Chapter 2
A Generic Knowledge Model for SME Supply Chain Based on Multiagent Paradigm

Jihene Tounsi, Julien Boissière, Georges Habchi and Van-Dat Cung

Abstract This chapter is dedicated to the generation of an agentified knowledge model for modeling and simulation of the supply chain in Small and Medium Enterprises (SME) context. For this purpose, the organization of this work is directed by the **ArchMDE** development process which is founded on Model Driving Engineering (MDE). The fundamental contributions concern two research areas. The first one concerning the industrial engineering scope, proposes a generic domain meta-model (i.e. supply chain integrating SME) identifying the functional concepts and their properties. The second one considering the computer engineering scope, highlights all steps necessary to integrate the dynamic behaviour into the domain meta-model built according to the multi-agent technology. In this perspective, this chapter describes the outcome of these artefacts from the study of the domain through the agentification process, to the implementation of the knowledge model.

J. Tounsi (✉)
University of Tunis, SOIE, 41, Rue de la Liberté cité Bouchoucha,
2000 Le Bardo-Tunis, Tunisia
e-mail: tounsi.jihene@yahoo.fr

J. Boissière
LISTIC-Polytech'Annecy, University of Savoie, 12, Chemin de Bellevue,
80439 74940 Annecy-le-Vieux, France
e-mail: julien.boissiere@univ-savoie.fr

G. Habchi
SYMME-Polytech'Annecy, University of Savoie, 12, Chemin de Bellevue,
80439 74940 Annecy-le-Vieux, France
e-mail: georges.habchi@univ-savoie.fr

V.-D. Cung
Grenoble INP/UJF/CNRS, Laboratory G-SCOP, 46, Avenue Felix Viallet,
38031 Grenoble Cedex 01 Grenoble, France
e-mail: van-dat.cung@grenoble-inp.fr

H. K. Chan et al. (eds.), *Decision-Making for Supply Chain Integration*,
Decision Engineering 1, DOI: 10.1007/978-1-4471-4033-7_2,

Keywords Knowledge model · Multi-agent system · Decentralised supply chain · Small and medium enterprises (SMEs)

2.1 Introduction

The supply chain concept was born in the 90's when management techniques in the business world were evolving from independent to collaborative logistics. It is well known that the supply chain is a complex macro system. This complexity is firstly due to the variety of the involved organizations and the diversity of relationships between them, and secondly it results from the decision-making mechanisms between these companies. Thus, the success and subsistence of a company in the economic market rely not only on its ability to integrate managerial processes but also on coordinating all the related actors (Drucker 1998; Lambert and Cooper 2000). Our work in this chapter is mainly focused on Small and Medium Enterprises (SME). These companies evolve in an unstable and complex network. In order to guarantee its role in a supply chain, SME must be able to support the inherent requirements of the supply chain (low lead times, high level consumer satisfaction, etc.) and the external requirements due to the environment (unpredictable mutation, competition, etc.). Consequently, SME have to collaborate together in order to achieve their goals without losing their autonomy and identity (Julien 1997; Villarreal et al. 2005).

According to some field investigations, three major features of the supply chain which integrates SME clusters arise. Firstly, a supply chain is a complex system, in particular in the SME context. This complexity is due to the number of autonomous actors and the number of SME networks which are linked and work together to achieve given processes and goals. Secondly, the SME are not often located in the same geographical area as it could be in a more classical supply chain. Finally, they face a lack of visibility over the entire supply chain as a result of the two previous characteristics. Indeed, sites only have local visibility but are coordinated with other sites through the flow of products. Due to this decentralized organization and limited view over the overall supply chain, studying the structure and the behaviour of the SME supply chain is a challenging task and even more so if sustainable considerations have to be taken into account.

Besides, the study of and the experimentation with the overall supply chain integration of SME clusters are difficult to implement on actual industrial systems without heavy investments from all the actors of the supply chain. Thus, in order to facilitate the analysis of the supply chain network, it is necessary to propose a modelling solution which reflects the actual system and is able to simulate its behaviour. In light of this perspective, this chapter proposes a knowledge model based on the Architecture Model Driven Engineering [ArchMDE (Azaiez 2007)] development process that aims to identify and model the domain concepts using the multiagent system. Hence, the work described here is a combination of two research areas. The first one (industrial engineering scope) proposes a modelling approach using different layers that represent different views of the system (i.e. the system refers to a supply chain). The representation of the domain concepts within the models allows one to capitalize on the know-how and then facilitates the re-use of the supply chain concepts in different contexts. The second one (the computer engineering scope) outlines the transition from the identification step of the domain structure to the study step for the dynamics behaviour of the domain concepts. This work aims to combine the domain concepts with the multiagent ones.

In this chapter, we highlight the research work through these two areas. Hence, the contents of this chapter are organized around four main issues:

- How can we build a reusable model?
- Which methodology to adopt for domain conceptualization and what are the concepts?
- What are the steps to follow in order to agentify the domain metamodel?
- How to move from the modelling stage to the implementation?

To answer these questions, Sect. 2.2 introduces the ArchMDE development process and its contribution to this work. In the Sect. 2.3, we present the different steps applied to generate the conceptual metamodel. Section 2.4 highlights the agentification process and the dynamic behaviour integration of the different agentified concepts based on multiagent theory. Finally, Sect. 2.5 describes the transition from the modelling phase to the implementation one.

2.2 How to Build a Reusable Knowledge Model?

The heart of this research work is the modelling of SMEs supply chain by using multi-agent systems in order to build a reusable, flexible and secure knowledge model. To reach this aim, advances in the field of computer engineering, especially those dealing with multi-agent paradigm appear to be a promising approach.

To this end, we have adopted a modelling approach called ArchMDE and proposed a PhD thesis (Azaiez 2007). This approach is based on Model Driven Engineering (MDE) (Kent 2002) which founds its developing process on producing several interrelated models. In the ArchMDE approach, two types of metamodels are identified: a domain metamodel that describes functional concepts and properties related to a particular domain (e.g. a SME supply chain) and a computer modelling metamodel (e.g. a multiagent system). The combination of both metamodels will generate an agentified metamodel, that constitutes the starting point of the conceptual models. From this last metamodel, different functional models are described in order to introduce the functionalities of the system (Fig. 2.1). Finally, the use of a platform metamodel is necessary to generate the program code.

This approach is of great interest to fill in the existing gap between the design and the implementation phases. The following sections describe the step-by-step approach through those main axes.

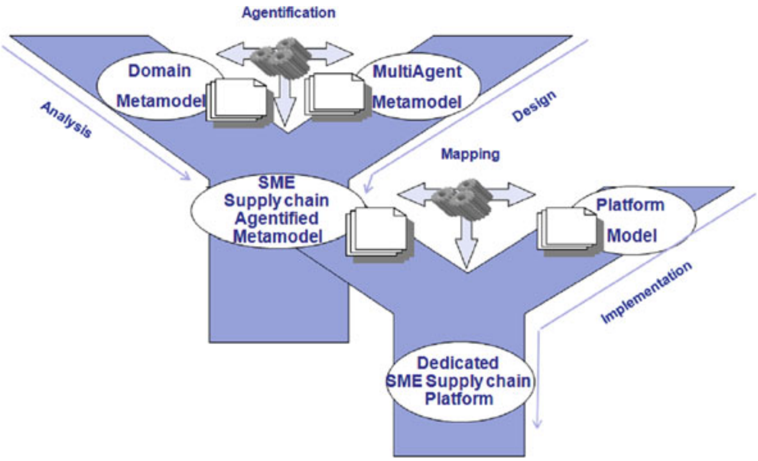


Fig. 2.1 ArchMDE development process (Tounsi et al. 2009c)

- The methodology to generate the domain metamodel.
- The agentification process which is introduced through the multiagent metamodel and the analogy between the multiagent concepts and the domain ones. The agentification is achieved by integrating the dynamic behaviour into the agentified SME supply chain metamodel.
- The implementation phase focuses on the transition from the modelling step to the encoding one.

