A Norm-Governed Holonic Multi-Agent System Metamodel

Patrizia Ribino[1], Carmelo Lodato[1], Salvatore Lopes[1], Valeria Seidita[2,1], Vincent Hilaire[3], and Massimo Cossentino[1]

[1] Istituto di Reti e Calcolo ad Alte Prestazioni,
Consiglio Nazionale delle Ricerche, Palermo, Italy
{cossentino, c.lodato, s.lopes, ribino}@pa.icar.cnr.it
[2] Dip. di Ingegneria Chimica Gestionale Informatica Meccanica,
University of Palermo, Italy
valeria.seidita@unipa.it
[3] System and Transport Laboratory,
University of Technology of Belfort Montbéliard, France
vincent.hilaire@utbm.fr

Abstract. Modeling and designing systems that require a high level of coordination, control and automation is a very difficult task. The problem is the lack of design processes able to cover all the features these systems present. This paper presents an extension of the ASPECS metamodel for supporting organizational and normative principles and it allows to define models not only from an holonic agent viewpoint but also from a normative organization perspective. Moreover, our work emphasizes and makes explicit those norms that regulate the structural, behavioral and finally adaptive aspect of an organizational system. The extended metamodel was experimented creating a Virtual Enterprise model for the optimization of distributions inside the logistic districts. This organizational model is implemented using JaCaMo.

1 Introduction

Nowadays, a lot of researchers in the field of artificial intelligence and intelligent systems aim to develop software systems able to act in full autonomy such as a human beings do in reaching their objectives. During their daily activities, human beings pursue multiple goals that sometimes interleave and overlap; in doing that, they often communicate and coordinate with other entities of the world they live. We are far from having tools to create systems completely acting as they were in a daily "human routine". Nevertheless, literature proposes a way for developing goal-driven systems using the knowledge these systems have of their environment in order to react to environment changes. The ability of coordinating, controlling and making autonomous the activities of all the different involved entities is a strong requirement for this kind of systems. These issues can be faced with the use of the Multi Agent System (MAS) design paradigm and with organizational models. The latter is covered by only a restricted set of agent oriented design processes[4] of which only few cover the whole design process life cycle, from analysis to implementation. To the best of our knowledge, among the agent-oriented processes only ASPECS [6] manages abstractions, such as holon, group and goal, for modeling and implementing organizational structures like holarchy. Instead, among organizational models, only $\mathcal{M}$oise+ [13] and OMNI [26] address the normative aspect of a multi-agent organization.

The novelty introduced by our work is merging the strength of ASPECS and $\mathcal{M}$oise+ in order to create a complete support for developing MASs structured organizations, such as holarchies, ruled by norms. Actually ASPECS does not include the possibility of designing norm-based systems. The need of introducing norms arose from a design requirement, we needed the possibility of modeling constraints in form of institutional rules (Norms) defined outside the agents. Defining external rules (like social rules, laws, company procedures, etc...) allows to face all the problems related to management, coordination and control of different holons. The result was an extension of the ASPECS metamodel in order to include all the elements providing abstractions for managing the normative issues along with the definition of some new norms that regulate the structural, behavioral and adaptive aspect of an organization. Moreover, we instantiated this new metamodel in a specific logistics business model in order to create an optimized representation of the distribution processes inside a supply chain. We have implemented this model as a norm-governed holarchy using Jason [5], $\mathcal{M}$oise+ and Cartago[22], unified in the JaCaMo framework [24].

The rest of the paper is organized as follows. In Section 2 an overview on the theoretical background of the work is presented. Section 3 is the core of the paper. It presents the extended metamodels along with the definition of norms we have introduced and the conceptual mapping among the elements of the Agency Domain and JaCaMo metamodel. In Section 4 we show the addressed case study. Finally some conclusions are presented in Section 5.

2 Theoretical Background

An agent is an *autonomous*, *reactive* and *proactive* entity that pursues individual goals interacting with the environment and others agents by means of *social ability* [28]. A multi-agent system [9] (MAS) is a software providing a tool to model and reproduce the interaction and social structures observed in real world organizations. It allows to adapt human organizational patterns in agent-based systems that become a virtual counterpart of real organizations.

As well-known, there are different kinds of organizational schema, such as hierarchies, holarchies, teams, coalitions and so on. Each organizational schema is usually defined by means of roles adopted by an agent, relationships, rules and norms defining the agents behavior and organizational structure.

In this paper we adopt holarchies as an organizational structure of the agents societies. The concept of Holarchy adopted as an Enterprise model has its origin

[4] We use the term design process and methodology as synonyms because here it is not important to highlight the differences among them.

from the work of Koestler [16]. During his research on self-organization in biological systems, Koestler discovered nested hierarchies of self-replicating structures (*holarchies*). He coined the term *holon* to describe the elements of such systems.

A holon is, commonly, defined as a self-similar structure composed of holons as sub-structures. For this reason, it can be seen from different perspectives, either as an autonomous *atomic* entity or as an *organization* of holons. A holon is a whole-part composed of other holons and at the same time, a component of a higher level holon. A holon acts basically as an autonomous entity, although cooperating to form self-organizing hierarchies of subsystems (such as the cell/tissue/organ/system hierarchy in biology) in order to achieve the goals of the holarchy. In addition, holons can simultaneously belong to different super-holons and can be regulated by rules. These rules not only allow to define a system as a holon with an individuality of its own but also to determine its structural configuration, functional patterns and behavioral regulations [25].

Holonic systems, while modeling complex systems, are able to efficiently manage their resources and to adapt themselves to changes occurring in the environment. A useful way to implement holarchies in software system is by means of the Holonic Multi-Agent System (HMAS) paradigm. As shown in [10], HMAS paradigm allows to represent a holonic system where individual agents are driven by coordination mechanism according to the cooperation rules of the holon the agent is member of. In HMAS a holon is a set of individual agents organized according to different organizational models (see [10] for more details).

