How to specify an simulation model of an industrial system in $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$

Stphane Galland, Frdric Grimaud, Jean-Pierre Campagne
Department *Mthodes Scientifiques pour la Gestion Industrielle* (MSGI)
cole Nationale Suprieure des Mines
158, Cours Fauriel, Saint-tienne, 42023 Cedex 2, France
e-mail: {galland,grimaud,campagne}@emse.fr

KEYWORDS
Industrial system simulation, Distributed simulation, Modeling, Multi-agent systems, Methodological approach

Abstract

We are located in the context of industrial system simulation. These system are complex and distributed in operational, decisional and informational terms.

This article is divided into two parts. The first has for object to position us within the framework of a modeling and specification methodology, for which a study was suggested by [Galland, 1999]. We briefly expose the problems and the considered solution: a methodological approach called $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$. The following section presents the first major stage of the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$'s life cycle: specification of abstract simulation models. It allows to create industrial system models that will be sufficiently generic to be independent from the visions brought by the existing simulation tools. This phase is based on the concepts resulting from UML and the multi-agent systems. In addition we introduce new modeling elements that taking into account the operational, the decisional and the informational distributions of the industrial systems. We conclude this article by a short presentation of a specific editor and two applications of the theories released by $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$.

1 Introduction

Simulation is a tool, which is privileged and adapted to the modern industrial problems. It takes into account the dynamic aspects during the study of the production system behavior. The operational, informational and decisional distributions are still seldom managed within same tools. Moreover modern simulation tools are seldom accompanied by a complete and adapted methodology. In [Galland, 1999, Galland, 2000a, Galland, 2000b], we propose a multi-agent methodological approach for simulation ($\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$ [1]) that takes into account the three aspects of the distribution. However we want to mention the existence of interesting methodologies on which we based our work: ASCI [Laizé, 1998a, Grimaud, 1996], CM [Nance, 1981]...

$\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$ is a methodology based on a life cycle, whish is composed by five main stages [Galland, 2000b]. The first one is the analysis. It permits to write a needs' specification describing the goals of the simulation model under development, and the means necessary to satisfy those. The specification is the second stage during which an abstract simulation model was carried out. This last corresponds to a representation of an industrial system, whish is independent of any tool or methodology for simulation. The third major stage is the conception. A multi-agent oriented simulation model is generated from the abstract simulation model. This new model takes account of a distributed architecture, based on the multi-agent concepts [Ferber, 1995]. But it is independent from any particular agent platform. The next stage of the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$'s life cycle is the implementation. It permits to instance the multi-agent oriented simulation model on platforms and simulation

---

[1] **M**ulti-**A**gent **M**ethodological **A**pproach for **S**imulation

tools (SWARM, Arena®...). The experimentations compose the last major stage of our approach's life cycle.

In this article we present the phase of specification. We define a formalism based on Uml metamodels [Booch, 1997]. It takes into account the operational, decisional and informational subsystems of a production systems.

The following section is a short outline of the context in which our work is. In the section 3, we present the phase of specification of an abstract simulation model. Then we briefly expose related aspects : an editor and already carried out applications. Finally we conclude and expose our perspectives.

2 Problems and propositions

In this section we describe the framework of our research : the simulation of industrial systems. Then we present the various problems, which appear interesting to us. Finally we conclude by a brief presentation of our proposals.

2.1 Research domain

Since several ten years emerged a new concept that is able to contribute to the resolution of complex problems in various fields. This concept of systemic approach results from work from Von Bertalanffy, Wiener, Shannon, Forrester [Durand, 1975] and De Rosnay [De Rosnay, 1975]. It defines a system as a "set of elements in interaction, organized according to a goal". Within the framework of our research tasks, we are interested more particularly in the class of the industrial systems of production with discrete flows and resource sharing [Leroudier, 1980]. These systems must fulfill the functions of finished or semi-finished product manufacturing; of product or raw material transport; and storage. These systems are composed of active or passive resources (tools, workmen...), of flows (routing, nomenclatures...) and of supervising mechanisms (sensors, actuators...).

The production systems are confronted, at their design and exploitation times, with a certain number of structural, functional and organisational problems. They can be gathered in six categories:

- dimensioning (determination of the machine capacities...),
- operation ("Does bottleneck exists ?"...),
- productivity (minimization the number of resources...),
- maintenance,
- risks or breakdowns of the resources (behavior of the system in degraded mode...),
- scheduling (scheduling of the production...).

