Chapter 5

MODELING AND SIMULATING DISTRIBUTED INDUSTRIAL SYSTEMS

A Multi-Agent Methodological Approach

Stéphane GALLAND[1], Frédéric GRIMAUD[2],
Philippe BEAUNE[3], Jean-Pierre CAMPAGNE[4]

[1] *Systems and Transportation Laboratory (SeT)*
University of Technology of Belfort-Montbéliard (UTBM)
F-90010 Belfort, France
Email: stephane.galland@utbm.fr

[2] *Scientifical Methods for Industrial Management Team (MSGI)*
Higher National School of Mines (ENSM-SE)
F-42023 Saint-Etienne, France
Email: grimaud@utbm.fr

[3] *Multi-Agent System Team (SMA)*
Higher National School of Mines (ENSM-SE)
F-42023 Saint-Etienne, France
Email: beaune@emse.fr

[3] *Productic and Computer Science for Manufacturing Systems (PRISMA)*
National Institute for Applicative Science (INSA)
F-69621 Villeurbanne, France

Abstract: We are located in the context of the industrial system simulation, which are complex and distributed in operational, informational and decisional terms. In this chapter, we present the problems and a methodological solution. This methodology is based on the systemic approach and on multi-agent systems. It allows the modelling of distributed industrial systems such as enterprise consortiums. Moreover, it proposes a software platform architecture whish is currently instanced with Arena and dedicated agents.

Keywords: Industrial System, Discrete-Event and Distributed Simulation, Multi-Agent Systems, Decision-making process.

1. CONTEXT AND PROBLEMS

The simulation is a tool adapted to the studying of modern industrial problems and more precisely the dynamic behaviour of industrial systems [4]. In this context, the support of the new industrial organizations is particularly focused. An example of new industrial organization is the enterprise consortiums, which is a whole of companies related the ones to the others by a cycle of production. The bond is neither legal, nor structural; it has often the form of simple agreements. These companies have in common a powerful system of functional cooperation [2].

Even if the simulation is powerful, some problems always exist. This chapter is concentrated around four of them. First, the simulation tools are still seldom packaged with a dedicated methodology. They have a strong influence to the designers' point of view. For example, Arena® and Simple++® offer two modelling views which are similar and different in the same time: the modelling concepts are similar but they are not used or defined in exactly the same way. This formalisation problem is partly solved by existing methodologies

The second problem is the poor support of the component-based or modular modelling.

Next, the strong relationship between the physical, the informational and the decisional aspects of an industrial system is also highlighted. Currently, the simulation models include these two kinds of flows. But they don't highlight each of them. Then the understanding is still difficult according to the necessity to mentally distinguish them. Another example is when a designer wants to update the decisional (e.g. the management policy). Then, in most of cases, he must remodel and rewrite all the models to include this change.

Finally, the last problem is about the difficult to model the new industrial organizations, such as enterprise consortiums or virtual enterprises. This problem has two sides: the modelling and the simulation. The modelling of distributed industrial systems is not naturally supported by the tools. Moreover, all the tools do not accept to simulate on a computer network. This constraint is for instance introduced by the confidentiality imposed by the consortium members.

To solve these different problems, a methodological approach is proposed: *MaMA-S* (Multi-Agent Methodological Approach for the Simulation of industrial systems) [6,7]. It offers a modelling framework that is independent of any software platform (simulation tool or multi-agent system). In the rest of this chapter, the major concepts attached to this methodology are presented. More precisely, the life cycle and the major propositions on this methodology are explained.

2. METHODOLOGICAL APPROACH

This section presented the major propositions about *MaMA-S*. See [6,7] for more details.

2.1 Life cycle of the *MaMA-S* models

The development of a methodological approach passes by a first major stage: the definition of the life cycle of the models. This section is devoted to the definition of this life cycle for the models of distributed industrial systems. Starting from the assets of the software engineering and the works already completed in the field of simulation, an extension of the existing approaches is proposed to take the new enterprise organizations into account.