2.3 Conceptual Domain Metamodel

According to the ArchMDE development process, the first modelling step involves the definition of the domain conceptual model. This step leads to the identification of the main concepts of SME supply chains. To achieve this objective, we follow a methodology based on existing conceptual modelling visions in the literature (Tounsi et al. 2008). In this methodology, the visions are organized into three steps. Each step addresses concepts related to supply chains. These concepts and their relationships will then be gathered within a domain metamodel that will be expressed using Unified Modelling Language (UML). The following section presents this methodology.

2.3.1 Conceptual Modelling Methodology

To identify the properties and concepts of the supply chain domain, an incremental methodology combining three visions is proposed: *product vision*, *structure vision*

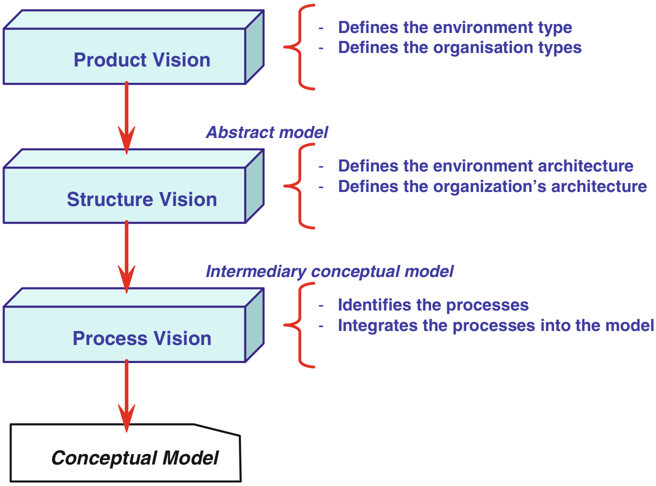


Fig. 2.2 Conceptual modelling methodology framework (Tounsi et al. 2008)

and *process vision*. In each step, a vision is applied to build or to refine the conceptual model. The result of each step (intermediate model) is the input of the next one. Therefore, at the end of the three steps, a final architecture of the conceptual model is generated (Fig. 2.2).

Step 1: *Product Vision*

This vision considers the supply chain dedicated to a particular product (or a family of products) from the raw materials through to the final goods. It focuses on the product flow to define the environment and organizations involved in its management (Thierry 2003). In the methodology framework, the *Product Vision* leads to the construction of a first abstract model of the supply chain involving the environment and organizations:

- The environment is characterized by the physical flows and the different steps of the product transformation as well as the related disturbances.
- The organizations are the entities carrying out one or several product transformation stages and the physical flow management. The organizations involved can be a network of firms that collaborate to accomplish one or several transformation stages.

Step 2: *Structure Vision*

This vision has been proposed by Cooper et al. (1997). It considers the architecture of the supply chain, made up of: actors (decision-making actors and synchronization actors), network structure (roles in the network and the number of actors for each role) and relationship characteristics between actors. Thus, on the basis of the abstract model provided by the previous step, the *Structure Vision* details the organizations involved and the physical environment:

- The environment is the part containing the physical flow. Therefore, the product flow and the resources used to achieve its transformation have to be described.
- The organization consists in identifying and prioritizing the actors in the network according to their involvement in the different levels of decision-making as well as the tasks they will be assigned. The information flow management that depends on the decision-making level is also considered in this approach.

At this step, a more detailed intermediate model is built.

Step 3: *Process Vision*

This vision is based on the process classification according to the decision-making levels (Stevens 1989; Chopra and Meindl 2001): strategic, tactical and operational.

While applying *Process Vision*, the various categories of processes are identified and integrated into the previous intermediate model. This can be done according to the decision level but it also depends on the relationships between the actors. These relationships can be classified into two categories:

- Synchronization: contains processes for exchanging information and physical flows according to a process program developed and predefined by the decision-making layers.
- Management and control: contain processes that ensure suitable decision implementation in the perspective of a continuous improvement of processes in terms of added value.

This step leads to a refined conceptual model of the supply chain.

2.3.2 Domain Model Concepts

This section presents the concepts that constitute the domain model. By applying the methodology described in the preceding section, several concepts, processes and the architecture of the model were identified. Based on these concepts, a metamodel of supply chain is proposed.

Step 1: *Applying Product Vision*

By applying the *Product Vision*, a first abstract model of the supply chain is built. It is composed of (Fig. 2.3):

- *Environment*: represent the part allocated to the product flow and management through the internal resources as well as the external elements able to influence supply chain activities.
- *Sub Supply Chain* (*SSC*): represents a group of SME which collaborate to achieve an internal aim and/or the overall objective of the supply chain. The SSC is responsible for the management of the product flow in a certain stage of its life cycle.

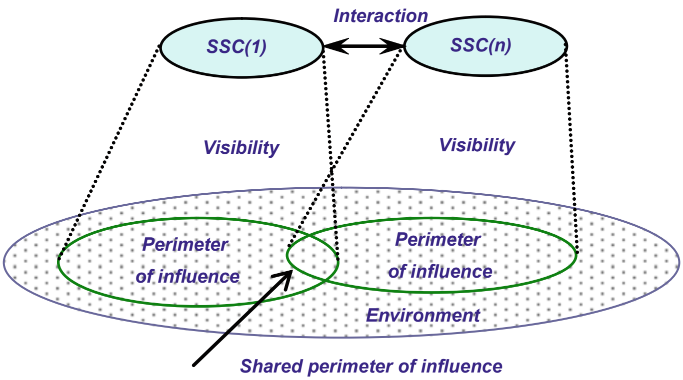
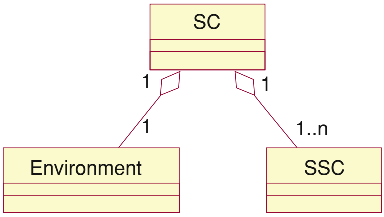


Fig. 2.3 The abstract model (Tounsi et al. 2008)

Fig. 2.4 Abstract domain metamodel (Tounsi et al. 2009c)



- *Perimeter of influence*: represents the visible part of the environment to the SSC on which it can act by internal conferring (if the action does not disturb the environment located outside its visibility) or by conferring with another SSC.
- *Shared perimeter of influence*: represents the area of the flow transfer between two SSCs. It is a shared zone where SSC coordinates their activities to allow the flow transfer.

Figure 2.4 shows the domain metamodel which reflects this conceptual abstract model using UML.

Step 2: *Applying Structure Vision*

By applying *Structure Vision*, the previous abstract model is refined. The internal architecture of the SSC and the visible part of the environment (the perimeter of influence) are described. As showed in Fig. 2.5, the SSC model and its environment are based on three layers representing the different decision-making levels.

