DOI: 10.2478/ijasitels-2024-0011

Categorical Mechanisms in Modelling Multiagent Systems

Daniel C. Crăciunean[1]

[1]*Computer Science and Electrical and Electronics Engineering Department, Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania*
daniel.craciunean@ulbsibiu.ro

Abstract

The main contribution that this paper brings is the specification of multiagent systems, at the metamodel level, using an appropriate categorical sketch. We will see that category theory provides all the necessary ingredients for the formal specification of multiagent systems. In our approach, a multiagent system is specified by a static dimension and a behavioural dimension. For both dimensions we have defined a metamodel based on the categorical sketch. To specify the static dimension, we used a categorical sketch whose models are the states of the multiagent system. To specify the behavioural dimension, we introduced a Kripke-type categorical metamodel, which is based on a categorical sketch with constraints equivalent to the specified logical axioms.

Keywords: multiagent system, categorical sketch, Kripke model, metamodel, modal logic

1 Introduction

A formal definition of the concept of agent, unanimously accepted by the community in the field, does not exist, but a multitude of characterizations have been issued, each with their pluses and minuses [1]. Although these characterizations of the agents are different, depending on the specific applications, they all include the notions of the environment in which the agents evolve and their autonomy. Autonomy means the ability of an agent to respond to environmental changes through various actions.

If object-based modelling is characterized by the encapsulation of an object's attributes and access to them only through methods, agent-based modelling goes a step further and encapsulates the methods as well, so that access to the methods can only be done indirectly through messages. This approach leaves the freedom of the agent to decide which methods to use to achieve an objective.

Because the property of autonomy was quite confusing, in the definition of the concept of agent, it was later replaced with the property of flexible autonomy. The resulting agent concept is that of an intelligent agent that has three important characteristics [2]: reactivity, pro-activeness and social ability.

When designing a system, the designer seeks to achieve some global objectives that the resulting system must fulfil. Most of the time there is no agent capable of fulfilling these general objectives. An agent is able to fulfil local objectives. Global objectives can only be achieved by aggregating several agents, so that the fulfilment of their local objectives leads to the achievement of these global objectives. The result of this operation of adequate aggregation of agents, in order to fulfil some general objectives, is a multiagent system.

In general, the achievement of some objectives can be expressed by satisfying some logical formulas. One of the important objectives of logic is to provide languages and formal mechanisms for specifying reasoning on specific situations for the development of models. These languages and mechanisms must be, on the one hand, as intuitive as possible and on the other hand, endowed with rigorous syntax and semantics, without ambiguities so that they can be executed by a machine. In this context, modal logic plays a decisive role because it reduces the complexity of the language and at the same time preserves rigor.

Most of the time, in these models, besides the concepts of space, time, events, human or artificial agents appear and therefore the concepts of knowledge, action, belief. Each of these concepts requires its own way of reasoning and as a result of this difference, multimodal logic was introduced, which allows reasoning with several modal operators.

The reference model for specifying modal semantics is the Kripke frame, introduced by Saul Kripke in 1959, which nowadays has become a standard in specifying the semantics of multiagent systems, based on modal logic. We will introduce, in this paper, a categorical Kripke model for specifying the behavioural dimension of a multiagent system.

The main contribution that this paper brings is the specification of multiagent systems, at the metamodel level, using an appropriate categorical sketch. Category theory provides all the necessary ingredients for the formal specification and analysis of models [6, 4]. In our approach, a multiagent system is specified by two dimensions, namely: the static dimension and the behavioural dimension. For both dimensions we have defined a metamodel based on the categorical sketch.

Section 2 introduces some general notions and notations used in section 3 which presents the metamodel and the categorical model of a multiagent system. Section 4 concludes the paper with some conclusions and future papers.

2 Overview

A category $\boldsymbol{C}$ is a mathematical construct made up of two types of atomic components, namely formal functions that we call arrows, and objects that are the domains and codomains of the formal functions to which the function composition operation is added. In addition, the multitude of functions together with the composition operation form a monoid structure, i.e., it respects the associativity property and there is an identity function for each object. If $\boldsymbol{C}$ is a category, we will denote with $\boldsymbol{C}_0$ the set of objects of this category, and with $\boldsymbol{C}_1$ the set of arrows of the category.

In this paper, we will use especially the category that has sets as objects and as arrows functions with domains and codomains these sets, which we denote with Set, and we will also use the category that has graphs as objects and as arrows homomorphism between these graphs, which we denote by Grf. In this context, we will specify the multiagent systems, using as a metamodel, the categorical sketch, which is a mathematical object with precise syntax and implicit semantics. To specify the semantics of these systems, we will use the Kripke structures, which we will specify, also through a categorical sketch.

