

A Heuristic for Problem Formalization in Agent Based Simulation Studies

Massimo Cossentino[‡], Carmelo Lodato[‡], Patrizia Ribino[‡] and Valeria Seidita^{§‡}

[‡]Istituto di Reti e Calcolo ad Alte Prestazioni

Consiglio Nazionale delle Ricerche

Palermo, Italy

Email: {cossentino,c.lodato,ribino}@pa.icar.cnr.it

[§]Dip. di Ingegneria Chimica Gestionale Informatica Meccanica, University of Palermo

Palermo, Italy

Email: {valeria.seidita}@unipa.it

Abstract—Agent Based Modeling and Simulation (ABMS) is considered an effective approach for conducting simulation studies in many fields. In order to develop high quality simulation models, methodological approaches are demanded. In such direction we are moving by proposing a heuristic for the formalization of agent based simulation problems. The proposed heuristic is based on some guidelines developed for identifying the main elements of the problem domain description by analysing verbs and their common taxonomy in grammar.

I. INTRODUCTION

Simulations are used in several contexts for evaluating the behaviour of complex systems and for understanding how the numerous variables constraint the performance of such systems. Especially, a simulation model gives the opportunity to make experiments and to identify errors in such a way that would often be infeasible in the real world. A simulation study is a process that allows to define a simulation model of a real system and to make experiments with that model.

Among simulation models, Agent Based Modelling and Simulation (ABMS) is becoming a widely used approach for conducting simulation studies in many disciplines (such as social sciences, economy, traffic and transportation) that deal with complex systems characterized by the presence of autonomous and active entities [3, 9, 11, 13, 20, 24, 25].

Such increasing interest is mainly due to the possibility to include in such models some aspects that are commonly not considered by traditional approaches, such as local interactions, entity organizations, impact of environmental changes on entity behaviours, entity heterogeneity and so on.

Whether on the one hand these features make the ABMS paradigm very useful for the representation of highly quality models, on the other hand it requires appropriate methodological approaches for addressing such a kind of simulation study. At present, there are some proposals in such direction but several issues are still open and a comprehensive methodological approach is still lacking.

The work we present in this paper is part of a more ambitious objective: the development of a complete methodological approach (we named MAMAS, *Methodological Approach for Multi-Agent Simulations*) that covers the entire life cycle of a simulation study taking into account each facet of the agent

based modelling paradigm. In so doing, the experience we collected in the latest years about methodological approaches in the field of agent oriented software engineering (hereafter AOSE) consolidated our opinion that a solid foundation for design processes lies on the use of system metamodels. This is also recognized in the field of agent based simulations where some proposals about system metamodels [14, 16, 22, 23] for creating a common ground for several agent based simulation domains are proposed. For these reasons, we are defining MAMAS approach starting from the metamodel we proposed in [22]. Specifically we are specializing the set of activities that are commonly defined for generic simulation studies in the ABMS context by using the metamodel as activity's benchmark. Thus we are defining specific guidelines and work products for these activities wherever are not defined yet.

The importance of the problem formalization in order to produce complete and well defined simulation models is well recognized in literature [5, 17]. In our work, problem formalization assumes a very important role for the identification of simulation goals and all the multi-agent system goals influencing the simulation goal itself. The concept of goal, in our approach, spreads over the whole MAMAS process. For this reason we spent a great effort in studying a formalized way for identifying and modelling goals. We also studied how they influence the whole simulation study, starting from the very early stages of simulation analysis i.e. the *problem formalization*.

In such a landscape, the aim of this paper is to provide guidelines for performing a core activity of MAMAS process: *Problem Formalization*; the proposed contribution lies on the heuristic developed for identifying the main elements of the problem domain by using verbs and their common taxonomy in grammar.

The rest of the paper is organized as follows: in section II we detail the motivations for our work against the related works in literature, in section III we present a brief overview on the MAMAS process; in section IV we illustrate the proposed heuristic and finally some discussions and conclusions are drawn in section V.

II. MOTIVATION AND RELATED WORK

The advantages of simulation models for studying the behaviour of complex and dynamic systems are widely known. In particular multi agent models are among the most interesting ones due to some useful properties showed by this type of modelling [11]. Moreover, multi agent simulations is considered one of the killer applications of agent oriented technology [15]. Nevertheless, as far as we know, to date there is no comprehensive methodological approach that can provide guidelines for the development of such models. Indeed, traditional approaches are conceived for generic simulation studies and they do not adapt well to agent based simulations because they overlook some essential aspects that are fundamental in an ABMS approach, such as for example the role of the environment or agent interactions.

Only to cite a few, Balci in [1] suggests a process based on 10 phases and 13 credibility assessment stages. The credibility assessment starts from the first phase of problem formulation until the last phase of the presentation of simulation results. In [2] instead Balci states the importance of the VV&T (Validation Verification and Testing) techniques throughout the life cycle of a simulation study showing how they are categorized. Although these two works established a baseline for the development of credible simulation studies they are conceived for generic simulation approaches. We think that, from the perspective of agent-based simulations, the life cycle of a simulation study has to take into account some typical features of agent oriented technology. For example, it is important to focus on the environment model that plays a fundamental role in such kind of simulations where it commonly represents the real context in which the real actors work.

In [5] John S. Carson defines the human actors that have to compose the simulation team relating them to the common steps of a simulation study and providing some useful hints in order to conduct the activities of the process. Just to give an example, Carson states that during *Problem Formulation* the team should develop a list of specific questions the model should address and he also states that it has to focus on model boundary and scope, level of detail and project scope. While during the *Model Development* phase he suggests two main activities have to be performed: (i) development of data structures to represent the data needed by the model and (ii) translation of the modelling assumptions in a specific document written in the language or representation required by the adopted simulation package. Hence, although the guidelines of *Problem Formulation* are surely helpful also in the context of agent based simulations, the second one is not applicable due to the different technological simulation paradigm.

In [17] Averill M. Law proposes a seven-step approach for conducting a simulation study. In his work, Law highlights the importance of the validity of a simulation model. Thus, he presents practical techniques and guidelines for developing valid and credible models. In particular, he declares that the keystone for building valid and credible models lies on (i) formulating the problem precisely; (ii) interacting with the decision-maker on a regular basis throughout the simulation project to ensure that the correct problem is being solved and to promote model credibility; (iii) using quantitative techniques to validate components of the model; (iv) performing sensitivity analysis to determine important model factors; (v) comparing

model and system results for an existing system (if any); (vi) using a Turing test to compare model and system output data; (vii) reviewing of model results and animations to see if they appear to be reasonable. As well as the process proposed by Balci in [1], the process proposed by Law covers the whole life cycle of a simulation study providing right-minded guidelines for each activity. But as the previous ones, such guidelines are not effective enough for agent based simulation studies because too general.

