Improving Skills Management using Objectives within a Multi-Agent System

Olivier Chator
Conseil Général de la Gironde
IMS Laboratory, CNRS, IPB
Bordeaux, France
o.chator@cg33.fr

Jean-Marc Salotti
IMS Laboratory, CNRS, IPB
Bordeaux, France
jean-marc.salotti@ensc.fr

***Abstract*—Whatever the professional sector of activity, achievements are the results of dedicated processes and skill usage. If a project is successful, the knowledge and know-how that have been used have to be capitalized for a possible use in other projects. However, this is a dynamic process. Skills are always evolving. The issue is how to improve the management of skills to make them more efficient over time. A new approach is proposed to address this problem. It is based on the management of objectives through a multi agent learning system where skills are represented as agents. Being autonomous entities, and having their own learning abilities, skills will enforce project efficiency, capitalizing on past iterations. The key point of our approach is the appropriate exploitation of past objectives to help the user identifying the required skills for the new project.**

Keywords-Multi-agent systems; objectives; skills; governance

I. INTRODUCTION

Whatever the goal is (e.g., improve the energy balance of a house by installing photovoltaic panels on the roof or win a rugby match next Sunday), the problem is always stated as doing something to achieve something. Then, the project comes into place. Goals and objectives are statements that describe what the project will accomplish. Each project is defined (structured) by a set of resources and processes according to a specific schedule. The processes are based on a set of required skills. Importantly, the definition of the project may evolve according to the environmental conditions, the past experiences, meaning that the list of skills required to reach the goal may not always be the same. Let us consider the goal of "building a house". Even if the process is almost the same, different construction materials (like cinderblocks or bricks) may be used. This implies different skills for each building project. Considering past experience in the domain, a dynamic dimension is observed for each project over time.

Then, the point is how to improve the management of skills to make them more efficient, throughout projects, and over time. If a user wants to address a new goal, how to define the project and what are the skills needed to implement it? How to get benefits from past projects? In order to solve the problem, it is suggested to work on the goals and skills of past projects using a learning multi-agent system [1] [2] [3], where skills are autonomous agents [4].

Our proposal is based on a "Skill sharing" multi-agent system (MAS). It has already been presented in a previous paper [4] and implements learning abilities that are close to "Case Based Reasoning" mechanisms [13]. In this new paper, Section II starts with a short reminder of the MAS. A new method is then proposed to help the user defining a new project based on previous objectives, objective domains and skills used to implement past projects. The first results are shown in Section III. Section IV concludes this document.

II. MODEL

A. The issue

Let us present one of the missions of "Conseil Général de la Gironde" (CG33). As a local authority, CG33 defines policies and practices for the Sustainable Development (SD) of the "département" (a territorial division lower than regions). In Gironde, 61 of the local authorities are part of an "SD Network". When looking for feedbacks about SD projects, each Network Member (NM) uses an information system, where they share experience and skills [4]. When defining an (potentially new) SD project, each NM needs to identify the skills needed to make his project a success. It is also interesting to get benefits from past experience on similar projects to build the new one. Let us illustrate with the objective: "*I want to put photovoltaic panels on the roof of my house*". I have to find the skills required for this new project. At time T, taking into account my needs and experience of past projects, there are two options:

- I find a past project that reflects exactly what I want to do. Thus, what I need is to find a way to retrieve all the skills of this project, and proceed to my new project creation using this list.
- I find a past project, but it is not exactly what I want to do. Thus, what I need is to retrieve a subset of the skills of this project, and proceed to my new project creation using this restricted list.

The two points above are efficient if the user finds projects that already required all or part of the skills required to implement the new one. However, this is not always the case. Skills may be scattered throughout various projects. Thus, the point is to find a way to answer to a limited expression of needs at specification level. For example, consider we have already in the system the two past projects: "wind turbine implementation" and "hydraulic micro power implementation". They both belong to the same "domain": "new means of energy production". If the user wants to build a new project, in the same domain (e.g., the implementation

 ISBN: 978-1-61208-340-7

of solar panels), an interesting idea is to look for skills used in all the past projects of this domain. Thus, the system will suggest integrating the skills that were fluently used across all projects in the domain. It is possible to generalize this example. Each professional sector of activity has procedures and processes, each of them used to reach specific goals or objectives. Each objective may be declined throughout concrete projects, themselves composed by skills. Generally speaking, it may be difficult to make processes evolve according to environmental constraints. When successful, the knowledge and know-how that have been used should be capitalized for a possible use in further projects. Finally, an important issue is to find a way to improve the management of skills to make projects more efficient over time. To do this, it is possible to build new projects, working on past projects or objective domains. The goal of this work is thus to dynamically improve new projects definitions and to identify the skills needed to make them a success.

