Towards to an Agent-Oriented Meta-Model for Modeling Vehicular Systems with Multi-Configuration Ability

Mohamed Garoui
PRINCE Research Unit, ENSI,
Manouba, Tunisia
garouimohamed2010@gmail.com

Belhassen Mazigh
Department of Computer Sciences
FSM, Monastir, Tunisia
belhassen.mazigh@gmail.com

Béchir El Ayeb
PRINCE Research Unit
FSM Monastir, Tunisia
ayeb_b@yahoo.com

Abderrafiaa Koukam
IRTES-SET, EA 7274, UTBM,
F-90010 Belfort cedex, France
abder.koukam@utbm.fr

***Abstract*— Agent technology is a software paradigm that permits to implement large and complex distributed applications. In order to assist the development of multi-agent systems, agent-oriented methodologies (AOM) have been created in the last years to support modeling more and more complex applications in many different domains. By defining in a non-ambiguous way concepts used in a specific domain, Meta modeling may represent a step towards such interoperability. In the transportation domain, this paper proposes an agent-oriented meta-model that provides rigorous concepts for conducting transportation system problem modeling. The aim is to allow analysts to produce a transportation system model that precisely captures the knowledge of an organization so that an agent-oriented requirements specification of the system-to-be and its operational corporate environment can be derived from it. To this end, we extend and adapt an existing meta-model, Extended Gaia, to build a meta-model and an adequate model for transportation problems. Our new agent-oriented meta-model aims to allow the analyst to model and specify any transportation system as a multi-agent system. In this paper, we aim to propose an Agent-Oriented meta-model adequate to any Transportation Systems with multi-configuration ability problem.**

Keywords-*Agent technology; Transport domain; Meta-model; multi-agent system.*

I. INTRODUCTION

The purpose of the Agent-Oriented Software Engineering is the creation of a path towards integration and interoperability of methodological approaches for Multi-Agent Systems (MAS) development. This involves the definition of a common framework for MAS specification, which includes the identification of a minimum set of concepts and methods that can be agreed in the different approaches. The tool for defining this framework is meta-modeling. The principle of meta-modeling has been already used in other fields of software engineering, for instance, in the specification of Unified Modeling Language (UML) [1] by Object Management Group (OMG) [1], to describe the elements of the language, their constraints and relationships.

In platooning systems Research, such as [2] and [3], each vehicle determines its own position and orientation only from its perceptions of the surrounded environment. In this context, the reactive multi-agent paradigm is well adapted to specify and analyze this system. The interest of those approaches results from their adaptability, simplicity and robustness. In this case, platoon configuration can be considered as the result of the self-organization of a Reactive Multi-Agent System (RMAS). A platoon multi-agent system can then be defined as a set of agents, each one corresponding to a vehicle. Two agent roles can be distinguished: leader and follower agents. The Leader agent interacts only with its environment (road, obstacles, etc.).

Our problem here is that when we model the vehicular system, we need an agents-oriented meta-model that gives us a set of basic concepts. These concepts are necessary to model the entire of transport system problem in different ***environment*** (Urban, Agricultural, and Military) and with various ***navigation policies*** and its ***behavior***. In addition, as soon as we obtain the system model, it will be easy to implement our multi-agent system by using agent oriented programming.

In this paper, our contribution is to provide an agent-oriented meta-model adequate to transportation system problem, which allowed us to model the vehicular system in their navigation environment. Our proposed meta-model has been built by *adopting* and *extending* the existing Extended Gaia meta-model [4] and thus, we define two levels of models inspiring from PASSI meta-model [5]. This seems to us coherent with the most accepted definition of meta-model: a meta-model is a "model of a model", and it provides an explicit representation of the constructs and relationships needed to build specific models within a domain of interest. This proposition arises by remarking that in the field of transport doesn't occur any Agent oriented meta-model to clearly specify and analyze any transport system in the form of multi-agent systems.

We choose to use the Extended Gaia meta-model as it is well adapted to organizational structures such as *teams*, *congregations*, and *coalitions* which are used in clustering and collaborative missions of the platoon entities. Furthermore, the proposed approach must take into account, in their meta-model, the concept of environment and different social structures associated with different application areas (Urban, Agricultural, Military), as indicated in Table I. Extended Gaia specifies the notion of the environment by Environment concept. The abstraction of the environment specifies the set of entities and resources of a multi-agent system can interact with, limiting interactions using the authorized shares.