At the same way, as far as we know, the methodological approaches currently proposed for agent oriented modelling give a rundown of the overall life cycle of a simulation study and only in some cases they focus on single steps, such as for example model definition and validation.

In [7] the authors present a model-based methodological framework for designing multi-agent simulations. The aim of the authors in this paper is the introduction of a consistent use of the agents during the entire life cycle of a simulation study. In particular they individuate three roles involved in a MABS (Multi-Agent Based Simulation) the thematicians that are practically the domain experts, the modellers that have the responsibility to design the model and finally the computer scientist that have to implement the model. Each role deals with abstractions of the agent at different levels (i.e: real agents, conceptual agents and computational agents) thus defining the specific kind of model related to the level he belongs to (i.e: domain model, design model and operational model). At that time, in their operational model the computational agents do not own the peculiar features of agents such as proactivity or autonomy due to the lack of development language supporting them. At the time was common the use either object-oriented or procedural or functional languages to implement the specifications described in the conceptual agents. These deficiencies are nowadays overcome by the several agent-oriented development paradigm. Hence according to Drogoul *et.al*, we believe that a methodological approach for simulation study has to be based on model transformations from different levels of abstractions but we also want to fill the gap between the conceptual model and the operational one.

In [12] the authors proposed a standard protocol for conceiving agent-based models, named ODD (Overview, Design concepts, Details). Such protocol provides guidelines in order to structure the information needful for agent-based models in an established sequence following a top down approach. It is composed of seven steps grouped in three main blocks: Overview, Design concepts and Details. The core of the protocol is the description of the design concepts following a detailed check-list of questions that allow to examine particular aspects of the system to be simulated such as emergence, adaptation, prediction etc...

The easyABMS methodology [8] is based on an iterative process that covers the common activities of a simulation study and produces several models according to the specific phase of the simulation study. Such models are produced according to a reference meta-model that is characterized by elements that are quite common in the AOSE context. It lacks of explicit representation of space, entity features and organizational structure.

In [6] Cioffi introduces a general methodology for social

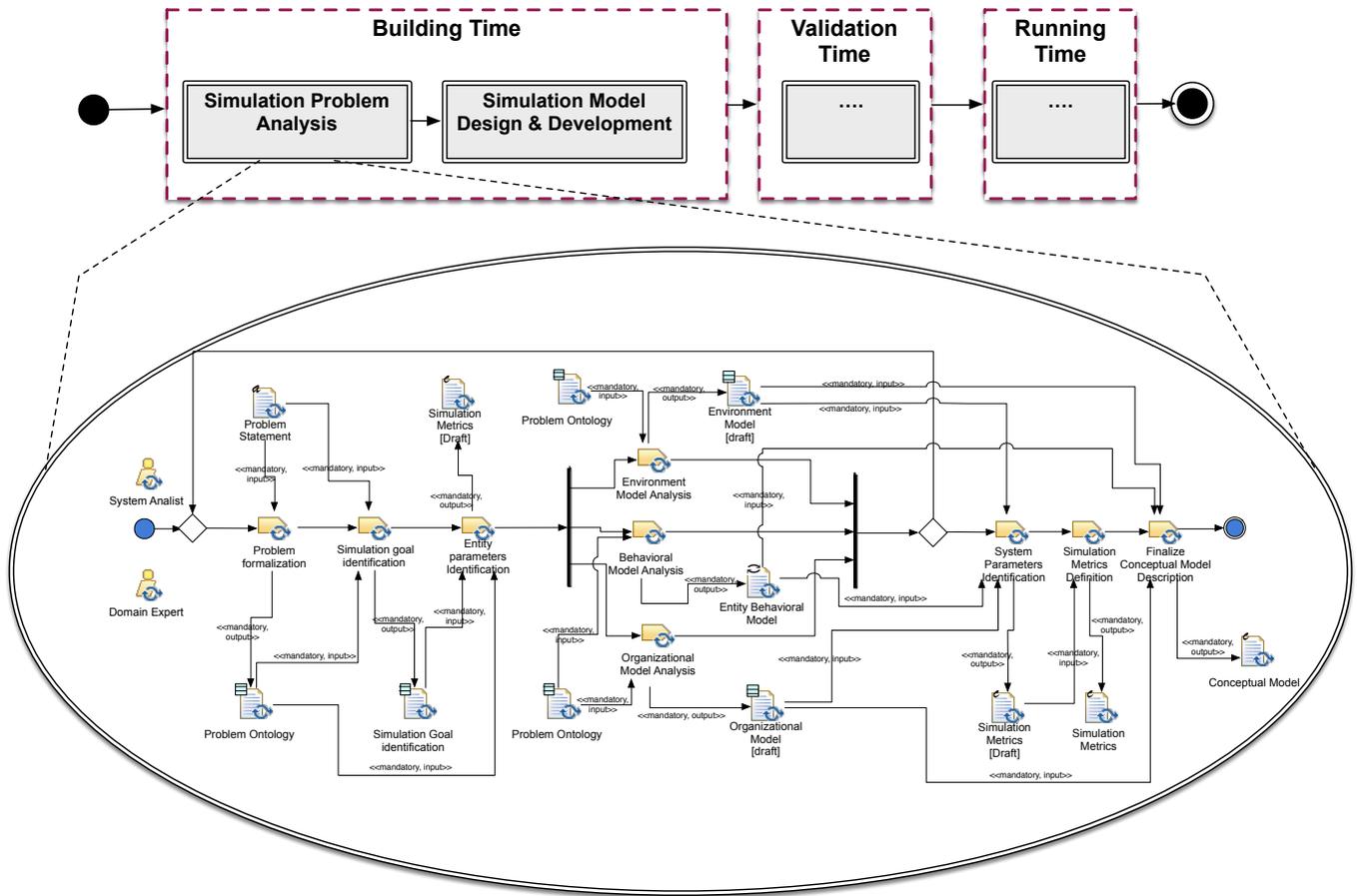


Fig. 1. A sketch of MAMAS process with zooming on Simulation Problem Analysis Phase

simulations. It is an iterative process that begins with a referent system in the real world (he names *explanandum*). Then techniques of abstraction, formalization, programming are used to develop a simulation model (he names *explanans*). As well as techniques for Verification and Validation are reported such as Code Walk-Through, Parameter Sweeps, Histograms and so on. Social simulations find a natural representation by means of agent oriented techniques. As Cioffi states, agent based models have a more sophisticated landscape and actors that come closer to emulating humans through various aspects of reasoning, decision-making, and behaviours. But he only focuses on three main components of the agent based model (i.e: agents, rules, and environments) thus adapting the methodology only for this aspects.