Each layer involves particular concepts and plays a specific role in the SSC:

- The *Monitoring System* is the intelligent layer of the SSC. It controls and monitors the two other layers through the information provided by the *Execution System*. Monitoring Actors (MA) modelling the intelligent actors of SSC are the main elements of this layer. They establish metrics to evaluate the performance of the group and consequently act on the other two layers. Hence, MA are the components responsible of controlling and decision-making into a SSC and of the coordination of the activities for the overall supply chain.

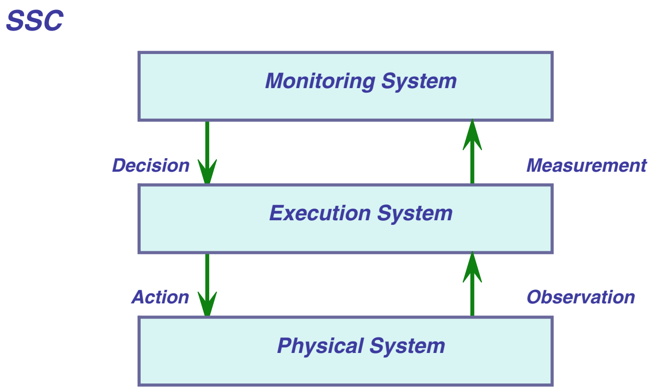


Fig. 2.5 Layers of the SSC (Tounsi et al. 2008)

- The *Execution System* is the reactive layer of the SSC. It has two main roles: (1) it ensures the synchronization of the physical flow according to the information gathered from the *Physical System*, (2) it observes and corrects the *Physical System* if a disturbance occurs. In abnormal situations, the *Execution System* refers to the *Monitoring System* for coordination and decision-making. Executive Actors (EA) are the main entities of this layer. An EA mainly models the reactive actor in the Execution System. However, occasionally MA can appear in this layer with reactive behaviour.
- The *Physical System* is the visible part of the SSC environment. It corresponds to the SSC's perimeter of influence. This layer is composed of non-decisional elements controlled by the other two layers of the SSC. Two main concepts are identified: the Moving Entity (ME) modelling the product flow and the Resource modelling production means.

Figure 2.6 shows the first conceptual abstract model refined in a domain metamodel. On one hand, we have integrated the identified concepts of each layer. On the other hand, an abstract class "Actor" is added to the metamodel for implementation purposes. Indeed, the "Actor" class defines the structural characteristics and behaviour of a decisional entity. Thus, both the EA and the MA inherit from this class. However, the "EA" class defines the specific characteristics of an executive actor and likewise for the "MA" class.

Step 3: *Applying Process Vision*

The object of the last step is to identify and integrate the different kinds of processes into the model. Table 2.1 gives a classification of the processes identified according to their role in decision-making. In the *Physical System*, the Physical Processes (PhP) have been identified. A PhP describes the sequence of the processing stages of a product. It is a concept to be integrated within a domain metamodel in order to define the tasks that can be handled by the *Execution System*.

Table 2.1 Process classification

SSC Layer	Process family	Role
Monitoring system	Strategic processes (SP)	Coordinate long term decisions
	Monitoring and control processes (MCP)	Monitor SSC activities
		Drive and evaluate SSC performance in the overall supply chain
Execution system	Operational control processes (OCP)	Synchronize and control the physical system
Physical system	Physical processes (PhP)	Define the transformation routings of products

The processes identified in both *Monitoring* and *Execution Systems* are management processes. Hence, they represent the dynamic behaviour of the SSC. This behaviour is induced by control and monitoring decisions that come from either the SSC or the overall supply chain. This behaviour is basically a communication mechanism (coordination, collaboration or cooperation).

In order to model management processes and communication mechanisms, more informational elements are needed for EA and MA to ensure their role in the domain model. Thus, decisional actors of the SSC (EA and MA) need three conceptual elements that consolidate their internal architecture:

- *Indicator*: is used by actors for two different tasks. Indeed, the EA control and detect Physical System deviation by comparing the value of an indicator with its fixed objective. As for MA, they evaluate the internal performance of the SSC but also in the overall supply chain.
- *Action*: Actors apply actions when facing indicator deviation.
- *Organizational Knowledge*: is an actor's database that stores information about his acquaintances. For example, if an actor "A" is an acquaintance of an actor "B" this means that "B" owns information about the identity, the behaviour, the capabilities and the resources of the actor "A". Reciprocally, the actor "A" owns the same information about the actor "B". According to this, each actor (EA and MA) owns knowledge about resources of all actors in the same SSC. However, the MA involved in the overall supply chain have additional internal acquaintances, each MA owns limited knowledge about the other MA of the overall supply chain. Note that this knowledge requires continuous updating.

In the same way, the intelligent behaviour of the MA requires the definition of other conceptual components:

- *Objective*: models the strategic goal of the SSC. According to this aim, the SSC coordinates its activities with other SSCs in the overall supply chain.
- *Constraint*: is a knowledge that an actor must consider to reach the goal of the overall supply chain or the SSC's one.

Through the *Process Vision*, the previous metamodel and its concepts are refined by integrating identified concepts. Figure 2.7 presents a UML

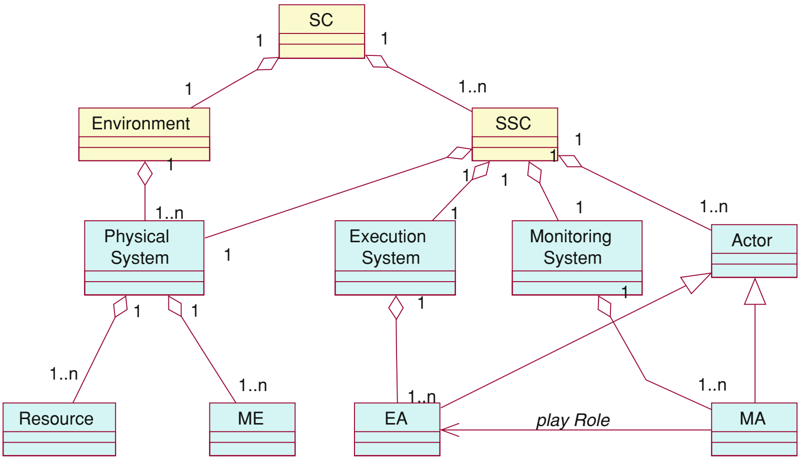


Fig. 2.6 Intermediary domain metamodel (Tounsi et al. 2009c)

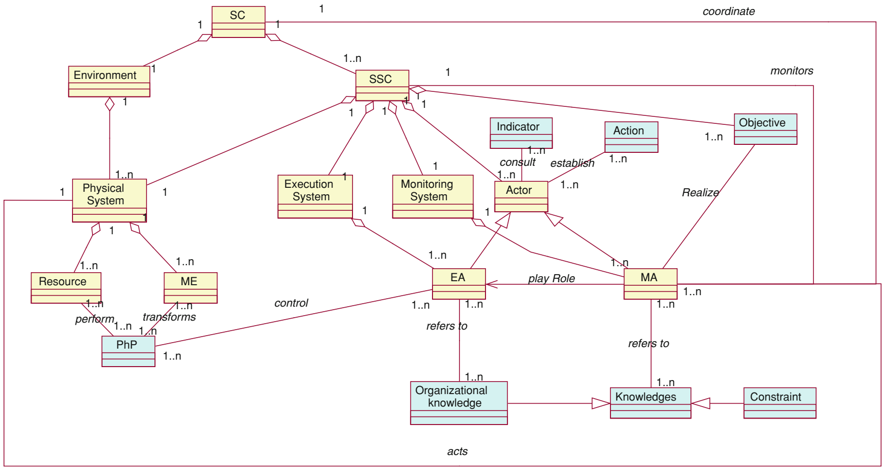


Fig. 2.7 Final domain metamodel (Tounsi et al. 2009c)

representation of the final domain metamodel for the supply chain in the SME context. It corresponds to the final conceptual model with its associated concepts regardless of computer technologies.

