A study case with a methodological approach for complex and distributed industrial system simulation

Stéphane Galland[♯♭], Frédéric Grimaud[♯],
Philippe Beaune[♭], Jean-Pierre Campagne[♯]

Department of *Scientifical Methods for Industrial Management* (♯)
Department of *Multi-Agent Systems* (♭)
École Nationale Supérieure des Mines
158, Cours Fauriel, Saint-Etienne, 42023 Cedex 2, France
e-mail: {galland,grimaud,beaune,campagne}@emse.fr

Abstract

We are located in the context of the simulation of industrial systems, whish are complex and distributed in operational, decisional and informational terms. In this article, we present through a study case the whole of the concepts, formalisms and models suggested by our methodological approach $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$. It takes into account the distribution, the modularity, and the reusability during the modeling and the simulation of industrial systems. In addition to various formalisms suggested, we present a model of a multi-agent system, whish is able to carry out simulation.

Keywords: Distributed simulation, Multi-agent systems, Methodology, Systemic

1 Introduction

Simulation is a tool adapted to the modern industrial problems. It permits to take account of the dynamic aspects during the study of the production system behavior. But certain problems always remain. We particularly concentrate ourself on four categories. First, we consider that the simulation tools are still seldom accompanied by methodologies. These last would facilitate the modeling of the systems. Then, simulation tools make difficult any modular conception of models. It is difficult to re-use part of already-developed models without carrying out significant adaptations. The third problem is the very strong interconnection of the operational, informational and decisional aspects within simulation models. For example, if you want to simulate two different kind of management *e.g.*, pushed- or pulled-flows, you must write two completly different models event if only the decisional aspect change. Lastly, we think that it is increasingly difficult to simulate current industrial systems (virtual society, consortium, ...). Indeed, they evolve to increasingly decentralize structures.

To answer these various problems, we proposed in [Galland, 1999] a methodological approach : $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$ [1]. It offers a modeling background to solve the various problems mentioned above. It provides an modeling approach, whish is independent of any simulation tool. The data-processing distribution of simulation models is managed via the introduction of the multi-agent concepts (MAS). Finally differentiation of the industrial system flows is carried out according to the systemic theories [Le Moigne, 1992] and to MAS.

The two following sections present respectively our methodological approach and the multi-agent framework in which we are located. In section 3 we illustrate through a study case the main concepts of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$. This study case corresponds to a production line of an automobile manufacturer. Finally we conclude and present our perspectives.

2 Methodological approach for simulation

To develop a simulation model and to answer the problems evoked in the introduction, we use the methodological approach, which we propose in [Galland, 1999], named $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$. The figure 1 illustrates, according to a formalism close to SADT, the life cycle of this approach.

The first stage is the analysis. It has a result that must be the description of the properties and constraints attached to the modeling or to the simulation of the studied industrial system. Even if this

[1]Multi-Agent Methodological Approach for Simulation

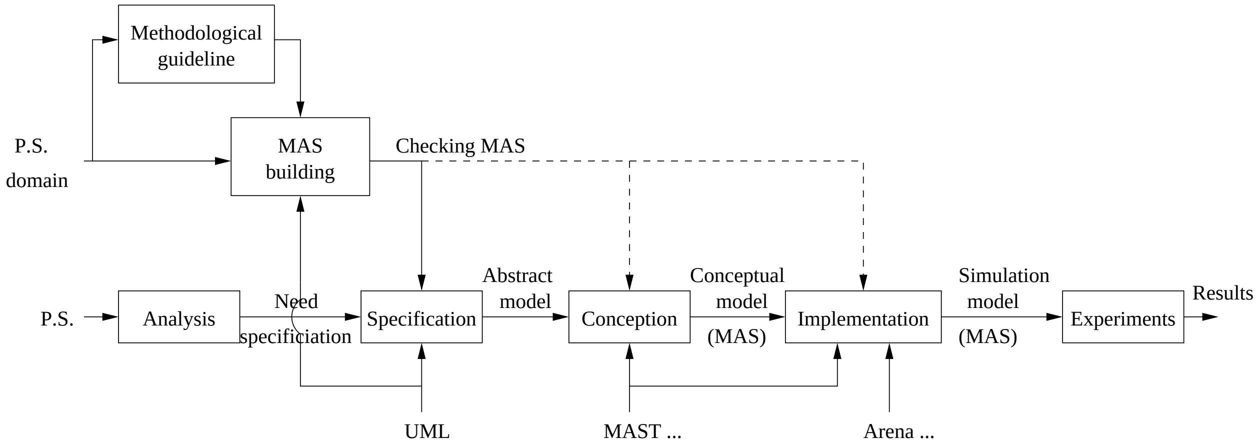


Figure 1: Methodological approach

phase was not the object of our works, we propose within the framework of this article an approach drawn from work on the extensions of the methodology B [Galland, 1998]. This approach consists in describing a whole of rules. They should appear in the continuation of the modeling and more especialy during the specification phase.