B. Working with objectives: theoretical proposal

1) Main concepts

Let us consider an example to illustrate the use of "objective" in our MAS: "*To make energy savings, I want to put a better insulation into the walls of my flat*". This example drives us to the definition of five concepts further used in this paper.

a) Environment

An environment is viewed as a professional sector of activity. In our example, "*sustainable building sector*" is the environment in which the user request occurs. This is the highest level of abstraction and it is related to the professional sector of activity.

b) Objective domain

An objective domain is a group of objectives, concerned by the same thematic of activity. In our example, "*thermal insulation improvement*" is the objective domain in which the user request occurs. Another domain could be "*air tightness improvement*". The idea is to position objectives into one or several objective domains.

c) Objective

An objective is a simple textual description of a goal to reach (the term "goal" would have been more appropriate, but "objective" was chosen from the start for convenient reasons and links with the French language). In our example, the objective is to "*put a better insulation into the walls*". This objective is part of the "*thermal insulation improvement domain*". Considering this simple question from the user, no constraint about materials or skills used to reach the goal is expressed. Another objective could be "*improve air tightness in my flat*". In our implementation, an *Objective* agent is available. To transpose an objective in "real life", it is mandatory to firstly define first a project.

d) Projects

A project is defined by an objective, a start date, an end date, resources (like human actors) and processes to schedule the list of skills to use. In our example, a project defined by the "*integration of glass wool into walls*" is proposed. It will start next week, will stop in 15 days, and requires several skills. Finally, the project is implemented and evaluated at the end using the Elementary Competencies (ECs) of each skill.

e) Skills

As already mentioned in [4], the skill is the ability to exploit some knowledge and know-hows in order to solve a class of problems. It is different from a competency, which is generally accepted as a set of behaviors or actions that have to be successfully implemented within a particular context [8]. Skills are used into projects to reach the goal (the objective).

2) Objectives' type

For a user, the problem is to reach a goal. It is usually defined by a simple assertion like "*I want to do something*". In order to reach the goal, the user will define in our system a new project. He does not often have the knowing of all skills that have to be used into the project. As it is difficult to give a unique answer, depending on the user request at time T, two approaches are proposed to build new projects. The first one is based on completed projects and their objective. The second consists of building the new project using only concerns about the objective's domain, and eventually the environment. This case occurs when the user wants to do actions in a particular domain of activity, but does not exactly know what to do.

C. Skill Agents

1) Integrating feedbacks from experience

As described in a previous work [4], at CG33 and in "real life", numerous SD projects have been realized. To improve future SD projects efficiency, feedbacks show that it is important to enhance human cooperation, and that each actor has only a partial knowledge of the capabilities of the other. An information system, accessible by all actors, offers to each of them a better view of their various expertise areas. Even if the final goal is to know who is able to do something, defining the need as precisely as possible is the key Thus, the proposed information system is centered on skills and not on actors [8]. Furthermore, as skills always evolve, and it is often difficult to explicit them, the proposal is to use a multi-agent system with skill agents. Each of these agents considers human actors as resources, tries to determine its elementary skills over time (self definition), and aims to be involved in SD projects. Generalizing, the challenge is to allow each stakeholder of a project to share and learn more about the expertise and know-how of the others. A traditional approach is to consider direct links between actors and skills, in a "static point of view". Here, the central role of skills, and not of actors, is explicit. Thus, at CG33, an online collaborative skills sharing tool has been elaborated.

 ISBN: 978-1-61208-340-7

2) Definition

It is assumed that a skill is unique and can be implemented as an agent in a multi-agent system. A skill agent is cognitive, non-conversational and non-dialogic [1] [7]. It has resources (a list of physical actors) and its own life cycle. For the sake of simplicity, it has been assumed that a skill is a sum of ECs [8] within projects. Each EC is evaluated after realization and contributes to the learning abilities. Doing so, defining the "embodiment" [5][6] of skill agents, an EC is part of the global skill definition. The skill agent environment is defined by indirect (through another technical agent called *WebRequester*) interactions with the users. It has:

- *Perception*: It listens to information broadcasted by other agents or environmental evolutions, e.g., a new project starts.
- *Internal attributes*: It is defined by a list of elementary competencies, a creation date, a domain(s) of activity, and a specific "*age*".
- *Action*: It updates its behavioral rules and the weight of elementary competencies that define it.