In this paper we use the HMAS and the Virtual Enterprise paradigm to model a holonic framework applied to the cited logistic problem. According to Uliero *et al.* [25] we refer to Virtual Enterprise (VE) as a new organizational form that can be characterized by a collection of geographically apart individuals, groups or entire organizations depending on electronic communications in order to collaboratively work and to provide a service or to realize a common goal.

Multi agent systems can be developed using several frameworks (JADE [2], JADEX [20][27], PRACTIONIST [18] etc.) based on different approaches. In this paper we adopt the JaCaMo approach in order to implement a Holonic Multi-Agent System. JaCaMo [24] is a programming platform that integrates three levels of multi-agent abstractions: an agent programming language (Jason), an organizational model (Moise), and an environment infrastructure (CArtAgO).

Jason [5] is a Java-based interpreter for an extended version of the AgentSpeak language [21], an abstract agent language founded on the BDI(Belief-Desire-Intentions) model. A Jason agent is described by means of a set of plans the agent is able to follow in some situations.

$\mathcal{M}$oise+ [13][14] is an organizational model for MAS which specifies the structural, functional and normative aspect of MAS organizations. Each aspect is defined in a specification set.

CArtAgO [22] is a general purpose framework/infrastructure that allows to program and execute virtual environments for multi-agent systems. It is based on the concept of Artifacts intended as resources and tools dynamically constructed, used, manipulated by agents to support/ realise their individual and collective activities.

3 A Norm-Governed Holonic MAS Metamodel

In order to have means for developing norm-governed multi-agent systems structured by holonic organizations we need a metamodel containing all the abstractions to be treated during the phases/activities of the design process devoted to develop such systems. In this section we illustrate the metamodel we created by adding to the ASPECS metamodel all those concepts from $\mathcal{M}$oise+ metamodel useful for modeling MASs under a normative point of view.

The ASPECS metamodel [6] is divided in three parts: the problem, agency and solution domain. The first contains the elements useful for the description of the problem under an organizational point of view. The second domain provides an agent oriented solution to the problem. Finally, the last provides the concepts for the implementation in a specific platform. As also stated in [15], ASPECS is one of the most complex and complete organizational approaches. It covers all the organizational aspects considered in other design processes *"(roles, tasks, plans, goals, organizations, resources, agents and, in this case, holonic structures), rich interactions (communication, protocols, messages) and a formal definition of the domain knowledge (ontology)"*.

Nevertheless, ASPECS does not cover some aspects such as those related to the tasks to be accomplished by the organization and the rules to observe in order to ensure the profitable achievement of the goals of the organization. For these reasons the $\mathcal{M}$oise+ metamodel (deduced from [12]) was taken into account. In particular, we have paid attention to the following $\mathcal{M}$oise+ concepts: the Role constraining the agent's behavior; the Organizational Link regulating the social behavioral part of agents and the group, to which agents belong; the Norms, which rule the set of roles and missions that agents can do.

Our work consists in the definition of a new metamodel that emphases the normative aspect of a HMAS. To do this: (i) we have extended the ASPECS's Problem and Agency Domain metamodels with the previous $\mathcal{M}$oise+ concepts; (ii) we have specialized the concept of Norms into three categories: Behavioral, Structural and Adaptive Norms; (iii) we have mapped these new extended metamodels in the Solution Domain provided by JaCaMo platform. In the following subsections, we explain these three steps. For the sake of clarity, in Fig. 1 and Fig.2, we have differently colored the new concepts to highlight the differences with ASPECS metamodels. In the following, we give a detailed description of new concepts referring to ASPECS metamodel for those not mentioned in this paper.

3.1 Problem Domain Metamodel

The extended Problem Domain metamodel is shown in Fig. 1. According to ASPECS, an *Organization* can be an aggregate of other sub-organizations. Each

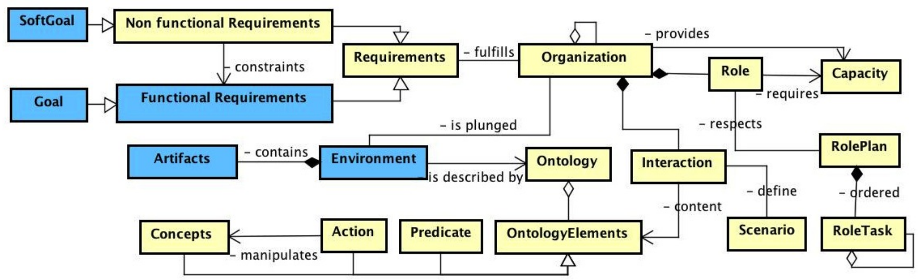

Fig. 1. Problem Domain metamodel.

organization is composed of *Roles* which specify the *Capacities* that should be owned by an agent to play them. *Interactions* between *Roles* define *Scenarios* where each *Role* contributes to the achievement of organizational objectives (*Requirements*). Unlike ASPECS, we highlight that an *Organization* is plunged in an *Environment* composed of *Artifacts* that can be either passive elements (e.g. resources) used by agents and normative elements (e.g. social laws) imposed on organizations and their members in order to fulfill their goals. Each element of the *Environment* is described by means of an *Ontology* providing a common vocabulary and a machine-readable knowledge.

In the following we give a brief description of the new concepts we have introduced in the Problem Domain.

Functional Requirements describe the functions the software has to execute. In some context, often also in agent-oriented systems, they are also known as *Capabilities*[1].

Nonfunctional Requirements [1] are seen as constraints or quality requirements of the solution to be adopted.

Goals and **Softgoals** are a specialization of functional and nonfunctional requirements respectively. A *Goal*, representing an actor' strategic interest, can satisfy a system requirement. While *Softgoals* are generally considered as goals for which it is difficult to decide whether they are satisfied or not. In our model we use *Softgoals* to constrain *Goals*.