One of the means, which had the managers of production systems to answer these problems, is the use of simulation and more especially of discrete event simulation [Leroudier, 1980]. It allows the symbolic modeling of production systems, the support of stochastic phenomena and dynamisms of systems. Finally this category of simulation authorizes the use of events to change the system state. All these characteristics are adapted to the modeling of production systems.

2.2 Problems

A certain number of problems are met during modeling and simulation of industrial systems. In this section, we describe only those, which interest us.

2.2.1 Formalization

One of the first problems encountered during the phase of modeling for the manufacturing system's simulation is the lack of formal definition of the elements constituting the system. Thus the simulation tools strongly influence the vision of designers. For example, the Arena® and Simple++® tools do not offer the same modeling vision: the modeling elements are different. Moreover the quality of simulation models depends on the designer's competences. Rules, whish defined the structural and semantical modeling constrains, are rarely defined. The existence of these rules would be useful to help the designers to define and check their models. The problem of formalization is partially solved by existing methodologies as ASCI [Kellert, 1998a, Kellert, 1998b, Grimaud, 1996, Laizé, 1998b, Laizé, 1998a], CM [Nance, 1981, Nance, 1994a, Nance, 1994b, Laizé, 1998b, Page, 1994] or

(a) Company grouping

(b) Change of a simulation model

Figure 1: Examples of problems

IDEF [US Air Force, 1993, Mayer, 1992]. But they allow the modeling of industrial systems with a point of view that is not easily translatable into simulation models.

2.2.2 Modularity

Another problem currently encountered by the designers of simulation models is the low modularity. Indeed, even if the concepts of model and sub-model are often present in simulation tools, there remains difficult to build completely modular models. For example, the use of a sub-model already developed often forces to carry out a copy of it and incorporate it in the new simulation model. This duplication, though that useful, does not allow automatic reflect of changes in all the instances of the sub-models. It is necessary to modify each of them manually.

2.2.3 Highlighted of flows and subsystems

A production system is traversed by flows of entities and flows of information [Le Moigne, 1977, Le Moigne, 1992]. These flows are distinct even if there are very strong interactions between them. Currently, simulation models include these two types of flow without highlighting them one compared to the other. This vision of the modeling of production systems does not go without posing some problems. Indeed, the comprehension of the simulation model remains difficult because it is necessary to make an often consequent effort to differentiate entity flows and informational flows. Another problem is highlighted when the designer wishes to carry out a modification in one of the flows *e.g.*, change the management from pushed flow to pulled flow. In this case the strong overlap of flows imposes a complete rebuilding of the simulation model.

2.2.4 Centralization

A major problem encountered at the same time by methodologies and simulation tools is the centralization of models and simulation processes.

Currently, knowledge is often gathered within a same specialized team, or by only one designer. This vision can be sufficiant for many cases but sometimes it poses some problems inherent to centralized systems. For example, the designers of simulation models can have a too restrictive or incomplete vision of the production system. Moreover it is necessary to be able to centralize all information necessary to the good work of the simulation. Many problems of communications can exist between designers and people who really know the system (workshop manager, workman...).

The centralization of the simulation process not only poses knowledge problems but also generates increasingly significant difficulties: requirements in calculative resources are more and more important because of the simulate system dimensions, needs for managing the problems of confidentiality...

2.2.5 Examples

In order to better fix the significance of the problems than we wish to study, we will present two typical examples.

The first one, illustrated by figure 1(a), corresponds to a consortium. A grouping is interesting by its capacity to associate several distinct companies within the same organization. This type of structure characterizes well the whole of problems that we mention at the beginning of this part. Indeed, the members of the grouping having a autonomous life, they do not share the whole of existing information. In addition, certain members can want to keep secret some aspects of their organization or their management. Under such conditions, how to simulate the total behaviour of the grouping ? It is impossible to collect sufficient significant information to implement a coherent simulation model. In addition the dynamism inherent in the groupings *i.e.*, the evolution of the relations, forces the simulation model to be able to quickly adapt himself. Unfortunately, the current tools often force to rebuild the simulation models for each change in the simulated system.