TABLE I. SOCIAL STRUCTURE ACCORDING TO THE APPLICATION AREAS

Application Area	Suitable Social Structure
Urban	Congregations, Coalition
Agricole	Congregations, Teams, Coalition
Military	Teams, Congregations, Coalition

The Extended Gaia meta-model adds some organizational based concepts. The organization itself is represented with an entity, which models a specific structure (or topology). The organizational rules are considered responsibilities of the organization. They include safety rules (time-independent global invariants that the organization must respect) and a liveness rules (that define how the dynamics of the organization should evolve over time).

This paper is structured as follow: in Section II, we present a state of the art about the existing agent-oriented meta-model used for modeling and specify multi-agent systems. Section III presents our Proposed Agent-oriented Meta-model for transportation systems. Then, Sections IV illustrates our Agent-oriented Meta-model with an application of urban public transportation systems. Finally, Section V concludes by giving a list of possible future works.

II. STATE OF THE ART

Many agent-oriented meta-model have been proposed for modeling of multi-agent system. The first version of the Gaia methodology, which modeled agents from the object-oriented point of view, was revisited 3 years later by the same authors in order to represent a MAS as an organized society of individuals [6, 7].

Agents play social roles (or responsibilities) and interact with others according to protocols determined by their roles. With that approach, the overall system behavior is understood in terms of both micro- and macro-levels. The former explains how agents act according to their roles, and the latter explains the pattern of behavior of those agents. These constraints are labeled organization rules and organization structures, respectively.

A central element of the meta-model of Gaia is the agent entity, which can play one or more roles. A role is a specific behavior to be played by an agent (or kind of agents), defined in terms of permissions, responsibilities, activities and interactions with other roles. When playing a role, an agent updates its behavior in terms of services that can be activated according to some specific pre- and post-conditions. In addition, a role is decomposed in several protocols when agents need to communicate some data. The environment abstraction specifies all the entities and resources a multi-agent system may interact with, restricting the interactions by means of the permitted actions.

The Extended Gaia meta-model adds some organizational based concepts. The organization itself is represented with an entity, which models a specific structure (or topology). The organizational rules are considered responsibilities of the organization. They include safety rules (time-independent global invariants that the organization must respect) and liveness rules (that define how the dynamics of the organization should evolve over time). Given the aggregation association defined from agents with respect to organizations rules and from organizations with respect to organization structures, Gaia permits to design a hierarchical non-overlapping structure of agents with a limited depth. From the organizational point of view, agents form teams as they belong to a unique organization, they can explicitly communicate with other agents within the same organization by means of collaborations, and organizations can communicate between them by means of interactions. If inter-organization communication is omitted, coalitions and congregations may also be modeled.

Process for Agent Societies Specification and Implementation (PASSI) [8] is an iterative-incremental process for designing multi-agent systems starting from functional requirements that adopts largely diffused standards like UML (as the modeling language, although extended to fit the needs of agents design) and FIPA (as the agent platform). PASSI covers all the phases from requirements analysis to coding and testing with a specific attention for the automation of as many activities as possible with the support of PASSI Toolkit (PTK), a specifically conceived design tool.

The PASSI MAS meta-model [8] is organized in three different domains: the Problem Domain (where requirements are captured), the Agency Domain that represents the transition from problem-related concepts to the corresponding agent solution (that is a logical abstraction), and the Solution Domain (where the implemented system will be deployed).

The Problem Domain deals with the user's problem in terms of scenarios, requirements, ontology, and resources; scenarios describe a sequence of interactions among actors and the system. Requirements are represented with conventional use case diagrams. The system operating environment is depicted in terms of concepts (categories of the domain), actions (performed in the domain and effecting the status of concepts) and predicates (asserting something about a portion of the domain elements), the environment also includes resources that can be accessed by agents.

The Agency Domain includes the agent that is the real centre of this part of the model; each PASSI agent is responsible for accomplishing some functionalities descending from the requirements of the Problem Domain. Each agent during its life can play some roles; these are portions of the agent social behavior characterized by some specificity such as a goal, or providing a functionality/service and in so doing it can also access some resources. The Service component represents the service provided by a role in terms of a set of functionalities (including pre- and post-conditions as well as many other details mostly coming from the Ontology Web Language (OWL) specifications), and can be required by other agents to reach their goals. Agents could use portions of behavior (called tasks) or communications to actuate the roles aims.