The phase of specification permits to carry out a formal simulation model (or abstract simulation model). This one is independent of any simulation tool or MAS platform. In order to answer the problems of flow disctinction within the system, the model is based on a systemic approach [Le Moigne, 1977, Le Moigne, 1992], and separately describes the operational, informational and decisional aspects. Moreover, the approach suggested in $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$ takes into account the data-processing and conceptual distributions of simulation models. The data-processing distribution corresponds to the distribution and the communication of models within a computer network. These problems are partly solved by technologies like HLA [US Department of Defense, 1996].

The stage of conception is based on the use of the abstract simulation model previously defined. It is completely or partially translated into a multi-agent model, which is able to carry out the simulation. We decided to use the MAS concepts because they are strongly adapted to the management of the physical and decisional aspects of industrial systems [Burlat, 1996]. The result of the conception phase is a MAS model using the Vowel approach [Demazeau, 1995]. However this model is always independent of any multi-agent platform and simulation tool.

The last major stage of our methodological approach is the implementation. It permits to instance the MAS model described above. This instanciation can use a multi-agent platform (MAST, Matkit...) as well as a simulation tool (Arena®, Simple++®...).

Our methodological approach provides also a set of stages that allow to make some experiments on the simulation models. The results should be used to validate the model, or to retrogress in the modeling life-cycle to update or to correct models.

Finally the methodological guideline and the construction of a MAS are internal stages of the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$ development. They respectively provides a modeling guideline with $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$ and the implementation of a multi-agent system whose role is to check the integrity of the various models produced by $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$.

3 Case study: workshop of production of sensors

In this section, we present a study case that allows us to study and validate the concepts of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$–$\mathcal{S}$.

3.1 Abstract presentation of the system studied

This study case is based on the problem tackled in [Hacid, 1999, Campagne, 2001]. It is about an equipment supplier that realize sensors for the car industry. The unit "sensors" is organized in lines. Our work concerns one of these lines, which produces two different kind of sensors A and B. The line is made up of four machines respectively carrying out winding, the weldings and the installation of legs, the weldings and the rotation of the sensors, and controls of them.

Sensor A follows a route made up of the first, the second and the fourth machines. Whereas the sensor B follows its own route made up of the first, the third and the fourth machines.

Structural architecture of the line imposes an organization in the form of two distinct workshops being in different places in the building. The first workshop contains only the first and the second machine whereas the second workshop is composed of the two last machines.

The human organization of the line consists of the use of three machinists' competences. The first can work only on the second and the third machines, whereas the two following can work on the first and third machines. The management mode of these human resources is based on the "first in first out" principles.

3.2 Analysis

We will name the four machines M_1, M_2, M_3 and M_4. Figure 2 illustrates the studied system. This phase is not the subject of our work. However we propose an approach based on the expression of rules and constraints. This approach, whish is the result from works on the method B such as [Galland, 1998], permits to follow the evolution of the properties and constraints during the modeling.

Within the framework of our case study, we release about fifteen rules of which some are presented in the continuation of this section. We consider that the drafting of these rules was empirical *i.e.*, we do not propose method or technic that should facilitate their drafting.

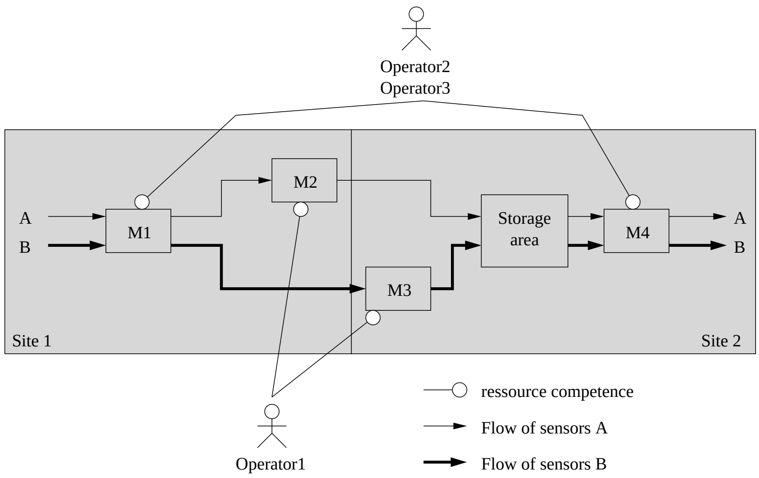


Figure 2: Production line of a automobile manufacturer

RULE 1 :

The line of production is made up of four machines M_1, M_2, M_3, M_4.

RULE 2 :

The line of production is able to produce a first type of sensors A. The latter must pass successively by the machines M_1, M_2 and M_4.

RULE 3 :

The operator `Operator1` *is able to work on the machines M_2 and M_3.*

RULE 4 :

The first operator is a member of the team on which depends directly from the first site. He is regarded as a local resource of this site.

