MULTI-AGENT SYSTEMS AND ENTERPRISE MODELING

G. Pepiot[1], R. Glardon[1], M. Pouly[1]
[1]*Laboratory for Production Management and Processes*
Department of Mechanical Engineering
Swiss Federal Institute of Technology at Lausanne
CH-1015 Lausanne, Switzerland
E-mail: gregoire.pepiot@epfl.ch

Manufacturers operate today in constantly moving market, characterized by a short product life cycles and increased demand for flexibility. In the same way, the ability to respond rapidly to changing market opportunities by quickly providing the required combination of design and manufacturing capabilities becomes a key issue to maintain competitiveness. In this context, the project oriented enterprises may be modeled by a multi-agent system representation. Such an approach allows modeling the heterogeneous actors of the real world in terms of agents, focusing on communication and collaboration between them. This paper shows how a multi agent approach may be used in the modeling of project oriented enterprises.

1. INTRODUCTION

Short product life cycles and increased demand for flexibility characterize production constraints today. This is a true challenge for the whole company and particularly for the production and logistics departments. In order to cut response time to customer needs and increase flexibility, enterprises need to base their organization around know how and capabilities of individuals or groups of employees as well as on technological performances of other resources (machines, computers, etc.) These human and machine resources and capabilities must be frequently rearranged and grouped in different ways, depending on the changing customer needs. This is particularly true in project oriented enterprises in which capabilities must be regrouped on an order-by-order basis. Access to updated and accurate information is essential for reaching this goal with quality and efficiency. Information must also be adapted to the task to be accomplished. There should be neither more nor less information than necessary. In such situations the enterprise becomes closer to horizontal network of resources and capabilities that need to exchange information and communicate in a project specific way than to a vertically, functionally oriented organization. Rigid, functionally oriented models supporting software systems that are difficult and expensive to reconfigure are not adequate. Multi-Agent systems offer the potential of modeling flexible and heterogeneous organizations[1]. This paper shows how a multi-agent approach could help to bring the required flexibility in the modeling of project oriented enterprises. It shows, focusing on communication and collaboration between agents, how such an approach facilitates modeling the heterogeneous actors of the real enterprise world in terms of agents.

2. MULTI-AGENT SYSTEMS

2.1 Agent definition

The concept of an agent seems to be one of the most suitable formalisms for modeling project oriented enterprises; even if this concept is not clearly defined: As stated in [2] *"The overuse of the word "agent" has tended to mask the fact that, in reality, there is a truly heterogeneous body of research being carried out under this banner "*. As a matter of fact, there are many different concepts: MuBot Agent, AIMA Agent, Maes Agent, KidSim Agent, Hayes-Roth Agent, IBM Agent, Wooldridge & Jennings Agent, SodaBot Agent (MIT AI Lab), Foner Agent, Brustoloni Agent, etc... [3]. The concept of agent considered in this work is best defined according to Wooldridge and Jennings[4] who state that an agent is a hardware or (more usually) software-based computer system that enjoys the following properties:

- *autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;*
- *social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language;*
- *reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;*
- *pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative. "*

Let us notice that an agent can, in its simplest version, be reduced to a wrapper, that is, an entity which allows encapsulation and communication of heterogeneous software. In summary, it can be said that an agent is an entity which owns skills, is autonomous and concurrent (since it is executed in parallel with others agents), communicates by means of protocols that may be complex, and which cooperates by inserting itself into a population of agents.

2.2 The communication among agents

An agent in a multi agent system must be able to collaborate with other agents (software and humans) .

- Collaboration is a process in which two or more participants coordinate their actions towards achieving shared goals. **Figure 1.** represents the structure of a generic collaboration between a software agent and a human user.

Collaboration generally requires communication and knowledge sharing.

- Communication is the exchange of information between agents (in **Figure 1.**) Between a software agent and a human user). It includes observations and/or reporting of their respective actions, as for example, the message "I have done action X".
- Discourse is a technical term for an extended communication between two or more participants in a shared context, such as a collaboration.

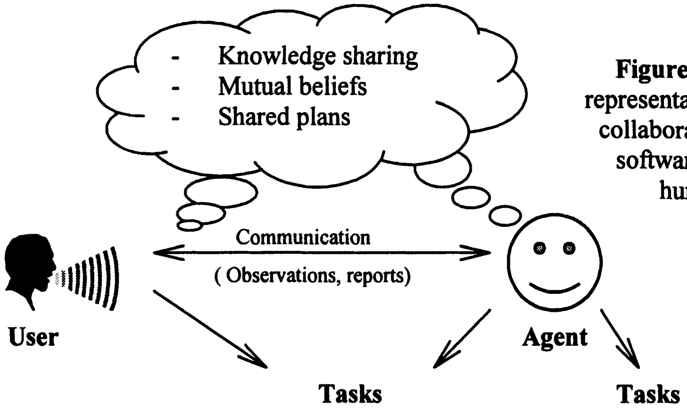


Figure 1. Schematic representation of a generic collaboration between a software agent and a human user.