2.4 Agentified SME Supply Chain Metamodel

In this section, the domain metamodel is merged with an agent metamodel using the ArchMDE methodology (Azaiez 2007). Thus, this section is divided into two parts: the first one outlines the properties of each multiagent concept according to the Agent-Interaction-Environment-Organization (AIEO) approach and the second part highlights the agentification of the domain metamodel.

2.4.1 Multiagent Metamodel

The AIEO approach breaks the whole multiagent system down into four views: Agent view, Interaction view Environment view and Organization view. Figure 2.8 shows multiagent concepts according to each view and the links between them.

The Agent view defines the agent metamodel composed of the following concepts:

- "Agent" identifies different kinds of agent according to the decision-making capacity of the agent (reactive agent, cognitive agent and hybrid agent).
- "Cognitive agent" defines an agent with cognitive abilities. The metamodel highlights the main concepts modelling the BDI agent (Belief, Desire, Intention = plan).
- "Reactive agent" defines an agent with reactive abilities to respond to unpredictable events.
- "Hybrid agent" defines an agent with hybrid intelligence (cognitive and reactive abilities).
- "Goal" defines the aim that an agent should achieve.
- "Knowledge" and "Norm" define all the knowledge and norm necessary for the agent to achieve its goal.
- "Plan" represents an action plan implemented by the agent. The plan is composed of one or several elementary actions.
- "Reactive action" is an action implemented by the reactive agent.

The Interaction view describes the dynamic relations between the agents. This interaction is a structured exchange of messages between the agents through a specific protocol or language. Thus the interaction metamodel highlights the following concepts:

- "Interaction protocol" represents the interaction protocol adopted by the agents.
- "Communicative action" represents an elementary action of communication that is part of the "interaction protocol".
- "Message" is a set of information exchanged between the agents through the "interaction protocol". The agent interprets the message based on the communicative action.

The Environment view focuses on all the elements external to the agent allowing it to reach its goal or activate its behaviour through events. The elements belonging to the environment metamodel are as follows:

- "Active resource" represents the resources that activate the behaviour of the agent by generating events or triggers.
- "Passive resource" defines the resources the agent needs to accomplish its task.

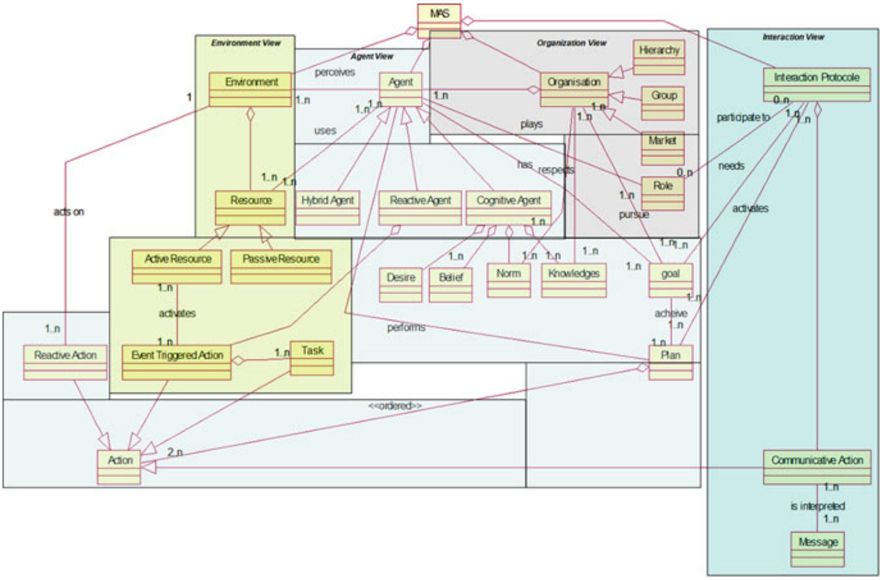


Fig. 2.8 Agent metamodel (Azaiez 2007)

- "Event triggered action" represents events that resources activate. An event is composed of one or more tasks.

Finally, the Organization view describes the structure of the whole system. The organization metamodel is made up of the following concepts:

- "Organization" defines the system topology (hierarchy, group or market).
- "Role" represents different roles that the agent could play.

2.4.2 Agentification of the Domain Metamodel

This step of the ArchMDE methodology consists in merging the multiagent metamodel with the domain metamodel. Hence, on the one hand, a metamodel defines a multiagent system according to the Vowel approach (Fig. 2.8). On the other hand, a domain metamodel describes the supply chain in SME context (Fig. 2.7). A correspondence between the multiagent concepts and those of the domain is then carried out according to their properties and their roles in the metamodel. Table 2.2 summarizes the correspondence between these concepts in order to achieve the agentified metamodel for SME supply chains.

After the agentification process, we obtain an agentified domain metamodel as presented in Fig. 2.9. The domain metamodel and the multiagent one are separated for more clarity.

Table 2.2 Correspondence between domain and agent concepts

Domain concepts	Multiagent concepts	Description
Supply chain (SC)	Multi-agent system (MAS)	By analogy, the root of the domain metamodel corresponds to the root of the multiagent system
Environment	Environment	In both metamodels, the environment is the physical space defining all things that are external to the agents and necessary in order to manage the SC
Sub supply chain (SSC)	Organization	It is an organization made up of two groups of agents
Physical system	Resource	It is all the resources needed for one agent or a group of agents to manage the group (perimeter of influence)
Resource	Passive resource	It is a resource allocated to the agent to perform its task
Moving entity (ME)	Active resource	The ME represents the product in circulation. It activates the behaviour of the reactive agents.
Physical process (PhP)	Task	It is a task or a physical activity to be handled by reactive agents
Monitoring system	Group	It is a group of cognitive agents which collaborate in the SSC and coordinate the activity of the organization with other organizations
Execution system	Group	It is a group of reactive agents which collaborate in the SSC
Actor	Agent	An actor can be a cognitive agent or a reactive agent according to its decisional characteristics
Executive actor (EA)	Reactive agent	EA perceives the physical system and acts on it according to the observation
Monitoring actor (MA)	Cognitive agent	According to the collected information and the history of the situation and action, the group of MA monitors the SSC to reach a goal and accomplish its activity
Objective	Goal desire	A SSC has a goal to reach. This goal is coordinated with other nodes' goals. In addition, each MA has a personal aim for each indicator. This kind of "Objective" is modelled by the "Desire" of the BDI agent (MA)
Indicator	Belief perception	The agents act on the environment according to the indicator measures. In this case, an indicator is modelled by the "Perception" of an agent. However, a MA monitors the SSC according to the history of these measures. So, an "Indicator" is modelled by the "Beliefs" of BDI agent
Action	Plan	It is an action or a set of actions to apply when facing a disturbance
Knowledge	Knowledge	It is all the knowledge needed by the agents to act in an appropriate way
Organizational knowledge	Knowledge	Each agent has a list containing the information about other agents from the same SSC or the overall SC. This list stores knowledge about the name of the agent, the task that it performs and its resources

(continued)

Table 2.2 (continued)

Domain concepts	Multiagent concepts	Description
Constraint	Knowledge	The MA make decisions according to their objectives and their beliefs. At the same time, there are some constraints (about product or other SCs where the group is involved) that the group of MA must take into account when making decisions

2.4.3 The Integration of Processes Into the Metamodel

Up to now, the static part of the domain metamodel has been created. In this section, we define the dynamic behaviour of the concepts based on the multiagent tools and theory. Indeed, this dynamic is described by the implementation of interaction protocols according to the process vision and the communication mechanisms.

