

# Multi-agent System for Skills Sharing in Sustainable Development Projects

Olivier Chator

Conseil Général de la Gironde  
IMS Laboratory, UMR 5218, ENSC/IPB  
Bordeaux, France  
o.chator@cg33.fr

Pierre-Alexandre Favier, Jean-Marc Salotti

Ecole Nationale Supérieure de Cognitique  
IMS Laboratory, UMR 5218, ENSC/IPB  
Bordeaux, France  
pierre-alexandre.favier@ensc.fr, jean-  
marc.salotti@ensc.fr

**Abstract**— A local authority, the “Conseil Général de la Gironde” in France, works in the field of Sustainable Development and coordinates public and private partners. Sharing skills between them is one of the identified problems. Another difficulty is that nobody has a global vision of all know-hows of each partner. This work addresses these problems by building a collaborative multi-agent system called “sustainable development skills sharing”. One of the innovations of this framework is that skills are represented as agents, not just as capabilities, as it is usually the case.

**Keywords**—Multi-agent systems; sustainable development; skills; governance

## I. INTRODUCTION

A local authority, the “Conseil Général de la Gironde” (CG33) is responsible for public actions for 1.5 million inhabitants. Numerous domains are concerned: school transportation, management of middle schools, tourism development, solidarity, integration and support for elderly people. One of the CG33 missions is to define policies and practices for the Sustainable Development (SD) of the department (a territorial division lower than regions). For example, the objective could be to transform a neighborhood into an eco-district [1]. Experience shows that this type of project is very complex and requires the collaboration of many public and private actors under the supervision and management of a “maître d'oeuvre” (MO), for instance an architectural firm. Each actor has only a partial knowledge of the capabilities of the other and some information is sometimes lacking, but the MO has to take decisions anyway. In addition, the objective is often to minimize the costs and to obtain energetic or ecological labels, which are typically antagonist objectives. For the MO, it is often difficult to understand the impact of each parameter. The preferred option is usually the one that is better understood at the expense of other options because there was insufficient knowledge on their impact, cost, and implementation. In order to help the actors, and especially the MO, to find the best partners, the CG33 decided to build a database of skills and actors. For example, it should be possible for a MO who wants to renovate some buildings to identify skills and actors in various domains such as thermal insulation, thermal simulation, air tightness, and installation of different types of photovoltaic panels on the roof. In turn, the partner who has an expertise in thermal insulation may require the help of

another partner who is specialized in the use of specific insulation materials. Thus, the challenge is here to allow each stakeholder of an SD project to share and learn more about the expertise and know-how of the others.

A traditional approach could be to build a simple database with a direct link between actors and skills. However, considering the central role of skills and the needs for constant evolution and modifications of the data, a research project has been carried out in our laboratory to find and implement a better solution. It is suggested here that a multi-agent system (MAS) is more appropriate [2, 3, 12].

The model is described in Section II. The first results are shown in Section III and Section IV concludes this document.

## II. MODEL

### A. Problematic

Let us introduce the problematic with a concrete example of an SD project that aims to “transform a neighborhood into an eco-district”. Let us assume that a MO has to build a HQE building. HQE stands for “Haute Qualité Environnementale” and is a standard for green building in France [13]. For this kind of project, the MO needs:

- A definition of the objective.
- Skills such as “integrating insulation materials” in order to meet the HQE objectives.
- Actors such as private building companies to implement the skills.

Using our “SD skill sharing” system, the MO should be able to identify a list of possible partners. Intuitively, we could think that this list could be simply sorted according to the most experimented partners for the given task. However, other criteria should be taken into account than just experience: price, quality, duration, localization, expertise with specific materials, etc. The system may suggest a partner according to this list of criteria. In addition, it also has to select different companies over time. The problem is to determine a good strategy to achieve that goal.

### B. Defining a Skill

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 needed to successfully be

performed within a particular context [4]. In this study, for the sake of simplicity, it is assumed that a skill is a sum of elementary competencies.