Agent-Based Social Simulations are also the core of MAIA [10]. MAIA is a framework based on a meta-model that supports conceptualization of a agent-based simulations. It provides some guidelines in order to adequately capture, analyse, and understand the domain of application. It helps the modellers to explicitly report the motivations behind modelling choices. Such framework is focused on social and institutional structures and its meta-model aims to describe those systems where the key components are individuals and institutions. We think that some other model components have to be considered when we address agent based simulation studies. In [22] we conducted a study that highlighted the presence

of an organizational aspect in many practical agent based simulations. According to the authors of [6] and [10] we claim that a methodological approach should certainly consider agents, rules, norms and environments but also other aspects such as the organizational one.

In [19] the focus is on the use of simulation in biological systems. In such a context the authors propose a revision of the Agent and Artifact (A&A) metamodel for adaptation to the Systems Biology, they made experiments using the case study of glycolysis and shown how to model and simulate the metabolic pathway.

In [18] the authors propose a method to integrate simulation-based approach within AOSE methodologies as a new fragment of the methodology. The main idea behind the use of simulation in the software engineering process is that the simulation can be used to predict the dynamics of the system to be developed and also the run-time properties that can be induced by design, before the system is completely developed.

From the literature it arises that several issues have to be still addressed for the definition of an appropriate methodological approach for agent based simulation studies. This work goes in such direction by addressing a crucial aspect: the *Problem Formalization*.

III. AN OVERVIEW OF MAMAS PROCESS

In order to understand the aim of this paper we need to briefly introduce the context in which this contribution is plugged in. As we previously said, we are working on the development of a complete methodological approach for addressing agent based simulation studies (hereafter MAMAS) taking into account specific facets of agent based simulation paradigm.

From a literature review (see Section II), it arises that the main stages of a generic simulation study are: (i) the simulation model development, (ii) its validation and (iii) the execution of experiments on such model with the related analysis. We have defined in the MAMAS process three macro phases according to the stage they refer, we named them respectively *Building Time*, *Verification&Validation Time* and *Running Time* (see Fig. 1).

For the scope of this paper, let us focus on the *Building Time* phase. The final aim of this phase is to produce the agent based simulation model. This usually requires the transition from real-world system to a simulated one throughout four main artefacts:

- the *Problem Domain Description*, also called *Problem Statement*, containing an informal description both of the real-world problem to be addressed by the simulation study and of the simulation objectives;
- the *Domain Model*, also called *Conceptual Model*, that is the model of the system to be simulated emerging from the study of the *Problem Domain*. This model contains a more formalized form of knowledge about the *Problem Domain*. In particular, it may contain knowledge about real-world agents, behaviours, rules and so on;
- the *Design Model* is the model derived from the Domain Model by describing the details of the architecture of the agent based system that have to be realized for simulating the real system. A Design Model contains knowledge about design agents, interaction, behaviours that is a formal refinement of the previous one.
- the *Computational Model* is the implementation of the Design Model on a specific agent platform. It handles the computational agents that are implementation on a specific agent platform.

For supporting this transition in a systematic way, several activities are necessary. In MAMAS process, we grouped them in two phases: *Simulation Problem Analysis* and *Simulation Model Design & Development* respectively. The former supports the transition between the *Problem Domain Description* to the *Domain Model*, while the second produces the *Design Model* and the *Computational Model*. These last phase could be performed by using classical AOSE approach.

Now, let us focus on the *Simulation Problem Analysis* phase (see the zooming of Fig.1). According to the ABMS literature, we recognize the importance of the problem formalization as a mean for producing well defined simulation models. We have already experienced (see [21]) in the AOSE context how to perform the problem formalization through the use

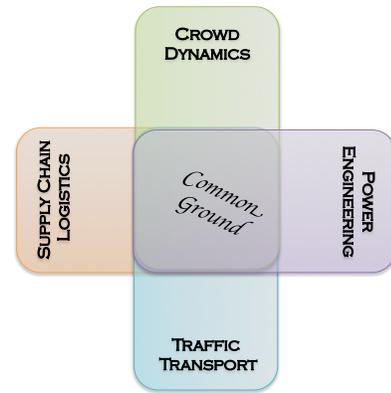


Fig. 2. Several simulation problem domains share a common ground that aims to provide a starting point for the development of an agent based simulation model.

of ontological approaches in a way useful for identifying goals of multi agent systems that are the leading elements for the development of design models. Such goal identification approach is based on the definition of a *Problem Ontology* starting from the *Problem Statement* of the multi agent system to be developed.

In the same way, we think that in the ABMS context simulation goals along with agent goals are leading elements for the development of the conceptual model. Thus we are adapting the approach proposed in [21] for the identification of simulation goals and all the agent goals that influence the simulation goals itself. It is worth noting that in this context the concept of goal assumes different meanings. Commonly in generic simulation studies the term *goal* is used for referring to the simulation objectives, namely what are the real issues the simulation has to address. In such a sense, simulation goals are linked to the interest that the simulation team sees in the results of the simulation study. Conversely, agent goals refer to the states of the world an agent wants to achieve. Thus, they are linked to the interest the single entities have in the system they live in. This system is the means the simulation approach uses to achieve the intended simulation goals.

Thus, in order to adapt the approach we proposed in [21] in the context of ABMS, we need to define appropriate guidelines for building the *Problem Ontology* starting from the *Problem Domain Description* of a simulation study. Such guidelines are founded on the metamodel for agent based simulation problems proposed in [22] that will be presented in the following sections.

A detailed description of all the activities shown in Fig.1 is out of the scope of this paper, however it is important to note the role of the problem Ontology as input of several activities thus making necessary the problem formalization we propose.