Firstly, the process vision allows us to define two scenarios: (1) the synchronization of the physical processes and (2) the monitoring and the control of processes.

Secondly, a communication mechanism is “a framework formalizing interaction between different actors in the network according to their managerial relationship characteristics” (Tounsi et al. 2010). The study of the domain identifies two kinds of communication framework. Indeed, in the overall supply chain, SSC coordinate their activities in order to achieve the common objective of the overall supply chain. Within the SSC, the actors collaborate to achieve a local goal.

This section describes the different protocols implemented in the agentified domain metamodel taking into account the different scenarios of the process vision and the communication mechanisms.

2.4.4 Synchronization of the Physical Processes

The SSC is responsible for the synchronization of the Physical System involved to achieve its task. This activity consists in applying a communication protocol relative to the nature of the interaction framework. In this section, the collaboration and coordination processes are described in order to be implemented in the Execution System and Monitoring System, and then to synchronize the Physical System.

2.4.5 Integration Into the Execution System

In accordance with the agentified domain metamodel, the Execution System is responsible for the synchronization of the physical process (PhP) in common situations. Indeed, Executive Actors (EA) which are reactive agents, synchronize PhP by taking into account the availability of resources.

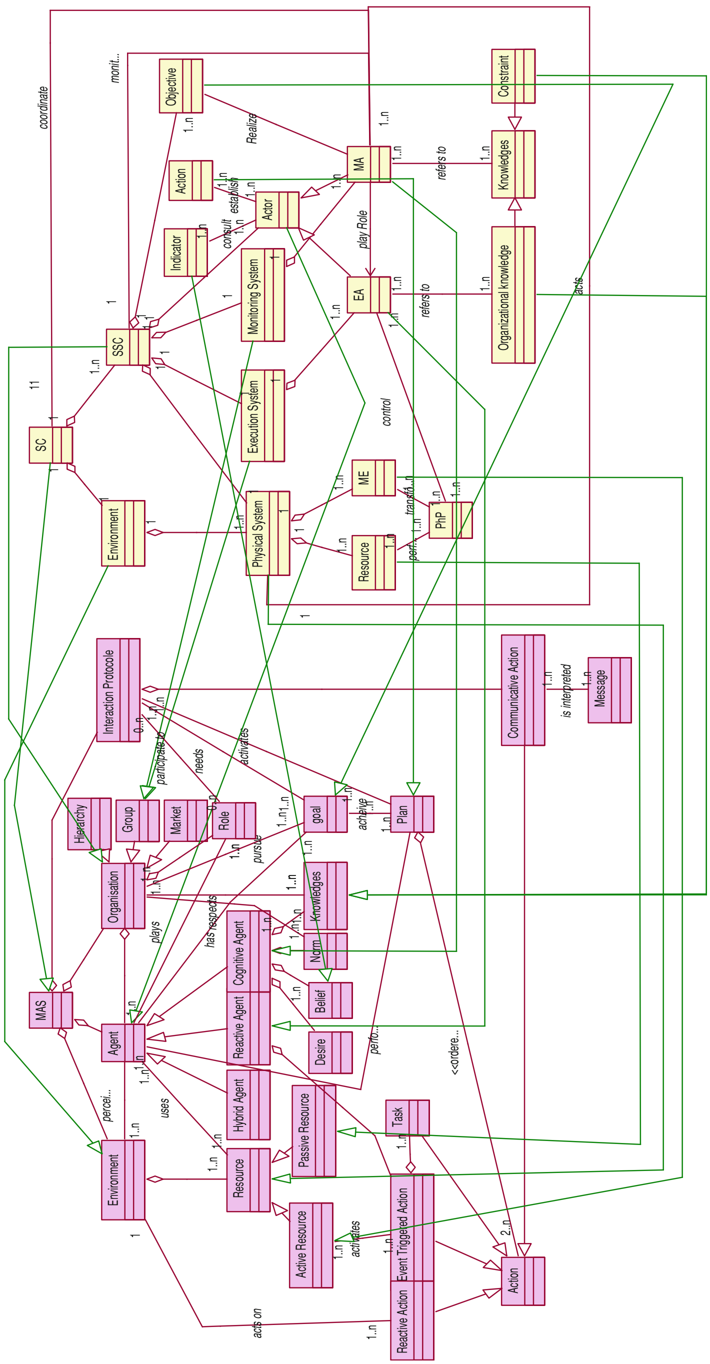


Fig. 2.9 The agentified SME supply chain metamodel (Tounsi et al. 2009c)

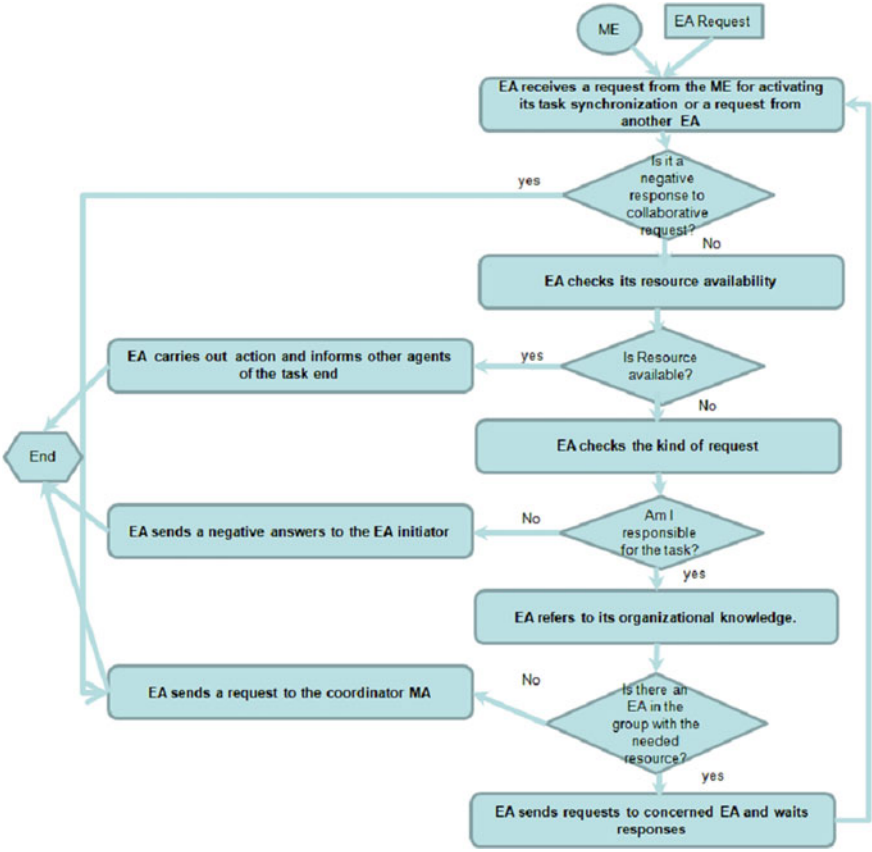


Fig. 2.10 EA synchronization behaviour (Tounsi et al. 2009a)

Figure 2.10 describes the Executive Actor's (EA) behaviour in its role of synchronizing the physical flow (ME). An EA receives a request and reacts according to its type. Three types of request can be distinguished: (1) a ME request, (2) a collaborative request from another EA or (3) a negative response to a collaborative action initiated by the agent itself.