- Knowledge sharing is a prerequisite for a successful collaboration. It can be characterized by the following main conditions that must be fulfilled by the communicating agents which should:
  - -have a common goal
  - -have agreed on a sequence of actions to reach this common goal
  - -each be capable of performing their assigned actions
  - -intend to do their assigned actions
  - - be committed to the overall success of the collaboration

All these exchanges rely on an Agent Communication Language (ACL) between agents, such as KQML for Knowledge Query Manipulation Language[5].

2.3 The specific agent used

In this work, a specific agent based on the PVS98 agent has been used. The PVS98 agent has recently been developed for application mainly in the field of production planning and control and enterprise modeling [6]. It relies on a KQML-like language for communication. Every agent belongs to an agent class and the different agent classes are organized into a hierarchy allowing inheritance between them. So, every agent type inherits from the predefined *Agent* class and in particular inherits the attributes*: **name, type, competences, goals, state and properties.*** The root class "agent" defines the common characteristics for all agents.

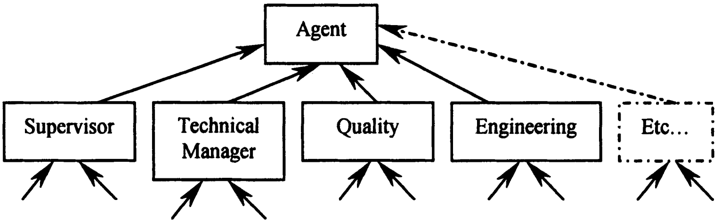


Figure 3. Classes Definition

The agent used in this work is an entity which owns:

- **<u>dynamic knowledge</u>** called skills, represented by programs. The agent has therefore the ability to encapsulate one or several programs inside its structure

- **static knowledge** divided into attributes, relations, a local knowledge base and a local ontology
- a control procedure in order to manage its different knowledge, the received messages and the **negotiation schemes** (collaboration).

The proposed agent uses the formalism illustrated in **Figure 2.** It is made up of three main parts: a knowledge part (shaded in **Fig. 2**) a communication part and a cooperating part.

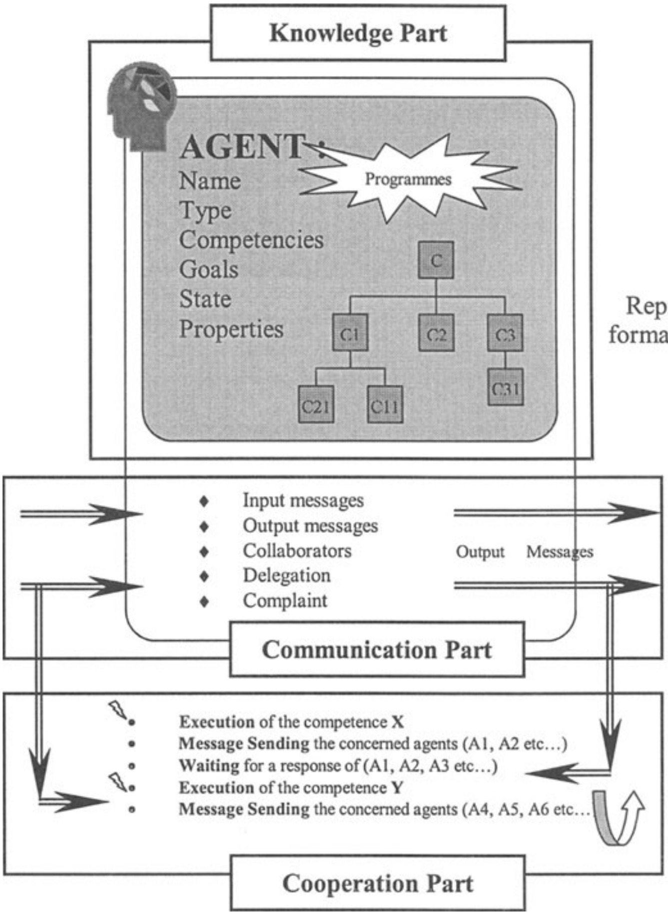

Knowledge Part
AGENT
Name
Type
Competencies
Goals
State
Properties
Programmes
C
C1
C2
C3
C21
C11
C31
Input messages
Output messages
Collaborators
Delegation
Complaint
Output Messages
Communication Part
Execution of the competence X
Message Sending the concerned agents (A1, A2 etc...)
Waiting for a response of (A1, A2, A3 etc...)
Execution of the competence Y
Message Sending the concerned agents (A4, A5, A6 etc...
Cooperation Part

Figure 2. Representation of the formalism used to define an agent.