The **Environment** is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interactions among agents and the access to resources. The passive components of the system, such as resources and objects, that are shared and used, cooperatively or competitively, by agents to support their activities or norms, rules, physical and social laws that act on the environment or govern its living entities are represented by means of **Artifacts** [19]. For this reason, we see the environment as a set of artifacts that form a context in which agents perform their tasks and pursue their goals. A special kind of artifacts that we considered in this paper are the norms, which will be deeply explained in the next section.

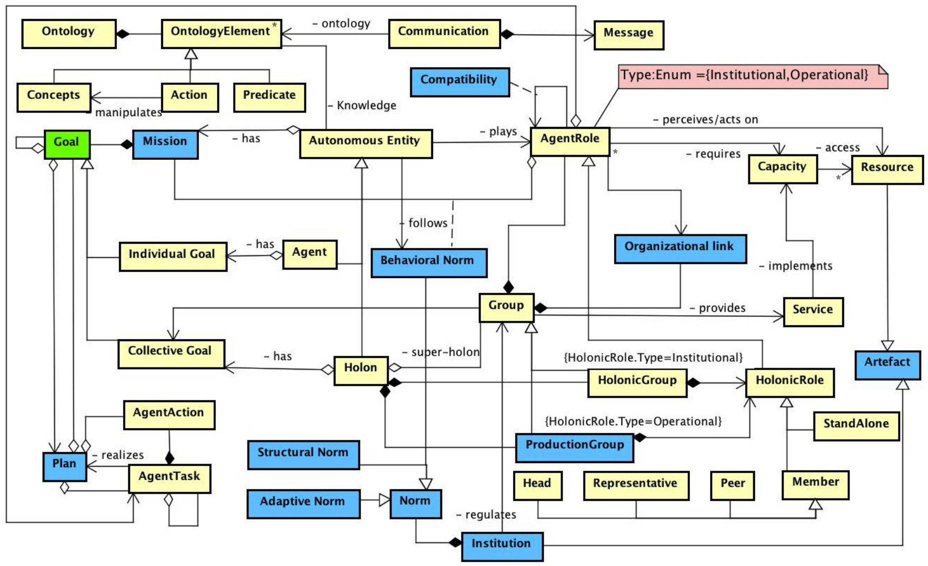

Fig. 2. Agency Domain metamodel.

3.2 Agency Domain Metamodel

Several complex dynamic systems, naturally occurring as well as those created by the society, show common features. Such as (i) the composition of entities operating in parallel *(nerve cells into brain, individuals or enterprise in a market economy, etc...*), (ii) different levels of organization (*proteins and lipids form a cell, cells form tissues, tissues form organisms and so on*), (iii) the continuous adaptation of their components through the process of evolution (*adaptation involves the recombination of the component elements or the generation of a new one*), only to name but a few. Adopting an organizational approach to model these kinds of system is a very useful way to represent their structure as well as all the elements indispensable to define a solution.

The Agency Domain metamodel shown in Fig. 2 describes an organizational solution from the agent-oriented perspective along with normative concepts. As ASPECS does, we consider *Holons* the base elements of the organizational solution. The *Holons* are recursively composed of other Holons and, at the same time, each of them is composed of groups. In our extended model we considered two different kinds of *Groups*: the **Holonic Group** and the **Production Group**. At the first level of abstraction, members of the *Holonic Group* play *Institutional Roles* to which are assigned the task to regulate the organizational aspect of the system and to enforce the norms. In that, we accepted the position of V. Dignum *et al.* who say [8] that *Institutional Roles* are roles needed in order to keep the society going. Differently, members of the *Production Group* play *Operational Roles* to which is assigned the task to perform activities necessary to pursue the organizational objectives in accordance with the *Behavioral Norms* and their *Mission*. A **Mission** is "*a set of constraints that the agent must take into account when it wants to execute parts of this task. It defines an allowed behavior as a consistent set of authorization related to goals to be achieved, plans to follow, actions to execute and resource to use*" [12]. A set of missions to which an agent must obey is assigned to each *AgentRole*. A **Plan** is defined as an oriented graph where each node can be a simple *Agent Action* or *Agent Task* or a set of sub-goals. It represents the way to reach the organizational objective. In this context a *Goal* is seen as an aggregate of *Plans*. Roles inside different *Holons* are linked by means of *Organizational Links*. The **Organizational Links** define the way in which the social exchanges between *Agent Roles* occur [12].

The most significant difference compared to ASPECS is the introduction in the Agency Domain of the **Norms**. A general definition of Norm is *an authoritative standard or model*. We have specialized this concept making explicit different kinds of Norm: *Behavioral*, *Structural* and *Adaptive Norms*.

We called **Behavioral Norm** what Boella et.al [3] define "*a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behavior*", similar to the concepts of *Regulative norms* described as *the expected contributions to the social system* [23]. In our model, a *Behavioral Norm* regulates the way a *Agent Role* performs a *Mission*. Two main types of *Behavioral Norms* are *Obligation* and *Permission*. When an **Obligation** is established between an *Agent Role* and a *Mission*, the Autonomous Entity playing the *Agent Role* is obliged to execute the *Mission*. Instead, when the Behavioral Norm is a **Permission** then the Autonomous Entity playing the *Agent Role* can decide to execute the *Mission* or not [12].

The *Structural* and *Adaptive Norms* are instead two new kinds of norm we propose in order to regulate the static and dynamic aspects of an organization separately. The **Structural Norms** define the static structural aspect of the system at the design time, that is the initial composition defined by the designer for the organization to fulfill its objectives. The **Adaptive Norms** govern the state transition of the organization from a given configuration to a new one according to needs emerging from environmental changes. By means of adaptive norms the agent society spontaneously evolves toward another optimal configuration for the new state of the world.