The figure 2.1 illustrates the life cycle used by *MaMA-S*. Their contributions are restricted to the adaptation of the *specification*, the *design*, and the *implementation*. Two special stages are also included: the *methodological guidelines* and the *coherence checking.*

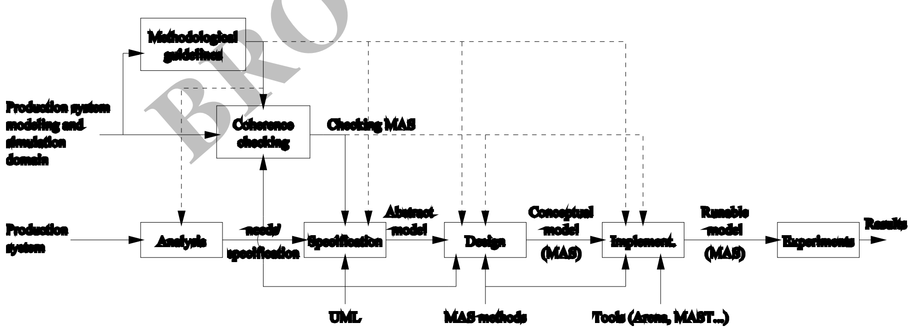

Figure 2-1. Life cycle of the *MaMA-S* models

The methodological guidelines are written during the stage of the same name. It permits to specify the principles of the methodology (life cycle, modelling elements, methods...). The guidelines are, at the same time, the specifications of and a user guide. Currently, they are limited to the specification of from [6,7]. It will evolve according to the progresses of the works on *MaMA-S*.

The coherence checking aims to check the coherence of the different simulation models. This stage is not presented in this chapter. You could read [6,7] for more details.

The other adapted stages are presented in the following sections.

2.2 Phase of Specification

The phase of specification is crucial in *MaMA-S*. Indeed, it corresponds to the moment when the first formally expressed model must be produced. In this section, the methodological bases and the subjacent principles of the abstract model's specification are presented.

The modelling elements are used within the framework of the specification for the creation of an abstract simulation model. This building must be carried out starting from the information collected and exposed in the needs' specification. This methodological approach considers that the distributed production system can be broken up according to the systemic approach proposed by Jean-Louis Le Moigne [8]: an operational subsystem, an informational subsystem and a decisional subsystem. Some modelling concepts are proposed for each of these subsystems.

a) Physical subsystem

The physical subsystem is the whole of the industrial infrastructures of the modelled system. The basic concepts supported by are partly from [1]:

- **Composition**: the concepts of model and sub-model;
- **Critical resources**: the resources which can stop the physical flow.
  - active resources: used to realize an activity (processing units, human resources, transportation means…)
  - passive resources: used by the active resources to realize these activities.
- **Queue**: it represents an ordered list of physical entities waiting for a specific event;
- **Structural modelling of the physical flow**: a set of additional modelling elements that permit to define the paths used by the physical entities (links, junctions, forks, jumps, exit points and entry points).
- **Distribution**: the whole of modelling elements that allows the definition of a distributed model. They are defines in [6,7].

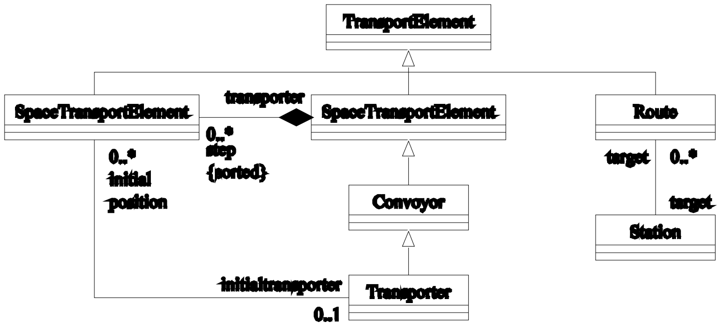

Figure 2-2. Part of the physical sub-system meta-model

The modelling artefacts are defined in an extension of the UML meta-model. The figure 2.2 illustrates a part of the physical subsystem meta-model. It corresponds to the definition of the modelling elements for the transport means. The roads (Route) permit to reach a destination. The elements of type TransportElement contain a stochastic law for the transport duration. Thus, a road supports the temporal aspect of the transport. The two other kinds of transport means (conveyors and transporters) extend the concept of road by including a spatial aspect. The difference between a conveyor and a transporter is the limitation of the transportation resources in the second.

