. Distribution of this global task on the various agents to derive the agents' individual tasks. This produces a set of synchronized CPN, which each one constitutes a description of an agent's individual task.

This demarche makes it possible to derive, starting from the organization (Structure, Objective), autonomous, co-operatives and pro-actives agents. The pro-activity means that agents are equipped with individual decision processes and the cooperation means the achievement of these individual tasks makes it possible to achieve what is equivalent to the global task's implementation.

5.3 Agents and their coordination in MASA

In MASA, the communication base is the speech acts exchange. To a speech act reception, we associate an entry event, and to its emission, we associate an exit event, in CPN-S.

In MASA, we use Java thread objects to implement or, rather, to simulate agents. We materialized the speech acts emission and their reception by remote method invocation protocol (RMI). This consideration of synchronization and communication is what ensures the agents coordination activities.

5.4 Human and interface agents in MASA

In MASA, we consider the human, from conceptual view, like an agent of the complete multi-agents system. To integrate the human agents with software agents, in MASA, we associate to each of them an interface agent.

6. Interface agents in MASA

In MASA, agent characteristics are the following: (1) Sensors and effectors, (2) Resources, (3) Mental state, (4) Expertise, and (5) Individual task. In the following sections, we describe how we can derive these characteristics for interface agents.

6.1 Interface agent

We attribute the human agent's characteristics to the interface agent as follow:

1. Sensors and effectors are models of those attributed to the roles that human agent must assume in the MAS,
2. Resources: The interface agent has not the human agent's resources but it acts on via the human agent it self by demanding him to act on it,
3. Mental state is a part of the human mental state. We explain how we choose this part of the mental state in section 6.2,
4. Expertise: The expertise's procedures are not exactly those, which the human uses. We explain the procedures' derivation in section 6.3,
5. Individual task: We explain this in the section 6.4.

6.2 Agent's state detention

For the state's detention, we can consider two solutions: (1) Memorize the state by the interface agent or, (2) Its detention by the human. The difficulty of the first solution is in that the interface agent is not able to memorize any type of information. The difficulty of the second is that the software system will not assist the human in memorizing and presenting certain data and/or results.

In MASA, we propose a combination of the two approaches: Memorizing all information, which the representation is acceptable (cost consideration, feasibility...) and leave the others with human. We left the decision, on the adoption of a solution or the other for particular information, to the MAS's designer.

6.3 Expertise's procedures

A procedure of an interface agent's expertise must make possible to communicate, to the human, the procedure name and, to present to him procedure's data and/or to receive its execution results.

If communication's means, with the human, are the usual ones (keyboard, screen, mouse...) the data's display and/or the results' entry can be complex. They can be formatted texts, graphs and/or images. In this case, the associated procedure must use tools enabling him to display these data and/or to assist the human to enter these results.

An expertise's procedure of an interface agent, which is implemented like a method of an active object, can be then described as follows: (1) Display procedure name, (2) Display its data if there is any and (3) Recover results if there is any.

6.4 Individual task

In MASA, we derive the individual task from the CPN-S description of the global task and transform it into Java program, indifferently, as those of all agents. We do not consider this derivation and this transformation in this paper to respect the

authorized size of the paper. However, we show these aspects in the application example (See section 7).

7. Example: Project's head as interface agent in the programming process

The programming process belongs to the software development process. This last knew a significant turning in the middle of the Eighties with the article of Osterweil [12] entitled "Software processes are software too". Osterweil considered the software process as being a complex software system whose development must follow the software development process.

The majority of researchers consider the many classical proposals for software process modeling (analytical, systemic and object-oriented) as insufficient. In MASA-Applications, we tackled this problem by a multi-agents approach. For the development of programming process, we used MASA-Method.

We applied the programming process, which we describe here, with the MERISE's conceptual models where the role of team's head (see the following sections) is ensured by a project head that a human agent. We also applied it to the development of an environment, for MAS development, of MASA-Method. In the following sections, we focus on the way in which we employed the interface agents.