The last element introduced in the model is the *Institution*. **Institutions** [7] provide the social and institutional backbone of the agent society and they are the place where social norms are explicitly specified.

In the following subsection we highlight some general structural norms that a holonic organization must comply with. As regard the adaptive norms, in this paper we provide only a preliminary introduction without discussing any theoretical details that will be argued in another specific work.

Structural Norm

When we want to adopt a solution based on organizations (without going into the details of a methodological approach for holonic organization design), the organizational structure is the first element to be defined. The choice of the appropriate organizational schema is related, first of all, to the global objective of the system. Its performance depends on the way tasks are distributed among individuals, how their responsibilities (assigned to Roles) are defined and how they could be aggregated in groups. For instance, organizational groups can be created as functional units responsible to execute either a process or some of its phases, depending on the interdependence of groups involved in the work flow execution. In the following, we exemplify an organizational schema by means of structural norms. In particular, we distinguish the norms needed for the design of holons from those used for the definition of organizational schema.

The following list shows a sub-set of structural norms that allow us to define a holarchy [17]:

1. A generic holonic structure must contain at least three levels of representation. The level (n) represents a holon as a *whole* with its unique characteristics, the level (n-1) contains the holons subordinated to the previous one, finally, the level (n+1) holon is a *super-holon* containing the level (n) holon (and others if required).
2. A *top holon* is not included in any holon of level(n+1).
3. A *bottom holon* does not include holons of level (n-1).
4. A *stand-alone* holon is a non-member holon. It can be seen as a top and bottom holon at the same time.
5. Holons of the same level cannot be included in each other.
6. The number of holons at level (n) cannot be greater than that of the holons at level (n-1).
7. Holons at level (n) can be part simultaneously of holons at level(n+1).
8. Holons at level (n) that are not decomposable can be brought to a lower level(n-1) by means of virtual holons (see Fig. 5).

While, in order to define the organizational schema such as for example a *moderated group* [11] three roles are necessary. The *Head* role identifies the decision maker of the holon. The *Representative* role is the interface of the holon outside the world. Finally, the *Peer* role identifies the default members, generally they perform tasks assigned by the Head. This organizational schema will be well-formed applying the following structural norms:

1. A moderated group must contain at least one individual playing the *Representative* role.
2. It must contain at least one individual playing the *Head* role.
3. It can include from zero to a generic number of *Peer* players.
4. Head and Peer are exclusive roles.
5. Members belonging to only one super-holon adopt the *Part* status.
6. Members of the moderated group can belong to more than one super-holon, adopting the *Multi-Part* status.
7. The Part status is adopted by default.
8. Part and Multi-Part are exclusive status.

We have adopted these norms to define the structural specification for our case study.

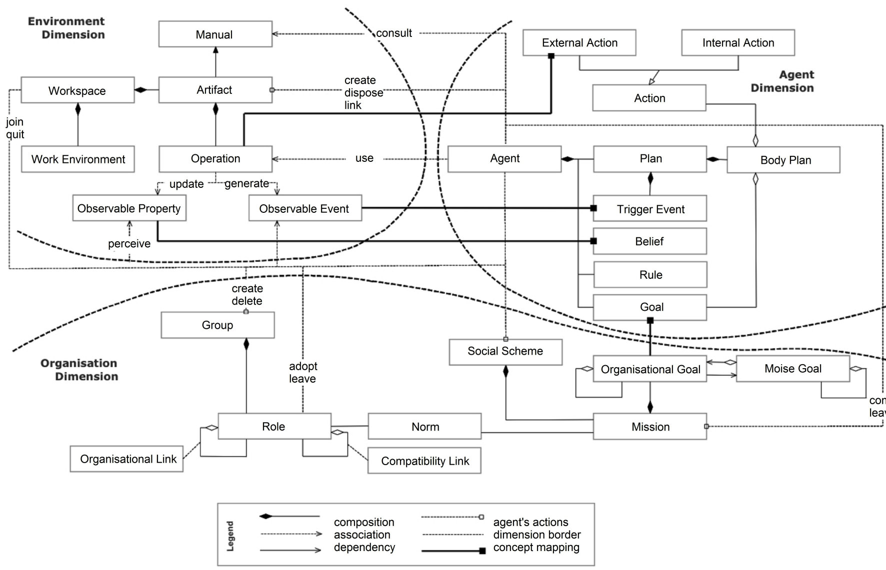

Fig. 3. The JaCaMo Meta-Model adapted from[4].

3.3 Conceptual Mapping in a Solution Domain

Designing systems normally results in realizing a possible implementative solution in a given platform. The aim of this section is to give a possible implementative solution to a holonic organization using the platform called *JaCaMo*[4]. Among several existing platforms, we have chosen JaCaMo because it natively supports key concepts such as organizations and norms.

JaCaMo is a framework providing a set of programming abstractions that allow us to implement a holonic organization of BDI agents in a shared distributed artifact-based environment. It gives an integrated vision of three fundamental aspects for the implementation of a multi-agent system, such as: the agents belonging to the MAS; the organizational structure on which the MAS is based; the environment in which agents are plunged. Fig. 3 shows an adapted version of JaCaMo metamodel [4].

In *JaCaMo*, an *Agent* is an autonomous entity owning *Beliefs*, *Plans* and *Rules* that allows him to pursue its *Goals*. The *Beliefs* represent the knowledge owned by an agent about itself and the environment in which it lives. The *Rules* are ways to infer new knowledge starting from some *Beliefs*. The *Goals* are the states of the world the agent wants to reach. Finally, the *Plans* are ways to reach goals. The *Trigger Event* defines the circumstances in which a plan should be considered. The *PlanBody* is the core of a plan. It contains *Actions* and others sub-goals to be performed/achieved in order to fulfill the goal a plan was defined for. *Actions* are simple tasks that an agent can perform. There are two kinds of action: *Internal Actions* (that does not produce changes in the environment) and *External Actions* (changing the environment).