The following sequence highlights the EA's behaviour:

- If the request is a negative response for the collaborative demand that the EA initiated, the EA sends a request to the coordinator agent (MA coordinator) of the monitoring system.
- If the request is a synchronization request coming from the Moving Entity (ME) or a collaborative request coming from another initiator agent, the EA checks the availability of the resources concerned.

- If the resource is available, the EA carries out its task, updates the state of the ME and informs the other agents within the executive system and the coordinator agent at the end of the action.
- If the resource is unavailable and the EA has been solicited by another executive agent to achieve the task, it sends a failure request to the initiator.
- If the resource is unavailable and the EA is in charge of the task, then it seeks within its organizational knowledge an agent from the Execution System of the SSC that might have the necessary resource.
- If the agent finds another agent within its organizational knowledge that can handle the task, it delegates the responsibility of the task. In this case, the collaboration process of the agent concerned will be activated and follows the same sequence.
- If the agent does not find another agent who has the resource necessary to handle the task, it sends a request to the coordinator agent. This agent is a monitoring agent (MA) that receives requests from the Execution System. The MA sends the information to other monitoring agents in the SSC in order to find a solution.

2.4.6 Integration Into the Monitoring System

In unusual situations,[1] the Executive System refers to the Monitoring System. In this case, the group of MA evaluates the situation according to the defined objective and establishes an action plan. If the objective is not reached, the MA needs to consult other SSCs to find a suitable solution. Thus, the agents adopt the protocol based on Contract Net Protocol to provide the coordination of the objectives. The synchronization protocol can be described according to the following steps:

- In the Monitoring System, a monitoring actor is responsible for checking all the requests received and sending them to other MA in the layer. Three kinds of requests can be distinguished: (1) EA request, (2) reply to a help request or (3) a help request from another SSC in the overall supply chain.
- If a MA coordinator receives an EA request then it sends the information to other MA. In this case, the group evaluates the situation according to the SSC's objectives. Two cases may arise: (1) the problem has no impact on satisfying the SSC's objectives or (2) the objective is deviated.
- If there is no impact on the objective, the MA tries to find an internal solution according to its desire, belief and constraints. If a solution can be found, the MA coordinator sends the actions plan to the Executive System.
- If the objective is deviated or an internal solution cannot be reached, the group of MA sends a Help Request to other SSCs via the MA coordinator and waits for the responses.

[1] Unusual situation is occurred when Executive System cannot propose a solution for a happened problem in the physical system.

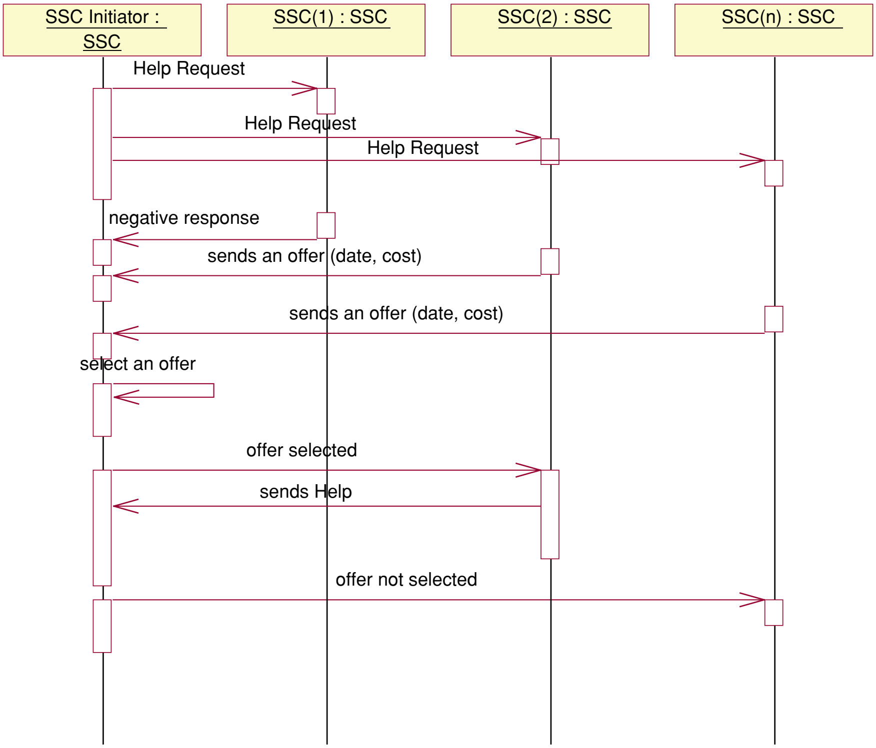


Fig. 2.11 SSC coordination process (Tounsi et al. 2009b)

- When all the replies have been received, the MA coordinator ranks them according to the reception date of the request. The list of responses will be sent throughout the Monitoring System. According to their beliefs, desires and constraints, the group of MA chooses the most suitable answer and diffuses the action plan to the Execution System. In this case, the EA updates the state of the ME.
- If the request is a help request from another SSC, the MA coordinator sends the request to the Monitoring System. In this case, the Monitoring System evaluates the demands according to internal criteria (objective, constraint, belief, and desire). If the SSC can provide assistance, it makes an offer to the SSC initiator or it sends a negative response.
- The SSC initiator chooses the suitable offer and sends a confirmation to the selected SSC and a cancellation response to other bids.

The following figure (Fig. 2.11) shows the sequence of messages between the SSCs. This diagram represents the coordination process in the overall supply chain in order to synchronize the physical flow in the case of a disruptive case (SSC cannot reach the internal aim).

2.4.7 Monitoring and Control Protocol

This protocol describes the conditional preventive (based on measurements) and corrective (in case of disturbance) monitoring and control in the SSC or in the overall supply chain. The monitoring and control protocol is based on performance evaluation in both layers of the SSC: the Execution System and the Monitoring System.