Rule 5 :

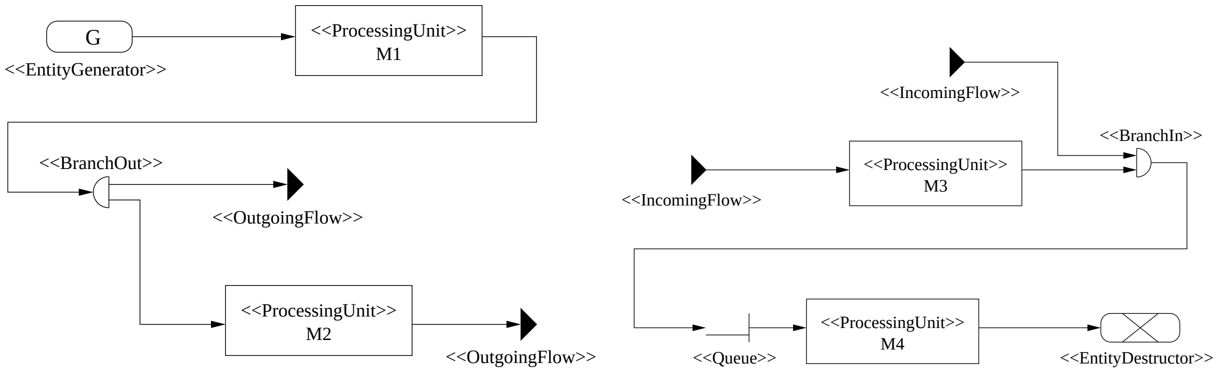
The machines M_1, M_2, M_3 and M_4 have processing times according to a normal statistical law of reason 10.

The rules, which we have just stated are not exhaustive but allow us to illustrate the use of this analyzing approach in the continuation of this article.

3.3 Specification of the simulation model

In this section, we define the abstract simulation model describing the industrial system presented in section 3.1. The abstract model corresponds to a representation of the industrial system, which is independent of any simulation tool and any multi-agent platform. The model is divided into three sub-models corresponding respectively to each subsystem of the systemic approach suggested by [Le Moigne, 1977, Le Moigne, 1992] : operational, informational and decisional.

3.3.1 Operational subsystem model



(a) First site (b) Second site

Figure 3: Abstract model of the operational subsystem

The operational subsystem is modelled by a simple description of the physical structures. Figures 3(a) and 3(b) illustrate the models of the first and second site.

These diagrams enable us to be compliant with the rule 1. We represent also the only storage area of our system using a queue.

As illustrated in figure 2, the first site contains only the machines M_1 and M_2, whereas the second site contains the two other machines (M_3 and M_4) as well as the queue before the last machine.

We consider that the model representing the operational subsystem should contain only the physical structures of the production system [Le Moigne, 1992]. In particular it is impossible to indicate the processing times of the various production units or the behaviors used by resource allocation (FIFO, LIFO...). This various information fall within the competence of the informational or decisional substystems.

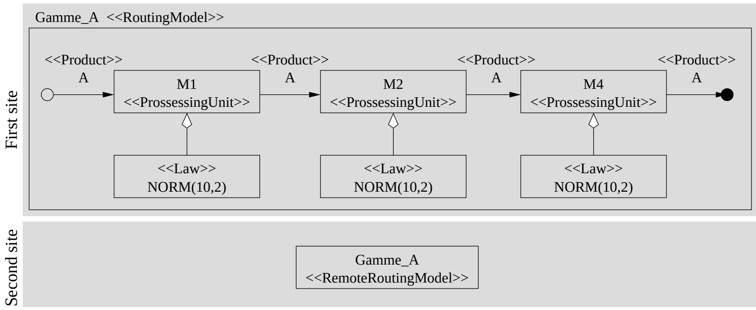
3.3.2 Informational subsystem model

We define the informational subsystem from the models of the nomenclatures and of the product routing.

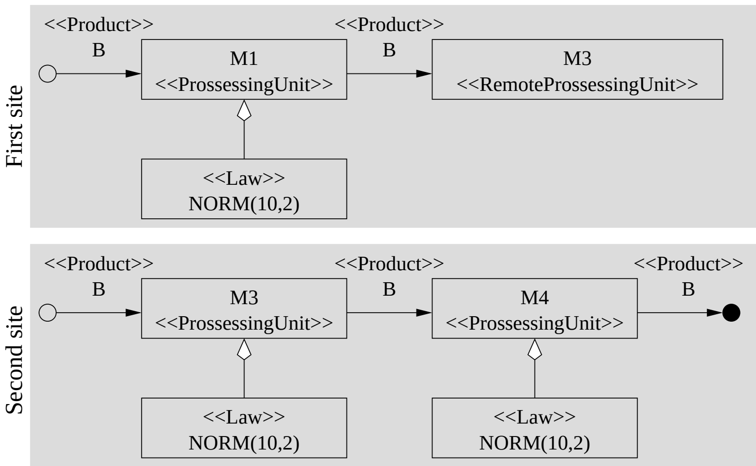
Nomenclature model: Within this model, we describe the list of the products and sub-products (components) transformed within our production system, as well as the list of the composition relations between these various products.

In this study case, the production line treats only sensors A and B. Our system not handling any sub-product, the nomenclature models of these two products are simple. However we consider that a sensor is composed by components that are not used by our model (*cf.* rule 5).