An *Agent* interacts with the *Artifacts* (non living entities) in the environment performing *Operations*. An operation generates *Observable Events* and it updates an *Observable Property* of the Artifact.

<table>
<tr><th>Agency Domain Element</th><th colspan="2">JaCaMo Element</th><th>Code Portion</th></tr>
<tr><td>Holon</td><td colspan="2">Group</td><td><organisational-specification id=[Holon ID]> … </organisational-specification></td></tr>
<tr><td>Holonic Group</td><td colspan="2">Group</td><td><sub-groups>
<group-specification id=[Holonic Group ID] > … </ group -specification>
</sub-groups></td></tr>
<tr><td>Production Group</td><td colspan="2">Group</td><td><sub-groups>
< group -specification id=[Production Group ID]> … </ group -specification>
</sub-groups></td></tr>
<tr><td>AgentRole</td><td colspan="2">Role</td><td><role id=[Role Name]></role></td></tr>
<tr><td>Organiz. Link</td><td colspan="2">Organizational Link</td><td><link from=[Role Name] to=[Role Name] type=[Autority | Acquaitance | Communication] scope= [intra-group | inter-group] extends-sub-groups=[True | False] bi-dir=[True | False]/></td></tr>
<tr><td>Compatibility</td><td colspan="2">Compatibility Link</td><td><compatibility from=[Role Name] to=[Role Name]
scope=[intra-group | inter-group] extends-sub-groups=[True | False]
bi-dir=[True | False]/></td></tr>
<tr><td>Resource</td><td colspan="2">Artifact</td><td>class [ArtifactName] extends Artifact {
void init() { defineObsProp ([ObservablePropertyName], 0);} … }</td></tr>
<tr><td>Service</td><td colspan="2">Operation</td><td>@OPERATION
void changeObservableProperty (int PropertyValue) {
int c =getObsProperty([ObservablePropertyName]).intValue();
if (PropertyValue > c)
updateObsProperty([ObservablePropertyName],PropertyValue);}</td></tr>
<tr><td>Ontology Element</td><td colspan="2">Belief</td><td>functor(term1, ..., termn)[annotation1, ..., annotationm]</td></tr>
<tr><td>Comunication</td><td colspan="2">Internal Action</td><td>.send([AgentName], [Performative], [Content])</td></tr>
<tr><td>Individual Goal</td><td colspan="2">Goal</td><td>! functor(term$_1$, ..., term$_n$)</td></tr>
<tr><td>Collective Goal</td><td colspan="2">Organisational Goal</td><td><goal id=[Goal ID]></td></tr>
<tr><td>Agent</td><td colspan="2">Agent</td><td>[AgentName] agentArchClass jmoise.OrgAgent;</td></tr>
<tr><td rowspan="2">Plan</td><td colspan="2">Plan</td><td>Triggering Event : Context <- PlanBody.</td></tr>
<tr><td colspan="2">Organizational Plan</td><td><plan operator=[sequential | parallel |choice] >… </plan></td></tr>
<tr><td>Agent Task</td><td colspan="2">Body Plan</td><td>PlanBody</td></tr>
<tr><td rowspan="2">Agent Action</td><td rowspan="2">Action</td><td>Internal Action</td><td>. actionName(term$_1$, ..., term$_n$)</td></tr>
<tr><td>External Action</td><td>actionName(term$_1$, ..., term$_n$)</td></tr>
<tr><td>Mission</td><td colspan="2">Mission</td><td><mission id=[Mission ID] >…</mission></td></tr>
<tr><td rowspan="2">Behavioral Norm</td><td colspan="2">Norm</td><td><norm id=[Norm ID] type=[Obligation | Permission]
role=[Role Name] mission=[Mission ID] /></td></tr>
<tr><td colspan="2">Rule</td><td>functor(term$_1$, ..., term$_n$) :- Logical Expression.</td></tr>
<tr><td>Structural Norm</td><td colspan="2">Formation Constraints on Group, Role, Mission</td><td><role id=[RoleName] min=[0…N] max=[0…N]>
<mission id=[Mission ID] min=[0…N] max=[0…N]> …</td></tr>
<tr><td>Adaptive Norm</td><td colspan="2">Belief</td><td>adaptiveNorm ([RoleName], [Entry_Condition], [Plan])</td></tr>
</table>

Fig. 4. Conceptual mapping among Agency Domain and JaCaMo elements.

Finally from the organizational viewpoint, an agent can adopt *Roles* defined into a *Group*. Agents playing different roles can interact each other only if their roles are connected by *Organizational Links*. An Agent can also play two or more compatible roles at the same time. When an agent adopts a *Role* it is committed to a *Mission* by means of *Norms*. A mission is responsible of a set of *Organizational Goals* reachable by means of *Organizational Plans*. The *Social Scheme* groups *Missions* and it defines the functional aspect of an organization.

The table in Fig. 4 shows the conceptual mapping among the Agency Domain and the JaCaMo metamodel elements along with a codified solution. In particular, we want to underline that *Plan* and *Behavioral Norm* elements do not have a unique mapping with the elements of the solution domain. This is due to the dual nature of a holonic MAS we want to implement. In the solution domain, in fact, Plans and Behavioral Norms are defined differently when they refer to the holon as a whole or as a part.

As we previously said, we adopted the HMAS paradigm in order to implement holarchies in software systems. In HMAS a holon is a set of individual agents organized according to an organizational model. When we want to model an HMAS we usually define the *Collective Goals* of the entire holon as well as the *Individual Goals* of the members of the holon (single agents). In our Agency Domain both *Individual* and *Collective Goals* can be reached by means of *Plans* (hereafter *Agency Domain Plan*). In the Solution Domain, the *Agency Domain Plan* concept is associated to two different elements (see Fig. 4): *Plan* and *Organizational Plan*. An *Agency Domain Plan* can be mapped in a *Plan* of the JaCaMo agent dimension in order to define as an agent could reach its own *Goal* (see Fig. 3). The *Agency Domain Plan* can be mapped in an *Organizational Plan* in order to define as an entire holon could achieve its own *Organizational Goal* (see Fig. 3).