Description of the knowledge part:

The knowledge part is an entity which describes the agent, his name, his goals, his properties. This part mainly focuses on the capabilities of the agent. All these capabilities are defined in terms of competencies. It is to some extent a representation of the knowledge of the agent translated into competences.

Description of the communication part:

An agent is a communicating entity, in other words able to receive and transmit messages: task execution demand, report of task execution are realized only by message exchange. The characteristics representing the communication part of an agent are:

- Input message and Output message

All messages are structured into three layers: **the content level, the performative level, the communication level.**

> **The content level:** The content level, corresponding to the "content" keyword, describes the information sent between agents: the product or the task description, technical data, etc...
>
> **The performative level:** The performative level specifies the sender's intention about the content. The same content can be transmitted with different intentions. The values of the performative keyword correspond to the different allowed speech acts: *ask, achieve, tell etc.*
>
> **The communication level:** This level specifies who is the receiver or the receivers of the message (keyword "to"). They can be clearly identified (an agent or a list of agents)

- Collaborators or Collaborate to

The treatment of the received messages by an agent may need the collaboration of other agents, these will be considered as ***collaborators*** . Reciprocally, an agent can be used by other agents, it ***collaborates to*** the implementation of these agents.

- Delegation and Complaint

The communication with message exchange needs to take into account the "misunderstood" messages i.e. the messages for which there is no treatment. The two modules *delegation* and *complaint* are designed for this:

> *Delegation*: When the agent cannot answer a message, it delegates it to one or several agents that constitute its delegation.
>
> *Complaint*: When the agent cannot answer a message and doesn't own delegation, it redirects it towards a "complaint" agent for error treatment.

Description of the cooperation part

This agent is also a **cooperating entity** based on the collaboration of others agents. The main function of this entity is to treat the requests initiated by the reception of messages. The processing of a request (received message) is carried out according to a set of rules called: "negotiation schemes". The negotiation schemes are a succession of operations that could be for example (See **Figure 4.**):

- Execution of local competencies
- Sending of messages to other agents, collaborators of the agent
- Waiting for a response to a message sent (etc..)

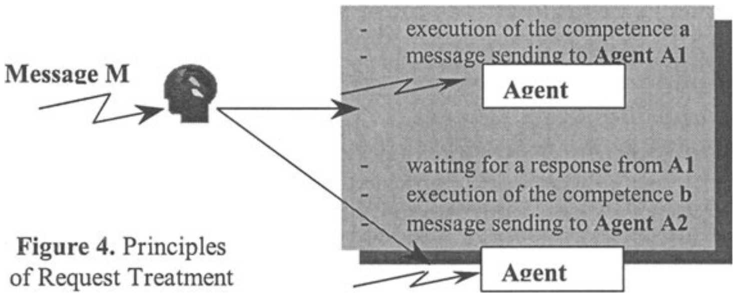

Message M
- execution of the competence a
- message sending to Agent A1
Agent
- waiting for a response from A1
- execution of the competence b
- message sending to Agent A2
Agent

Figure 4. Principles of Request Treatment

With each negotiation scheme are associated its conditions of application. They can be represented by a set of rules of the type: **If** *conditions* **Then** *negotiations* (see **Figure 5.**). Each agent owns a database of rules.

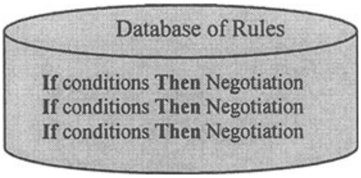

Database of Rules
If conditions Then Negotiation
If conditions Then Negotiation
If conditions Then Negotiation

Figure 5. Database of rules describing the conditions of applications associated with each negotiation scheme

Finally, it must be noted that same negotiation scheme can be used for the processing of different requests just as several negotiation schemes can be used to treat the same message.