A categorical sketch is defined as a graph $\mathcal{G}$, together with a set of constraints $\mathcal{C}(\mathcal{G})$ imposed on the models through the components of the sketch graph. So, the categorical sketch $\mathcal{S}$ is a tuple $\mathcal{G}=(\mathcal{G}, \mathcal{C}(\mathcal{G}))$ [5, 3]. The constraints on the models can be specified by commutative diagrams, limits and colimits in the classical form of the sketch or by logical predicates in the case of the generalized sketch [5, 3]. Commutative diagrams, limits and colimits have a great advantage in modelling, because they are generic constructs and can also be used in the generalized sketch for specifying predicates.

A model of a categorical sketch $\mathcal{S}=(\mathcal{G}, \mathcal{C}(\mathcal{G}))$, is the image of the graph $\mathcal{G}$, through a homomorphism of graphs, in the category Set; $M:\mathcal{G}\rightarrow Set$, image that is subject to the constraints $\mathcal{C}(\mathcal{G})$. A wide range of diagrammatic models used in software engineering can be defined as categorical sketch models [3].

Functors are similar to graph homomorphisms only that they respect the monoid structure of the set of functions with the composition operation, i.e., associativity and conservation of the identity function. Graphs can also be extended to free categories by composing arrows and adding identity arrows to nodes. To simplify the exposition, we will continue to use the name functor even when it is a homomorphism of graphs.

To define the constraints, on the graph structure of the models, we need the diagram concept. A diagram D is a functor d, defined on a shape graph $\mathcal{P}$, with values in a category $\mathcal{C}$, i.e., a functor $d:\mathcal{P}\rightarrow\mathcal{C}$. A diagram has the property that its image in the category $\mathcal{C}$ preserves the shape graph $\mathcal{P}$, even if several nodes have the same label or several vertices have the same label [5, 12]. This means that the category could only be ambient for the image of diagram D, without this image mapping exactly on a portion of the category. The role of the diagrams is to link the formulas of the first order logic (FOL) to the components of the models.

We will specify the predicates that represent the constraints of the categorical sketch through the concept of diagram predicate signature. A set of predicates Π, together with an application $ar:\Pi\rightarrow Grf_0$, defines a diagram predicate signature. The application ar, maps each $P\in\Pi$ to a graph, from the category Grf, which is called shape graph arity of P. The images of the application ar in Grf_0 will be shape graphs for the diagrams that will map them to the components of the sketch graph $\mathcal{S}$ and therefore the specified constraints by predicates will propagate on the models through the diagrams.

Example 2.1. Let's suppose that we want to set the condition that the graph structure of all models of a sketch has the property that between any two nodes there is only one arc. This condition can be put by including in the diagram predicate signature, the predicate $P_1(x,y,z,r_{zx},r_{zy})=(\forall a_1,a_2\in z\Rightarrow((r_{zx}(a_1)=r_{zx}(a_2)\wedge(r_{zy}(a_1)= r_{zy}(a_2))\Rightarrow a_1=a_2))$ where the shape graph arity is $ar(P_1(x,y,z,r_{zx},r_{zy}))=Span(x,y,z,r_{zx},r_{zy})=(x\xleftarrow{r_{zx}}z\xrightarrow{r_{zy}}y)$ and $ar(x)=x$, $ar(y)=y$, $ar(z)=z$, $ar(r_{zx})=r_{zx}$, $ar(r_{zy})=r_{zy}$.

Shape graph $Span(x,y,z,r_{zx},r_{zy})$, will then be mapped by a diagram to the sketch graph components. Through the functor that defines the model, these constraints will reach the components of the model. The role of the shape graph construct is to keep the shape of the graph signature at the model level.

Intuitively, we can interpret the diagram predicate signature as a collection of procedures that implement the constraints defined by predicates having as parameters, formal parameter graph, and diagrams, by means of some functors, associate formal parameter graph to actual parameter graph, nodes to nodes and arcs to arcs.

In this paper we will use the categorical sketch to specify, at the metamodel level, the static dimension of a multiagent model and also to represent the behavioural dimension of the system.

Each model of the categorical sketch that specifies the static dimension of the system is a state of the system and is characterized by the graph structure of the system and the values of some attributes attached to the components of the model, at a given moment [18]. The sketch models that represent the behavioural dimension are Kripke type models, which have the states of the system as possible worlds. The transitions of the system are the result of the actions of the agents, who act to fulfil some local objectives. Each type of agent is endowed with a specific modal logic.

From a syntactic point of view, the basic modal logic language contains well-formed formulas, with the classic propositional logic operators to which two unary modal operators $\Box$ and $\Diamond$ are added. Depending on the specified modal logic, the two modal operators can have various interpretations, for example required for $\Box$, and possible for $\Diamond$. The two operators are linked by the relation $\Diamond\varphi=\neg\Box\neg\varphi$, where φ is a logical formula, and therefore, they are not independent.