 ISBN: 978-1-61208-319-3

Agent-oriented Software Process for Engineering Complex Systems (ASPECS) [9] provides a holonic perspective to design MAS. Considering that complex systems typically exhibit a hierarchical configuration, on the contrary to other methodologies, it uses holons instead of atomic entities. Holons, which are agents recursively composed by other agents, permit to design systems with different granularities until the requested tasks are manageable by individual entities.

Being one of the most recent methodologies, it takes the experience gained from previous approaches, such as PASSI and RIO [10], as the base to define the meta-model and the methodology.

The goal of the proposed meta-model is to gather the advantages of organizational approaches as well as those of the holonic vision in the modeling of complex systems. A three layer meta-model, with each level referring to a different aspect of the agent model, is proposed: The *Problem domain* covers the organizational description of the problem. An organization is composed by roles which interact within scenarios while executing role plans. Roles achieve organizational goals by means of their capacities (i.e., what a behavior is able to do). The organizational context is defined by means of ontology. This meta-model layer is used mainly during the analysis and design phases. The *Agency domain* defines agent-related concepts and details the holonic structure as a result of the refinement of the elements defined in the Problem domain. Each holon is an autonomous entity with collective goals and may be composed by other holons. Holonic groups define how members of the holon are organized and how they interact in order to achieve collective goals. At the finest granularity level, holons are composed by groups and their roles are played by agents, which achieve individual goals. A rich communication between agent roles (which are instances of organizational roles) is also supported, specifying communicative acts, knowledge exchange formalized by means of the organizational ontology, and protocols specifying sequences of messages.

III. OUR PROPOSED META-MODEL: PLATOONING META-MODEL

The UML is based on the four-level meta-modeling architecture. Each successive level is labeled from M_3 to M_0 and are usually named meta-meta-model, meta-model, class diagram, and object diagram respectively. A diagram at the M_i-level is an instance of a diagram at the M_{i+1}-level. Therefore, an object diagram (an M_0-level diagram) is an instance of some class diagram (an M_1-level diagram), and this class diagram is an instance of a meta-model (an M_2-level diagram). The M_3-level diagram is used to define the structure of a meta-model, and the Meta Object Facility (MOF) belongs to this level. The UML meta-model belongs to the M_2-level.

After studying the Extended Gaia meta-model, we observe how this explicit and useful models of the social aspect of agents. Although, it was not designed for open systems, and it provides little support for scalability and simplicity to allows improvements to facilitate with its relative concepts. It models both the macro and micro aspects of the multi-agent system. Gaia believes that a system can be regarded as a company or an organization of agents.

In this section, we try to solve our contributions mentioned from the start. It manifests itself to extend and adapt an existing meta-model to build a meta-model and an adequate model for transport problems.

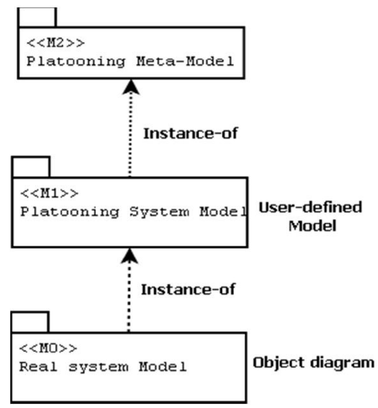

Figure 1. Model instantiation checking

In Fig. 2, the classes presented in black color are the base classes of Extended Gaia meta-model. For against, the blue classes are the classes added to the existing meta-model to be adapted to platooning applications and then help us to implement our own methodology for modeling and dependability analysis. Table II presents the definition of the added new concepts.

TABLE II. DEFINITION OF THE ADDED NEW CONCEPTS

Concept	Definition
Functional Requirement	A function that the software has to exhibit or the behavior of the system in terms of interactions perceived by the user
Non-Functional Requirement	A constraint on the solution. Non-functional requirements are sometimes known as constraints or quality requirements
AgentModel	Abstract description of a formal model which gives an abstract view about ***the agent behavior***.
OrganizationModel	Abstract description of a formal model which gives an abstract view about ***the organization behavior***.