3. PROJECT ORIENTED ENTERPRISE

3.1 Characteristics of project oriented enterprises

A company is project oriented when its customer orders require a significant adaptation of its products and services compared to the previous orders. These enterprises must be able to respond rapidly to new customer requirements. A project oriented enterprise may be characterized by a network of capabilities composed of different resources such as individuals, groups of people, machines, other enterprises, joining their competences in order to realize a common objective which may exceed the ability of each simple capability. In other words a project oriented enterprise is a network of interacting entities that consist of people, machines, computer systems, etc. all these entities having to cooperate in a particular way, specific to each project. The main challenges faced by project oriented manufacturers in today's economic environment can be summarized as follows:

- Fast reaction to customer demand / Re-organization capability
- Integration of heterogeneous entities / Knowledge sharing and exchanging
- Communication between heterogeneous software systems

Conventional project management based on a deterministic planning process is too rigid and modifications are too cumbersome to be really helpful. Moreover it does not sufficiently support the dynamic allocation of capabilities that are not necessarily identical to functional company units. It is also of no help for the integration of heterogeneous software systems, a challenge of growing importance in a globalized economy.

3.2 Analyzed enterprises

Several companies were involved in the study; however, two of them were more intensively analyzed and the results presented in section 4 and 5 are essentially based on these two cases. They are a producer of railway vehicles and a steel construction company.

Railway vehicle producer: Products range from passenger vehicles to freight wagons. The design and the manufacture of passenger railway vehicles require a detailed understanding of objectives and requirements of the end user. Therefore each order constitutes a unique project.

Steel construction company: This enterprise, specializing in mechanical engineering and steel construction, manufactures a wide range of diversified products and in particular the most exciting major rides in the world: Free Fall, Swing Boats, 360° turn Swiss bobs etc. Obviously, in this company too, each new order must be handled as a specific project requiring a particular allocation of resources and special capabilities (dynamic, calculations, material selection, welding, non-destructive testing, etc.) As the short description of the two previous typical cases shows, launching of a project can be characterized by the creation of a new network of capabilities in the enterprise: people, machines, heterogeneous software etc. This must be done quickly (dynamically) in order to fulfill the normally short response times imposed by the market. Moreover, the network of capabilities must constantly be reorganized and reallocated during the project life as the company situation is by nature very dynamic.

4. ENTERPRISE ANALYSIS

The goal of the enterprise analysis is to identify the main actors who will be represented by agents in the model. These can be individuals, groups of people, machines, computer programs etc. which own specific capabilities, as described above, called skills. The major difficulty of the analysis resides in the fact that the description, in terms of main actors, is not identical to the usual functional description of the enterprise. In order to avoid, as much as possible, a bias in the results the analysis procedure described below was put together and strictly followed.

4.1 Analysis procedure

The procedure is centered around the analysis of the business process which is carried out, successively, at three different levels: strategic, planning and execution. It is further divided into two main phases: I. Identification of the main actors and II. Description of the characteristics of each actor

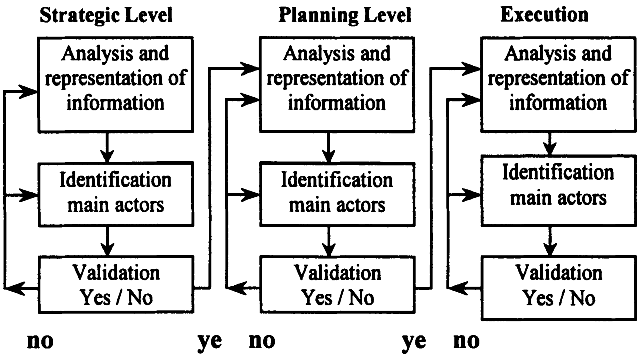
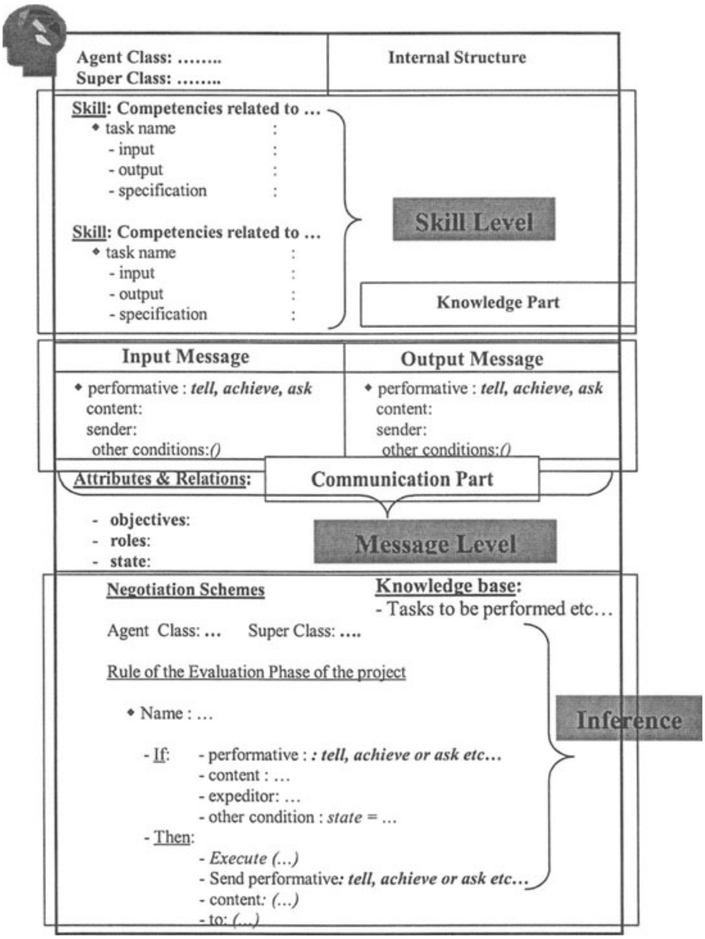