The standard for interpreting the formulas of modal logic are the Kripke models. A Kripke type model is a tuple $M=(\mathcal{K},\mathcal{P},\pi)$, where $\mathcal{K}=(\mathcal{W},\mathcal{R})$, is a graph, which is called a Kripke frame, $\mathcal{W}$ is a set of possible worlds, $\mathcal{R}$ is the accessibility relation on the set $\mathcal{W}$, $\mathcal{P}$ is a set of atomic propositions and $\pi:\mathcal{W}\to 2^{\mathcal{P}}$, is an evaluation application that returns for each possible world $w\in\mathcal{W}$, the atomic propositions satisfied in the respective world.

If we denote by $\mathcal{R}(v, w)$ the arc in $\mathcal{R}$ that connects the world $v\in\mathcal{W}$ to the world $w\in\mathcal{W}$, then we can check if a well-formed formula is satisfied in the world $v(v\Vdash\varphi)$ of the model M, inductively as [15]:

$M,v \Vdash p \Leftrightarrow p \in \pi(v)$;

$M,v \Vdash\neg\varphi \Leftrightarrow M,v \nVdash \varphi$;

$M,v \Vdash \varphi \wedge \psi \Leftrightarrow M,v \Vdash \varphi$ and $M,v \Vdash \psi$;

$M,v \Vdash \varphi \vee \psi \Leftrightarrow M,v \Vdash \varphi$, or $M,v \Vdash\psi$;

$M,v \Vdash \varphi \to \psi \Leftrightarrow M,v \Vdash \varphi$ implies $M,v \Vdash\psi$;

$M,v \Vdash \varphi \leftrightarrow \psi \Leftrightarrow (M,v \Vdash \varphi \Leftrightarrow M,v \Vdash \psi)$;

$M,v \Vdash\Box\psi \Leftrightarrow$ (for each $w\in\mathcal{W}$ with $\mathcal{R}(v,w)$, we have $M,w\Vdash\psi$);

$M,v \Vdash\Diamond\psi \Leftrightarrow$ (there is a $w\in\mathcal{W}$ such that $\mathcal{R}(v, w)$ and $M,w \Vdash \psi$).

In general, the evaluation of modal formulas depends on the axioms we impose. In modal logic, one starts from an axiom, called axiom K:$\Box(\varphi\to\psi)\to(\Box\varphi\to\Box\psi)$. Other important axioms that have been imposed in modal logic are: T:$\Box\varphi\to\varphi$; B:$\varphi\to\Box\Diamond\varphi$; D:$\Box\varphi\to\Diamond\varphi$; 4:$\Box\varphi\to\Box\Box\varphi$ and 5:$\Diamond\varphi\to\Box\Diamond\varphi$.

In Kripke models, as we can see, the evaluation of a modal formula depends a lot on the accessibility relation $\mathcal{R}$. There is an equivalence between axioms, or other logical formulas and the structure of the accessibility relation $\mathcal{R}$. In our approach, we will take advantage of this equivalence and impose the axioms through constraints on the relation $\mathcal{R}$, in a categorical sketch.

3 The Categorical Model of a Multiagent System

Agents are autonomous physical or logical entities that can perform actions in a certain environment in order to fulfil certain objectives. In general, they can observe environmental changes caused by other agents in a certain context and will make decisions based on these changes. We assume that each agent is endowed with a finite set of actions that it can perform. The performance of some actions is conditional on the state of the environment, which it perceives through the associative preconditions of each action and the objectives it pursues.

Based on the preconditions and objectives pursued, the agent will have to choose from the possible actions those that satisfy his objectives in optimal conditions. This choice implies a certain logical reasoning in each state. In our approach, each agent will be endowed with a certain modal logic, based on which the agent will reason for decision-making.

When designing a multiagent system, the designer pursues a general objective, which cannot be known by each agent. Agents only have local objectives. Therefore, the system designer will have to make a convenient aggregation of a lot of local objectives to achieve the general objective.

In order to achieve the objectives, most of the time agents must communicate and cooperate with each other. Therefore, a good structuring of a multiagent system will have to allow the encapsulation of cooperating agents in appropriate substructures that could cooperate or compete with other substructures depending on the general objectives.

There are many and quite different approaches related to the organization of agents, some more flexible others less flexible [13, 14]. The categorical sketch is a formal construction, suitable for structuring agents and flexible enough to allow a great diversity of structuring multiagent systems.

Each type of component is characterized by attributes and behaviour. A state of a component is represented by the values of the attributes at a given moment. A state of the system is represented by the graph structure of the system and the states of all its components at a given moment. The transactions of the system from one state to another are done by the actions of the components that can modify both the values of the attributes and the graph structure of the model within the limit allowed by the constraints $\mathcal{C}(\mathcal{G})$.

Even if the categorical sketch will only specify static models, i.e., states of the multiagent system, it facilitates the specification of dynamic components through other related mechanisms.