The concept *Functional Requirement* is a function that the software has to exhibit or the behavior of the system in terms of interactions perceived by the user. This concept allowed us to identify our system requirements. The *Non-Functional Requirement* concept provides a constraint on the solution. Non-functional requirements are sometimes known as constraints or quality requirements. *AgentModel* concept gives an abstract view about *the Agent behavior*. *OrganizationModel* gives an abstract view about ***the*** *organization behavior*. Behavior is described by formal state-based models [11].

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-319-3

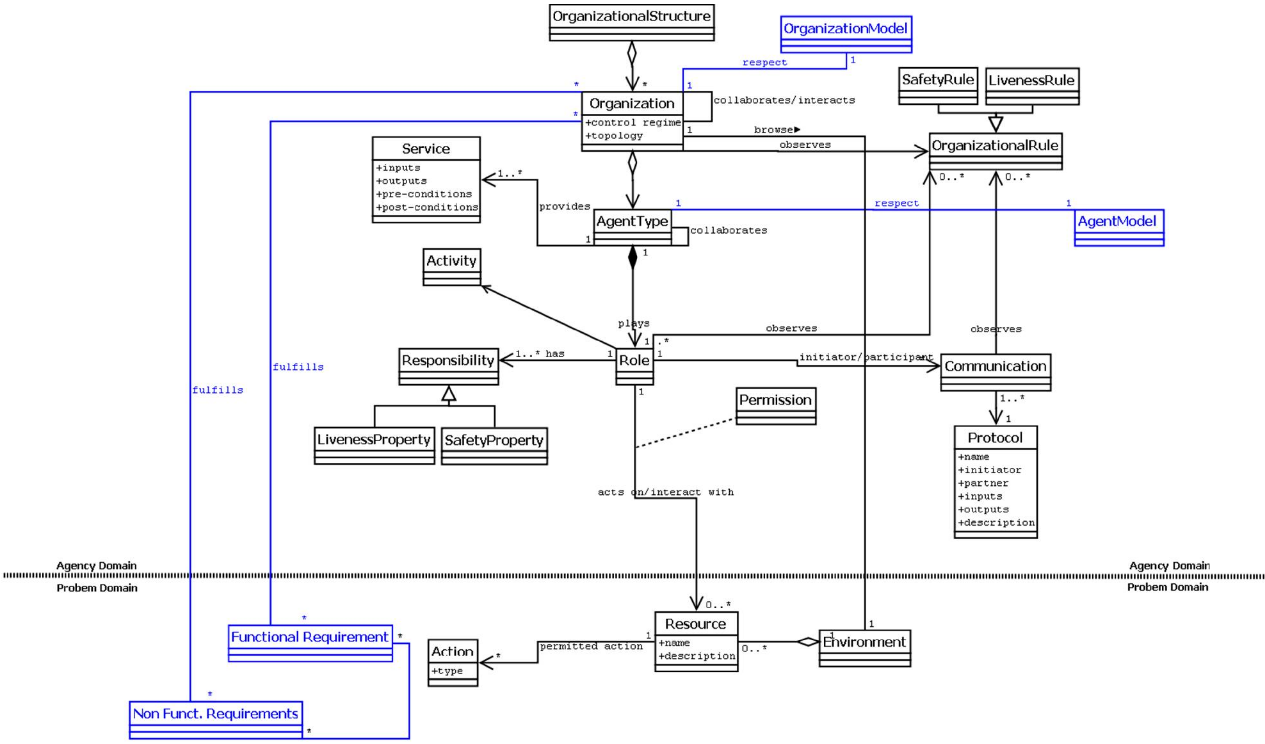

Figure 2. Our proposed Meta-model: Platooning Meta-model.

By inspiring from PASSI [8] and ASPECS meta-models [9], we tried to organize our meta-model in two areas: Problem Domain and Agent Domain. Problem Domain involves elements (Fig. 2) are used to capture the requirements problem and perform initial analysis. Agent Domain includes elements (Fig. 2) are used to define an agent-oriented solution to the problem described in the previous step.