Routing model: In this model we describe the means and the resources necessary for the realization of sensors A and B. Figure 4 illustrates the four models necessary to describe the route of the two products out of the two sites of production. The formalism used not having been definitively chosen yet, we propose



(a) Routing for the sensor A



(b) Routing for the sensor B

Figure 4: Abstract model of product routing

a representation based over transformation units (`ProcessingUnit`), transitions (`Product`) and processing times (`Law`).

We chose to define the route of sensor A only on the first site. It is necessary to have a coherence between the routing model and the nomenclature model. Indeed, if the product whose manufacture is described in the route requires the use of sub-products, the latter must obligatorily appear in the nomenclature model of the same product.

In this study case, the products not being composed of sub-products, the routing model must reveal the sensors A and B respectively in the models of the first and second products. Rule 2 enables us to build a model of the ranges for the sensor A, which is composed of machines M_1, M_2 and M_3. Moreover the processing times on these last (*cf.* rule 5) are explicitly mentioned in the form of normal statistical laws.

In the routing model of the sensor B, we arbitrarily decided to distribute information within the two sites. Thus we consider that sites 1 and 2 contain routing sub-models respectively so that each site has information necessary and sufficient concerning the route of the sensor B. This choice requires the implementation of checking mechanisms of the coherence, which we have described in the specifications of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$ [Galland, 2000a]. The processing times of the machines are deduced from rule 5.

Considering that the routes are made up of transformation units, we define the processing times necessary to carry out their task. A `Law` object is thus attached to each processing unit.

3.3.3 Decisional subsystem model

The decisional subsystem contains the whole of the decision-making centers necessary to the management of the industrial system [Burlat, 1996]. Within the framework of this study case, we propose the decisional model presented in figure 5.

This decisional model includes only the operational level. The tactical and strategic levels do not have any particular significance in the current state of the problem. It is not necessary to make them appear in this model.

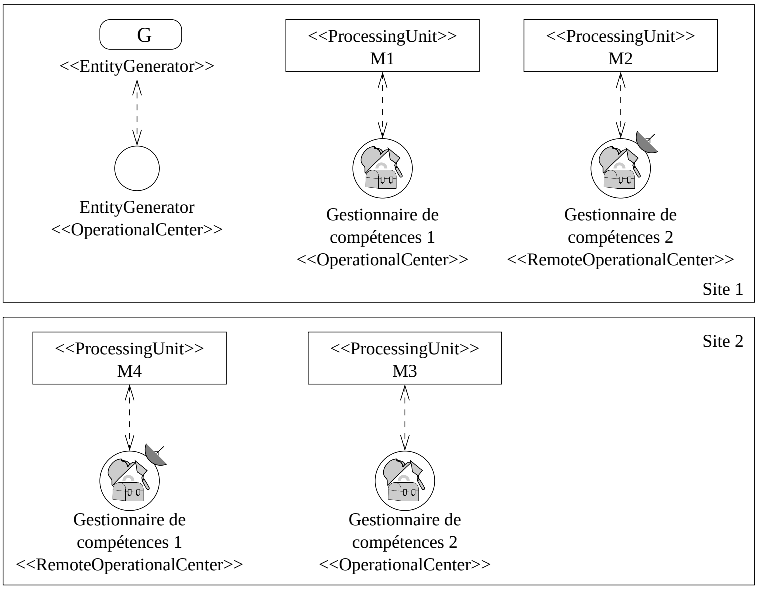


Figure 5: Abstract model of the decisional subsystem

We defined a decision-making center for the function of "raw material" generation, and for each comptetence group playing a role in the system. `Operator1` is a resource, which is used as soon as it is available. To model this management mode, we create an operational decision-making center, which associated with machines M_2 and M_3, enables us to model the algorithm of attribution of the resource `Operator1`. Indeed we consider that the resource management is under the competence of the decisional subsystem rather than the operational subsystem. We define the behavior of the decision-making center by the way of a reactive behavior: the center will answer favorably an allowance or release of the resource according to the as-used condition of it. With this intention we use the groundwork of a reactive behavior given by $\mathcal{M}_{\mathcal{A}}\mathcal{M}\mathcal{A}$-$\mathcal{S}$. It enables us to define by the way of pseudo code the reactions of the decision-making center according in the messages, which it will receive from the machines B and C. But before illustrating these behaviors, we define a value or variable allowing us to represent the resource on the level of the decision-making center : `CONTEXT Resource adminitrator 1 DEFINE resource`$_operator1$: $Semaphore.Asofnow,wecanasfollowsdefinethebehaviorofthemanagementcenterofcompetence1(weuseapseudocodederived$
$\mathcal{S}$[`Galland, 2000a`]) :

```
CONTEXT Resource administrator 1
WHEN RECEIVE SYNCHRONOUS
     AllocationOfResource
FROM ATTACHED( M )
THEN
   self.resource_operator1.get()
   REPLY Operator1
END

CONTEXT Resource administrator 1
WHEN RECEIVE SYNCHRONOUS
     ReleaseOfResource
WITH PARAMS ( ID: Identificator )
FROM ATTACHED( M )
THEN
   self.resource_operator1.release()
END
```

These two reactive behaviors permits the decision-making center to allocate the resources according to a management mode FIFO.