As concerns a *Behavioral Norm* (see Fig. 4), it can be translated in the Solution Domain in two different elements: *Norm* and *Rule*. The former regulates the agent's behavior playing *Roles* inside a holonic system. The latter may regulate an agent 's behavior in the environment independent from the Role it plays.

As regards the *Structural* and *Adaptive Norms*, the JaCaMo meta-model does not support natively these kinds of norms. Thus, we have mapped the *Structural Norm* in the formation constraints imposed on Group, Role and Mission elements of the JaCaMo metamodel. We are currently working for the definition of a new element in the solution domain that may directly implement a Structural Norm.

Conversely, we have already defined a way to represent an Adaptive Norm in the Solution domain (see table in Fig. 4). They are codified as a Beliefs with the following specific notation:

adaptiveNorm ([**RoleName**], [**EntryCondition**], [**Plan**])

where *[RoleName]* identifies a list of roles to which the adaptive norm can be applied. *EntryCondition* represents a set of environment changes to whom the agent (playing the RoleName) tries to adapt itself. *Plan* define how the agent could adapt itself.

For space concerns, we omit to detail the remaining elements of Fig. 4, which are however easily understood because they have a direct codification in JacaMo framework. In the following section we present our case study.

4 Case Study: Virtual Enterprise for Logistics

The work presented in this paper was carried out under the IMPULSO[5] project and it represents the solution we have studied for it. IMPULSO - *Integrated Multimodal Platform for Urban and extra urban Logistic System Optimization* - offers an integrated system for goods management within the logistic districts, for their storage in special metropolitan distribution centers and finally for distribution within the cities. The development of the IMPULSO system was the test-bed for evaluating and assessing the newly created metamodel with all its concepts. Indeed through the enactment of the design activities devoted to instantiate each concept we were able to create the model of the system (the Figures from now on are parts of the artefacts composing such model) on the base of the right specification provided by the metamodel. We experienced the completeness of the proposed metamodel, indeed both the domains contain all the useful concepts for representing the problem we were dealing with and for describing the solution in terms of holons. Moreover we were able to analyze and establish the behavior of each part of the system through the use of the identified norms.

In the following subsections only three artifacts of the development process are illustrated. They deal with the concepts of holon, group, role and norm.

The Holonic Architecture. The whole Impulso System was modeled as a Virtual Enterprise that is a holarchy of collaborative systems, where each system is a holon itself. Each of them is composed of other systems that act according to the same organizational schema, at the same time they perform different functions at lower levels of resolution. For space concerns, we show only a member of Impulso Holarchy: the Yard Management System(YMS).

The YMS deals with goods traffic inside logistic districts. It manages the automatic container loading and unloading by means of the use of AGVs (Autonomous Guided Vehicles) which move independently but are coordinated in accordance to predetermined patterns by a remote control center. Fig. 5 shows the holonic architecture we have designed for YMS. As we can see, the YMS is composed by three holons: the YMC (*Yard Management Central*), the Freight Forwarders and the YMP (*Yard Management Peripheral*). These holons interact to fulfill the goal of their organization, the YMS, although they themselves are autonomous entities with personal objectives. The holonic enterprise framework, which connects enterprise entities, allows information exchange through communication channels and resources management.

Groups, Roles and Norms. In this subsection we define the entire composition of the holonic organization of the YMS (see Fig. 5). In particular, we define its structural and functional aspects correlated to its normative features.

According to the metamodel shown in Fig. 2, there are two aspects that overlap in a holon. The first is the *holonic aspect* that is directly related to the

[5] Further information available at http://www.vitrociset.it - Section Ricerca&Sviluppo

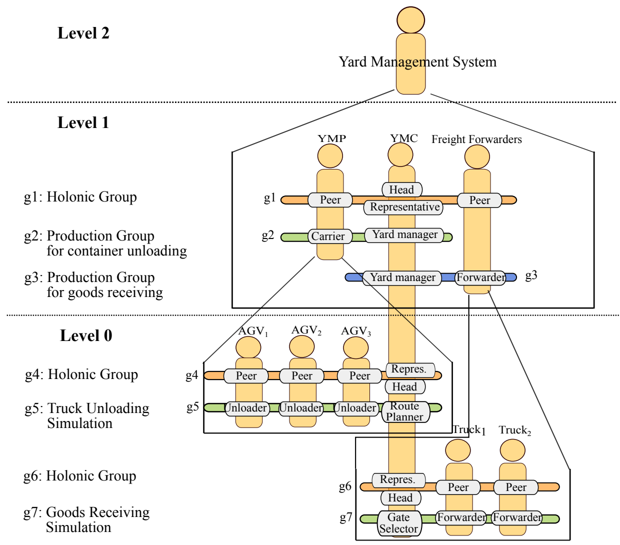

Fig. 5. Roles/Groups of the Yard Management System.

holonic character of the entity, i.e. a holon (super-holon) is composed of other holon members. As Fig. 5 shows, the YMS super-holon is an entity on its own although composed by members. So, the holonic aspect refers how members organize and manage their representative super-holon (i.e. how they form the Holonic Group). We adopted the *moderated group* configuration as organizational structure for the Holonic Group of each super-holon. Thus, each Holonic Group is created according to the structural norms defined in the section 3.2 which related to the moderate group formation. The second aspect of the holon is related to the problem the members are trying to solve (we will call this the production aspect in order to maintain a uniform nomenclature). The production aspect refers to how members of the holon are organized to pursue their goals according to the global objective of their super-holon. This holonic representation, by means of holonic and production groups, allows to clearly distinguish the different features and functionalities to be attributed to each member.