To permit an easier design of the simulation model, a graphical language attached to each object defined in the meta-model is proposed in [6,7].

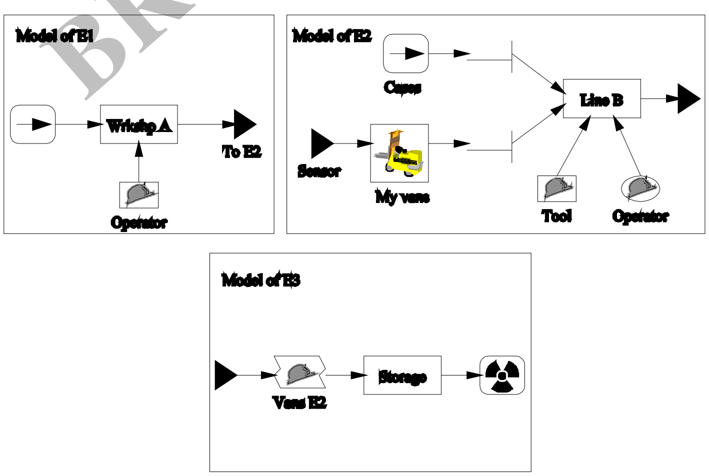

Figure 2-3. Example of a physical sub-system model

To illustrate the operational subsystem modelling, we propose to create a model of a simple distributed system: a consortium of three enterprises E_1, E_2 and E_3. The objective of this consortium is to produce movement-detecting cameras.

The enterprise E_1 produces sensors in its workshop *A*. They are sent to the second enterprise. This last assembles in its workshop *B* the sensors with the cases which are locally manufactured. Then, E_2 forwards the resulting detectors to the third consortium member which must only store the final products. The transport between these three enterprises is exclusively carried out by the transport services of E_2. Additionally, the workshop *A* uses a critical resource: an operator, and the workshop *B* uses two resources: a critical resource which permits to assemble the cameras, and a passive resource representing an supervisor. Moreover, the agreements between the consortium members specify that E_1 does not know how the sensors are transported, and that E_2 does not force E_3 to use its transport services. These considerations enable us to put the transportation modelling artefacts in the models of E_2 and E_3. The figure 2.3 graphically illustrates the three resulting models.

b) Decisional subsystem

The decisional subsystem is the whole of the organisational structures and decision-making processes of the industrial system. The UML meta-model of *MaMA-S* defines a language that permits to describe the relational structures between the *decision-making centres.* The centres can take operational, tactical or strategic decisions. The relationship between the centres can be hierarchic or cooperative. This point of view is issued from the works on the organisational structures in industrial systems and in multi-agent systems.

Each decision-making centre includes at least one *behavioural model.* Each of them could be a protocol based on state-transitions, on stimuli or on both of them.

Let us take again the example of the consortium of three enterprises. Here, we are interested exclusively in the decisional subsystem modelling. First, the focus is on the management policy. The enterprise E_3 is in direct relation with the “market”. Each time its stock of cameras does not permit to answer to an order, this enterprise sending a production order to E_2. This one has pulled flow management policy. For each production order coming from E_3, it creates a corresponding production order to E_1. This last launches the production of the number of sensors claimed by E_2. The production process uses a pushed flow

management policy. The figure 2.4 illustrates the decisional models for each of the three members of the consortium.
Each decisional centre has its own behaviour (the used language is defined in [6,7]). For instance, the production controls of E_1 and E_2:

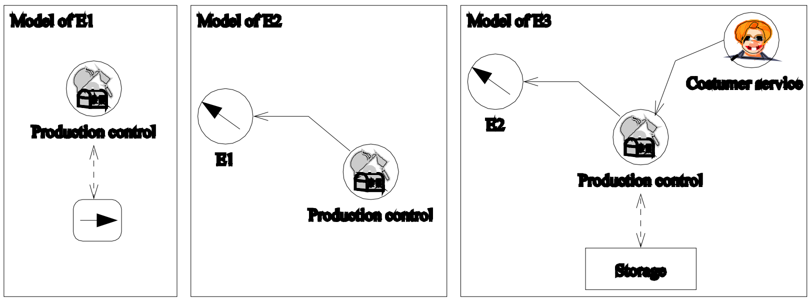

Figure 2-4. Example of a decisional sub-system model

– **Production control of E_1:**
This centre generates a physical entity for each sensor having to be produced. Its behaviour can be defined as follow:

```
CONTEXT “Production control”
WHEN RECEIVE “Production order”
WITH PARAMS ( “size of the PO” )
THEN
  i = 1;
  WHILE ( i <= “size of PO” )
  DO
     OPERATION( generate-entity,
        INFORMATION(entity) ) ;
     i = i + 1 ;
  DONE
END
```

– **Production control of E_2:**
This centre generates a production order of sensors towards E_1 for each order coming from E_3:

```
CONTEXT “Production control”
WHEN RECEIVE “Production order”
WITH PARAMS ( “size of the PO” )
THEN
  SEND TO E1
     TYPE ORDER
     NAMED “Production order”
     DATA “size of PO” *
        INFORMATION(
```

```
            bill-of-material-camera
            “quantity of sensors” ) ;
END
```

c) Informational subsystem

The informational subsystem contains the information used by the two other subsystems. In the *MaMA-S* modelling language, the concepts of *bill of material*, *manufacturing routing* and *entities* (physical or decisional) are defined.

Let us take again the example of the already presented consortium. Consider the models of bills of materials (figure 2.5). Each enterprise has its own vision of the products. However, the sensor used by E_2 is in fact the definition being in the model of E_1. This is a simple usage example of a distant product definition. The bill-of-material model of E_3 is the same as the model of the enterprise E_2.

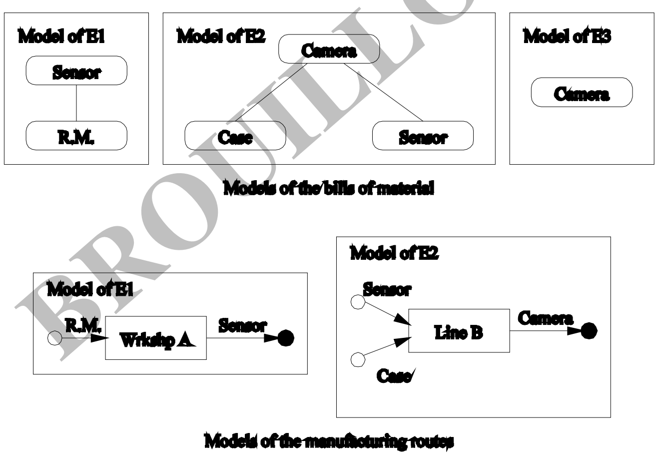

Figure 2-5. Example of an informational sub-system model

Consider now the manufacturing routing. In our example, only the first two enterprises must define a manufacturing routing model. Indeed, E_3 does not carry out any transformation on the products. The figure 2.5 illustrates the graphical representation which is proposed within *MaMA-S*. Thus, in E_1, the raw material (`R.M.`) is transformed into sensors by the workshop *A*, and in E_2, the cameras are obtained starting from the assembly of the sensors and the cases. Note that, for each manufacturing routing model, the treatment units must be associated to the corresponding processing units from the operational subsystem (they must have the same name). In addition, a treatment must define a whole of times necessary to model the processing durations.