3) Learning mechanisms

The main concerns of the learning mechanism, for a skill agent, are actors' selection (identifying a human person for the embodiment [5][6] of the skill), links establishment with other skill agents, and improving skills management using objectives. The link establishment decision is based on similarity studies, like common involvement into projects over time. Behavioral rules' evolution is a consequence of those learning mechanisms. Skills management improvement concerns will be detailed further.

D. Building new projects from past Projects

Two cases are available to build the new project, duplicating, or customizing existing ones. From our previous example, we will suppose that the user finds the project "*integration glass wool into walls*". One may then:

- Think that it is exactly his objective. Thus, he will duplicate this project and all its skills, without creating a new one, changing only contextual information like the start date for example.
- Think that it is not exactly his objective, but is very close. He decides to create a new project and customizes the list of skills associated to the existing project. A new objective is thus created with a new list of skills built from a subset of the previous one.

Finally, in both cases, a new project instance is generated from the objective. In our implementation, those operations are done throughout our unique *Objective* agent into the MAS. As those actions are based on historical data, no communication with skill agents themselves is needed.

E. Building new projects from an objective domain

Let us assume that a new project has to be built according to the user needs. There is sometimes a limited expression of needs at the specification level and the objective might be met for the first time. The user would probably not know how to address the problem and how to exploit past projects. There is nevertheless a solution to help the user. The idea is to determine all the skills involved in past projects in the same objective domain, or eventually in the same environment. In our system, a skill "***wants to be involved***" in the new project according to a degree of involvement in past projects (see further). In contrast to what we have proposed in paragraph D, this method is not limited to an exchange with the *Objective* agent. In cognitive science, effective agents are obtained by the embodiment of mind [5] [6]. If a skill alone has no perception, no motivation and no means to perform an action and change its environment, it is always possible to define them artificially. Several types of motivations have been integrated in skill agents: contribution to new projects, determine the list of elementary competencies that define themselves [4], and determine their relationships with the other skills [4]. Let us develop an example showing the motivation of being involved in new projects. A user defines a project in the "*thermal insulation improvement*" objective domain. No more detail is forwarded to the system. "*Thermal insulation improvement"* is an objective domain and is part of the environment "*sustainable building sector*". The answer of our system is defined by the following process:

- The *Objective agent* receives the user request.
- The *Objective* agent sends (broadcasts) the request to all skill agents within the MAS.
- Each skill agent computes a "*relevance coefficient"* according to the request content, and returns an answer to the Objective agent (see next paragraph).
- The *Objective* agent consolidates all answers from skill agents, and returns the list of candidates to the user

Each skill agent is autonomous and decides if it wants (or not) to contribute to the new project. The key point is the computation of the relevance coefficient. It is the percentage of projects in which the skill has been involved in the past among all projects of the objective domain. If it exceeds a threshold, the skill agent wants to be involved in the new project.

III. IMPLEMENTATION AND RESULTS

A. The MAS architecture

The model has been implemented using the JADE MAS and standard multi-agent tools [4][9][10][11][12] (see Figure 1, where the focus is on exchange of flows). Synthetically, the Web *Requester Servlet*, *Gateway*, and *WebRequester Agent* are java components used for the management of the exchanges between human users and the MAS itself. The *WebRequester Agent* is in charge of the interactions with the human user, forwarding requests to other agents, and sending back their answers. It guarantees (FIPA compliance [11]) that no direct exchange is possible between human users and skill agents. The *Objective* Agent, according to Ferber's classification, is reactive [1]. When a

 ISBN: 978-1-61208-340-7

new project creation request is received from the user, through the technical agent *WebRequester*, this agent retrieves "existing projects" or sends broadcasts to all skill agents. At CG33, 110 skill agents are active and running into the JADE MAS [4].