Strategic Level
Planning Level
Execution
Analysis and representation of information
Identification main actors
Validation Yes / No
Analysis and representation of information
Identification main actors
Validation Yes / No
Analysis and representation of information
Identification main actors
Validation Yes / No
no
ye
no
ye
no

Figure 6. Schematic representation of the analysis procedure used for the identification of the main actors

In phase I, the following method was applied (**See Figure 6.**)

- ✓ Analysis and representation of the information exchange
- ✓ Identification of the main actors (usually concentration point of the information exchange representation)
- ✓ Validation of the representation with the enterprise collaborators

The result of this first phase is a list of main actors and a rough description of their information exchange. In phase II, the characterization of each identified actor is detailed. The problem consists in finding a comprehensive and homogeneous description of an heterogeneous population of actors. To help reduce this difficulty, a standard analysis grid was designed and systematically used. It consists of three levels (**See Figure7.**): ➢ the **skill level** which makes it possible to describe the actor competence, i.e. the means required by and the tasks executed by the actors ➢ the **message level** which makes it possible to describe the information exchange ➢ the **inference level** which allows description of simple deductions, rules starting from the observation of competences and the objects that they handle. As in phase I, an iterative validation / correction process was used to fill in all the required analysis grid. The final result is a list of main actors with detailed characteristics identified in standard grids.



Agent Class: ....
Super Class: ....

Internal Structure

Skill: Competencies related to ...
- ◆ task name :
  - input :
  - output :
  - specification :

Skill: Competencies related to ...
- ◆ task name :
  - input :
  - output :
  - specification :

Skill Level

Knowledge Part

Input Message	**Output Message**
◆ performative : *tell, achieve, ask* content: sender: other conditions:*()*	◆ performative : *tell, achieve, ask* content: sender: other conditions:*()*

Attributes & Relations:

Communication Part

- **objectives:**
- **roles:**
- **state:**

Message Level

Negotiation Schemes

Knowledge base:
- Tasks to be performed etc...

Agent Class: ... Super Class: ....

Rule of the Evaluation Phase of the project

◆ Name : ...

- If: - performative : *: tell, achieve or ask etc...*
  - content : ...
  - expeditor: ...
  - other condition : *state* = ...
- Then:
  - *Execute (...)*
  - Send performative*: tell, achieve or ask etc...*
  - content*: (...)*
  - to: *(...)*

Inference

Figure 7. A standard analysis grid

5. ENTERPRISE MODELING BY AGENTS

Modeling the enterprise consists in creating a community of software agents representing the population of actors identified in the real world. The analysis grids defined in the previous section are used for the definition and specification of the software agents.