7.1 Multi-agents system for programming conceptual models

The organization's global task must answer the question "how to program a design?" We can make this by a team of programmers who program conceptual models. We choose here an organization in team in which a team's head supervise the programmers.

The global task, which its formal description is given in the figure 2, can be described informally as follows: (1) Break up the design into components, (2) Allot components to programmers, (3) a programmer, holding a component, program it and produces a module, (4) when all components were programmed, the team's head integrates modules and produces resulted software.

7.2 Interface Agent: Project's head

We describe, in this section, the interface agent associated with the Project's head, which ensure the team's head role. We give its individual task in figure 3. The team's head comprises the following procedures: Br, Ex, St and Intgr. We show here how we derive actions plan of the interface agent and how we program an expertise's procedure for project's head.

The actions' plan is a transformation of the CPN-S of figure 3 associated to the project's head. However, the base of the derivation of the expertise's procedure is how it displays the procedure name and its entry parameters and how it recovers the

- *Types*: *Dt*, Design, *Sct*, Set of components, *Nt*, Natural numbers, *Ast*, Components, *Mt*, Modules, *St*, Software, *Smt*, Set of modules,
- *Places*: *D*, Design, *Sco*, Set of components, *Nc*, Count of extracted Component, *Ec*, Extracted components, *Pm*, Programmed modules, *Nm*, Number of modules, *Spm*, Set of programmed modules, *Rs*, Resulted software,
- *Initial marking*: *Design*, Design, *Nc*, 0 (zero), *Nm*, 0, *Spm*, ϕ , Int, 0,
- *Transitions*: *P1*..., *Pn* and *Chef* represent programmers and team's head. *Br*, *Ex*, *Intgr*, *Mem* and *Pg* represent expertise's procedures of programmers and team's head roles,
- *Functions of the arcs*: *De*, Design, *Sc*, Set of components, *R*, Remainder of design components, *C*, *Cl*... *Cn*, Component of design, *M*, Module, *Sm*, Set of modules, and *Sr*, Software result, *i*, Extracted components count, *j*, Programmed modules count, *k*, Extracted and programmed modules component count,
- *Guards*: $V(Sc)$, *Sc* is empty, $NV(Sc)$, *Sc* is not empty.

Fig. 4. The abbreviations used in the CPN and CPN-S descriptions of global task and individual tasks of figures 2 and 3

```

public ProjectHeadClass Extends Thread {
    /* State */
    DesignCl Design;  CompDesignCl DesignComp;
    ModulesSetCl Modules; SoftwareCl Software;
    /* Sensors and effectors */
    SensorsCl S-P1, ..., S-Pn; EffectorsCl E-p1..., E-Pn;
    /* Expertise */
    void Br; void Ex; void Mem; void Intg;
    /* Objective */
    public void run()/* See the remainder of the figure */

    void Br { /* The expertise's Break up procedure */
        Message.Text('Break up that design:');
        Message.Display; Co.Display; Eco.Entry;
    }

    public void Run() { /* Result of CPN-S transformation */
        Die; i.Init;
        While NV(Eco) {
            i.Inc;
            ... }
        ... }

```

Fig. 5. The Java code skeleton associated to the project's head agent, the expertise's procedure Br and the run method representing the individual task

8. Related works

The cooperation-based approach of interface agents' development, that we described here, is a new approach. We view other interface agents' approaches [2, 4, 6, 9, 10, 15] as a particular case of the cooperation based approach. We can justify this by introducing agents, in the MAS, in accordance to the cooperation-based approach, that assure particular roles of assistance, supervising.... In addition, our approach allows remedying disadvantages of usual approaches and it has many advantages, like which we explained in the section 3. Among advantages of this approach, other those explained in section 3, we can cite the simplification of the human-software system interaction. The approach reduces this to the only communications of the operations names, data and results.