To illustrate one second time the problems quoted above, let us take a simulation model of a production workshop. Figure 1(b) represents a workshop made up of three couples queue – machine. If we want to add two machines, it is advisable in general to completely rebuild the simulation model.

2.3 Propositions : $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$

2.3.1 Methodological approach

To partly solve the problems of the decentralization of information and knowledge as well as those of modeling and simulation delay, the scientific community works on implementations of distributed simulations according to two major approaches: the data-processing distribution of the models *e.g.*, synchronization problems [Filloque, 1992]; and knowledge distribution *e.g.*, the representation of the companies in a world context [Burlat, 1996]. Distributed simulation comprising these two axes takes account of the international characteristics of companies *i.e.*, the problems of technical culture, knowledge and geographical distribution can be fixed by a distributed simulation model. The existing tools are strongly dependent on a domain of activity *e.g.*, ARÉVi[2] concentrates on the virtual representation of systems [Duval, 1997, Chevaillier, 1997], SWARM is an simulation environment adapted to the artificial ways of life [Burkhart, 1994]. Finally certain tools take into account only one aspect of the model distribution *e.g.*, HLA [US Department of Defense, 1996] is an architecture that permits to make communicating simulation models but, in order to be usable in the major part of the cases, it is limited to the data-processing distribution of the simulation models.

We propose to conceive a methodological approach [Galland, 1999] based on the multi-agent concepts [Ferber, 1995]. We use the Voyelles approach (or AEIO) defined by [Demazeau, 1995]: a multi-agent system (MAS) is defined according to four major axes, which are the **A**gents, the **E**nvironment, the **I**nteractions and the **O**rganization. The multi-agent systems are adapted the three aspects of the industrial system distribution:

- **operational distribution**: The autonomy and the capacities of interactions of the agents allow to carry out, at the same time, the distribution within a data-processing network, and the distribution of the various parts of the industrial system by associating each agent with one of them;
- **informational distribution**: The cognitive and interactional capacities of agents permit to distribute information;
- **decisional distribution**: The cognitive mechanisms allow to set up the processes of catch and propagation of decisions.

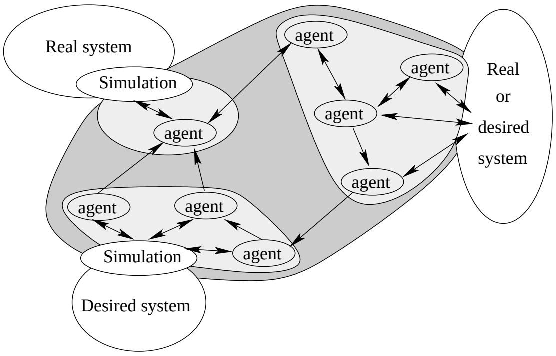


Figure 2: Example of a MAS architecture

Moreover the modularity generated by the use of the multi-agent systems permits us to answer another crucial point at the industrial level: the re-use of knowledge and the already known tools. Figure 2 is an illustration of a multi-agent architecture created during our methodology. It is made up of simulation models representing real or desired systems. These various models can be either the models handled by existing simulation tools as Arena® or Simple++®, or a society of agents carrying out the same task. These various models are inter-connected with another agent societies. The latter allow the transmission of information and entities between the remote simulation models.

2.3.2 Life cycle

$\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$ is based on the life cycle illustrated by figure 3 on the following page. It is composed of five main stages:

- **Analyze**: allows to write a needs' specification containing the abstract definition of the production system to be simulated, as well as the various objectives having to be reached by the data-processing simulation model;
- **Specification**: is the phase during which the designers build an abstract simulation model *i.e.*, a model representing the production system, whish is independent of any MAS-platform or simulation tool;
- **Conception**: has the role to transform the abstract model into a MAS oriented simulation model;
- **Implementation**: consists in choosing the simulation tools and MAS platforms that will instance the model produced during the conception;
- **Experiments**: is the last major stage of our life cycle. It consists of the experimental-design application on the data-processing simulation model.

In the following section, we describe one of the stages of the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$: specification of a abstract simulation model.

3 Specification of an simulation model

We present in this section the first significant stage of the life cycle of our methodological approach $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$. We start by defining what is the phase of specification. Then, we present a sufficiently general formalism to allow the modeling of complex and distributed industrial systems. We conclude this section by a significant aspect of the modeling, which is the checking of structural and semantical coherences of models.