Each agent acts with the aim of achieving an objective: The objectives that an agent must achieve are specified by logical formulas, which will have to be satisfied in the following states. That is, an objective will be represented by a logical formula φ which is not satisfied in the current state but will have to be satisfied in the state after the action. The possibility of performing an action is also conditioned by the satisfaction of a logical formula in the current state. These decisions can be taken by an agent through a logical reasoning with which it is endowed, as will be presented in the next section. The transformations produced by the actions of the agents will be made within the limits allowed by the constraints of the categorical sketch.

Also, agents can move from one substructure to another, or new agents can appear in the system or some agents can disappear from the system, thus modifying the initial structure of the model. These structural changes can be specified by graph transformations of the model within the limits allowed by the constraints of the categorical sketch.

In our approach, the agents' actions can have the effect of changing the values of the attributes as well as changing the graph structure of the model, within the limits allowed by the constraints of the categorical sketch. Therefore, we will specify a multiagent system on two dimensions, the static dimension and the behavioural dimension, each dimension with its own syntax and semantics.

3.1 The Static Dimension of the Multiagent Model

The syntax of the static dimension will be specified through a categorical sketch, and the semantics through the mapping of attributes to data domains and graphic structures to structures with known semantics (join, fork, etc.).

The behavioural dimension is given by the actions of the agents that can modify the values of the attributes and the graph structure of the model. The syntax of actions will be represented by action signatures, and the semantics by mapping signatures to the double pushout algorithm and to algorithms that transform the values of the attributes.

To define a categorical sketch, we need a diagram predicate signature which is a construct $\Theta=(\Pi,ar)$ formed by a set of predicates Π and an application $ar:\Pi\rightarrow Grf_0$, which associates to each predicate P from Π an object from the Grf category, object called graph arity of P.

A categorical sketch $\boldsymbol{\mathcal{S}}$ is a tuple $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}(\Theta))$, where $\boldsymbol{\mathcal{G}}$ is a graph, called the sketch graph, Θ is a diagram predicate signature and $\boldsymbol{\mathcal{D}}(\Theta)$ is a set of diagrams indexed by the set of predicates Π, which have as shape graph, the images of the application ar in the Grf category [3]. That is, $\boldsymbol{\mathcal{D}}(\Theta)=\{d_P:ar(P)\rightarrow\boldsymbol{\mathcal{G}}|P\in\Pi\}$. Thus, $\boldsymbol{\mathcal{D}}(\Theta)$ defines the signatures of the predicates that represent the constraints of the sketch on the structure of the models and maps them to the components of the sketch graph. A model of a sketch $\boldsymbol{\mathcal{S}}$ is the image of a functor $M:\boldsymbol{\mathcal{G}}\rightarrow Set$, which satisfies all the constraints specified by $\boldsymbol{\mathcal{D}}(\Theta)$ and represents an instance of the multiagent system, i.e., a state of this system.

Next, we will consider that a directed graph is specified by a set of objects X, called nodes, a set of arcs Γ, and two functions $s,t:\Gamma\rightarrow X$, which associate the source and target nodes to the arcs.

In our approach, the nodes of the sketch graph represent concepts of the model such as agents, objects, groups of agents and roles, as well as elements of organizing and structuring of these components to form an adequate system. Sketch graph arcs represent sketch operators meant to help define constraints. Therefore, the sketch graph will have to be a type graph [7, 10, 11]. We will consider that the label of each node or arc of the sketch graph will correspond to the label of its type. That is, if we have a sketch $\boldsymbol{\mathcal{S}}$ and a model $M:\boldsymbol{\mathcal{S}}\rightarrow Set$, then, for each component c of the model $type_M(c)=C$ where $c\in M(C)$

and C is a component of the graph $\mathcal{G}$. Thus, this categorical sketch becomes a metamodel for the static dimension of a multiagent system.

Therefore, the sketch graph nodes represent types such as agent type, object type, group type, role type, etc., or types of relationships between them. These types can be decomposed into subtypes, if the constraints on the structure of the models require this. In Fig. 1, we have an example of a categorical sketch graph. We denoted with A the agent type, with O the object type, with G the group type and with R the role type. We also denoted with γ_{XY}, the type of relationship between type X and type Y, meaning that concepts of type X can contain concepts of type Y. Of course, we cannot present a graph of the sketch that satisfies the requirements of all applications, and we do not intend to do so.

As we have already mentioned, the generalized sketch defines the constraints in the form of signatures of logical predicates that have as variables the nodes and arcs of shape graphs. These special graphs, called shape graphs, are mapped by diagrams to the sketch graph components (Fig. 2).