2.3 Phase of Design

The conceptual model is a multi-agent system (MAS) model [5] that corresponds to a translation of the abstract model shown in the previous section. It describes the structural organization of the agents. But, it does not force to use a particular multi-agent platform or a particular simulation tool. The only one constraint is that it must respect a specification according to the approach “Vowels” (or AEIO) [3].

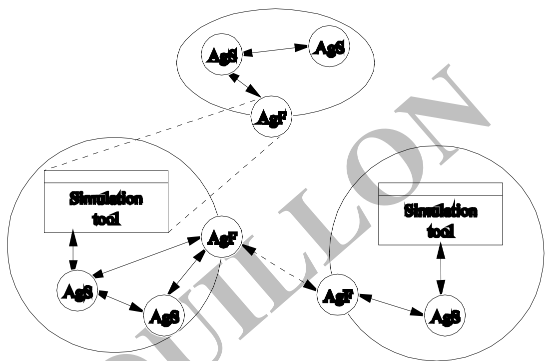

Figure 2-6. Multi-Agent Architecture allowing simulations of industrial systems

An infrastructure is proposed to allow a simulation process based on an agents' society (illustrated by the figure 2.6). It is mainly composed of interconnected agents. This principle permits to distinguish two classes of agents:

– the facilitators (AGF) facilitate the exchange of messages between simulation agents. They are intermediaries between agents' sub-societies. Moreover, the facilitators dynamically manage a knowledge database of the resources and the available services (resources, processing units, decisional centres...). They allow a better modularity and better dynamic’s evolution support.
– the agents for simulation (AGS) are used during the simulation process for the decisional centres. The architecture is based on “white boxes” that must be filled with the behaviour defined in the abstract simulation model. This kind of agent is not entirely specified in *MaMA-S*. The proposed architectural skeleton is based on the AEIO's facets [3]. This skeleton includes the interactions between the facilitators, and, in most of

cases, between a facilitator and the environment's objects (such as a simulation tool) [6,7].

The figure 2.6 illustrates another aspect of the *MaMA-S* approach: the recursive architecture. In fact, each agent or each environment object can be also a multi-agent system.

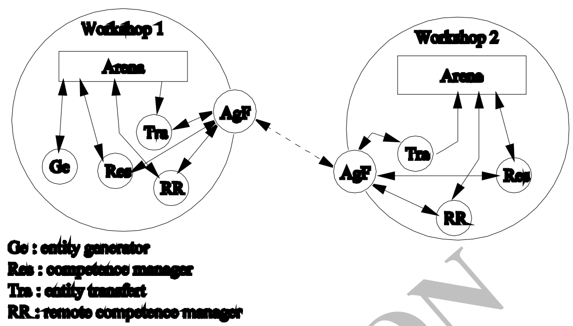

Figure 2-7. Example of the architecture of a multi-agent model

From the previous example shown for the specification, the following agent classes are highlighted:
– the agents corresponding to resource management decision centres,
– the agent corresponding to the entity generation centre,
– the agents that permits to send physical entities from a simulation model to another,
– the agents representing the “remote” resource managers.

The figure 2.7 illustrates the structure of the resulting multi-agent system. *MaMA-S* considers that agents are white boxes, which must be filled with the parts of the abstracts models. The means to interact between agents is proposed by *MaMA-S* (message syntax, service specification...).

2.4 Phase of Implementation

The implementation is the last phase which is adapted to the support of the modelling and the simulation of distributed industrial systems. Here, the objective is to translate the multi-agent model previously obtained into a computer model. Within this intention, this phase aims to choose the tools which will have to carry out the simulation (Arena®, Simple++®, QNAP…)and the multi-agent platform being used as support of the agent execution (SWARM, MadKit, Zeus, CORMAS, ARéVi...).

To tackle the translation of the multi-agent model, we propose a set of constraints that must be respect by the software. The major of them are:

– **for the simulation tools**:

The simulation tools must propose a communicating interface usable by the agents.

They must implement a set of behaviours whish are strictly equivalent to those awaited in the conceptual model.