IV. PROBLEM FORMALIZATION

In [22] we already addressed the definition of a metamodel for agent based simulation problems. In that work we adopted a systematic approach to review the existing literature about agent-based simulation studies in order to identify what elements are commonly used for describing simulation problems and what elements are used in specific application domains. In

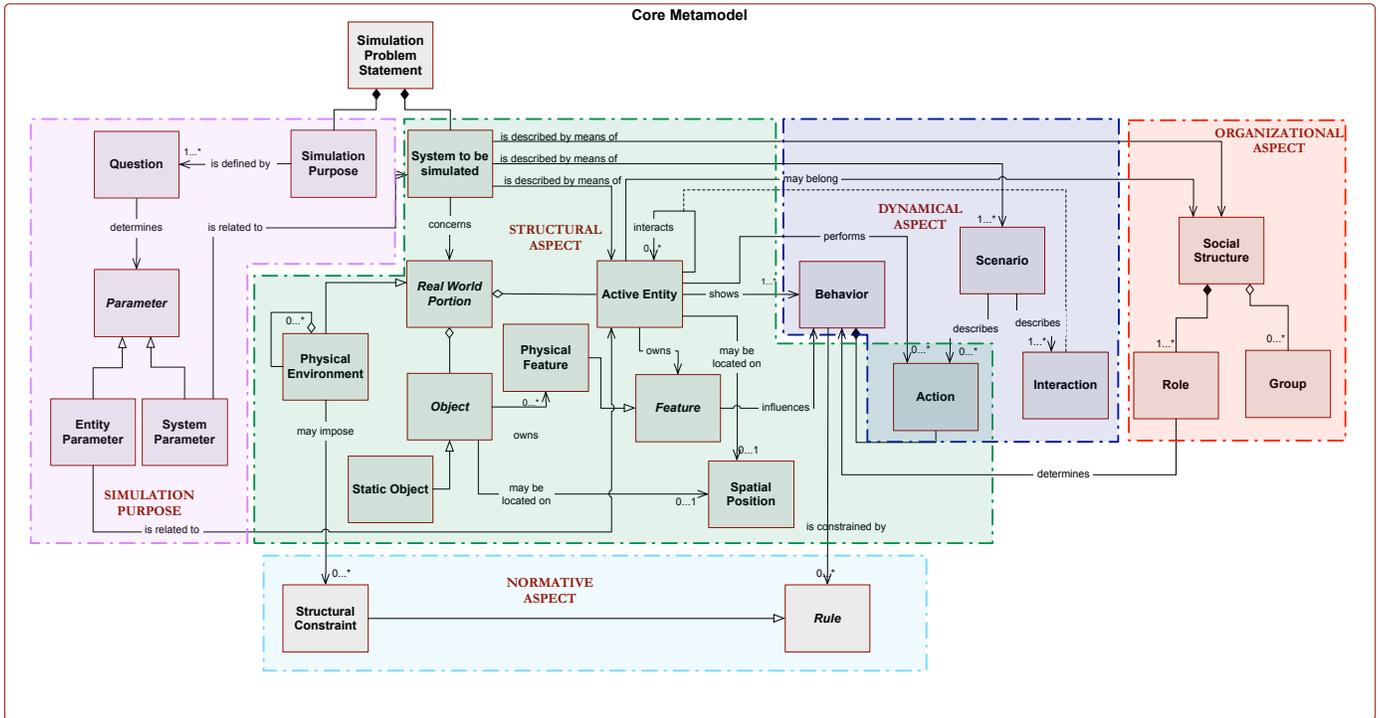


Fig. 3. The Core metamodel for agent based simulation problem.

so doing, we identified and analysed some problem domains where agent based simulations are frequently employed, thus determining a common ground formalized in a *Core Meta-model* that contains all the elements shared among the domains under analysis. Fig.2 illustrates the domains we covered in that review and the result is the metamodel shown in Fig.3. Such a metamodel shows that real-world systems commonly simulated by means of multi agent systems share some features we grouped in five key categories:

- the *Simulation Purpose* grouping elements that are related to the questions defining the goal of the simulation;
- the *Structural Aspect* namely the configuration of the real world influencing the simulation problem;
- the *Dynamical Aspect* represented by actions, interactions performed by someone/something for producing something or for achieving a particular end;
- the *Organizational Aspect* represented by the element that defines the social structure by means of group and roles;
- the *Normative Aspect* given by regulations that commonly constrains the dynamic or physical aspects of the system such as structural constraints or social norms.

Such metamodel underpins the *Simulation Problem Analysis* phase and each activity is devoted to manage some portion of it (see zooming of Fig.1). In particular, in this paper we detail the *Problem Formalization* activity that is devoted to instantiate the elements belonging to the structural aspect of the *Problem Ontology*. In the following we introduce the

guidelines for performing this activity by means of a case study.

A. Case Study

During the latest years we carried out several experiments in the field of goods management within logistic districts, studying how to store them in metropolitan distribution centres and the way to efficiently deliver them throughout the city. By working on this problem we had the possibility to conduct several experiments on agent based methodological approaches for simulation studies. The focus of our studies was on one single node of the supply chain; the node was responsible for managing containers automatically unloaded by means of some AGVs (Automatic Guided Vehicle) provided with forklift for pallet handling; that is representative for a large number of real logistic warehouse-related problems.

Throughout the following subsections we use this case study in order to illustrate the proposed approach to problem ontology definition; an excerpt of the Problem Statement document is reported below.

a) *Problem Statement*: A logistic district is a large area composed of several warehouses where some freight forwarders may deliver their container. Inside a logistic district, several articulated lorries, coming from extra urban areas, arrive all the time and are arranged in warehouse bays. Each articulated lorry transports one 40-foot standard ISO (*International Organization for Standardization*) type container. Whenever a lorry reaches the warehouse, it docks into a bay for unloading its cargo. Commonly, a container holds several kinds of goods contained in boxes and grouped in pallets. These latter are *EUR pallets* (a standard ISO pallet

measures 800x1200 mm). Each pallet must be unloaded from the container and carried into a well defined warehouse area dedicated to the sorting of goods, this operation is carried by the AGVs. In this area, each pallet is opened and its contents (packages of goods) placed on different sorters according to their destinations. These operations averagely last 5 minutes per pallet.