In the following, we describe only the lowest level of abstraction of the YMS architecture. At this final (finer grained) level of decomposition (see Fig. 5), the holons are represented by groups of agents which play institutional and operational roles at the same time. We focus only on operational roles and production groups, since the institutional roles of the holonic groups have been already described in the section 3.2.

For the simulated scenario, we have defined two production groups (*Truck Unloading Simulation* and *Goods Receiving Simulation*) of two high-level holons. The *Truck Unloading Simulation* is a group formed by the Unloader and Route Planner roles. The Route Planner can be played by YMC agents, which have the capacity to perform the related task. While the Unloader is played by AGV agents which emulate the behaviour of real automated guided vehicles. The *Goods Receiving Simulation* group is formed by the Forwarder and Gate Selector roles. The Forwarder is a role adopted by agents emulating the behaviour of the trucks. As we can see from Fig. 5, the Route Planner player in the Truck Unloading Simulation group adopts the role of Gate Selector in the Goods Receiving Simulation group, at the same time. This is allowed by the structural norm concerning the multi-part status previously defined (see section 3.2).

In the following list we only show some structural norms we have defined for regulating the formation of Truck Unloading Simulation groups. Then, we present their codification in JaCaMo framework. We avoid to list them all because the presented subset provides enough information in order to understand the purpose of the structural norms.

1. The Route Planner role must be played by at least one agent.
2. The Route Planner role can be played by only one agent.
3. The Unloader role must be played by at least one agent.
4. The Unloader role can be played by at most 10[6] agents.
5. At least one Truck Unloading Simulation group must be active.
6. It can not be created more than 10 Truck Unloading Simulation groups simultaneously.

The norms related to the Route Planner (i.e: norms 1. and 2.) and the Unloader (i.e: norms 3. and 4.) role are codified respectively in

```
<role id="route_Planner" min="1" max="1"/>
<role id="unloader" min="1" max="10"/>.
```

While the last two rules concerning the Truck Unloading Simulation groups (i.e: norms 5. and 6.) are translated into

```
<group-specification id="truck_Unloading_Simulation" min="1" max="10">.
```

As concern the functional aspect of Yard Management System, it is defined by set of plans and missions the agents can commit into a Social Scheme (see Fig.3). It describes how an organization can achieve its global goals. The Fig.6 gives an overview of the Social Scheme of the organization shown as a goal decomposition tree. The root goal of the Yard Management System is sorting out goods toward metropolitan centers. To do that, the members of two production groups can play the permitted roles according to the structural norms and commit to some missions according to behavioral norms described below. Groups perform their activities independently. Holonic groups are responsible for managing their respective production groups and their coordination. For everything else, the Fig.6 is self explanatory.

In the following we show those behavioral norms, which represent JaCaMo Norm elements, according to following template:

[6] This is a constraint of the project because the AGVs are costly resources.

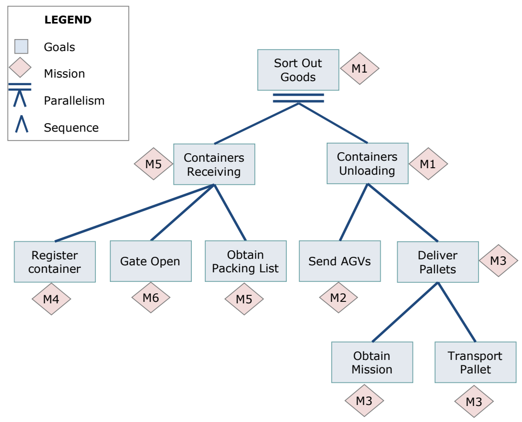

Fig. 6. The functional view of Truck Unloading simulation group represented by means of a goal/mission decomposition tree.

```
norm<id> : type=[Obligation | Permission]
role=<RoleName> mission=<MissionName>
```

As we have previously said, these norms impose agents to commit to certain missions when they choose to play a role. Some of them are reported below:

```
norm 1: type=Obligation role=unloader
        mission=AgvMission
norm 2: type=Permission role=forwarder
        mission=ForwarderMission
norm 3: type=Permission role=representative
        mission=ManagementMission
norm 4:  type=Permission role=representative
        mission=RecruitmentMission
```

Finally, we have introduced some adaptive norms that allow us to regulate the dynamic evolution of the Yard Management System. We have defined a set of norms that allow the adaptation of the holon to perceived environmental changes. Three examples are:

- If the workload grows beyond some limit (for instance a new truck arrives to be unloaded), the Representative holon creates a new Truck Unloading Simulation group.
- If the workload decreases (for instance unloading operations of a truck are over), Truck Unloading Simulation groups are removed proportionally by the Representative.
- If all role-players leave Representative roles an election has to be made for new players.

According to the definition given in section 3.3, the codified solution of the listed adaptive norms are:

1. *adaptiveNorm*(representative, workload(W) & W >Treshold, @HolonReorg +! createUnloadingSimGroup <− .createGroup(GroupSpec, IDHolon, ID))

2. *adaptiveNorm*(representative, workload(W) & W <Treshold2, @HolonReorg2 +! removeUnloadingSimGroup <− .removeGroup(IDHolon, ID))

3. *adaptiveNorm*(Role, violated(RepresentativeStructNorm), @HolonReorg3 +! newElection <− vote(Player)), where RepresentativeStructNorm is

   ```
   <role id="representative" min="1" max=N />
   ```

   that is the codification of the structural norm 1. of section 3.2 related to moderated group formation.

We want to point out that all upper-case terms are variables that can assume different values during the running of the system. This means that the same adaptive norm can be triggered by different conditions. For example for the first norm, the threshold is a variable value according to the number of created group. In fact the first time the threshold has a defined value according to the amount of work the members of Truck Unloading Simulation group are able to perform. Thus when this norm is applied not only a new group is created but a new value of threshold is automatically defined. This avoids the creation of groups not necessary. Analogous considerations can be made about the second norm. Moreover, if the second norm is applied, the number of Truck Unloading Simulation groups can not become less than one, because it violates the above defined structural norm (i.e: norm 5.) of this production group.