After this, we pass to M_1-level (Fig. 1) describes the *Platooning System Model* (see Fig. 3) which constitutes of instance of the concepts of M_2-level model. This model includes all the basic concepts and necessary for us to model any type of application to platooning with their bodies, interaction, environment, their geometric configuration and formal models associated with each component platoon. The table below provides the concepts related to platooning System Model and their relationship with the concepts of the M_2-level model.

Table III gives an idea about the basic concepts of *Platooning System Model* (Fig. 3) which is instances of our meta-model that shown in the Fig. 2. The *Platoon* concept represents the main element in our model which is an instance of meta-concept *Organization*. Any Platoon is modeled as a set the *Entity*. There are two kinds of entities: Leader and Follower, which are modeled by the two concepts *Leader* and *Follower*. The tow concepts *Entity_Model* and *Platoon_Model* are used to describe the behavior of entity and platoon in the environment. The concept *Area* model the environment notion. In our transportation problem, there are three types: Urban, Agricultural, and Military. The concepts *Parameters*, *Entity_Parameters* and *System_Parameters* provided a general idea about the parameters of the entities and of the system. These parameters are necessary and useful for Dependability Evaluation in our future work.

TABLE III. THE RELATED CONCEPTS TO PLATOONING SYSTEM MODEL

Concept	Instance of
Platoon	Organization
Structure	OrganizationalStructure
Geo_Configuration	OrganizationalRule
Navigation_Policy	OrganizationalRule
Interaction	Communication
Entity	AgentType
Leader	AgentType
Follower	AgentType
Parameters	OrganizationalRule
Entity_Parameters	--
System_Parameters	--
Model	--
Entity_Model	AgentModel
Platoon_Model	OrganizationModel
Area	Environment

 ISBN: 978-1-61208-319-3

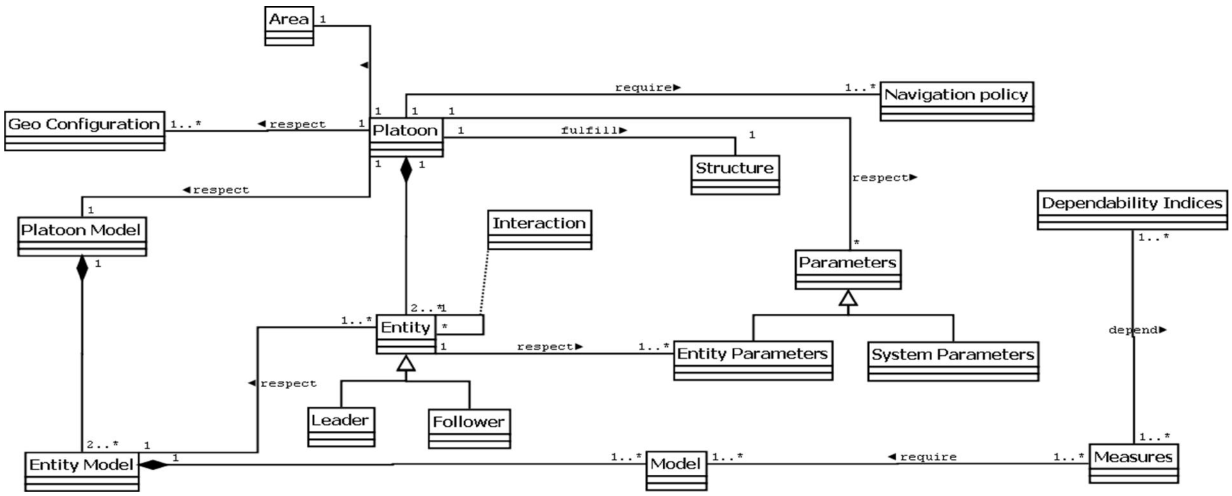

Figure 3. Platooning System model

IV. A CASE STUDY: URBAN PUBLIC TRANSPORTATION SYSTEM

According to the agent-oriented meta-model, we try to specify a transport applications in urban environment. The convoy adopts a *line* configuration (see Fig. 4a) Longitudinal gap (Inter-distance) between vehicle 0 meter and 2 meters in lateral gap. For these scenarios, the convoy will have a fixed number of vehicles between two and three and will move on a track with a radius of curvature ranging from 15 m to infinity. The train moves at a maximum speed of 50 km/h with an acceleration of 1 m/s^2 and a deceleration of -3 m/s^2 on a maximum distance of 1000 meters. From these settings, two scenarios are proposed. The first is to evolve a convoy of vehicles with fixed-line configuration. During the movement, the convoy can change its geometric configuration from Line configuration to Echelon configuration (see Fig. 4).