It is our opinion; that our approach allows, not only the integration of humans with software agents, but also a harmonious integration of any kind of physical agents such as robots. This is not the case of other approaches. We did not study the robot integration with human and/or software agents, for the moment. However, we think that it requires only a communication interface definition based on speech acts between robots and computers. The objective of such integration is to have a model of an enterprise in which humans, robots and software agents cooperate to realize a global task (Objective), which is that of to serve clients or manufacture products.

9. Conclusion

We validated the cooperation-based approach in a multi-agents modeling of software development process as explained earlier, in this paper. The interface agent development systematic and causes no particulars difficulties. The multi-agents development process, based on MASA-Method, directs this development. This later is a difficult task that we work to simplify by systemizing, assisting and/or automating stages of this development. This is the MASA project's objective.

9. Bibliography

- [1] Henry Lieberman, "Integrating User Interface Agents with Conventional Applications", International Conference on Intelligent User Interfaces, San Francisco, January 1998.
- [2] Henry Lieberman, "Autonomous Interface Agents", In *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI '97* (pp. 67-74). ACM Press, 1997
- [3] Henry Lieberman, Christopher Fry, Louis Weitzman, "Exploring the Web with Reconnaissance Agents", COMMUNICATIONS OF THE ACM August 2001/Vol. 44, No. 8.
- [4] Maes, P., and Robyn Kozierok, "Learning Interface Agents", AAAI Conference, 1993.
- [5] Max Edward Metral, "Design of a Generic Learning Interface Agent", Department of Electrical Engineering and Computer Science, Bachelor of Science in Computer Science and Engineering, MIT, Thesis supervised by Patricia E. Maes, May 1993.

- [6] Yezdi Lashkari, Max Metral, Pattie Maes, "Collaborative Interface Agents", *Proceedings of AAI '94 Conference*, Seattle, Washington, August 1994.
- [7] Charles Rich, "Window Sharing with Collaborative Interface Agents", *SIGCHI Bulletin*, Vol. 28, No. 1, January 1996, pp. 70-78.
- [8] Charles Rich, Candace L. Sidner, "Segmented Interaction History in a Collaborative Interface Agent", *Third Int. Conf. on Intelligent User Interfaces*, Orlando, FL, January 1997.
- [9] Charles Rich, Candace L. Sidner, "COLLAGEN: A Collaboration Manager for Software Interface Agents", TR-97-21a March 1998, MERL-A MITSUBISHI ELECTRIC RESEARCH LABORATORY, <http://www.merl.com>
- [10] Charles Rich, Candace L. Sidner, "COLLAGEN: When Agents Collaborate with People", *First Int. Conf. on Autonomous Agents*, Marina del Rey, CA, February 1997.
- [11] Neal Lesh Charles Rich Candace L. Sidner, "Using Plan Recognition in Human-Computer Collaboration", TR-98-23 December 1998, MERL-A MITSUBISHI ELECTRIC RESEARCH LABORATORY, <http://www.merl.com>
- [12] L. J. Osterweil, "Software Processes are Software Too", In *Proceedings of the Ninth International Conference of Software Engineering*, pages 2-13, Monterey CA, March 1987.
- [13] Anandeeep S. Pannu, Katia Sycara, "A Learning Personal Agent for Text Filtering and Notification", The Robotics Institute, School of Computer Science, Carnegie Mellon University, pannu+@cs.cmu.edu
- [14] Ipke Wachsmuth, Britta Lenzmann, Tanja J'ording, Bernhard Jung, Marc Latoschik, Martin Fröhlich, "A Virtual Interface Agent and its Agency", AI & Computer Graphics Lab, University of Bielefeld, ipke@techfak.uni-bielefeld.de
- [15] Britta Lenzmann, Ipke Wachsmuth, Yong Cao, "An Intelligent Interface for a Virtual Environment", University of Bielefeld, Faculty of Technology, britta@techfak.uni-bielefeld.de
- [16] Stuart E. Middleton, "Interface agents: A review of the field", Technical Report Number: ECSTRIAM01-001, ISBN: 0854327320, <http://www.ecs.soton.ac.uk/~sem99r>, Intelligence, Agents and Multimedia group (IAM group), University of Southampton.