Commonly, a real logistic warehouse is basically composed of the following elements: gates, recharging areas (for AGVs), sorting areas, sorter places, buffer areas, paths and waypoints. The *Gate* corresponds to the warehouse bay and it is holds an unloading platform; lorries park here waiting for unloading their cargo (pallets). The *recharging area* is a specific area provided with sockets for recharging AGV's batteries when necessary, in particular the AGV moves towards the nearest free recharging area when its battery has depleted to a specific level. The *Sorting Area* is the place in which goods, contained in the pallets, are processed by the Sorter, an automatic device able to collect incoming goods and to forward toward a new destination. AGVs deliver goods to the Sorter using several input points (called *Sorter Places*). Goods are placed on hold when the Sorter is busy (all Sorter Places occupied by AGVs delivering some good) in the *Buffer Area*, this area is also used for temporarily parking AGVs that are waiting for a new mission. AGVs move following some *Paths* realized by means of optical guides connecting particular floor markers (*Waypoints*). AGVs have a value for their speed, their turning radius, loading capacity and type of guidance and lie in a particular area called parking area when unused.

A Waypoint has a unique ID and may be recognized by an AGV during its movement; a Waypoint is positioned outside each area of interest (Sorting Areas, entrance of unloading platforms and so on) or two meters before each cross between optical guides thus pointing out a stopping position for each AGV willing to pass a cross.

Paths are divided in path sections that are delimited by waypoints; there may be two kind of waypoints: waypoints positioned at the end of a path and the one positioned in the mid of a path, in the first case the waypoint is adjacent to one other waypoint only in the second case to two ones.

An automatic warehouse may be configured with several physical layouts (different disposition of optical lines, different number of entrances or exits, etc.) and equipped with some machinery. Each layout may impose limits to the use of different resources, the number of AGVs working simultaneously may be bounded by the set of available optical paths and their lengths; at the same time, specific equipment (i.e. the number and the performance of the AGVs, sorter capacity, etc..) and the choice of specific business strategies, such as path reservation strategy or AGVs scheduling, may constraint the performance of the warehouse thus greatly limiting its ability to quickly unloading arrived trucks.

Smaller vehicles (eco-friendly trucks) move packages out of the warehouse to their new destination (usually in town). The transport of pallets toward the sorting area is committed to AGVs with optical guidance; AGVs move, in order to load or unload pallets, by following the guidelines painted on the floor and engage a path between two adjacent waypoints, only one AGV at a time may engage such a path.

Given a warehouse made as described above the main aim of this simulation study is: how may we improve the throughput (the number of pallets unloaded per hour) of the warehouse?

B. Guidelines for Problem Formalization

The aim of the Problem Formalization activity is to identify and instantiate all the metamodel elements composing the *Structural Aspect* of the domain under study hence all the real world concepts, or part of it, influencing the simulation problem. The result is a structural diagram, an UML class diagram, that we call Problem Ontology. This activity is deeply grounded on the analysis of the problem statement, hence on the analysis of portions of text from which the analyst has to identify the following elements: *active entity*, *object*, *feature*, *physical feature*, *spatial position* and *action*.

Before going on in the activity description, let us illustrate a verbs classification (shown in Table I) that proved useful in analysing the text for creating the Problem Ontology diagram.

Dynamic Verbs (or Action Verbs)	Dynamic verbs normally describe action that can be done or something that happens. They also describe action or events that may have a beginning and an end.	Transformational	They indicate events <i>without</i> a temporal extension and entail a <i>state change</i> .	wake up, leave, die...
		Continuous	They indicate events <i>with</i> a temporal extension and they <i>have not</i> an implicit <i>goal</i> to reach.	sleep, work, help, improve...
		Resulting	They indicate events <i>with</i> a temporal extension and they <i>finish</i> when a specific <i>goal</i> is reached.	learn, fall, build, complete...
Stative Verbs	Stative verbs refer to not changing or static condition.	Perception	They indicate perceptive processes	see, like, feel, listen, sound,...
		Cognition	They indicate cognitive process such as thinking, imaging, remembering and so on	think, suppose, remember,...
		Relation	They indicate a relationships among objects or person	have, consist of...

TABLE I. VERBS TAXONOMY

In grammar, verbs may be grouped in two main categories: dynamic verbs and stative verbs. A dynamic verb is a verb having a duration and referring to a continuous and progressive action of the subject, this is the opposite of stative verbs. Dynamic verbs are moreover distinguished in verbs showing a changing in the state of the element they refer to and/or verbs that make clear reference to a goal, an objective, to reach (for instance, work, help, sleep ...). In contrast, stative verbs refer to situations in which there is no obvious action (for instance, think, suppose, listen ...) or that relate different elements in a sentence (for instance, have, be, consist of, composed of ...).

We use this classification in order to identify in a text the elements of interest for our problem domain. As already said, the main focus when analysing a problem statement during a

simulation study is on all those elements having an influence on the simulation goals. This verb classification allows us to retrieve from the text elements who performs an action when that action is intentionally performed by some active entity in order to cause a state change in the domain. It is also useful for discriminating among all the entities that actively produce a state change and all the ones that may be considered objects, resources or everything else is also used to reach a specific goal. Moreover, the focus on verbs allows us to identify the right relations among entities in the domain and above all the qualities and parameters belonging to entities that have to be taken into account when identifying all the goals of the simulation study and the metrics to be used for measuring their fulfillment. Although the latest part of the previous statement is not in the scope of this paper, it is worth underlining that the construction of a complete and well built problem ontology is the foundation, in our approach, for representing the domain as well as for identifying the system goals that purposefully influence the simulation goal; we already reported our work on the retrieval of goals from ontology in [21].

The heuristic we propose for identifying the metamodel elements from the problem statement prescribes to analyse the problem statement text in order to retrieve nouns, adjectives, verbs and relationships among them. This approach may resemble the well known Abbot's technique for identifying objects and their attributes/operations/relationships from use cases description (see [4]). Moreover, particular kinds relationship, such as aggregation or composition, may be identified also through the genitive or possessive case.

The following subsections report the guidelines for implementing the heuristic, they start from the identification of *active entities*, *objects* and *actions* that are related and identifiable through subjects, verbs and direct objects in sentences. *Active entities*, *objects* and *actions* are drawn by means of classes.¹

It is worth noting the proposed guidelines help in identifying nouns, verbs, etc. that are candidate to become active entities, objects, actions and so on. The final decision whether they are really to be inserted in the Problem Ontology with that specific meaning is always left to the designer. The guidelines are not intended to define an algorithm for the automatic (i.e. non-supervised) production of a Problem Ontology diagram.

Fig. 8 reports a part of the Problem Ontology for the analysed problem statement. In the following subsections such ontology will be built step-by-step by applying the proposed guidelines.