Example 3.1. In Example 2.1. we specified the constraint that between any two nodes of the models there should be a single arc, by including in the diagram predicate signature, the predicate $P_1(x,y,z,r_{zx},r_{zy})$ with shape graph arity, $ar(P_1(x,y,z,r_{zx},r_{zy}))=(x \xleftarrow{r_{zx}} z \xrightarrow{r_{zy}} y)$. To map this shape graph to the components of the graph $\mathcal{G}$ of the sketch $\mathcal{S}$, we will use a diagram $d_1:(x \xleftarrow{r_{zx}} z \xrightarrow{r_{zy}} y) \rightarrow \mathcal{G}$, defined as follows: $d_1(x)=d_1(y)=X$, $d_1(z)=\Gamma$, $d_1(r_{zx})=s$, $d_1(r_{zy})=t$. Note that although the nodes x,y in the shape graph are mapped to the same node X of the graph $\mathcal{G}$, the image of diagram d_P keeps the graph shape.

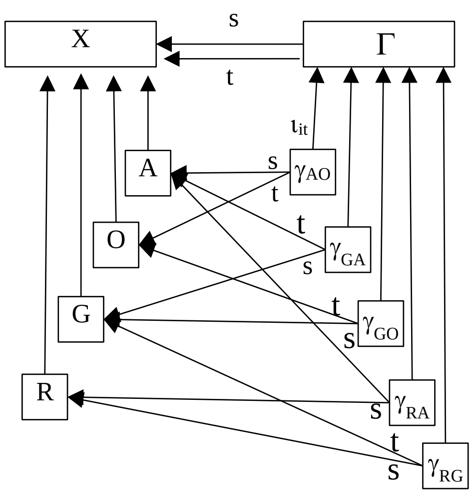

Fig. 1. An example sketch graph

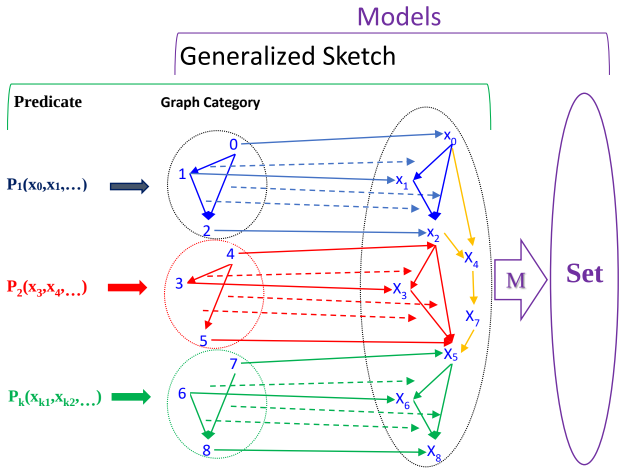

Fig. 2. Mapping diagram predicate signature to the model

Through the functor that defines the model, these constraints will reach the components of the model. The role of the shape graph construction is to keep the graph shape at the model level.

Although universal constructs such as commutative diagrams, limits and colimits, are used especially in the classical categorical sketch, many times they are also very useful in the generalized sketch to define predicates as we can see in Example 3.

Example 3.2. If we want to constrain the structure of any model of the sketch to be a connected graph, we can use a special limit, namely pushout, which is the limit of a span. It is known that the pushout of two functions f and g with the same domain of definition, which forms a span, $X \xleftarrow{f} Z \xrightarrow{g} Y$, coincides with the set of equivalence classes determined by the equivalence relation ρ induced by the relation: xρy, x∈X, y∈Y if and only if ∃ z∈Z so that f(z)=x and g(z)=y.

In the case of a graph, the pushout of the source s and target t functions from the definition of a graph coincides with the equivalence classes determined by the equivalence relation induced by s and t on the set of graph nodes, i.e., it coincides with the set of connected components of the graph. Therefore, the constraint that the graph of any model is connected can be put by the predicate $P_2(x,y,z,r_{zx},r_{zy})=|pushout(x \xleftarrow{r_{zx}} z \xrightarrow{r_{zy}} y)|=1$, i.e. the cardinal of the set $pushout(x \xleftarrow{r_{zx}} z \xrightarrow{r_{zy}} y)$, which represents the number of connected components of the graph to be 1, with the shape graph arity, $ar(P_2(x,y,z,r_{zx},r_{zy}))=(x \xleftarrow{r_{zx}} z \xrightarrow{r_{zy}} y)$, and diagram d_2, which coincides with diagram d_1, from Example 3.2.

A model of the categorical sketch represents a state of the multiagent system. The semantics of a state is characterized by the values of the attributes and the graphical structure of the model. Also, a state of the model can be endowed with a lot of atomic logical formulas that characterize the values of the attributes and the graph structure of the model.

3.2 The Behavioral Dimension of the Multiagent Model

The behavioural dimension of a multiagent system is characterized by its syntax and semantics. To specify the syntax of the behavioural dimension of a multiagent system, we will use as a metamodel, also a categorical sketch, as in the case of the static dimension, and to specify its semantics we will use a Kripke type model.