The behavior specification of the decision-making center managing the second group of competences (`Operator2` and `Operator3`) is carried out same manner as previously.

The decision-making center `EntityGenerator` has the role to periodically generate commands of entity generation. This center has a behavior imposing to him the use of an exponential law of average 10.

```
CONTEXT EntityGenerator
WHEN EXPO( 10 )
```

```
THEN
  LET aproduct = CHOOSE
      WITH NORMAL( 10, 5 ) IN P1,P2
  AND aquantity = CHOOSE
      WITH NORMAL( 10, 5 ) IN [1,]
  IN
  SEND GenerationEntity
       ( aproduct , aquantity )
       TO ATTACHED( EntityGenerator )
END
```

We have just described the abstract model of our industrial system. In the following sections, we present the model of a agent society corresponding to this abstract model. Then, we illustrate the choices of implementation, which were necessary to the data-processing realization of the simulation model.

3.4 Conceptual model : a society of agents

The conceptual model is a multi-agent model corresponding to the abstract model describes during the specification phase. It describes the structure of the agent society without however imposing the use of a multi-agent platform or a simulation tool. The only constraint is to respect a specification carried out according to the vowel approach [Demazeau, 1995].

Before presenting the conceptual model of our study case, we describe our MAS context, and then a model proposed by the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$ methodology and which is used as a basis for construction of the multi-agent model.

3.4.1 Multi-agent systems

Our methodological approach is based on the multi-agent concepts [Ferber, 1995]. We use the Vowel approach (or AEIO) proposed by [Demazeau, 1995]. Figure 6 [Boissier, 1999] illustrates the four facets composing approach AEIO.

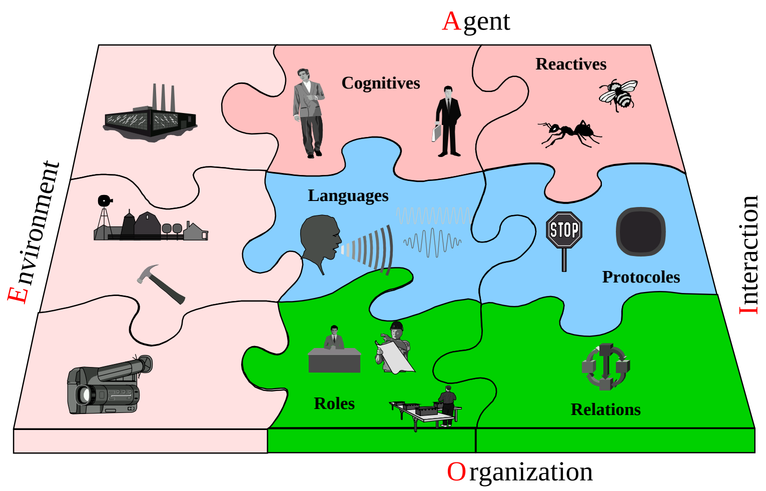


Figure 6: Vowel approach (AEIO)

The Agent facet corresponds to a structural, functional and behavioral description of the agents. The Environment is the whole of the objects existing in the MAS but which are not autonomous or deliberative. The Interaction facet permits to specify the whole of the languages and the protocols allowing the agents to exchange messages and knowledge. Lastly, the Organization is the description of the organisational relations (links of dependences or authority, roles, ...). They allow to specify the organisational structure of the agents within the system.

By their autonomy and their capacities of interactions, the multi-agent systems permits to carry out at the same time the distribution within a computer network, but also the distribution of the operational subsystem. Moreover cognitive capacities of agents authorize the distribution of the informational and decisional subsystems [Burlat, 1996]. The modularity generated by the use of the MAS enables us to answer another crucial point at the industrial level : the re-use of knowledge and the already controlled tools. In section 3.4, we briefly present multi-agent architecture proposed by $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$.