At the end of the synchronization protocol, each actor updates the indicator measurement by evaluating its activity and related resources. In addition, the Monitoring System evaluates the local activity of the SSC and participates in the improvement of the performance of the overall supply chain. Thus, the following sequence describes the monitoring and control protocol in the SSC's layers:

- At the end of its synchronization task, the EA evaluates the performance of its activity and related resources (the allocated space of the environment to the EA).
- According to this perception, the EA refers to the indicator base in order to detect a disturbance.
- If the EA finds a deviation, it seeks the cause of the disturbance.
- If the deviation is a common situation, the EA selects the appropriate action plan to solve the problem and applies it to the environment. After that, it sends measurement (or perception) to the MA coordinator which, in turn, sends the information throughout the Monitoring System. Then, each MA updates its belief.
- If a new situation occurs, the EA sends a failure control message to the MA coordinator. Then, the MA sends the information throughout the Monitoring System and each MA updates its belief.
- In this case, the Monitoring System analyzes the situation according to internal criteria (beliefs, desires, objectives and constraints).
- If the problem needs corrective maintenance, the Monitoring System generates an action plan and forwards it to the Execution System. The actors of the SSC update their bases of actions.
- If the disruption does not affect the SSC, the Monitoring System applies a preventive action plan to avoid future disturbance.

Figure 2.12 shows the protocol for monitoring and control through an UML sequence diagram:

2.5 The Implementation Phase

The last phase of the ArchMDE development process deals with the transition from the modelling phase to the implementation one. This last phase is mainly divided into two steps. The first one concerns the refinement of the agentified domain metamodel by integrating the information necessary for the implementation. The second step introduces the choice of the development platform. In the

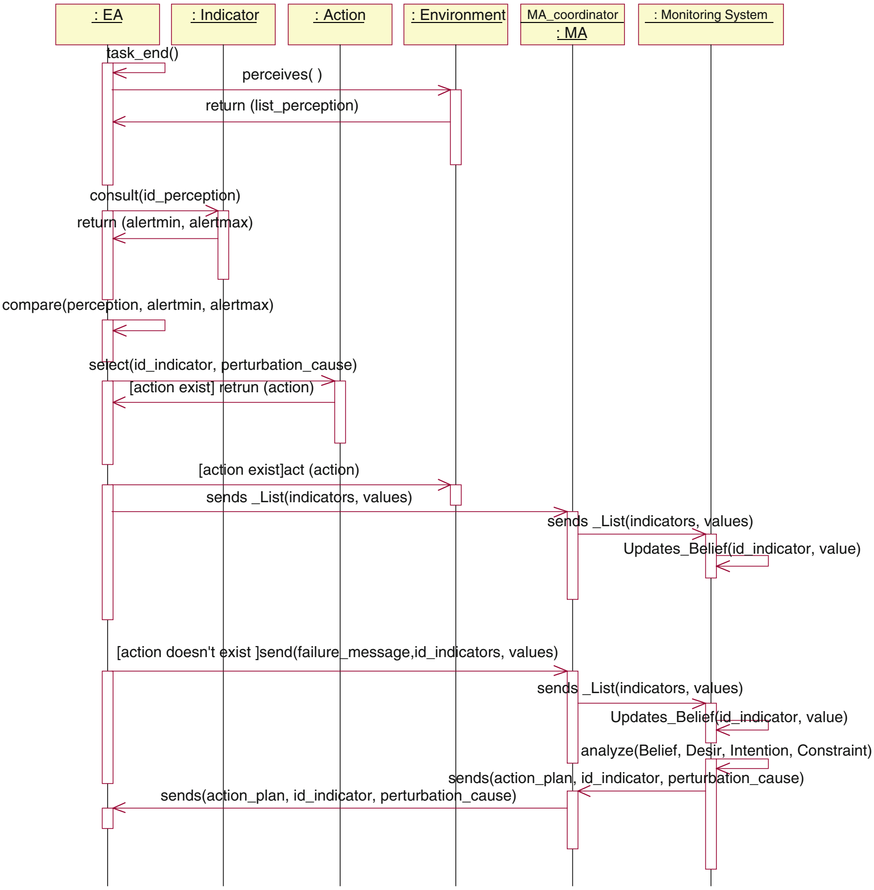


Fig. 2.12 Sequence diagram of the monitoring and control protocol (Tounsi et al. 2011)

next section, we focus on the two steps of the implementation phase of the ArchMDE development process.

2.5.1 Refinement of the Agentified Domain Metamodel

The steps of integration protocols identify the abstract patterns that describe the dynamic behaviour of the agentified domain concepts. The refinement consists of the integration of the attributes and the methods that define the architectural and the behavioural proprieties of each concept. This step provides a final class diagram, the Implementation Metamodel. The encoding of this last result leads to a dedicated simulation platform.

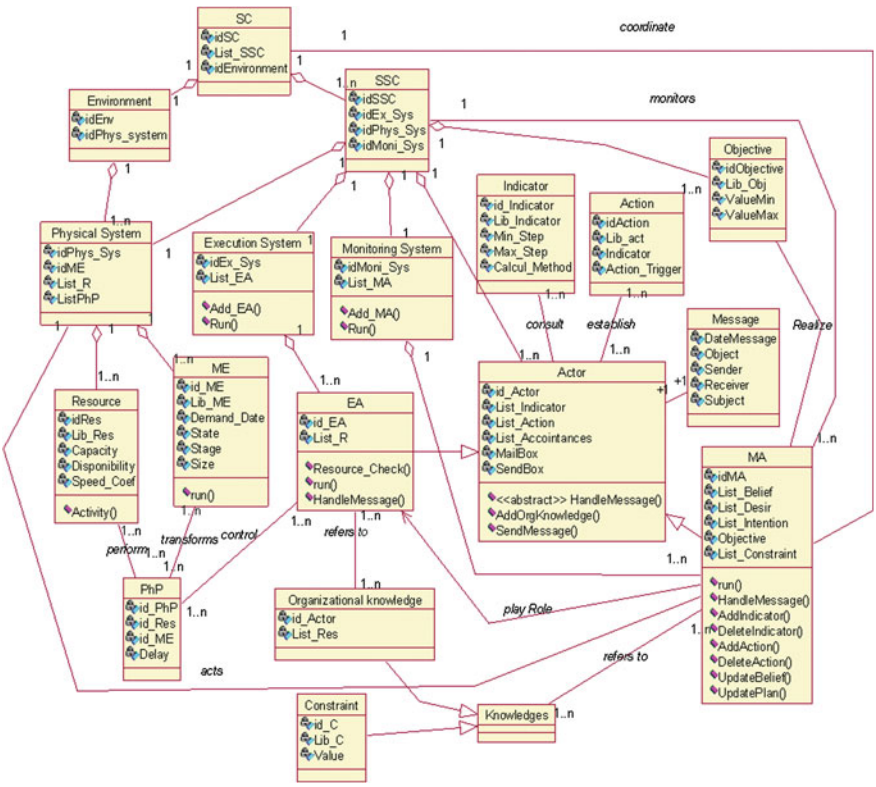


Fig. 2.13 The implementation metamodel (Tounsi et al. 2011)

In the implementation metamodel, we have refined the domain metamodel and integrated multiagent concepts that are essential to the agent's behaviour.

Figure 2.13 represents the Implementation Metamodel. The class diagram focuses on the main methods and attributes. Each private attribute has accessor (get) and mutator (set) methods. However, these methods are not visible in order to lighten Fig. 2.13. The Actor class defines the global architecture and behaviour of an agent. The EA (reactive agent) and MA (cognitive agent) inherit from Actor class the common characteristics. Nevertheless, each one implements the method run() that describes his behaviour and the method HandleMessage() that allow him to read and to construe the received message.