The main specifications of our application are to store information about the skills of possible participants to SD projects and to suggest interesting partners for a given skill. An important issue is to make the link between observations (e.g., “partner A has been assigned the role of task 1 and 2 in project X and succeeded in implementing solutions”) and skills, which do not correspond to the names of the task. Let us present an example:

Integrating glass wool for the insulation of northern walls in a specific building in a given project is related to the skill “integrating insulation materials”. However, integrating isolated wood panels under roofs might be very different from integrating glass wool in walls and the best expert for the first task might not be the best one for the second. The skills might be differentiated by small details, but, for the proposed application, it would be irrelevant. It is expected that the users of the application will ask general questions such as “who has skills in insulation materials”. The key problem is to find the appropriate level of details for each skill and to make the difference between an elementary competency that belongs to a skill and the skill itself. Then, assuming that a skill is defined at the right level and includes a list of possible elementary competencies, the question is to determine how each of them participates to the definition of the skill. For instance, for the skill “integrating insulation materials”, how important it is to have the know-how for isolated wood panels? In other words, there should be an associated weight for each elementary competency and there should be a mechanism integrated in the skill agent to learn them.

According to the needs of the project, a skill can be created at any moment, its definition (the list of elementary competencies) may evolve, it can eventually be split in different skills and it might even be removed. Such constraints cannot easily be handled in a standard database in which the actors and their skills would be stored. Because of the central role of the skills, it is suggested in this paper to consider the skills as agents of a multi-agent system. However, in most applications, agents are associated to models of actors in the real world, the skills defining the behavioral rules [2, 16]. The problem is that the skills have their own dynamics and are rather independent from the actors. The skills should be agents with their own life. In addition, if the skills and the actors are distinguished, it is difficult to define actors as other agents of the system. In cognitive science, the embodiment of mind is often considered a requirement to obtain an effective agent [9, 10, 11]. Skills alone have no perception, no motivation and no means to perform an action and change their environment. Nevertheless, it is possible to define these elements artificially. Intuitively, a skill can be motivated by the improvement of its own definition, e.g., a weighted list of elementary competencies and the clarification of its relationships with the other skills. The user of the system has another motivation: he wants to find a partner for his project. The system should provide some criteria and suggest an

actor for the required skill. The user makes his choice, then the work is carried out (embodiment of the skill) and an evaluation of the realization is performed. The key idea is to consider that a criterion is no more than an abstraction of a hidden list of elementary competencies. For example, the duration of a work is not a competency. However, implicitly, it is closely related to the ability to work fast, which is an elementary competency of the skill. Therefore, the skill can exploit the definition of the criteria, which evolve according to the evolution of the projects, to characterize its definition. Another issue concerns the links between the skills. Different skills may have several elementary competencies in common. If no actor is found for a given skill, an interesting idea is to make suggestions with actors associated to the skills that are closely related.

In addition, a database is still required for the storage of past observations (e.g., Actor A has been involved in project X for the embodiment of skill S with an evaluation of a list of criteria  $C_1..C_n$ ).

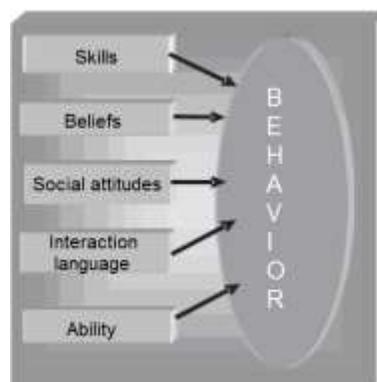


Figure 1. Agent's behavioural characteristics.

The details of our proposal are given in the next sections.

### C. Skill Agents: theoretical proposal

It is assumed that a skill is unique and can be implemented as an agent in a multi-agent system. It has resources (a list of physical actors) and its own life cycle. It can be created, can evolve and eventually be removed when not used anymore or replaced by another skill agent. Skills agents fit in a multi-agent system, where the environment is defined by the interactions with the users. They are cognitive, non-conversational and non-dialogic [2, 12]. They never directly communicate with human users. They react and evolve according to information modifications and requests from the user via a WebRequester agent. An important feature is their ability to learn how to define themselves and how they are linked to the other skills. Skill agents are defined by three main parts: perception, internal attributes and actions:

- Perception: Skill agents are listening to information broadcasted by the system after interaction with the users. It can be, for instance, an update after external observations (e.g., a new project is started or the result of a work for a given project is inserted in the

database) or a request is sent by another agent within the MAS.

- Internal attributes: A skill agent is determined by the list of elementary competencies that defines the skill, a creation date (appearance into the MAS), a domain(s) of activity and a specific “age” (see further).
- Action: If there is an update of an external observation that is linked to the skill, the agent updates its database and its weights according to a learning rule. It provides an answer to the WebRequester agent (which, in turn, informs the user) according to a strategy defined by behavioral rules.