The tools should not endanger the course of simulation. Thus, the simulation tool must be in conformity with the synchronisation policy of the models (the constraints of causality and vivacity must be respected).

– **for the agent platforms**:

The agents result directly from the conceptual model. Thus, the constraints of implementation are common to any multi-agent system (autonomy, interaction, distribution)

An implementation was proposed for the previous consortium example. It is based on the use of the simulation tool Arena®, and of Visual Basic® for the agents. The figure 2.7 also illustrates this choice of implementation.

Arena® permits to support the physical infrastructure of the system and to simulate the flow of physical entities inside this sub-system.

Visual Basic® is used to implement simple agents composed of a small communicating layer (based on sockets and message queues) and simple responding algorithms which correspond to the behaviours defined during the design. Moreover, those agents are temporally synchronized with a pessimistic approach.

3. CONCLUSION AND FUTURE WORKS

In this chapter, a methodological approach for the creation of simulation models of complex and distributed systems is proposed and presented. This approach, named *MaMA-S*, was previously specified in [6]. It facilitates the modelling and the simulation of decision-making processes, whether centralized or distributed. It also provides better reaction capacity in terms of modelling (systemic approach, etc.). Lastly, one of its strengths is its capacity to produce re-usable models.

Nevertheless, some applications (teaching application in the simulation scope and cyclic scheduling of a production system) enable to highlight certain weaknesses in our approach (see [6,7] for more details). First of all, collaborative modelling is not fully taken into account by *MaMA-S*. Indeed, only a basic architecture for the collaborative support is proposed. In addition, some modelling elements need to be developed and proposed inside *MaMA-S* (dedicated template, etc.). They should allow an easier modelling. Moreover, this chapter presents the first works completed on *MaMA-S*. It still remains of many points for which a study proves to be necessary (synchronization of the simulation models, methodological guidelines…).

Finally, this methodological approach needs to move towards the standardization attempts that concern our areas of investigation: (i) the Unified Enterprise Modelling Language (UEML) for the modelling of distributed production systems, (ii) the Discrete-Event systems Specification (DEVS) to support distributed and interoperable simulations, (iii) the Foundation for Intelligent Physical Agents (FIPA) whose aims is to define the set of components for a multi-agent system platform. The previous points are currently being developed within the frame of an extension of the *MaMA-S* methodological approach.

REFERENCES

1. J. Banks, 1999. Introduction to simulation. In: *Proceedings of the Winter Simulation Conference*. P.A. Farrington, H.B. Nembhard, D.T. Sturrock and G.W. Evans, editors. 7-13.
2. F. Butera, 1991. La métamorphose de l'organisation : du château au réseau. Chapter of *Les nouveaux modèles d'entreprises : vers l'entreprise réseau*. Editions d'organisation, Paris, France.
3. Y. Demazeau, 1997. Steps towards multi-agent oriented programming. In: *Proceedings of the 1st International Workshop on Multi-Agent Systems* (IWMAS), Boston, USA.
4. L. Dupont, 1998. *La gestion industrielle*. Hermès, Paris, France.
5. J. Ferber, 1995. *Les Systèmes Multi-Agents : Vers une intelligence collective*. InterEditions. Paris, France.
6. S. Galland, 2001. *Approche multi-agents pour le conception et le construction d'un environnement de simulation en vue de l'évaluation des performances des ateliers multi-sites*. PhD thesis, Ecole Nationale des Mines, Saint-Etienne, France. December 2001. http://set.utbm.fr/membres/galland/recherche/publis/.
7. S. Galland, F. Grimaud, P. Beaune and J.P. Campagne, 2003. *MaMA-S*: an introduction to a methodological approach for the simulation of distributed industrial systems. International Journal of the Production Economics, 85(1), 11-31. June 2003. http://set.utbm.fr/membres/galland/recherche/publis/.
8. J.L. Le Moigne, 1992. *La modélisation des systèmes complexes*. Editions Dunod, Paris, France.