2.5.2 *The Choice of the Development Platform*

This section focuses on the choice of a development platform according to several objectives of the implementation. Indeed, obtaining the final pattern (the implementation metamodel) completes the modelling process. At this level, the behavioural and architectural characteristics of the proposed concepts are implemented using a development platform. The most important modelling criteria that the implementation platform must allow are as follows: (1) the implementation of cognitive agents (MA) and reactive agents (EA), (2) the communication between agents (sending messages), (3) the integration of the agents group (Monitoring System and Execution System) and the modelling of the external environment (Physical System), and (4) a graphical user interface for easy setup and the observation of the simulation results.

According to these criteria, we have studied and compared two multiagent platforms: Jade (http://jade.cselt.it) and Madkit (http://www.madkit.net).

Jade (Java Agent Development Platform) is a multiagent platform fully encoded in JAVA. It allows the modelling of agents based on predefined patterns communicating through messages. The Jade software simplifies the implementation through a graphical user interface (remote GUI). However, it does not allow the implementation of a group of agents which is a major modelling criterion for our metamodel. Consequently, Jade does not correspond to our specifications.

Madkit is a modular and scalable multiagent platform also written in JAVA. The main reasons for taking an interest in this software are that it: (1) provides an API (Application Programming Interface) to enable the development of agents that communicate through sending messages, (2) allows one to develop agents located in groups and play roles in the organization and (3) offers a full set of facilities for launching, displaying, developing and monitoring agents and organizations (Gutknecht et al. 2000). However, Madkit does not allow one to draw the external environment as a set of objects (object in oriented-object programming theory). Indeed, each concept must be an agent in order to communicate into the application. Nevertheless, we can use the Madkit platform with a JAVA environment to integrate external classes. Thus, Madkit merged with JAVA environment can be considered as an implementation way. Due to lack of time and knowledge in this area, we have not investigated further, but this solution can be considered as a future perspective to implement the knowledge model.

Given our computer skills, we have decided to implement our own simulation platform using JAVA. In this platform, we have developed agents, groups of agents and structured a peer-to-peer communication between them. The elements external to the agent (i.e.: resources, indicator, action, PhP, etc.) are encoded as objects. The product (ME) is an active entity. It is encoded as an object that triggers events.

2.6 Conclusion

For reasons of complexity in terms of size, communication protocols and decision-making strategies with decentralized behaviours of actors, modelling and simulation of supply chains is a tricky task for researchers and supply chain managers. In this chapter, our objective is to propose a natural way for supply chain modelling while considering the organisational issues and the managerial relationships. The resulting knowledge model requires three main features: reusability, simplicity and genericity. Our main theoretical contributions are on this conceptual model and its building methodology.

To reach this goal, we have adopted ArchMDE as a development process. The first step of the development process focused on the identification of the major concepts. These concepts are consistent with both domains: the inherently decentralized supply chain and multi-agent systems. The high level of conceptualization of these concepts tends to provide genericity and reusability to the proposed model. In the second step, we have presented a mapping within the proposed concepts. This mapping translates the supply chain concepts into MAS concepts. This aims at simulating the dynamic behaviour of the supply chain based on multi-agent tools. Major communication protocols between actors (or agents) are also drawn to emphasize the importance of different kinds of exchange mechanisms. To keep the model as generic as possible only a few generic protocols are presented here.

In terms of implementation, the concepts and the protocols presented in this chapter have been developed as a generic Supply Chain Multi-Agent System using JAVA. A prototype of a 2-echelon Supply Chain with one manufacturer and one logistician including (1) respectively production and inventory planning processes, and (2) two-actor negotiation and coordination processes has been tested with success (Ogier et al. 2010).

Our aim is to be able to simulate a real multi-actor Supply Chain with dynamic behaviour. Thus, more internal optimization processes and coordination protocols are under study.

References

Azaiez, S. (2007). *Approche Dirigée par les modèles pour le développement de systèmes multi-agents*. Annecy le vieux, France: Thèse de l'Université de Savoie Spécialité Informatique. (11 Dec).

Chopra, S., & Meindl, P. (2001). *Supply chain management: Strategy planning and operation*. NJ: Prentice-hall.

Cooper, M., Lambert, D. M., & Pagh, J. D. (1997). Supply chain management: More than a new name for logistics. *International Journal of Logistics Management, 18*(2), 1–13.

Drucker, P. F. (1998). *Management's new paradigms* (pp. 152–177). Forbes Magazine.

Gutknecht, O., Ferber, J., & Michel, F. (2000). MadKit: Une expérience d'architecture de plate-forme multi-agents générique. 8ème Journées Francophones Intelligence Artificielle Distribuée Systèmes Multi-Agents JFIADSMA 2000. Saint-Jean Le Vêtre Octobre.

Julien, P. A. (1997). *Les PME bilan et perspectives* (2nd ed.)., Economica, Paris: France.

Kent, S. (2002). *Model-driven Engineering IFM 2002* (Vol. 2335)., LNCS, Springer-Verlag.

Lambert, D. M., & Cooper, M. C. (2000). Issues in supply chain management. *Industrial Marketing Management, 29*(1), 65–83.

Ogier, M., Cung, V.-D., Boissière, J., & Mangione, F. (2010). Supply chain performance in the case of decentralized planning. In *Proceedings of the 8th International Conference on Supply Chain Management and Information Systems* (*SCMIS2010*) (pp. 63–70). Hong Kong, China.

Stevens, G. C. (1989). Integrating the supply chain. *International Journal of Physical Distribution and Materials Management, 19*(8), 3–8.

Thierry, C. (2003). *Gestion des chaînes logistiques: Modèle et mise en œuvre pour l'aide à la décision à moyen terme*. Toulouse: University of Toulouse II. (Accreditation to supervise research).

Tounsi, J., Boissière, J., & Habchi, G., (2008). A conceptual model for SME Mechatronics supply chain. In *Proceedings of the 6th International Industrial Simulation Conference* (*ISC'08*) (pp. 273–280). France: Lyon.

Tounsi, J., Boissière, J., & Habchi, G. (2009a). Multiagent decision making for SME supply chain simulation. In *Proceedings of 23rd European Conference on Modeling and Simulation* (*ECMS*) (pp. 203–211). Madrid: Espagne.

Tounsi, J., Habchi, G., & Boissière, J. (2009b). A multiagent system for production synchronization in SME mechatronic supply chain. In *Proceedings of the 10th Middle Eastern Simulation and Modeling Conference* (*MESM*) (pp. 91–97). Beirut: Liban.

Tounsi, J., Azaiez, S., Habchi, G., & Boissière, J. (2009c). A multiagent approach for modelling SMEs mechatronic supply chains. In *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing*.

Tounsi, J., Habchi, G., & Boissière, J. (2010). A conceptual model for SMEs mechatronics supply chain. *International Journal of Computer Aided Engineering and Technology, 2*(4), 371–387.

Tounsi, J., Habchi, G., Boissière, J., & Azaiez, S. (2011). A multi-agent knowledge model for SMEs mechatronic supply chains. *Journal of Intelligent Manufacturing*, doi: 10.1007/s10845-011-0537-1.

Villarreal Lizarraga, C.L., Dupont, L., Gourg, D., Pingaud, H. (2005). Contributing to management of shared projects in SMEs manufacturing clusters. In *Proceedings of the 18th International Conference on Production Research* (*ICPR-18*). Salerno: Italy.