#### D. Learning mechanisms

##### 1) Actor selection

When the user asks the system to suggest an actor for a given skill, a list of criteria is presented to him with undetermined weights. For instance, if the user wants to know who has the best skill in “thermal insulation”, the system asks the user to define the weight associated to each criterion “wall insulation”, “roof insulation”, “wood based materials”, “diagnostic”, “price”, “duration”, etc. This information is used to update the definition of the skill. Let  $W_{1,t}..W_{n,t}$  be the list of weights associated to  $n$  criteria  $C_1..C_n$  for request number  $t$ . The skill is defined by a weighted list of elementary competencies (the criteria). Let  $E_1..E_n$  be the weights associated to them. They are calculated according to equation (1).

$$E_k = \frac{\sum_{i=1}^t W_{k,i}}{t} \quad (1)$$

Once the request is correctly specified, the agent returns a sorted list of possible actors. The notation is based on the evaluation of the work after its realization. Let  $A_{k,i}$  be the average evaluation of actor  $k$  for criterion  $i$ . For  $m$  actors, let  $A_{1,1} .. A_{m,n}$  be the past evaluations of the actors for each criterion associated to the skill. Each actor  $k$  is then evaluated for the new request according to equation (2).

$$Eval(k) = \frac{1}{\sum_{i=1}^n A_{k,i}} \sum_{j=1}^n A_{k,j} W_k \quad (2)$$

The first term allows a normalization of the notation, such that an actor with systematic lower evaluations is not penalized. In any case, the user can still select an actor who is not at the top of the list.

##### 2) Life cycle

Each skill agent has a “life cycle”. It is divided into 3 states called “ages”.

###### a) Childhood

The agent runs in a “learning” mode. During this age, the aim is to make the agent “grow”. When an agent is created, a list of criteria is assigned to it. A list of actors is also

associated to the agent but there is no evaluation in the database. If a user looks for a partner with this skill, the agent is not able to make relevant suggestions (childhood). It simply returns a list of potential actors ranked according to the number of times they have been involved in realizations (the most experimented at the end). Once the result of the work is available, the user evaluates the criteria associated to the skill and the data are stored in the database.

###### b) Teenage

When three evaluations are available in the database, the agent grows to the teenage age. If a user looks for a partner with this skill, the agent exploits the previous results to propose a list of actors sorted according to the criteria and their weights (equation (2)). If no data is available for an actor, default values are used.

The autonomy of the agent is rather limited. It communicates and tries to build relationships with other agents (see section II.D.3).

###### c) Mature

When the agent has accumulated sufficient knowledge (10 evaluations are available), it is able to make direct suggestions to the user as soon as a project is created. A list of skills is proposed with possible actors. Obviously, the user can still make modifications but he can save a lot of time if the choices correspond to his needs. In addition, the agent has a good knowledge of its relationships with the other agents (see next section).

##### 3) Learning agent links

A user who is not experimented may not necessarily know all the skills that are required for the realization of a project. He can ask the system to help him by the suggestion of a list of skills that may be of interest. In order to provide a relevant answer, it is possible to exploit the links that can be found among the skill agents that are at least in the teenage age. The search is based on similarities. Different features are examined: the domain of activity, the number of common elementary competencies and the number of projects in which the same skills have been involved. For each skill agent and each feature, a “proximity coefficient” is calculated. The list of linked agents is then returned to the user for validation (or not). Importantly, when a skill agent is in the mature age, no human validation is needed. Thus, the agent is completely autonomous, generates links according to its existing behavioral rules.

### III. IMPLEMENTATION AND RESULTS

#### A. Implementation: main points of interest

##### 1) Architecture

The model has been implemented using the JADE MAS and standard multi-agent tools [5, 6, 7, 8].

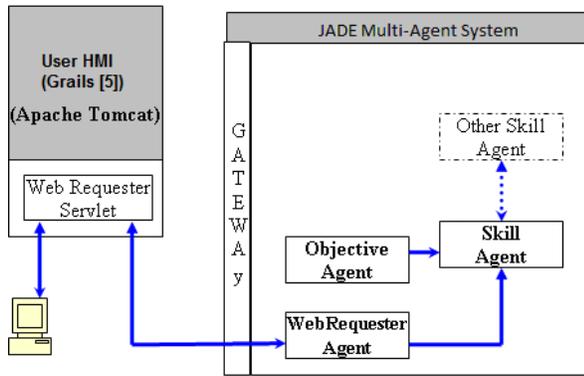


Figure 2. The MAS.