1) Preliminary Step-Nouns Identification: A preliminary step is required: it prescribes to analyse the problem statement and to prepare a list of all nouns present in the text. Obviously, not all the nouns found during this work are relevant for the problem, we are constructing a model hence a view on a specific area of interest. The problem domain we are analysing as a case study falls in the area of the simulation of processes. It takes place in a warehouse, and it aims to improve its throughput. Hence, we may say, for instance, that *logistic district* is not a noun of interest because we are focusing

¹It is worth noting that the guidelines for performing Problem Formalization may be associated to every kind of notation, we choose UML class diagram because its extended usage makes it easier to understand the composition of the Problem Ontology diagram.

on the warehouse that is a part of logistic district. The same is for *urban area* because, in this case, it clearly does not influence the real process in the warehouse and consequently our simulation study. It is also frequent that synonyms are used in the text. This is to be avoided when possible and existing synonyms are to be reported only once with the most immediately recognizable name. For instance, in the case study problem description text, it is possible to note that bay is used as a synonym of gate. The result of this activity in the reported case study produces the following list of nouns: lorry, pallet, AGV, path, gate, sorting area.

2) Active Entity Identification: The identification of active entities is performed in the following way:

For each noun in the previous list

- analyze sentences in which it is the subject of a verb
- if there exists at least one sentence in which it is the subject of a dynamic verb (or it is in some kind of relationship with a dynamic verb)
 - analyze the verb
 - if it is a *resulting* or a *transformational* verb (Table I) the considered noun is an active entity

Some active entities may be in composition/aggregation, specialization or association relationship with other active entities, this kind of relationships are identified by means of stative verbs, relation kind.

In the presented case study there are not evidences of stative verbs such as cognition and perception verbs, however, generally speaking, they both may be verbs whose subject is an active entity.

The result of this activity in the reported case study brings to the identification of sentences like:
 “AGVs *move*, in order to *load* or *unload* pallets, by following guidelines (hence paths) painted on the floor”
 “*Lorries park* here (ref. gates)”
 “AGV *moves* towards the nearest recharging area”

In these sentences two active entities may be identified: AGV and Lorry (the use of the singular form of the nouns is advisable). A first portion of Problem Ontology Diagram may now be built by reporting in it the two identified active entities (see Fig. 4).

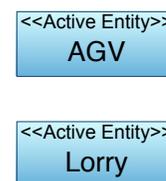


Fig. 4. Problem Ontology-Step 2 (active entities)

3) *Actions Identification*: The identification of actions is performed in the following way:

For each previously identified active entity, associate the corresponding action in the sentence and draw them in the diagram.

This step may be performed in an iterative fashion with the previous one (active entities identification).

The result of this activity in the reported case study brings to the identification of the following actions: move, load, unload, follow, park.

Such elements are reported in the Problem Ontology diagram thus obtaining its new release as reported in Fig. 5.

The notation we adopt for this diagram prescribes the direction of the relationships went from the active entity (the actor performing an action) towards the performed action.

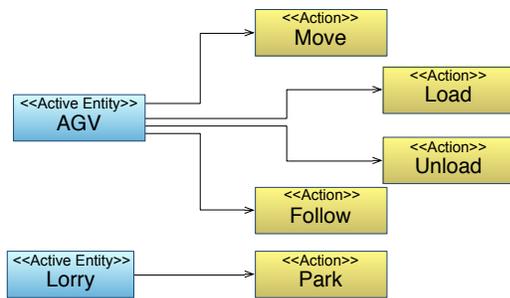


Fig. 5. Problem Ontology-Step 3 (actions)

4) *Objects Identification*: The identification of objects is performed in the following way:

All nouns that are direct objects in sentences where an active entity is a subject are *objects*. All the other nouns that are not active entities are candidate objects. Candidate objects that are subject of stative-relation or dynamic-continuous verbs are *objects*.

In order to definitively identify all the *objects*, the designer has to examine the structure of sentences and to analyse if a candidate object is related to another one or if it is related to adjectives, anyway she has to look through which kind of verbs they are related to. This is complemented by the following step. Moreover, as for active entities, objects may be related by association, composition/aggregation and generalization relationships.

The result of this activity in the reported case study brings to the identification of sentences like:
 “AGVs move, in order to load or unload *pallets*, by following guidelines (hence *paths*) painted on the floor”
 “Lorries park here (ref. *gates*)”
 “AGV moves towards the nearest *recharging area*”
 “Paths are divided in path sections that are delimited by waypoints; there may be two kind of waypoints...”

In these sentences the following objects may be identified: *Recharging Area, Pallet, Path, PathSection, Waypoint, MidWayPoint, FinalWayPoint, Gate.*

Some of them are related by composition/generalization relationships as described in the new release of the Problem Ontology diagram reported in Fig. 6.

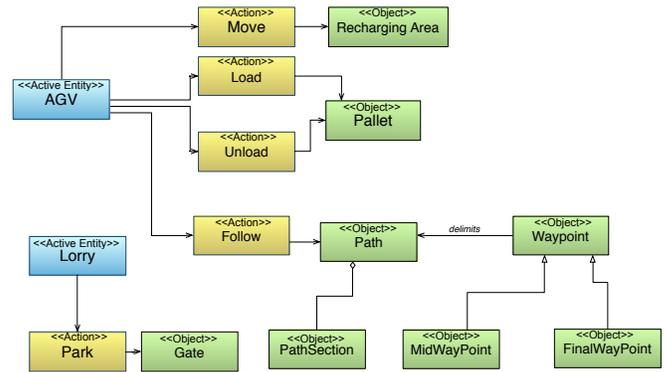


Fig. 6. Problem Ontology-Step 4 (objects)

5) *Feature Identification*: The identification of features is performed in the following way:

For each candidate object analyse if it is related to other candidate objects through relation or continuous verbs. We may find two cases:

- noun + relation or continuous verb + adjectives, the noun is an *object* and the relationship with the adjective is described with a predicate
- noun + relation or continuous verb + noun, the first noun is an *object*, the second one may be represented through another *object*, related with UML relationships, or through attributes.

The choice between representing relations as attributes rather than other objects depends on what the analyst wants to underline and on the specific problem domain. In the presented case study, for instance, speed is represented as an attribute of AGV and not as an object because we found not necessary to represent speed as an object while simulating processes in warehouses. The contrary happens for Sockets that in the real world own a specified position and physical attributes that, in the designer’s understanding, may influence the simulation goal (warehouse throughput optimization).