3.1 Specification stage

The specification allows to create an abstract simulation model. We consider that this model is sufficiently general to be able to be translated into and used by methodologies or existing simulation tools. In particular, in the continuation of the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$'s life cycle, we transform this abstract simulation model into a MAS-oriented simulation model.

3.2 Formalism

In order to be able to meet our aims, we were brought to define a formalism including the basic concepts of the production system simulation, as well as possibilities of distributing whole or part of the models. We use the possibilities offered by the UML metamodels to reach our goals [Muller, 1997, Lopez, 1998]. Metamodels allow a formal and object-oriented definition of formalisms [Booch, 1997]. This one is based on the representation of an industrial system given by [Le Moigne, 1977, Le Moigne, 1992] : an industrial system was composed by the operational, informational and decisional subsystems.

Initially, we describe UML and its capacities to define new formalisms. Then we present our UML's metamodel.

3.2.1 Uml principles

UML is a modeling language normalized by the Object Management Group [Booch, 1997]. It synthesizes the knowledge on object-oriented modeling (OMT, Uses cases, ...). The most known aspect of UML is the whole of the formalism defined in its specifications. We find there the class diagram, the use case, or the state machine diagrams. UML is divided into three major layers:

- the *meta-meta-model* corresponds to the definition of the concepts necessary to create diagrams of classes (class, inheritance...),
- the *metamodel* defines, by the way of class diagrams, one or more formalisms (class diagram, use case...),
- the *model* is the instanciation of a metamodel (use case of an application simulating a production workshop...).

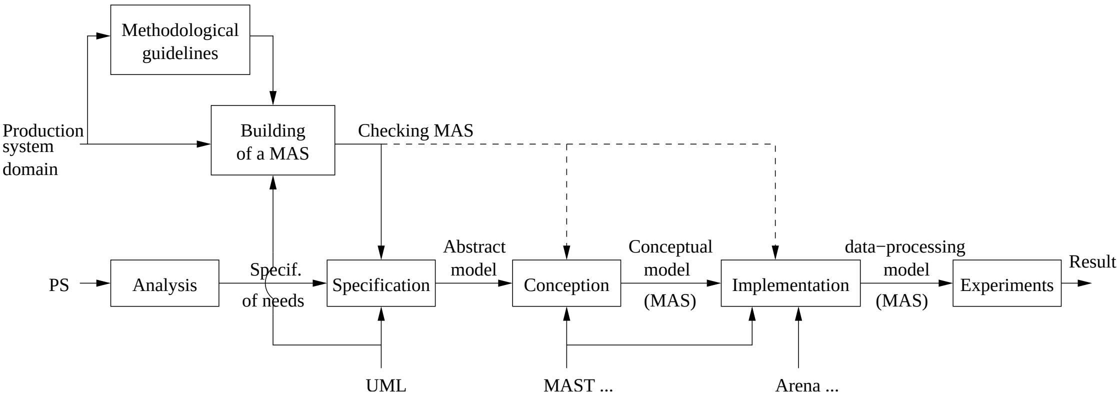


Figure 3: Life cycle of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$

With the concept of metamodel and mechanisms as stereotypes, it is possible to easily define new types of diagrams. In the continuation of this part, we present a metamodel allowing us to implement the concepts used during the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$'s specification phase.

3.2.2 Operational subsystem

We consider that the operational subsystem is the set of industrial infrastructures that compose the modelled system. These infrastructures can be modeled with basic concepts resulting from [Jullien, 1991] and softwares as ARENA® [Kelton, 1998] or SIMPLE++®:

- **Queue**: The **queues** permit to model the concept of tail in which the entities can wait until they can continue their way along the entity flow.
- **Server**: To be able to represent operational flows suitably, we introduce the concept of **processing units**. It corresponds to the modeling elements block the displacement of entities during a period of time. This one is determined by the way of a statistical law.
- **Ressource**: We introduce modeling elements corresponding to resources. We consider not-typified resources (workmen...) and typified resources. Among these last, we include the means of transport: station, road, conveyor and transporter. A **station** is a marker that allows to put a named element in the operational flow. Stations can be used to model the transport ways. A **road** is transport mean in whish only the temporal aspect is considerated. A **conveyor** is another means of transport holding account at the same time the space and the delay *e.g.*, length of the way. The characteristic of conveyors is that there has always a transport resource *e.g.*, a travelling carpet is a conveyor because it is defined by his space position, his speed of transfer and his availability in transport space. A **transporter** is a specialization of a conveyor. Like this last, the transporter takes into account the temporal and spacial aspects. But unlike a conveyor, the number of transport resources is limited *e.g.*, a pallet-lifting truck.
- **Flow managing**: The handling of flows corresponds to the whole of modeling elements carrying out a particular processing on the entities. The **generator of entities** permits to put in simulation models a point where the entities were created. On the other hand, the **destruction of entities** withdraws all the entities, which reach him. We include in this group the modeling elements allowing to represent flows: transitions, flow-junctions, flow-fork...
- **Entity**: The **entities** are the batches of raw materials, semi-finished or finished products forwarding in operational flow.
- **Hierarchisation**: The **models** are groundworks representing a operational subsystem and containing a whole of modeling elements. The latter belong to the other categories or are **sub-models**

i.e., a model whose entering and outgoing flows are used by an including model.

- **Statistic**: Finally the simulation models allow to include random factors via **statistical laws** used in the whole of the model. Moreover this group contains the elements that permit to collect statistical data.

In order to be able to handle these concepts, we defined a formalism according to a UML metamodel. The figure 4 illustrates the concept definition of models, statistical laws, attributes and remote objects in our formalism.

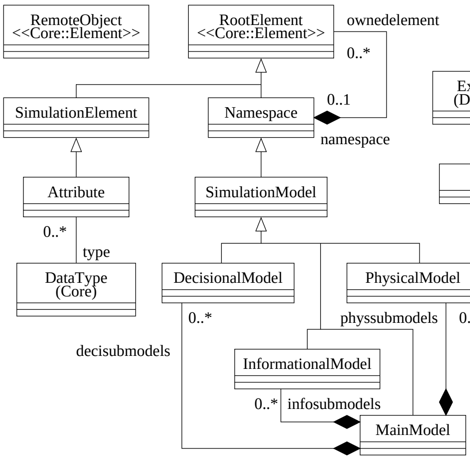


Figure 4: Part of the general metamodel

This metamodel defines initially a name space as an object containing modeling elements (ROOTELEMENT). Each model is defined as a specialization of a name space. There exists four standard models : one for each subsystem (PHYSICALMODEL, DECISIONALMODEL, INFORMATIONALMODEL) and one for representing the entire industrial system (MAINMODEL). Our metamodel defines also statistical laws (LAW) as well as modeling elements belonging to models (SIMULATIONELEMENT).

The formalism that describes the operational subsystem was built in the same way. The figure 5 illustrates a small part of it. It corresponds to the definition of the modeling elements corresponding to the transport means. As we indicated above, a road (ROUTE) allows to reach a destination. The elements of type TRANSPORTELEMENT contain a statistical law representing a duration, then the roads take into account the temporal aspect of transport. The two other types of transport (conveyor and transporter) are similar to the road but they support space constraints. These constraints are modelled using a path made up of “checkpoints” (TRANSPORTCHECKPOINT). As we already mentioned, the difference between a conveyor and a transporter lies in the limitation of the number of the transport resources. From a modeling point of view, only the checkpoint where the transport resource was initialy is interesting.

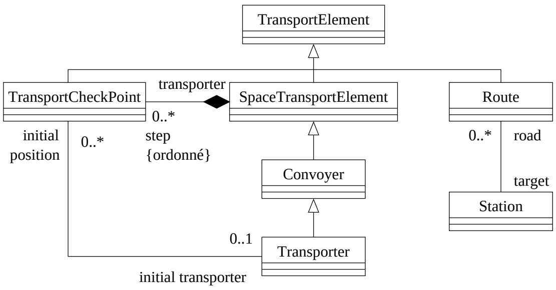


Figure 5: Part of the operational subsystem metamodel

3.2.3 Decisional subsystem

The decisional subsystem is composed by the organisational and the decisional infrastructures of industrial systems. Our UML metamodel defines the modeling formalism of this subsystem. It allows to represent a relational structure between organisational entities or **decision-making centers**. These last can make decisions with short (operational), medium (tactical) or long (strategic) horizon. The relations between these centers can have a hierarchical or simply relational nature. This vision results from [Kabachi, 1999] but also from work on the organisational structures in the multi-agent systems [Hannoun, 2000].