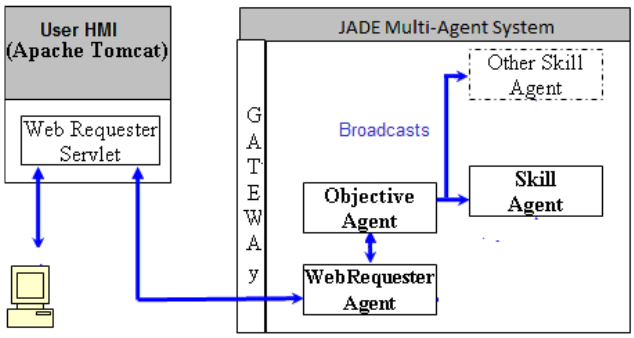

Figure 1. The MAS – Focus on broadcasts

B. Results

1) Real context

As already seen, in Gironde, there are 61 "SD Network" members. They share their experiences and skills through the collaborative "*Sustainable development Skill Sharing System*" application. Skills have been classified into a preliminary list of 9 objective domains: political wishes, sensitization, diagnostic, prospective, developing the strategy, elaborating the action plan, implementation of the action plan, evaluation, and continuous improvement. This application started to be used by the beginning of the year 2013. The first skills concern the management of SD projects. Thus, the results presented below are expressed in this context.

2) Case 1: new projects from past Objectives

The user request, through the *WebRequesterAgent*, is transmitted to the *Objective* agent (see figure 2).

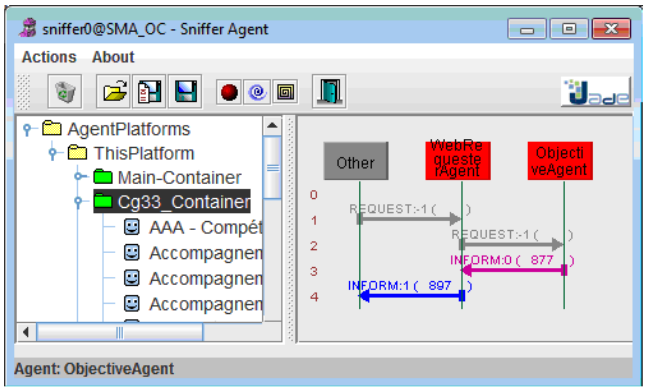

Figure 2. Case 1 - Exchanges between agents into the JADE MAS

Figure 2 is a screen copy of the JADE MAS Sniffer tool that monitors message exchanges among agents. The left part shows the MAS Agent tree where an agent is part of a container, a container belongs to a platform (*ThisPlatform*) and a platform is included in all agent platforms (*AgentPlatForms*). The right part shows three "boxes":

- *Other*: reflects other agents within the MAS, or in this case, the JADE gateway that manages exchanges with the external users
- *ObjectiveAgent*: see next paragraph for details.

The arrows (1 to 4) show the message exchanges, with their type (REQUEST or INFORM for the answer), and their directions (from sender to receiver).

The *ObjectiveAgent* ensures the treatments, based onto historical data, to retrieve projects instances and related skills. At the end, it processes the answer by means of an XML flow (see Table 1 of this flow).

TABLE I. LIST OF THE XML FIELDS INTO THE ANSWER FLOW

XML Tag	Comment
answers	Main tag encapsulating the answers
answer	Main tag for each answer, 1 for each project
newObjective	Boolean value indicating that the project is new. Value always false here because the project is over and taken into the historical
objective	Main tag for the past project
code	The past project code within the projects database table
description	The past project textual description within the projects database table
startDate	The past project start date
endDate	The past project end date
skills	Main tag encapsulating the skills list
skill	Main tag, 1 per skill
code	The current skill code within the skills database table
description	The current skill textual description within the database table

At the front-office user level, a list of projects and skills is proposed. The new user project is then generated according to one of the 2 identified methods: duplicate, or customize.

3) Case 2: new projects from objective domain

The user request is sent to the *ObjectiveAgent* that does a broadcast to all skill agents, consolidates their answer. At the end, the answer is also returned to the user as an XML flow (see Case 1 for details). Please note that a level of "genericity" for the broadcasting mechanisms is introduced (see figure 3).