The static dimension of the categorical model specifies the set of possible states of the system. Thus, if we have a categorical sketch $\boldsymbol{\mathcal{S}}$, there are many functors M:$\boldsymbol{\mathcal{S}}\rightarrow$Set, and each functor M represents a state of the model. We denote this set of functors by Mod($\boldsymbol{\mathcal{S}}$, Set). We assume that the set Mod($\boldsymbol{\mathcal{S}}$,Set) is at most countable and, therefore, we can index it with natural numbers.

The transition from one state to another of the behavioural model is done by the actions of the agents involved in this model [9]. Thus, if the initial state of the behavioural model is $\mathfrak{I}_0$, then the set of possible states $\mathfrak{I}$ of the behavioural model is: $\mathfrak{I}=\{\mathfrak{I}_k\in$Mod($\boldsymbol{\mathcal{S}}$, Set), $k\geq1$ | $\mathfrak{I}_k$ is the result of successive actions of the agents on $\mathfrak{I}_0\}$.

In this notation we will understand that a functor $\mathfrak{I}_k$, represents the image of the sketch $\boldsymbol{\mathcal{S}}$, through the functor $\mathfrak{I}_k$ in the Set category.

In our approach, an agent is characterized by a multitude of actions it can perform and by the logic it uses to make decisions. The performance of an action is conditioned by the state in which the model is located and by the objective it pursues in the following states.

An action can change the values of the attributes of the model, but it can also change the graph structure of the model because, as we mentioned before, agents can move from one substructure to another, new agents can appear or some of them can disappear. All these changes can only be made within the limits allowed by the constraints of the categorical sketch.

We will specify the structural changes of the models caused by the agents' actions through graph transformations. A graph transformation, $\boldsymbol{\tau}$=(L, R), is composed of two graphs, namely the left graph L and the right graph R, and a mechanism that specifies the conditions and the way to replace L with R. Because the structural changes, in our case, involves both deletions and additions, we will use the double-pushout variant (DPO), a graph transformation that is specified by three graphs L, R and K and two graph morphisms l and r: $\boldsymbol{\tau}=(L\xleftarrow{l}K\xrightarrow{r}R)$ where K is an interface graph contained in both R and L [10, 11, 7].

The components of the graphs L and R will be mapped to the components of the graph $\boldsymbol{\mathcal{G}}$, of the sketch $\boldsymbol{\mathcal{S}}$, by a pair of diagrams d_L and d_R, and will receive the types of the components of the graph $\boldsymbol{\mathcal{G}}$, and therefore will be shape graph typed. Now we can define an action of an agent as a pair a=($\boldsymbol{\tau}$, p), where $\boldsymbol{\tau}$ is a graph transformation $\tau=(L\xleftarrow{l}K\xrightarrow{r}R)$, and p is a procedure with the property that R=p(L). Therefore, performing an action a=($\boldsymbol{\tau}$, p), consists in applying the graph transformation $\boldsymbol{\tau}$ and calculating the values of some local attributes, associated with graphs L, R and K. Of course, the graph transformation could be an identity transformation that does not change the graph structure of the system in any way, but it is still useful because the graph L from the graph transformation $\boldsymbol{\tau}$ also has the role of locating the agent's action

To model the semantics of the behavioural dimension of multiagent systems, we will use Kripke-type structures, which have an implicit semantics. In our approach, each agent is characterized by a multitude of actions that it is capable of performing and by the logic that it uses to evaluate the logical formulas necessary for decision-making.

If all the agents use the same logic to evaluate the formulas, then, in our approach, a semantic model of a multiagent system is a tuple:

$M=(\mathcal{K}, \mathcal{P}, \pi, w_0, \mathcal{A}c, \alpha, \mathcal{A}g, \beta, \delta)$, where:

$\mathcal{K}=(\mathcal{W}, \mathcal{R})$ is a graph in which the set of nodes $\mathcal{W}$ represents the set of possible worlds and $\mathcal{R}$ is a set of arcs between the elements of $\mathcal{W}$;

$\mathcal{P}$ is a set of atomic propositions, and $\pi:\mathcal{W}\rightarrow 2^{\mathcal{P}}$, is an evaluation application, which associates to each world the set of valid atomic propositions in that world.

$\mathcal{A}c$ is a set of actions, and α is a surjective function $\alpha:\mathcal{R}\rightarrow\mathcal{A}c$, which associates an action to each arc;

$\mathcal{A}g$ is a set of agents, and β is a surjective function $\beta:\mathcal{R}\rightarrow\mathcal{A}g$, which associates an agent to each arc;

δ is a surjective function $\delta:\mathcal{A}c\rightarrow\mathcal{A}g$, which associates to each agent the set of actions it is capable of performing and which has the property $\beta=\delta\circ\alpha$.