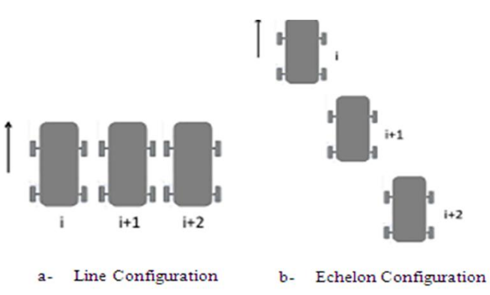

Figure 4. Urban Public Transportation Configuration

In Fig. 5, we present the object diagram which is an instantiation of the Platooning System Model (Fig. 3). The object diagram in the data modeling language UML used to represent instances of classes, that is to say objects. As the class diagram, it expresses the relationship between objects, but also the state of objects, thereby expressing execution contexts. In this sense, this pattern is less general than the class diagram. Object diagrams are used to show the state of object instances before and after the interaction, i.e., it is a photograph at a specific time and attributes existing object. It is used in the exploratory phase.

The object diagram of our study is a set of objects that have the attributes that characterize the system. The object *Convoy_Urban* is an instance of the concept *Platoon* which is it's an instance of *Organization* concept of the metamodel. The convoy adapt a line configuration and Line to Level navigation policy therefore we find an instance of *Geo_Configuration* named *Line_Configuration* and an instance of *Navigation policy* named *Line_to_Level*.

Our transportation system is constitutes of three intelligent vehicles: one Leader and two follower, thus the object diagram contains two items: *V_Leader* instance of the concept *Leader* with cardinality equal to 1 and *V_Follower* instance of *Follower* concept with cardinality equal to 2.

Our system has parameters regrouped in the two tables IV and V. These parameters are divided into two kinds: *Vehicle_Parameters* and *Convoy_Urban_Parameters*, which represent convoy entities parameters and the parameters of the overall system, respectively. They are used for dependability evaluation in our future works. Transportation system behavior is modeled by *Convoy_Urban_Formal_Model* object.

TABLE IV. VEHICLES PARAMETERS

Parameters	**Values**
Max speed	50 km/h
Acceleration/ deceleration	1 m/s^2/-3 m/s^2
Weight	500 kg

TABLE V. CONVOY PARAMETERS

Parameters	**Values**
Vehicle Number	3
Configuration	"Line"
lateral gap	2 meters
longitudinal gap	0

The behavior is described by state-based models which are used in system dependability evaluation. This model and her parameters are used in our future work to the dependability evaluation.

V. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed an Agent-Oriented meta-model adequate to any Transportation Systems with multi-configuration ability problem. The aim is to allow analysts to produce a transportation system model that precisely captures the knowledge and the behavior of an organization so that an agent-oriented requirements specification of the system-to-be. The most in our Agent-oriented meta-model against to others allows analysts to specify the system structure and behaviors because there is no meta-models specified twos. We illustrated our meta-model on urban public transportation system. We have tried to model our system as multi-agent system based on our proposed meta-model.

Future works will be devoted to several key points aimed at improving the proposed solution. On the one hand, we will work to provide a generic model for a methodology and will be suitable for all platonning application with different scenario in different transportation field. This model is used for the Dependability evaluation.

ACKNOWLEDGMENT

Works exposed in this paper are done in collaboration with Systems and Transportation Laboratory (IRTES-SET) with the support of the French ANR (National research agency) through the ANR-VTT Safe platoon project (ANR-10-VPTT-011). University of Technology of Belfort Montbeliard (UTBM), Belfort, France.

REFERENCES

[1] D. Gaevic, D. Djuric, and V. Devedic. Model Driven Engineering and Ontology Development, Springer Berlin Heidelberg, 2009.

[2] J.-M. Contet, F. Gechter, P. Gruer, and A. Koukam, Application of reactive multiagent system to linear vehicle platoon, in: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - Volume 02, ICTAI '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 67-70.

[3] M. El-Zaher, F. Gechter, P. Gruer and M. Hajjar, A new linear platoon model based on reactive multi-agent systems, in: Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 898-899.