5.1 Agent and classes

Based on the detail analysis described in the previous section, it was possible to extract common elements to all the studied project oriented enterprises. In particular, it appeared that the same agents, identically grouped in classes could be used to describe the various companies. For a first level modeling, it was found that a collection of agents grouped in 10 classes was sufficient to describe the major activities of project oriented companies. The following agent classes were identified: **Executive Agent:** Its main goal is to make the strategic decision for the enterprise (accept or reject the proposed project) **Supervisor:** Its main goal is to manage the different projects in order to satisfy the customers and the enterprise. **Planning Agent:** Its main goal is to globally plan and schedule the different project tasks. **Design Agent:** Its main goal is to find solutions to the technical requirements in order to satisfy the functions of the final product. **Production Engineering Agent:** Its main goal is to guarantee the links between the development and the manufacturing phases. **Negotiator Agent:** Its goal is to obtain the best advantages in terms of price, delay, quality for the company in the frame work of its collaboration with the outside world. **Execution Agent:** Its goal is to manage a production unit: to organize it locally according to the global project tasks. **Logistics agent:** Its goal is to manage the inventories and to maintain the item documentation. **Quality Agent:** Its goal is to guarantee the quality of the manufacturing equipment according to the technical requirements and the standards. **Service Agent:** Its goal is to support other agents involved in the projects. **Figure 8.** represents the main agent classes identified. Each agent and class requires a detail definition of all its constitutive elements as presented in section 2. For the sake of simplicity, only one example will be given in the following section.

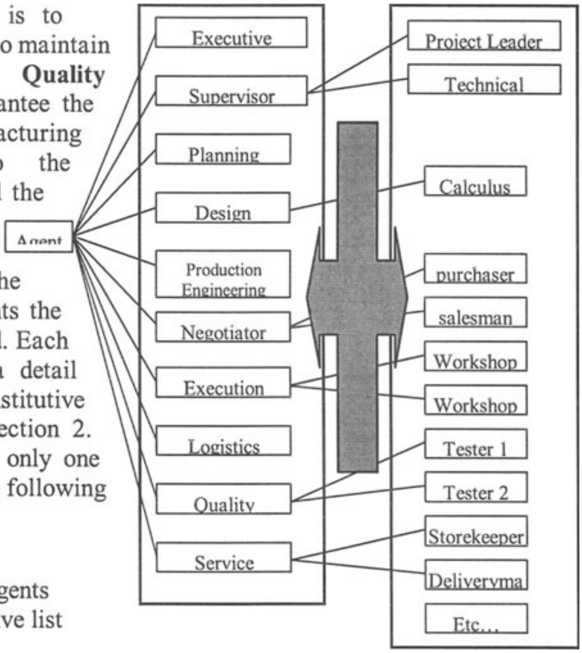


Figure 8. Structure of agents and classes Non exhaustive list

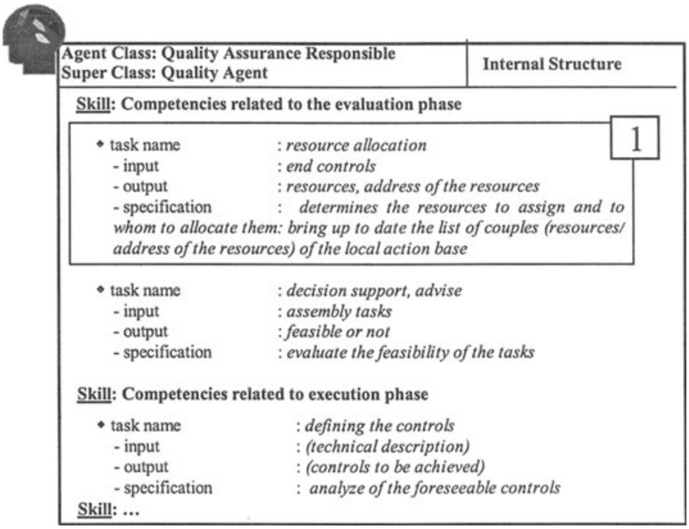
5.2 The "quality agent" internal structure

The "quality agent" is an agent whose function is to control the conformity of performed tasks in accordance with security norms and technical requirements, as well as with production and cost optimization rules. Its mission and responsibilities are: - verification of conformity to recognized specifications

- launching of actions to prevent the occurrence of non-conformity
- identification and recording of all quality problems
- finding and recommending improvement solutions in a general context
- control of QC staff abilities
- possibility to stop the realization of a project, or a fabrication, until a final decision is taken by the executive agent

The detailed definition of the quality agent is systematically represented in Figure 9. In part 1, the skill level, the competences are defined on the basis of the project phases. Each competence is structured into a task, (task name in **Fig. 9**), means (input in **Fig. 9**), task result (output in **Fig. 9**) and a short description of the task (specification in **Fig. 9**) For example ***in the evaluation phase***, a task named ***resource allocation*** has been identified. The means for this task is a definition of the action the resource is needed for. (***end control***). The result of this task is given by the output (***resources, address of the resources***). Under specification, a draft description of the task to be executed is described. In part 2, the message level, the various input and output messages are precisely defined using the three levels described in section 2. For example the input message (part 3) will initiate the processing of the request " ***evaluation of the assembly conditions***" in the inference level. In part 4, the request is executed according to the corresponding negotiation scheme. In particular a local ***competence "decision support, advise***" is executed and the output message ***"(feasibility, F)"*** is sent to ***the supervisor and production engineering agents*** shaded in part 5. Part 6 is purely descriptive, having no direct influence on the behavior of the agent.