Instead, as concerns the last norm, it is different from the previous ones because it can be adopted by different roles (Role in this norm is a variable) and it is triggered by a violation of a structural norm related to the formation of a moderated group. Thus when this violation occurs, all agents, playing roles inside holons, vote a player from a list of possible candidates according with some defined criteria.

5 Conclusions

In order to solve problems and engineering systems related to fields in which a high level of coordination, control and automation is needed we propose an extension of the ASPECS metamodel obtained by introducing some new concepts such as Norms. Norms are used to regulate holons' behavior, these norms separately deal with the behavioral aspect of the holonic members from the organizational one. From the agents viewpoint, behavioral norms impose constraints to their actions in order to maintain a social order. Conversely, from an organizational perspective it is useful to separate the static aspect from the dynamic one, in this paper this is done by respectively introducing Structural and Adaptive Norms. The formers define the static structural aspect of the system at design time and provide the initial composition required to the organization to fulfill its objective. The latters govern the state transition of the organization from a given configuration to a new configuration to fit the environmental changes.

The proposed metamodel fully supports, and we experimented that by developing the IMPULSO system, a methodological approach for holonic multi-agent system design in which the holons are ruled by means of norms. In the future we will improve the design process based on the new metamodel that is obviously an extension of ASPECS.

References

1. A. Abran, J. Moore, P. Bourque, R. Dupuis, and L. Tripp. *SWEBOK®: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2004.
2. F. Bellifemine, A. Poggi, and G. Rimassa. JADE–A FIPA-compliant agent framework. In *Proceedings of PAAM*, volume 99, pages 97–108. London, 1999.
3. G. Boella, L. Van Der Torre, and H. Verhagen. Introduction to normative multi-agent systems. *Computational & Mathematical Organization Theory*, 12(2):71–79, 2006.
4. O. Boissier, R. Bordini, J. Hübner, A. Ricci, and A. Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 2011.
5. R. Bordini, J. Hubner, and M. Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. Wiley-Interscience, 2007.
6. M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam. ASPECS: an agent-oriented software process for engineering complex systems. *Autonomous Agents and Multi-Agent Systems*, 20(2):260–304, 2010.
7. V. Dignum and F. Dignum. Modelling agent societies: co-ordination frameworks and institutions. *Progress in Artificial Intelligence*, pages 7–21, 2001.
8. V. Dignum, J. Meyer, H. Weigand, and F. Dignum. An organization-oriented model for agent societies. In *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications*, 2002.
9. J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 222. Addison-Wesley New York, 1999.
10. K. Fischer, M. Schillo, and J. Siekmann. Holonic multiagent systems: A foundation for the organisation of multiagent systems. *Holonic and Multi-Agent Systems for Manufacturing*, pages 1083–1084, 2004.
11. C. Gerber, J. Siekmann, and G. Vierke. Holonic multi-agent systems. Technical report, Université- und Landesbibliothek, 1999.
12. M. Hannoun, O. Boissier, J. Sichman, and C. Sayettat. MOISE: An organizational model for multi-agent systems. *Advances in Artificial Intelligence*, pages 156–165, 2000.
13. J. Hubner, J. Sichman, and O. Boissier. Moise+: towards a structural, functional, and deontic model for mas organization. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, page 502. ACM, 2002.
14. J. Hubner, J. Sichman, and O. Boissier. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3):370–395, 2007.
15. D. Isern, D. Sánchez, and A. Moreno. Organizational structures supported by agent-oriented methodologies. *J. Syst. Softw.*, 84:169–184, February 2011.

16. A. Koestler. The ghost in the machine. *Psychiatric communications*, 10(2):45, 1968.
17. P. Mella. The holonic revolution: Holons, holarchies and holonic networks: The ghost in the production machine, 2009.
18. V. Morreale, S. Bonura, G. Francaviglia, F. Centineo, M. Cossentino, and S. Gaglio. Reasoning about goals in BDI agents: the PRACTIONIST framework. In *Proceedings of the 7th WOA 2006 Workshop, From Objects to Agents (Dagli Oggetti Agli Agenti)*, volume 204 of *CEUR Workshop Proceedings*, pages 26–27. CEUR-WS.org, 2006.
19. A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, 2008.
20. A. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: A BDI reasoning engine. *Multi-Agent Programming*, pages 149–174, 2005.
21. A. Rao. AgentSpeak (L): BDI agents speak out in a logical computable language. *Agents Breaking Away*, pages 42–55, 1996.
22. A. Ricci, M. Viroli, and A. Omicini. Carta go: A framework for prototyping artifact-based environments in mas. *Environments for Multi-Agent Systems III*, pages 67–86, 2007.
23. G. Therborn. Back to norms! On the scope and dynamics of norms and normative action. *Current Sociology*, 50(6):863, 2002.
24. C. M. Toledo, R. H. Bordini, O. Chiotti, and M. R. Galli. Developing a knowledge management multi-agent system using jacamo. In *Proceedings of the 9th international conference on Programming Multi-Agent Systems*, ProMAS'11, pages 41–57, Berlin, Heidelberg, 2012. Springer-Verlag.
25. M. Ulieru, R. Brennan, and S. Walker. The holonic enterprise: a model for Internet-enabled global manufacturing supply chain and workflow management. *Integrated Manufacturing Systems*, 13(8):538–550, 2002.
26. J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing Multiagent Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, November 2005.
27. M. Winikoff. Jack™ intelligent agents: An industrial strength platform. *Multi-Agent Programming*, pages 175–193, 2005.
28. M. Wooldridge. *Reasoning about rational agents*. The MIT Press, 2000.