 ISBN: 978-1-61208-340-7

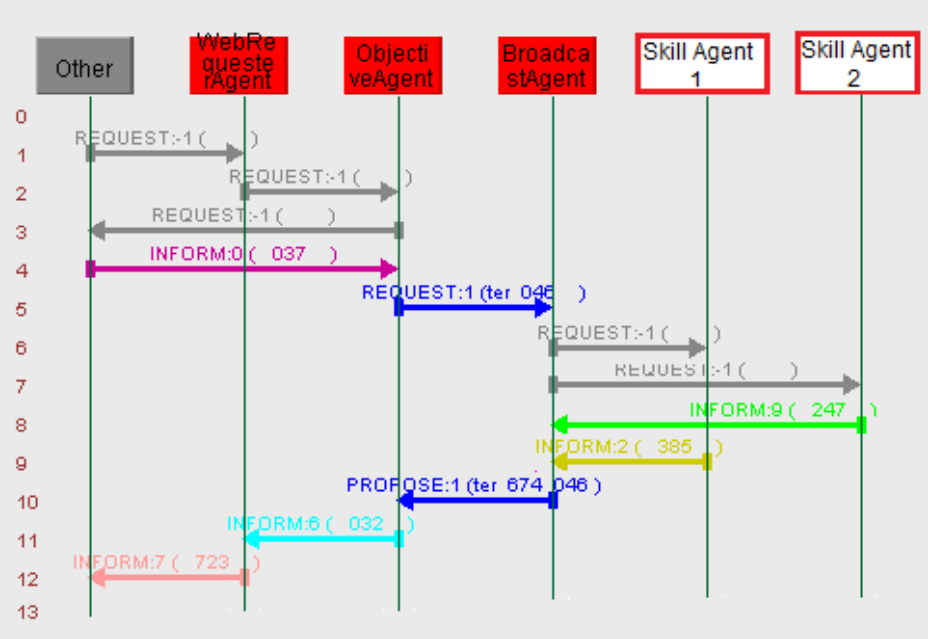

Figure 3. Case 2 - Exchanges between agents into the JADE MAS. See text.

Figure 3 is also a screen copy of the JADE MAS Sniffer tool. Only the right part of the window is shown to improve the visibility. The first three "boxes" are identical as those described for figure 2. Please note that *BroadcastAgent* is a technical agent. It receives a message from an original sender, broadcasts it to all agents, and consolidates all answers into a single one. The global answer, an XML flow, is then sent back to the original sender. *All other boxes on the right* are all skill agents within the MAS. The screen copy above shows only two skill agents.

For illustration, let us reuse the example where the objective domain is "*new means of energy production*". A project for a "*wind turbine implementation*" has 5 phases:

1. *Project management / implementation*
2. *Definition of power requirements*
3. *Selection of the best technology of wind turbine among models and worldwide suppliers*
4. *Logistic definition (transportation organization)*
5. *Wind turbine installation*

A second project, entitled "*hydraulic micro power implementation*", shares 3 (over 5) common phases with the first project, and has 5 *specific ones*:

1. *Project management / implementation*
2. *Definition of power requirements*
3. *Selection of the best technology of hydraulic micro power among models and worldwide suppliers*
4. *Logistic definition (transportation organization)*
5. *Hydraulic micro power installation*
6. *Technology transfer of appropriate designs to developing country manufacturers*
7. *Project formulation and appraisal for national and international aid agencies*
8. *Training on small-hydro technology and economics*

A third one, entitled: "*installation of a biodiesel generation system to power up a highway construction site*". Let us assume the project also shares the 3 common phases, and has *4 specific ones*:

1. *Identify the best site*
2. *Environmental benefits evaluations*
3. *Project management / implementation*
4. *Definition of power requirements*
5. *Selection of the best technology of biodiesel generation system among models and worldwide suppliers*
6. *Logistic definition (transportation organization)*
7. *Biodiesel generation system installation*

Considering these phases as skills (of course at a high level of abstraction), let us introduce into the system a new user request where the new project is "*solar panels implementation*". This project also belongs to the objective domain "*new means of energy production*". The requested minimum value for the relevance coefficient is 75%. Thus, according to our algorithm:

- The *ObjectiveAgent* send (broadcasts) the user request to all the skill agents within the MAS (here 3 common + 5 specifics according to the second and third project phases)
- Each skill agent computes its dedicated relevance coefficient. If this computed value is greater than the requested one, that means the skill agent "wants" to

 ISBN: 978-1-61208-340-7

contribute to the new objective, and the boolean value "*true*" is returned to the *ObjectiveAgent*

- The *ObjectiveAgent* consolidates the results where the answer is "*true*". Finally, it returns to the user the list of the skills as an XML flow:
  a - Project management / implementation
  b - Definition of power requirements
  c - Logistic definition (transportation organization)
- Through the front-office interface, the user then validates (partially or totally) the skills list to build its new project. This validation is stored into memory.