Each decision-making center is associated with a whole of protocolar or reactive **behavioral models**. These models are used according to the decisions taken or to the events occurring in the system. They also permits to define the semantical constraints having to be respected by the simulation model. For example, if a decision-making center uses the contract-net protocol and it plays the role of service provider, it is necessary that the simulation model contains at least one decision-making center that was playing the role of service consumer.

Figure 6 on the next page illustrates part

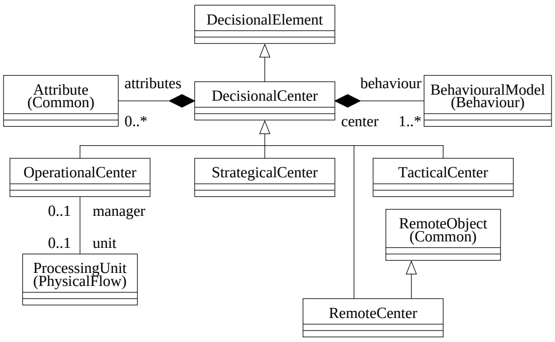


Figure 6: Metamodel part of the decisional subsystem

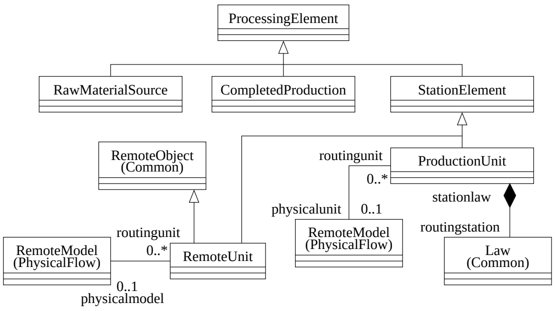


Figure 7: Metamodel part of the informational subsystem

of the metamodel that defining the formalism attached to the decisional subsystem. It represents the definition of the various types of decision-making centers: operational, tactical and strategical. We also see appearing the concept of “remote center”, which corresponds to a center whose real definition is in a remote simulation model. Each center can be attached to several behavioral models. These models are descriptions of the decision-making mechanisms. By default, we include in our metamodel two types of behavioral models : *reactive* models allow centers to *immediately* react to a set of stimuli, and *protocolar* models correspond to the definition of interaction protocols with other centers *e.g.*, contract-net protocol...

3.2.4 Informational subsystem

The informational subsystem is the description of information that are necessary to model and simulate an industrial system. We include in our UML metamodel the models of **nomenclatures** and **routing**. We include too in this subsystem the definition of some categories for decisional messages, whish are exchanged by the decision-making centers. The figure 7 illustrates the elements of the routing model: production units, raw materials sources...

3.3 Checking of the model

The creation of a simulation model imposes to check his coherence. The UML metamodel permits to carry out this checking *i.e.*, the simulation model produced by the user should not be in conflict with the definition of the class diagrams and with the construction rules expressed in our metamodel. Indeed, in addition to the class diagrams that graphically express the structural and semantic constraints models, the UML specifications allows to include rules of good construction that describe more precisely the constraints associated with the defined formalism. These rules are generally expressed with a formal language as the Object Constaint Langage [Booch, 1997] or the first-order logic.

The UML specifications not explicitly describe the implementation technics of the model checking, each tool uses its own (Rational Rose, Argo/UML, Mygale...). However we can note that all these tools obtain the same result : the validation of a model according to a metamodel.

The model validated by the UML checking is correct from the formal point of view. But it persists a problem concerning the use of the distributed modeling elements. Indeed, the UML coherence checking is not sufficient to be sure that the simulation model produced was correct. The use of a metamodel only allows a *local* checking of the model. The modeling elements representing the remote objects do not know the real values of the attributes of the latter. For example, when we use human resources defined in a remove simulation model, it is impossible only by the means of the UML metamodel to know if this type of resource is usable with a specific machine-tools.

The solution for these problems is the definition of a multi-agent system of which the goal was to carry out the consistency on the various distributed objects. The agents exchange correctness rules and thus are able to carry out tests. The multi-agent concepts are adapted to these problems of checking. They are dynamics: the properties of the elements supervised by the agents can change at any time; they are also open: modeling elements can be withdrawn or added; finally they are distributed: elements belong to simulation models being able to be distributed.