All agents identified in section 5.1 have been defined following the same procedure.



Agent Class: Quality Assurance Responsible
Super Class: Quality Agent

Internal Structure

Skill: Competencies related to the evaluation phase

1

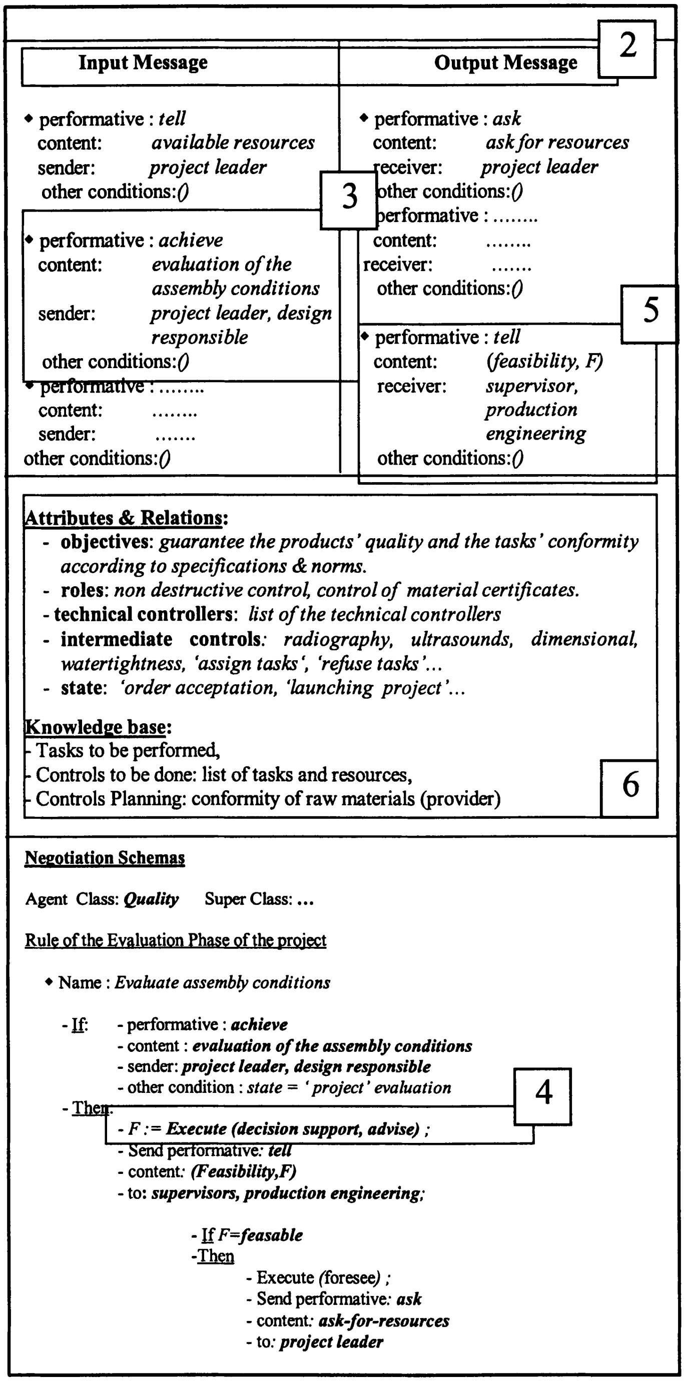
- task name : *resource allocation*
  - input : *end controls*
  - output : *resources, address of the resources*
  - specification : *determines the resources to assign and to whom to allocate them: bring up to date the list of couples (resources/ address of the resources) of the local action base*

- task name : *decision support, advise*
  - input : *assembly tasks*
  - output : *feasible or not*
  - specification : *evaluate the feasibility of the tasks*

Skill: Competencies related to execution phase

- task name : *defining the controls*
  - input : *(technical description)*
  - output : *(controls to be achieved)*
  - specification : *analyze of the foreseeable controls*

Skill: ...