The main components of the MAS are presented in Figure 2:

- *User workstation*: exchange using a web browser.
- *WebRequester Servlet*: This java component is used for the management of the exchanges between human users and the MAS itself.
- *Gateway*: It is a standard component of a JADE MAS standard component allowing dialogues among agents operating within the SMA and external programs (WebRequester Servlet) [8].
- *WebRequester Agent*: This agent is in charge of the interactions with the human user, forwarding requests to skill agents and sending back their answers. It guarantees (FIPA compliance) that no direct exchange is possible between human users and skill agents.
- *Objective Agent*: According to Ferber’s classification, the objective agent is reactive [2]. When a new project with a new objective is inserted within the system, information messages are broadcasted to all skill agents.
- *Skill Agents*: already presented.

2) Behavioral rules

The behavioral rules have been implemented in XML with a specific grammar (hierarchy, attributes, tags). See Table I for their description.

TABLE I. LIST OF THE BEHAVIORAL RULES

| XML Tag   | Attribute   | Mandatory | Comment                                      |
|-----------|-------------|-----------|--|
| rules     | description | X         | Main tag                                     |
| ruleGroup | description |           | Text describing the rule group               |
|           | weight      |           |  |
| rule      | description | X         | Text describing the rule                     |
|           | weight      |           | Weight of the rule in the rule group         |
|           | mandatory   |           | Value is 1 if rule is mandatory, 0 otherwise |
| when      | description |           | Text describing the condition                |
|           | sensor      |           | Sensor java class name used to verify rule   |

|           |             |   |  |
|-----------|-------------|---|--|
|           |             |   | condition  |
|           | params      |   | Parameters in format name=value, separated by   character. Passed to the sensor    |
|           | result      |   | Result variable name beginning with \$   |
|           | operator    | X | Logical operator used within condition expression                                  |
|           | table       |   | Table name from which we try to verify condition                                   |
|           | field       |   | Table field name or variable name from which we try to verify condition            |
|           | value       |   | Value of field attribute, expressed as a regular expression                        |
| otherwise |             |   | Used if <when> has not been verified   |
| do        | effector    | X | Effector java class name to start if rule condition is verified                    |
|           | description |   | Text describing the action   |
|           | params      |   | Parameters in format name=value, separated by   character. Passed to the effector. |
|           | result      |   | Result variable name, beginning with \$  |

3) Skill agent memories

Each skill agent owns a dedicated memory table in which it stores the incoming parameters and the related computed decisions. See Table II for a description of what is stored in memory.

TABLE II. MEMORY TABLE OF A SKILL

| Field          | Type         | Comment   |
|----------------|--------------|---|
| code           | Long integer | Memory unique key code  |
| ev_date        | Date         | Event date (record creation date)                                       |
| evt_id         | Long integer | Foreign key to event type table, describing the type of memorized event |
| agentid        | Long integer | The current skill agent unique id                                       |
| parametersin   | String       | Request parameters list in string format                                |
| decisionstring | String       | Computed decision in string format                                      |
| Humanvalidate  | Boolean      | Decision validated (or not) by human action                             |
| Comment        | String       | Free field  |

B. Results

In Gironde, 61 of the local authorities are part of an “SD Network”, where they share experiences and skills. They started to use and test the system by the beginning of the year 2013. The experiment concerned the management of SD projects. A preliminary study has been carried out, showing that most projects fall into 9 domains of activity. These domains are: political wishes, sensitization, diagnostic, prospective, developing the strategy, elaborating the action plan, implementation of the action plan, evaluation, and

continuous improvement. Skills are related to one or more domains. For instance:

- The skill “animation capability” is attached to the “political wishes” and “diagnostic” domains.
- The skill “identification and mobilization of expertise” is attached to the “prospective” domain.
- The skill “development of the sustainability report” is attached to the “continuous improvement” domain.

Currently, even if the experiment has just started, it is observed that the skills evolve in the system and provide answers. The positive point is that the skills provide valuable information to the actors who have poor understanding of the elementary competencies. The drawback, however, is that the initialization of the system is fastidious. The first definition of the skills requires a strong expertise of the domain. The updates can be done at any time, but it takes a long time to collaborate with experts in order to capitalize their knowledge and insert relevant skills and elementary competencies in the system. Therefore, it is difficult at the moment to conclude about the efficiency of our model because we are still in the early stages of the tests. We hope to present interesting results in a near future.