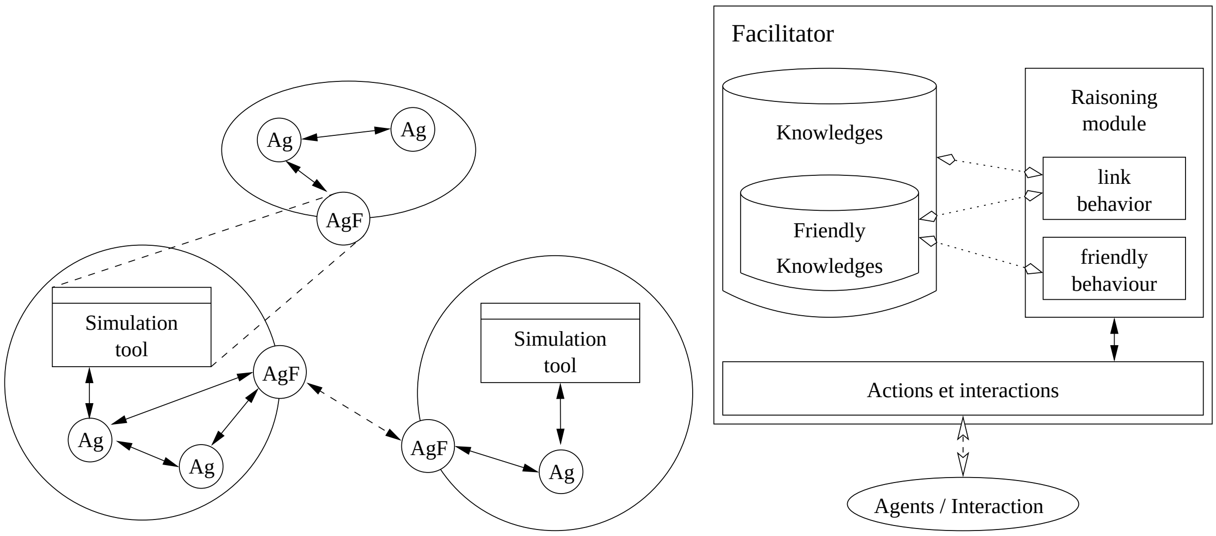
Wishing to concentrate us on the development of an multi-agent architecture adapted to the simulation of industrial systems, we use the results of work on the time expression in MAS [Carron, 1999], on the opening of MAS [Vercouter, 2000], on the organisational structures in MAS [Hannoun, 2000]. Within the framework of this article, we will approach in section 3.4 the concept of multi-agent opening system.

3.4.2 Agent model of simulation

In order to allow the installation of a process of simulation via agent society, we propose an infrastructure illustrated by figure 7(a).

It is made up mainly by inter-connected societies of agents. This principle enables us to distinguish two great classes of agents :

- the facilitators (AGF) have as the role to facilitate the transmission of the messages between the agents having to carry out the simulation process. Thus the facilitators are obligatory intermediaries to carry out the communications of a under-society of agents towards another. In addition, the whole of the facilitors maintain a knowledge base containing the whole of the resources and the services (resources, names of central processing unit, decision-making centers, ...) available in the system. AGF are charged to route the messages so that the latter reach at least one of the agents managing the corresponding services.
- the agents for simulation (AG) make the company of agents being able to simulate. We propose an architecture allowing the agent to communicate with the facilitators as well as with the other elements of its environment (agents and objects of the environment).



(a) Simulation agent infrastructures

(b) Facilitator's model

Figure 7: MAS architectures

The facilitators are the only static agents within $\mathcal{MAMA}$–$\mathcal{S}$ *i.e.*, their architecture and their interactions are defined in our methodological approach and cannot be extended by objects belonging to a simulation model. The latter can only influence the design simulation agents. Figure 7(a) illustrates also another significant aspect of our approach : recursivity of our architecture. Indeed, each agent or object of the environment of our multi-agent system can be in its turn a multi-agent system of lower level.

Definition of a facilitator

We define an agent facilitator according to four facets of approach AEIO :

- **Facet "Agent"**: An facilitator is composed of a behavioral module making it possible the agent to play the role of link or facilitator. We consider that the facilitators are friendly according to [Vercouter, 2000] with AGF. Figure 7(b) illustrates this architecture:
  - **Knowledge**: knowledge base necessary to the module of reasoning : services suggested by the simulation agents recorded at this AGF, ...
  - **Friendly knowledge**: together knowledge necessary to the friendly behavior of the agent. This knowledge is divided into two parts : the representation of other AGF and the knowledge common to all the agents (semantic of an action or a plan...) [Vercouter, 2000].

- **Friendly behaviour**: modulate allowing the agent to implement a friendly behavior with the other facilitators. When a AGF wants to enter the system, it is presented to the other facilitators by using the method suggested by [Vercouter, 2000]. Once introduced into the company, the new facilitator maintains the coherence of his knowledge on the services suggested by the simulation agents, which are associated to him, and his knowledge of other AGF.
- **Link behavior**: module allowing the agent to be used as link between various agent society. This module uses the part *representation of the others* of friendly knowledge. When the agent receives a request for message transmission on behalf of an simulation agent, it will seek the facilitator satisfying the needs and will send a message to this one. When it receives a message on behalf of another facilitator, AGF seeks an simulation agent able to meet the needs and if it finds any no, either it returns the address of another facilitator that could respond more efficiently to the answer, or it returns an error message.
- **Actions & interactions**: module allowing to implement the actions and the interactions decided by the module of reasoning.

- **Facet "Environment"**: The facilitator does not have any relationship with the environment. This facet is thus not considered in our model.
- **Facet "Interaction"**: The facilitator has two main categories of interactions: interactions with the other facilitators and the interactions with the simulation agents. Initially we define the ontology used in this facet : transmission of messages of simulation and modeling of industrial systems. In addition to the protocols necessary to the friendly behaviour, we define two protocols :
  - sending protocol of a message towards other AGF,
  - protocol of taking into account of a message coming from another AG, illustrated by figure 8(a).