Input Message

- performative : *tell*
  content: *available resources*
  sender: *project leader*
  other conditions:*()*
- performative : *achieve*
  content: *evaluation of the assembly conditions*
  sender: *project leader, design responsible*
  other conditions:*()*
- performative :........
  content: ....
  sender: ....
  other conditions:*()*

Output Message

- performative : *ask*
  content: *ask for resources*
  receiver: *project leader*
  other conditions:*()*
  performative :........
  content: ....
  receiver: ....
  other conditions:*()*
- performative : *tell*
  content: *(feasibility, F)*
  receiver: *supervisor, production engineering*
  other conditions:*()*

Attributes & Relations:

- **objectives**: *guarantee the products' quality and the tasks' conformity according to specifications & norms.*
- **roles**: *non destructive control, control of material certificates.*
- **technical controllers**: *list of the technical controllers*
- **intermediate controls**: *radiography, ultrasounds, dimensional, watertightness, 'assign tasks', 'refuse tasks'...*
- **state**: *'order acceptation, 'launching project'...*

Knowledge base:

- Tasks to be performed,
- Controls to be done: list of tasks and resources,
- Controls Planning: conformity of raw materials (provider)

Negotiation Schemas

Agent Class: ***Quality*** Super Class: ...

Rule of the Evaluation Phase of the project

- Name : *Evaluate assembly conditions*

```
- If:   - performative : achieve
        - content : evaluation of the assembly conditions
        - sender: project leader, design responsible
        - other condition : state = 'project' evaluation
- Then:
        - F := Execute (decision support, advise) ;
        - Send performative: tell
        - content: (Feasibility,F)
        - to: supervisors, production engineering;

                - If F=feasable
                -Then
                        - Execute (foresee) ;
                        - Send performative: ask
                        - content: ask-for-resources
                        - to: project leader
```

Figure 9. Quality Agent Definition

6. DISCUSSION AND CONCLUSION

Because the agent described in this paper is an autonomous entity which owns skills and is able to communicate, it can co-operate with its peers in an adaptive manner regardless of the modeled entities encapsulated in that particular agent. Each actor of the enterprise: person, department, workshop may potentially be described, and encapsulated into an agent. This modeling technique can support the flexibility required in putting together new combinations of competences quickly. However it must be noted that in a real environment, the main entities and the major communication modes should first be identified in order to allow for an optimal implementation of the model. This approach is required, as it is obvious that it would not be reasonable and economically feasible to model and integrate all the actors and communication modes of an enterprise.The collection of agents and agent classes proposed here constitutes a first attempt for a generic multi-agent description of project oriented enterprises. It has been established and validated on the basis of a restricted number of real cases and should therefore be further validated and possibly extended with additional company analysis. Nevertheless, it can be concluded that a multi-agent approach is particularly attractive for the modeling of very dynamic structures, such as those of project oriented enterprises. The modeling techniques presented in this work are focused on a relatively aggregated level of capabilities (decision functions, whole workshop, etc...). They could be extended to a much more detailed level by identifying the true competence building blocks of a company that could then be viewed as unit competences. Modeling an enterprise in terms of unit competences could be a very interesting approach as these are more and more recognized as key competitive factors. These unit competences need to be carefully managed, maintained and allocated in a flexible and dynamic way, particularly in project oriented enterprises. The use of multi-agent technology in the field of competence management seems therefore a promising field of investigation.

7. REFERENCE

1. Roche C, Fitouri S, Glardon R, Pouly M "The potential of multi-agents systems in virtual manufacturing enterprises". Proceeding of the Database and Expert Systems Applications, 1998.
2. Hyacinth S. Nwana "Software Agents: An Overview". Knowledge Engineering Review, Vol. 11 n° 3, pp1-40, Sept. 1996.
3. Franklin S, Graesser A "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents". Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages. Springer-Verlag 1996
4. Michael Wooldrige and Nicolas R. Jennings "Agent Theories, Architectures and Languages: a Survey" Wooldridge and Jennings eds. Intelligent Agents. Springer-Verlag, 1-22, 1995.
5. Finin T, Y. Labrou and James Mayfield. 1995. "KQML as an agent communication language". Software Agents (Jeff Bradshaw Ed.), MIT Press, Cambridge, Ma, USA.
6. Marik V, Pechoucek M, Lazansky J, Roche C "PVS'98 Agents: Structures, Models and Production Planning Application" International Journal of Robotics and Autonomous Systems special issue on Multi-Agent Systems Applications, pages 29-43, 1999.

7. ACKNOWLEDGEMENT

The authors wish to thank Pr. Marik from the Technical University at Prague for his helpful advice. Thanks are also due to Dr Ian Stroud and Dr Antonio Stagno for their comments. This work was carried out under the financial support of the Swiss Commission for Technology and Innovation.