In order to demonstrate the versatility of our proposal, other tests have been performed using another functional domain: the selection of the best players for rugby. In this application, each player's position is considered a different skill. Elementary competencies are for instance the ability to tackle and stop an opponent or to be accurate in the shoot of the ball. The evaluation of a player for the embodiment of a given skill is based on his performance for each criterion and on the number of selections. When the system is asked to suggest a player for a given skill, equation (2) is used. Then the propositions elaborated by each skill agent are validated (or not) by the user, the players are evaluated and the database is updated. The results are positive for the identification of players over the different iterations.

### C. Discussion

In most SMA applications, the skills are not agents. They are typically described by behavioral rules that determine the actions of the agents [16]. The difficulty is often to make the link between tacit and explicit knowledge and to learn from the real world [14, 15]. For instance, in other applications such as the management of skills in the context of e-learning, one of the main problems is to determine and explicit the tacit knowledge that has not been understood and to adapt the courses [14]. The advantage of our approach is that it is skill centered. The skills are learning agents and their motivation is to determine the list of elementary competencies that define themselves and their relationships with the other skills. These elementary competencies usually correspond to tacit knowledge and know-hows that cannot be easily defined. One of the key ideas of our model is to consider that the weighted list of criteria defined by the users to determine the best actor for a given skill are abstractions of a hidden list of elementary competencies. The system learns from the requests of the users.

## IV. CONCLUSION AND FUTURE WORK

A multi-agent system has been proposed for skills sharing between actors in collaborative projects in the domain of sustainable development. The key point of this work is the definition of skills as agents with their own rules for learning and evolving in an environment where actors are considered resources for the embodiment of the agents.

Several issues have been identified for future works. The current tests are preliminary. The system has to be tested with a comprehensive list of skills and elementary competencies provided by experts of the domain. A large number of evaluations is also required to test the evolution of the agents at different ages.

## REFERENCES

- [1] M. Bardou, “De la stratégie à l'évaluation : des clés pour réussir un Agenda 21 local” (“From the strategy to the assessment: the keys to succeed in the elaboration of a local Agenda 21”), Collection “Références” du Service de l'Économie, de l'Évaluation et de l'Intégration du Commissariat Général au Développement Durable, Paris, France, 2011.
- [2] J. Ferber, “Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence”, Addison Wesley, London (UK), 1999.
- [3] 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.
- [4] 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, 1, pp. 67-88, 2007.
- [5] G. Rocher, “The Definitive Guide to Grails”, Apress, New York (USA), 2006.
- [6] S. Goderis, “On the separation of user interface concerns: A Programmer's Perspective on the Modularisation of User Interface Code”, Ph.D. thesis, Vrije Universiteit Brussels, Belgium, 2007.
- [7] FIPA, “Foundation for Intelligent Physical Agents - Abstract Architecture Specification (Standard Version)”, publication des Technical Committee de la FIPA, Geneva, Switzerland, 2002.
- [8] 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.
- [9] R. Pfeifer and J.C. Bongard, “How the body shapes the way we think: a new view of intelligence”, The MIT Press, 2007.
- [10] L. Shapiro, “Embodied Cognition”, in Oxford Handbook of Philosophy and Cognitive Science, E. Margolis, R. Samuels, and S. Stich (eds.), Oxford University Press, 2010.
- [11] F. Varela, E. Thompson, and E. Rosch, “The Embodied Mind: Cognitive Science and Human Experience”, MIT Press, 1991.
- [12] G. Weiss, “Multiagent Systems, a Modern Approach to Distributed Artificial Intelligence”, MIT Press, 2013.
- [13] ADEME, “Bâtiment et démarche HQE” (“HQE building and methodology”), Collection Connaître pour agir, Paris, France, 2006.
- [14] A. Garro, L. Palopoli, “An XML Multi-Agent System for e-Learning and Skill Management”, Lecture Notes in Artificial Intelligence, 2592, Springer-Verlag, 2002.
- [15] S. Belkade, A. I. Cristea, T. Okamoto, “Measuring Knowledge Transfer Skills by Using Constrained-Student Modeler Autonomous Agent”, Proceedings of the IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society, 2001.

- [16] H. Friedrich, O. Rogalla and R. Dillmann, "Integrating skills into multi-agent systems", *Journal of Intelligent Manufacturing*, vol. 9, 2, pp. 119-127, 1998.