The result of this activity in the reported case study brings to the analysis of sentences like:

“in the first case the waypoint is adjacent to one other waypoint only in the second case to two ones.”
 “AGVs have a value for their speed, their turning radius, loading capacity and type of guidance.”
 “The recharging area is provided with sockets for recharging AGV’s batteries when necessary”

The first sentence brings to the identification of the IsAdjacent predicate between two waypoints. The second one lists some attributes of AGVs (speed, turning radius,...), the last one identifies two new objects (Socket and Battery) and the relationship between Socket and Battery.

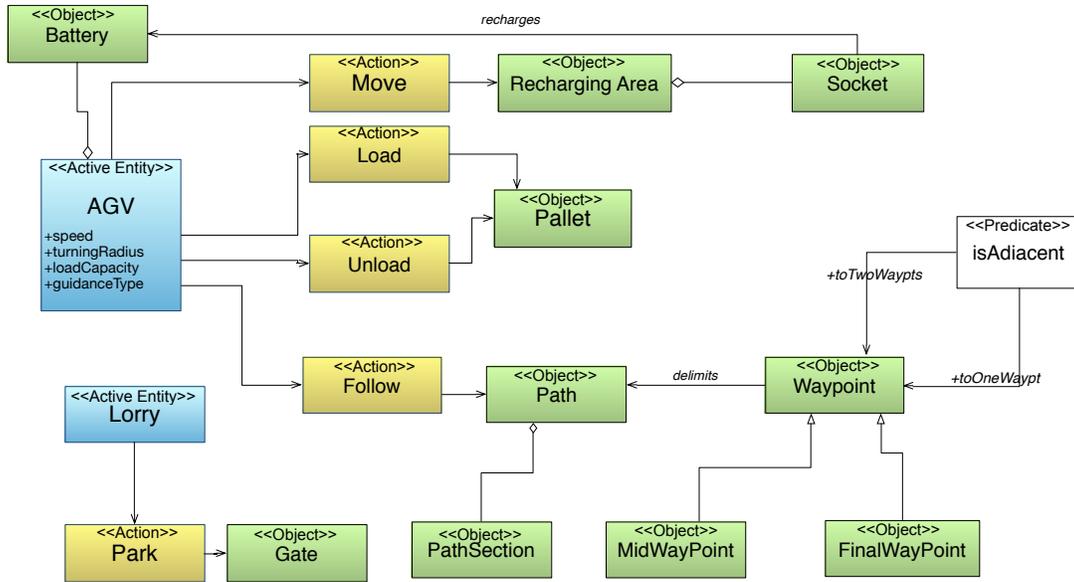


Fig. 7. Problem Ontology-Step 5 (Features)

Such elements are reported in the Problem Ontology diagram thus obtaining its new release as reported in Fig. 7.

6) *Spatial Position Identification*: The identification of features is performed in the following way:

Guidelines in this case are the same of *Feature*, the designer has to look at continuous or relation verbs denoting positioning of objects in the space and choose if they may be treated as objects, attribute or predicates.

The result of this activity in the reported case study brings to the analysis of sentences like: “a Waypoint is positioned outside each area of interest” “AGVs lie in a particular area called parking area when unused”

The first sentence allows to identify two relations, the first between the concepts Waypoint and InterestPoint whereas the second between the concepts AGV and ParkingArea; see Fig. 8 where, as already anticipated, the complete resulting Problem Ontology is reported.

V. CONCLUSIONS

Simulation studies conducted by using agent based modelling and simulation approaches had reported in the latest period good results, however they still lack a rigorous methodological approach for going, step by step, from simulation problem analysis to the implementation of the agent based simulation system. We may say that in the case of simulation studies the development of agent based systems is a part that resides within the overall simulation study and its requirements are greatly affected from the simulation requirements themselves. In this context an important role is covered by the simulation problem analysis phase that aims at identifying and

describing the domain under study along with the simulation goals.

We claim that simulation goals descending from the problem domain, are greatly related to the goals of the agent based systems, although they cover a different scope with respect to agent goals. Indeed, simulation goals are related to the interest that the simulation team sees in the results of the simulation study. Whilst agent goals refer to the states of the world an agent wants to achieve in the system they live. This system is the means the simulation approach uses to achieve the intended simulation goals. Thus, simulation goals constraint the scope of the problem to be addressed by the simulation study and they guide the definition of the problem elements that are useful to model.

In this context it is of high importance to have some guidelines for extracting the goal of the simulation and the system goals from the description of the problem domain. In some previous work [21] we already experienced how to extract goals from a formalized description of the problem domain and we want to somehow apply the same approach for the future development of our work on agent based simulation study.

In this paper we explore the step before the identification of goals and provide a set of guidelines for modelling the structural aspect, hence the configuration of the real word influencing the simulation goal, of the problem domain. We propose an heuristic based on the analysis of verbs and nouns in the problem statement.

This work is the logic progression of what reported in [22] where we identified which elements have to be present in the problem statement of a simulation study; the result of that work was a metamodel whose elements (the part related to the structural aspect) are instantiated in the simulation problem analysis phase by means of the guidelines we propose in the proposed contribution.