C. Discussion

The proposed information system proposes solutions to the problems of each SD Network Member at CG33, who needs to identify the required skills to make his project successful. Whatever the activity sector, project management is usually carried out through software applications, where tasks are defined and described in a static way. The definition of a new project requires the identification of human actor(s) for each of them. Two problems may be encountered:

- The first problem occurs when an evolution of the processes is considered. A traditional approach is to statically update the list of tasks for the project. Setting renewals have to be done by administrators, or advanced users, into the project management tool itself. This update is often generating costs, because in some cases an external help (e.g., of the software editor) is required. This way, there is a lack of efficiency, inducing at least a waste of time, and sometimes substantial costs overruns.
- Another problem is the management of the dynamic nature of projects according to the evolving needs of users. As a project reflects the user needs at a given date, there can be as many projects as user needs expressions within the system. The global skill sharing system presented in this paper is a collaborative and adaptive tool. The new projects are built "on the fly", from the real user needs. Thus, the list of new skills is built from those available in past projects and reflects the user needs at the time of the request. As the number of projects grows in our collaborative system, the global list of skills and the objective domains evolve and may converge. At a global level, our system learns from the requests of the users and reflects the evolution of activity over time. The observation of those evolutions will drive the management of the company to put the focus or to drop certain skills. Our proposal provides, therefore, an interesting solution to the problem of skills management, using objectives, at the project and organization levels.

IV. CONCLUSION AND FUTURE WORK

Over time, more and more objectives will be concretized through projects, and more and more information will be available to help the user. This work suggests interesting perspectives. From a professional point of view, concerning the problem of skills management, the analysis of skills applications through projects provides a wealth of information. After a period of running, managers and human resources management services will be able to identify the key skills, the cross-domain skills, the evolution of "sensibility" of each skill into the professional sector processes, and all of this over time. This work may introduce real benefits in human resources management to anticipate future evolutions of needs in terms of collaborators profiles.

In our proposal, the adaptation over time is made possible by the computation of a relevance coefficient value. This value is stored into the memory of each skill agent. One of the future directions of works is to implement complementary learning mechanisms to optimize the computation of this value over time.

REFERENCES

[1] J. Ferber, "Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence", Addison Wesley, London, UK, 1999.

[2] N. R. Jennings, M. Wooldridge, and K. Sycara, "A roadmap of agent research and development. Int Journal of Autonomous Agents and Multi-Agent Systems", Kluwer Academic Publishers, Boston (USA), 1998, pp. 7-38.

[3] L. Shapiro, "Embodied Cognition", in Oxford Handbook of Philosophy and Cognitive Science, E. Margolis, R. Samuels, and S. Stich (eds.), Oxford University Press, 2010

[4] O. Chator, J. M. Salotti, and P. A. Favier, "Multi-agent System for Skills Sharing in Sustainable Development Projects", proceedings of COGNITIVE 2013, the 5th International Conference on Advanced Cognitive Technologies and Applications, IARIA Conference, Valencia, May 27-June 1, 2013, pp. 21-26,

[5] F. Varela, E. Thompson, and E. Rosch, "The Embodied Mind: Cognitive Science and Human Experience", MIT Press, 1991.

[6] L. Shapiro, "Embodied Cognition", in Oxford Handbook of Philosophy and Cognitive Science, E. Margolis, R. Samuels, and S. Stich (eds.), Oxford University Press, 2010

[7] G. Weiss, "Multiagent Systems, a Modern Approach to Distributed Artificial Intelligence", MIT Press, 2013.

[8] M. Mulder, T. Weigel, and K. Collins, "The concept of competence in the development of vocational education and training in selected EU member states: a critical analysis", Journal of Vocational Education and Training , vol. 59, 2007, pp. 67-88.

[9] G. Rocher, "The Definitive Guide to Grails", Apress, New York, USA, 2006.

[10] S. Goderis, "On the separation of user interface concerns: A Programmer's Perspective on the Modularisation of User Interface Code", Ph.D. thesis, VrijeUniversiteit Brussels, Belgium, 2007.

[11] FIPA, "Foundation for Intelligent Physical Agents - Abstract Architecture Specification (Standard Version)", publication des Technical Committee de la FIPA, Geneva, Switzerland, 2002.

[12] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE – A FIPA-compliant agent framework", 4th International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-99), London (UK), 1999, pp. 97-108.

[13] J. L. Kolodner, "Case-Based Reasoning", Morgan Kaufmann Publishers, San Mateo, USA, 1993

 ISBN: 978-1-61208-340-7