We notice that the graph $\mathcal{K}=(\mathcal{W}, \mathcal{R})$, is, in fact, a Kripke frame, where $\mathcal{W}$ is a set of possible worlds, and $\mathcal{R}$ is the accessibility relation on the set $\mathcal{W}$. Also, the tuple $(\mathcal{K}, \mathcal{P}, \pi)$ specifies a Kripke semantic model for modal logic.

The theoretical results showed that there is a direct correspondence between the satisfaction of some schemes of modal logical formulas and the properties of the accessibility relation $\mathcal{R}$ [17, 8, 15, 16]. Thus, for axiom T to be satisfied, relation $\mathcal{R}$ must be reflexive, axiom B is satisfied if relation $\mathcal{R}$ is symmetric, axiom D is satisfied if relation $\mathcal{R}$ is serial, axiom 4 is satisfied if relation $\mathcal{R}$ is transitive, and axiom 5 is satisfied if the relation $\mathcal{R}$ is Euclidean. Also, if the relation $\mathcal{R}$ is functional, the formula $\Box\varphi\leftrightarrow\Diamond\varphi$ is satisfied, and if the relationship is linear, it satisfies the formula $\Box(\varphi\wedge\Box\varphi\rightarrow\psi)\vee\Box(\psi\wedge\Box\psi\rightarrow\varphi)$.

Although a Kripke frame $\mathcal{K}=(\mathcal{W}, \mathcal{R})$, does not contain the atomic formulas satisfied in every world $w\in W$, it is important to be able to specify a Kripke frame that satisfies certain formula schemes as a whole. Thus, a Kripke frame $\mathcal{K}=(\mathcal{W}, \mathcal{R})$, satisfies a formula scheme ϕ, i.e., $\mathcal{K}\Vdash\phi$, if for any evaluation application $\pi:\mathcal{W}\rightarrow 2^{\mathcal{P}}$, in every world $w\in\mathcal{W}$, $\mathcal{K},w\Vdash\phi$. Also, we have a theoretical result that says a Kripke frame $\mathcal{K}$, which satisfies a scheme of formulas, satisfies all substitution instances of that formula [15, 17].

Therefore, a model $M=(\mathcal{K}, \mathcal{P}, \pi, w_0, \mathcal{A}c,\alpha, \mathcal{A}g, \beta, \delta)$, of a multiagent system characterized by a certain logic can be specified by constraints on the accessibility relation. Thus, if we want the model to be characterized by the KT4 logic, we will introduce the constraints that the relation $\mathcal{R}$ be reflexive and transitive, if we want the model to be characterized by the KT45 logic, we will introduce the constraints that the relation $\mathcal{R}$ be reflexive, transitive and Euclidean, etc.

A model of a multiagent system may include concepts such as time, knowledge, belief, obligations, etc., which determine the formula schemes that must be satisfied. Therefore, in a multiagent system we will have to have for each type i of agents a distinct $\Box_i$ operator and, implicitly, a distinct $\Diamond_i$ operator. The i index of the agent type in modal operators implies distinct accessibility relations.

When designing a model for a multiagent system, it is important to establish precisely which formula schemes we need to be satisfied by each type of agent, and to specify constraints so that they are equivalent to these formula schemes.

As we have already mentioned, for the specification of a metamodel, of the behavioural dimension of a multiagent system, we will use the categorical sketch. We will denote this sketch by $\boldsymbol{\mathcal{S}}^B=(\boldsymbol{\mathcal{G}}^B, \boldsymbol{\mathcal{D}}^B(\Theta))$. In this case, the concepts involved in the metamodel are the agents Ag, with the corresponding subtypes, the relation R, with the sub-relations corresponding to each type of agent, and the actions Ac, with the subtypes corresponding to the types of agents. Thus, the graph of the categorical sketch, which specifies the behavioural dimension of a multiagent system, can be the one in Fig. 3.

In our approach, the set of agents Ag will be a disjoint union of subsets of different types of agents; Ag=$\coprod_{i=1}^{n} Ag_i$, the relation R is a disjoint union of sub-relations; R=$\coprod_{i=1}^{n} R_i$, and the set of actions Ac is a disjoint union of subsets of actions of different types; Ac=$\coprod_{i=1}^{n} Ac_i$. These constraints can be elegantly be imposed, in the categorical sketch, by colimits of discrete diagrams [5, 3], which we will not specify in this paper.

The constraints on the categorical sketch models are represented by the component $\boldsymbol{\mathcal{D}}^B(\Theta)$ by diagrams indexed by the set of predicates Π, which have as shape graph, the images of the application ar in the Grf category.