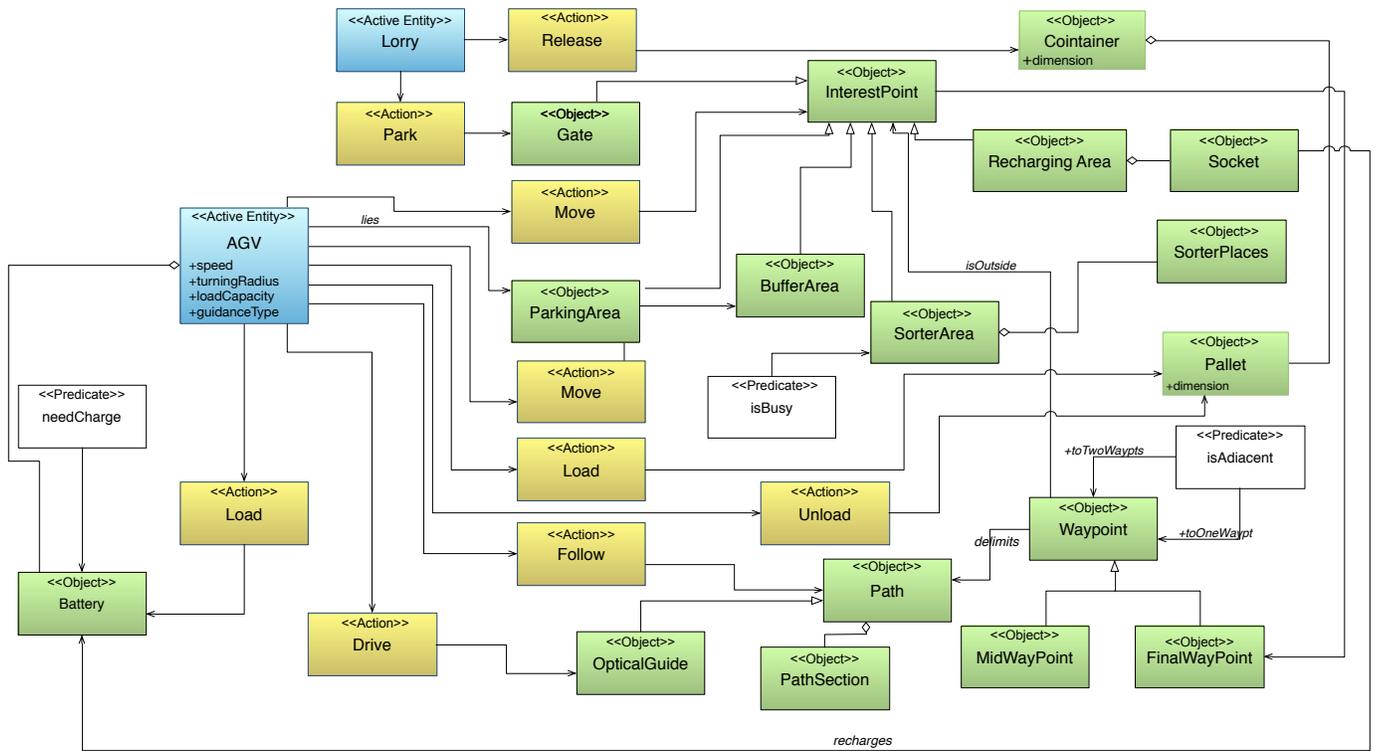


Fig. 8. The Problem Ontology diagram resulting from the application of the proposed heuristic for Logistic Warehouse case study.

For the future, we are going to complete all the activities of the simulation problem analysis phase as well as the whole methodological approach for multi agent simulation study.

REFERENCES

- [1] Osman Balci. Guidelines for successful simulation studies (tutorial session). In *Proceedings of the 22nd conference on Winter simulation*, pages 25–32. IEEE Press, 1990.
- [2] Osman Balci. Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of operations research*, 53(1):121–173, 1994.
- [3] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.
- [4] Bernd Bruegge and Allen H. Dutoit. *Object-Oriented Software Engineering Using UML, Patterns, and Java*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.
- [5] John S Carson et al. Introduction to modeling and simulation. In *Simulation Conference, 2005 Proceedings of the Winter*, pages 8–pp. IEEE, 2005.
- [6] Claudio Cioffi-Revilla. Computational social science. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3):259–271, 2010.
- [7] Alexis Drogoul, Diane Vanbergue, and Thomas Meurisse. Multi-agent based simulation: Where are the agents? In *Multi-agent-based simulation II*, pages 1–15. Springer, 2003.
- [8] Alfredo Garro and Wilma Russo. easyabms: A domain-expert oriented methodology for agent-based modeling and simulation. *Simulation Modelling Practice and Theory*, 18(10):1453–1467, 2010.
- [9] J.P. Georgé, M.P. Gleizes, P. Glize, and C. Régis. Real-time simulation for flood forecast: an adaptive multi-agent system staff. In *Proceedings of the AISB*, volume 3, pages 109–114, 2003.
- [10] Amineh Ghorbani, Pieter Bots, Virginia Dignum, and Gerard Dijkema. Maia: a framework for developing agent-based social simulations. *Journal of Artificial Societies and Social Simulation*, 16(2):9, 2013.
- [11] Nigel Gilbert. *Agent-based models*. Number 153. Sage, 2008.
- [12] Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K Heinz, Geir Huse, et al. A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1):115–126, 2006.
- [13] Dirk Helbing and Stefano Balietti. How to do agent-based simulations in the future: From modeling social mechanisms to emergent phenomena and interactive systems design. Chapter "Agent-Based Modeling" of the book "Social Self-Organization" by Dirk Helbing (Springer, Berlin, 2012), 2013.
- [14] Alexander Helleboogh, Giuseppe Vizzari, Adelinde Uhrmacher, and Fabien Michel. Modeling dynamic environments in multi-agent simulation. *Autonomous Agents and Multi-Agent Systems*, 14(1):87–116, 2007.
- [15] Franziska Klügl. Engineering agent-based simulation

- models? In *Agent-Oriented Software Engineering XIII*, pages 179–196. Springer, 2013.
- [16] Franziska Klügl and Paul Davidsson. Amazon: Abstract meta-model for agent-based simulation. In *Multiagent System Technologies*, pages 101–114. Springer, 2013.
- [17] Averill M Law. How to build valid and credible simulation models. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 24–33. IEEE, 2009.
- [18] Ambra Molesini, Matteo Casadei, Andrea Omicini, and Mirko Viroli. Simulation in agent-oriented software engineering: The soda case study. *Science of Computer Programming*, 78(6):705–714, 2013.
- [19] Sara Montagna, Alessandro Ricci, and Andrea Omicini. A&a for modelling and engineering simulations in systems biology. *International Journal of Agent-Oriented Software Engineering*, 2(2):222–245, 2008.
- [20] Michael J North and Charles M Macal. *Managing business complexity: discovering strategic solutions with agent-based modeling and simulation*. Oxford University Press, 2007.
- [21] Patrizia Ribino, Massimo Cossentino, Carmelo Lodato, Salvatore Lopes, Luca Sabatucci, and Valeria Seidita. Ontology and goal model in designing bdi multi-agent systems. *WOA@ AI* IA*, 1099:66–72, 2013.
- [22] Patrizia Ribino, Valeria Seidita, Carmelo Lodato, Salvatore Lopes, and Massimo Cossentino. Common and domain-specific metamodel elements for problem description in simulation problems. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1467–1476. IEEE, 2014.
- [23] Robert Siegfried, Axel Lehmann, Rachid El Abdouni Khayari, and Tobias Kiesling. A reference model for agent-based modeling and simulation. In *Proceedings of the 2009 Spring Simulation Multiconference*, page 23. Society for Computer Simulation International, 2009.
- [24] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.
- [25] Michael J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc. New York, NY, USA, 2001.