4 Editor

In order to check the applicativity of our theories presented in the preceding sections, we developed an editor allowing to build abstract simulation models. From a functional point of view, this editor allows to create a model for each subsystem composing the industrial system. He also implements the technics that check the coherence of distributed models. Local coherence that we proposed in section 3.3 is carried out according to our UML metamodel. Indeed, our editor is based on the library Mygale[3], which permits editions of graphs defined by the way of UML metamodels. The distribution coherence within models is checked by a multi-agent system running in parallel to our editor.

5 Applications

Within the framework of the $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$'s development, we used an iterative approach in order to allow us to gradually answer the various encountered problems. This approach illustrated by figure 8(a) on the following page enables us to iterativly develop the various aspects of our methodology : analysis, specification, conception, implementation, multi-agent system (AEIO)...

This approach enables us to develop a first implementatin version of our simulation agents. Those are primarily vectors of communications between several instances of the ARENA® software. They were used in the development of a teaching application [Galland, 2000c]. Figures 2 and 8(b) illustrate respectively the use and the architecture of these agents. However we will not go into the implementing details in this article.

Another application of our theories is presented in [Campagne, 2001]. It is about the evaluation of the cyclic production by the use of simulation in an automobile equipment supplier.

6 Conclusion et perspectives

Simulation is recognized as a tool adapted to the study of the industrial system behavior. But even if this technic takes into account the dynamic aspects, it allows only too seldom the operational, informational and decisional distributions. In addition, the modern tools are seldom accompanied by adapted methodologies. On the basis of this observation, we proposed in [Galland, 1999] a methodological approach for the simulation based on the concepts of the multi-agent systems: $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$.

In this article we have the results of the first phase of development. We suggest that a methodology integrating the operational, informational and decisional distributions must itself be distributed *i.e.*, the various submodels necessary to the development of a simulation model could be developed in parallel. Once to expose our point of view on these problems and the life cycle of a simulation model [Galland, 2000b], we present the first significant phase: specification. It permits to create an abstract simulation model *i.e.*, a model based on general components (machines, queues...) and independent of any simulation tool ans MAS platform. We base our approach on the UML metamodels, which allow the definition of formalisms and their basic structural checking. But these metamodels not directly take into account the distributed models directly, we introduce a multi-agent system that will realize the structural and semantic checks.

In the future, we propose to define the various concepts used and usable during the phases of conception (MAS-oriented model) and of implementation (MAS plateform and simulation tools). We will check the cogency of our theories by applications on industrial and teaching cases [Galland, 2000c, Campagne, 2001].

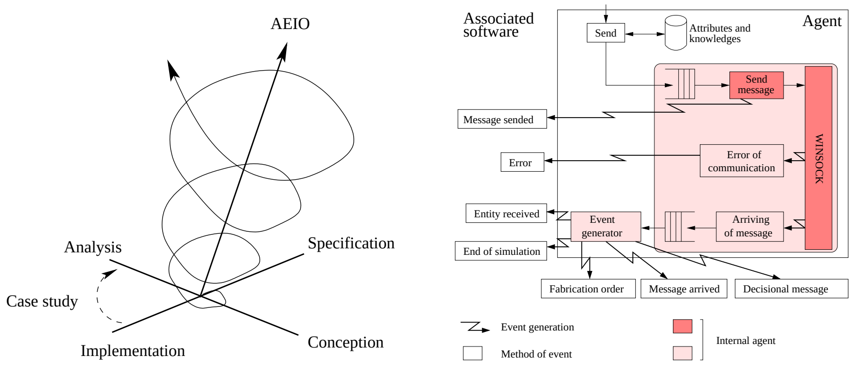
References

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

Burkhart R., 1994. The Swarm Multi-Agent

---

[3] `http://sgalland.multimania.com/tools/mygale/`



(a) Iterative developping approach

(b) Architecture of the first agents

Figure 8: Applications

Simulation System. *Proceedings OOPSLA Workshop on "The Object Engine"*.

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.

Chevaillier P., Querrec R., et Tiseau J., 1997. Modélisation Multi-agents d'une Cellule de Production. *Proceedings 2ème Colloque National de Productique*, Casablanca - Maroc, p.1-10.