(a) Receiving from others AG

(b) For the simulation agent

Figure 8: Protocols

These protocols are represented using state-transition graphs. The transitions are labelled by the messages sent (prefixed by +) or receipts (prefixed by !). In addition each message can be postfixed by arguments.
Other protocols can be used by these agents : protocol of search for service within knowledge of the facilitators...
The messages exchanged during its two protocols will be formalized using specifications of FIPA-ACL, FIPA-SL and [Carron, 1999].

- **Facet "Organization"**: The facilitators play the role of link between the various agent societies. They have links of dealings with the other facilitators (AGF) and of the links of authority with the agents (AG) composing the company with whish they are attached. The expression of such an organization with MOISE [Hannoun, 2000] does not put any problem.

Definition of an agent for simulation

This category of agents is not entirely specified by $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$. We propose only one partial architecture of the Interaction facet, which will permit to the agents to communicate with the facilitators. This architecture is made up of two protocols : sending and reception of messages towards AGF. Figure 8(b) illustrates the protocol making it possible to send a message towards another agent via AGF.

Our methodology proposes also a whole of definitions of simulation agents : agent transferring the entities from a model to another, resource management center... These various models are not usable directly. They can be compared to low-level business-objects.

3.4.3 A society of agents for the studied case

Within the framework of our study case, we release the following types of agents:

- agents corresponding to the decision-making centers managing the resources,
- the agent corresponding to the decision-making center for the generation of the entities,
- agents allowing to transmit the entities from a simulation model to another,
- agents representing the remote resources.

Figure 9 illustrates the structure of the agent society. $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$ considers that these agents are black-boxes (thereafter, we will propose an architecture for each agent that could be generated starting from the modeling elements of the abstract simulation model). The only constraint that we impose is that these various agents use the same technic of interaction (messages and protocols) to communicate with the facilitators. The description format of the services is also given by $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$.

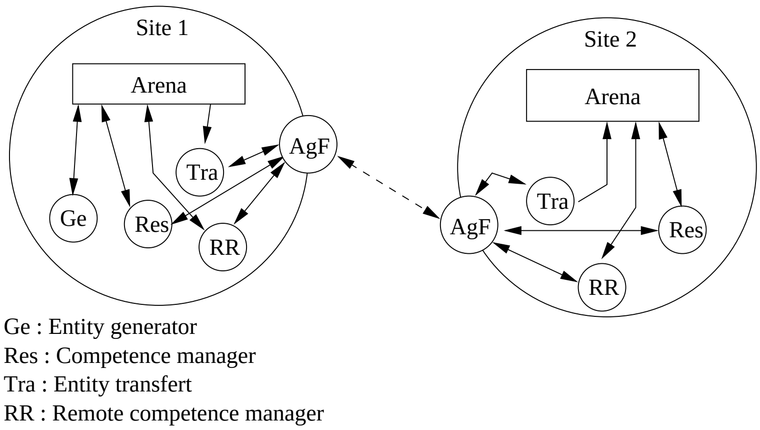


Figure 9: Structure of the SMA for the case study

3.5 Computer model : the implementation

During this phase the previous MAS model is instanced. For that, we use the simulation software ARENA® accompanied by the programming environment Visual Basic. Figure 10 illustrates the ARENA® model of the first site. $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$ proposes rules of translation from multi-agent model to a software architecture.

4 Conclusion and perspectives

We propose a methodological approach allowing to carry out simulation models of complex and distributed systems. This approach, named $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$, was introduced into [Galland, 1999, Galland, 2000b, Galland, 2000c] without however being the subject of a presentation integrating the whole of its concepts. In this article, we illustrate through study case the various concepts attached to $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$.

After having briefly presented the life cycle of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$: analysis, specification, design and implementation, we present these four phases through the modeling of a sensor production workshop for the car industry. This example enables us to concretely show the possibilities offered by $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$. We show that an modeling approach based on the systemic one was valid. We propose a model of multi-agent system able to carry out distributed and modular simulations. In addition, we can foresee that the use of a systemic modeling largely facilitates the modification of the simulation models. Indeed, any change in one of the operational, informational or decisional subsystems does not necessary imply a modification in the others.

In the future, we want to develop the phase of analysis. Moreover we think that the modeling elements currently proposed are not enough to allow an easy modeling of industrial systems. Thus we will study the concept of business-object as a possible extension of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}-\mathcal{S}$. Lastly, we will apply our theories to industrial applications.