[4] L. Cernuzzi, T. Juan, L. Sterling and F. Zambonelli, The gaia methodology: Basic concepts and extensions, in: Methodologies and Software Engineering for Agent Systems, Kluwer Academic Press, 2004

[5] M. Cossentino, S. Gaglio, L. Sabatucci ,and V. Seidita, The passi and agile passi mas meta-models compared with a unifying proposal, in: Proceedings of the 4th international Central and Eastern European conference on Multi-Agent Systems and Applications, CEEMAS'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 183-192.

[6] L. Cernuzzi and F. Zambonelli, Experiencing auml in the gaia methodology, in: ICEIS (3), 2004, pp. 283-288.

[7] F. Zambonelli, N. R. Jennings ,and M. Wooldridge, Developing multiagent systems: The gaia methodology, ACM Transactions on Software Engineering and Methodology, Volume 12 Issue 3, 2003, pp 317 – 370,

[8] D. Cabrera-Paniagua and C. Cubillos, Agent-oriented software engineering ix, Springer-Verlag, Berlin, Heidelberg, Ch. PASSI Methodology in the Design of Software Framework: A Study Case of the Passenger Transportation Enterprise, 2009, pp. 213-227.

[9] M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam, ASPECS: an agentoriented software process for engineering complex systems, Autonomous Agents and Multi-Agent Systems 20 (2), 2010, pp 260-304.

[10] N. Gaud, V. Hilaire, S. Galland, A. Koukam, and M. Cossentin, A Verification by Abstraction Framework for Organizational Multiagent Systems, In Proc. of the Sixth International Workshop AT2AI-6: "From Agent Theory to Agent Implementation", of the Seventh International Conference on Autonomous agents and Multiagent Systems (AAMAS), Bernhard JUNG, Fabien MICHEL, Alessandro RICCI, Paolo PETTA (eds.), 2008, pp. 67-73.

[11] J.M. Atlee and J.D. Gannon, State-Based Model Checking of Event-Driven System Requirements, IEEE Trans. Software Eng. 19(1):24-40, 1993.

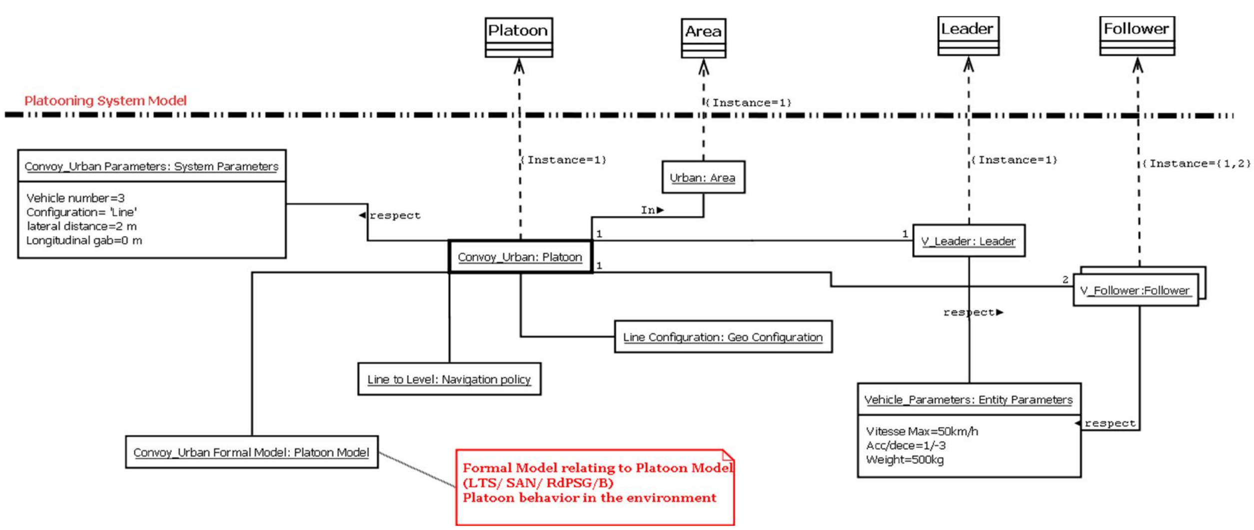

Figure 5. Object diagram relating to urban public transportation systems