De Rosnay J., 1975. *Le macroscope*, .

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

Durand D., 1975. La systèmique. *Proceedings Que sais-je ?*

Duval T., Morvan S., Reignier P., Harrouet F., et Tisseau J., 1997. ARéVi : Une Boîte à Outils 3D Pour Des Applications Coopératives. *"La coopération"*, 9(2), p.239-250.

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

Filloque J.-M., 1992. *Synchronisation Répartie sur une Machine Parallèle à Couche Logique Reconfigurable*. Thesis, Institut de Formation Supérieure en informatique et Communication - Université de Rennes 1.

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., Grimaud F., et Campagne J.-P., 2000a. Methodological approach for distributed simulation : General concepts for $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$. *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., 2000b. Methodological approach for distributed simulation : Life cycle of $\mathcal{M}_{\mathcal{A}}\mathcal{MA}$-$\mathcal{S}$. *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.

Galland S., Grimaud F., et Campagne J.-P., 2000c. Multi-agent architecture for distributed simulation : Teaching application for industrial management. *Proceedings Simulation and Modelling : Enablers for a better quality of life – 14th European Simulation Multiconference*, Van Landeghem R. (Eds.), Ghent, Belgium, p.756-762.

Grimaud F., 1996. *Conception D'une Base de Composants Logiciels Pour L'évaluation Des Performances Des Entreprises Manufacturières.* Thesis, Université Blaise Pascal, Clermont-Ferrand.

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.

Jullien B., 1991. *Simulation des Systèmes de Production, tome 1*, École Nationale Supérieure des Mines.

Kabachi N., 1999. *Modélisation et Apprentissage de la prise de décision dans les organisations productives.* Thesis, Université Jean Monnet / ENSM.SE, Saint Etienne, France.

Kellert P. et Force C., 1998a. Méthodologie de Modélisation Orientée Objets de Systèmes de Production – Application à nne Chaîne d'Étuvage de Bobines d'Allumage. *Journal Européen des Systèmes Automatisés*, 32(1), p.107-131.

Kellert P. et Ruch S., 1998b. Méthodologie de Modélisation Orientée Objets de Systèmes de Production – Un Processus de Construction/Validation du Modèle Générique Orienté Objets d'un Système de Production. *Journal Européen des Systèmes Automatisés*, 32(1), p.51-105.

Kelton W., Sadowski R. P., et Sadowski D. A., 1998. *Simulation with Arena*, McGraw-Hill.

Laizé E., 1998a. *Proposition D'une Méthodologie de Modélisation Orientée Objets de Systèmes Manufacturiers de Pneumatiques : La Méthodologie ASCI_SM*, , 4, p.93-146.

Laizé E., 1998b. *Méthodologie de Modélisation*, , 2, p.37-60.

Laizé E., 1998c. *Méthodologie de Modélisation de Systèmes Manufacturiers de Pneumatiques : Spécification de la Connaissance.* Thesis, Université Blaise Pascal.

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.

Leroudier J., 1980. La simulation à évènements discrets. *Proceedings Monographies d'informatique de l'AFCET.*

Lopez N., Migueis J., et Pichon E., 1998. *Intégrer UML dans vos projets*, Eyrolles.

Mayer R. J., 1992. *Information Modeling (IDEF1) – A reconstruction of the original Air Force wright aeronautical laboratory technical report.* Knowledge Based Systems, Inc., Technical report AFWAL-TR-81-4023, College Station, Texas, USA.

Muller P.-A., 1997. *Modélisation objet avec UML*, Eyrolles.

Nance R., 1981. *Model Representation in Discrete Event Simulation: The Conical Methodology.* Department of Computer Science, Virginia Polytechnic Institute and State University, Technical report CS-81003-R, Blackburg, USA.

Nance R., 1994a. Conical Methodology : An Evolutionary Convergence of Systems and Software Engineering. *Annals of operations research*, 53.

Nance R., 1994b. The Conical Methodology and the Evolution of Simulation Model Development. *Annals of Operations Research*, 53, p.1-45.

Page E. H., 1994. *Model Generation*, .

US Air Force, 1993. *Integrated Computer Aided Manufactured Definition Language (IDEF methods).* Department of Commerce, National Institute of Standards and Technology, Computer Systems Laboratory, Technical report, Gaithersburg, USA.

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.