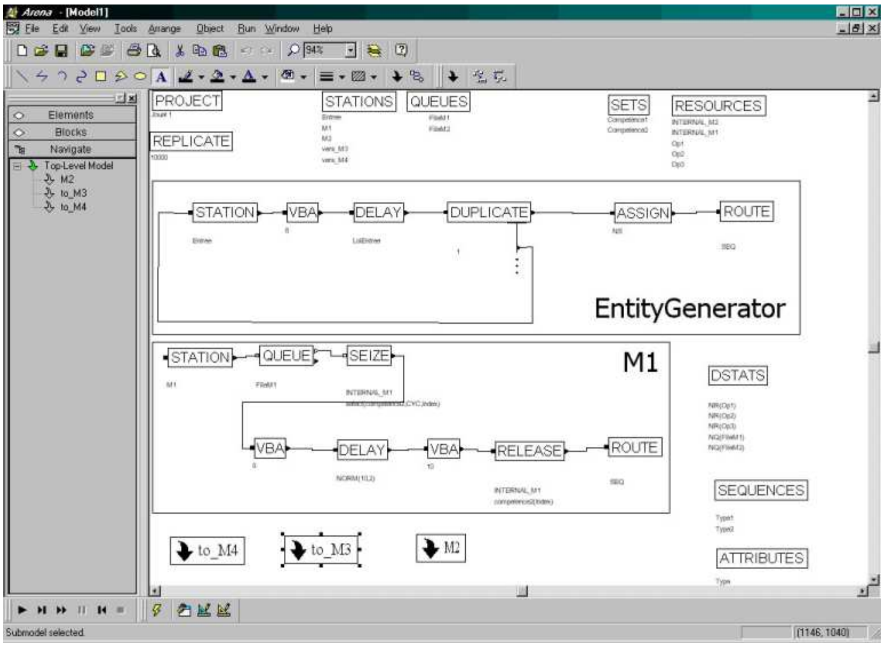


Figure 10: Simulation model of the first site

References

Boissier O., 1999. “Approche Voyelles AEIO” .

Booch G., Jacobson I., Rumbaugh J., et al., 1997. *Unified Modeling Language Specifications - version 1.1*. UML consortium - Object Management Group, Technical report.

Burlat P., 1996. *Contribution à l’Évaluation Économique des Organisations Productives : vers une modélisation de l’entreprise-compétences*. Thesis, Université Lyon 2.

Campagne J.-P., Grimaud F., et Hacid S., 2001. Production cyclique: Application et évaluation par simulation chez un équipementier automobile. *Proceedings 3ème Conférence Francophone de MOdélisation et SIMulation*, Troyes, p.965-972.

Carron T., Proton H., et Boissier O., 1999. A Temporal Agent communication language for dynamic Multi-Agent Systems. *Proceedings Modelling Autonomous Agents in a Multi-Agent World*, Spain, p.115-127.

Demazeau Y., 1995. From Interactions to Collective Behaviour in Agent-Based Systems. *Proceedings European conference on cognitive science*, Saint-Malo, France.

Ferber J., 1995. *Les Systèmes Multi-Agents - Vers Une Intelligence Collective*, InterEditions.

Galland S., 1998. “Classification de propriétés dynamiques et de leurs raffinements à partir de quelques études de cas” .

Galland S., Grimaud F., Beaune P., et Campagne J.-P., 1999. Multi-Agent Methodological Approach for Distributed Simulation. *Proceedings Simulation in Industry - 11th European Simulation Symposium*, Horton G., Möller D., et Rüde U. (Eds.), Erlangen - Germany, p.104-108.

Galland S., 2000a. *Rapport d’activité : $\mathcal{M}_{\mathcal{A}}\mathcal{MA}\text{-}\mathcal{S}$ - version 3.0-alpha*.

Galland S., Grimaud F., et Campagne J.-P., 2000b. Methodological approach for distributed simulation : General concepts for $\mathcal{M}_{\mathcal{A}}\mathcal{MA}\text{-}\mathcal{S}$. *Proceedings Simulation and Modelling : Enablers for a better quality of life – 14th European Simulation Multiconference*, Van Landeghem R. (Eds.), Ghent, Belgium, p.77-82.

Galland S. et Grimaud F., 2000c. Methodological approach for distributed simulation : Life cycle of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}\text{-}\mathcal{S}$. *Proceedings ASIM-workshop 20/21.3 2000 - Multiagentsystems and Individual-based simulation*, Klügl F., Puppe F., Schwarz P., et Szczerbicka H. (Eds.), Institut für Informatik, Würzburg, Germany, p.83-93.

Hacid S. et Campagne J.-P., 1999. Production cyclique à capacité finie : un exemple d’application. *Proceedings 3ème Congrès International de Génie Industriel*, Langevin A., Riopel D., et Ladet P. (Eds.), Montréal, Canada, p.661-670.

Hannoun M., Boissier O., Sichman J. S., et Sayettat C., 2000. MOISE : An organizational Model for Multi-agent Systems. *Proceedings Advances in Artificial Intelligence*, Monard M. et Sichman J. (Eds.), Brazil, p.156-165.

Le Moigne J.-L., 1977. *La théorie du système général. Théorie de la modélisation*, Presses Universitaires de France.

Le Moigne J.-L., 1992. *La modélisation des systèmes complexes*, Editions Dunod.

US Department of Defense, 1996. *High Level Architecture Federation Development and Execution Process (FEDEP) Model, version 1.0.* Defense Modeling and simulation Office, Technical report.

Vercouter L., 2000. *Conception et mise en œuvre de systèmes multi-agents ouverts et distribués.* Thesis, École Nationale Supérieure des Mines, Saint-Étienne, France.