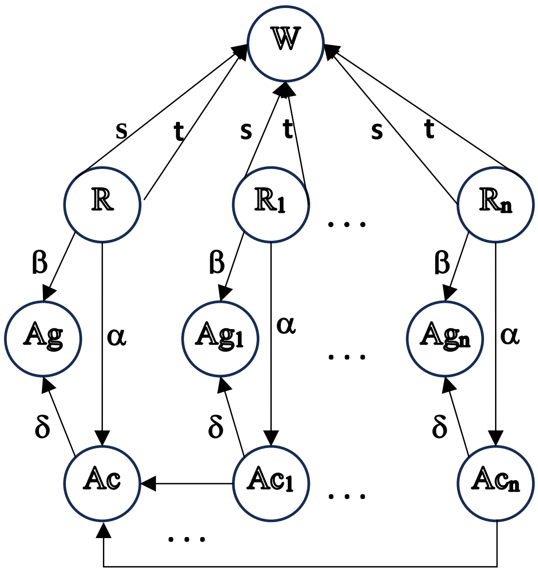

Fig. 3. The categorical sketch graph specifying the behavioral dimension

An important part of the predicates involved in such a metamodel are those that impose restrictions on the relation R and on the sub-relations R_i, $1 \le i \le n$. We will now exemplify some predicates that can impose constraints on the accessibility relations, conditions that can replace the logical formulas as mentioned above.

The constraint that a relation Ri, be reflexive can be put through the predicate:

$Q_1(x,y,s,t)=\forall w \in x, \exists r \in y, (s(r)=t(r)=w)$.

The symmetry of a relation Ri can be imposed by the predicate:

$Q_2(x,y,s,t)=\forall r_1 \in y, (s(r_1)=v \wedge t(r_1)=w \rightarrow \exists r_2 \in y. s(r_2)=w \wedge t(r_2)=v)$.

A relation is serial if the predicate is satisfied:

$Q_3(x,y,s,t)=\forall w\in x, \exists r\in y, \exists v\in x.\ s(r)=w\wedge t(r)=v.$

A relation is transitive if the predicate is satisfied:

$Q_4(x,y,s,t)=\forall u,v,w\in x,\exists r_1,r_2\in y.s(r_1)=u\wedge t(r_1)=v\wedge s(r_2)=v\wedge t(r_2)=w\rightarrow\exists r\in y.s(r)=u\wedge t(r)=w.$

A relation is Euclidean if the predicate is satisfied:

$Q_5(x,y,s,t)=\forall u,v,w\in x,(\exists r_1,r_2\in y.s(r_1)=u\wedge t(r_1)=v\wedge s(r_2)=u\wedge t(r_2)=w\rightarrow\exists r\in y.s(r)= v\wedge t(r)=w).$

A relation is functional if the predicate is satisfied:

$Q_6(x,y,s,t)=\forall w\in x, \exists! r\in y.\ s(r)=w\wedge t(r)=v.$

A relationship is linear if the predicate is satisfied:

$Q_7(x,y,s,t)=\forall u,v,w\in x,(\exists r_1,r_2\in y.s(r_1)=u\wedge t(r_1)=v\wedge s(r_2)=u\wedge t(r_2)=w\rightarrow\exists r_3\in y,s(r_3)= v\wedge t(r_3)=w).$

A relation is complete if the predicate is satisfied:

$Q_8(x,y,s,t)= \forall v,w\in x, \exists r\in y.\ (s(r)=v\wedge t(r)=w)\vee(s(r)=w\wedge t(r)=v).$

All these predicates have the shape graph arity $ar(Q)=x\underset{t}{\overset{s}{\leftleftarrows}}y$, defined as follows:

$ar(x)=x$, $ar(y)=y$, $ar(s)=s$, $ar(t)=t$. Also, all these predicates will be mapped to the components of the sketch graph through a diagram $\eta_1:ar(Q)\rightarrow\mathcal{K}$, thus: $\eta_1(x)=W$, $\eta_1(y)=R_i$, $\eta_1(s)=s$, $\eta_1(t)=t$, where i represents the relation on which we want to impose the constraint.

Therefore, the component $\mathcal{D}^B(\Theta)$, of the categorical sketch $\mathcal{S}^B$, will contain the diagram η_1, indexed by the predicates Q_i, $1\leq i\leq 8$.

Of course, these constraints can also be imposed through mechanisms specific to category theory, such as limits and colimits. For example, the condition that a relationship is total is equivalent to the condition that the graph corresponding to the relationship is connected. Therefore, the constraint that the relationship is total can be put through the predicate $Q(x,y,z,r_{zx},r_{zy})=|pushout(x\xleftarrow{r_{zx}} z \xrightarrow{r_{zy}}y)|=1$, which indexes the diagram d_1, from Example 2.

Other constraints imposed for the categorical sketch $\mathcal{S}^B$ refer to the functions α,β,δ, respectively $\alpha_i,\beta_i,\delta_i$, $1\leq i\leq n$. These functions must be epimorphisms. The necessary and sufficient condition for an application $f:A\rightarrow B$ to be an epimorphism is that the pushout of f and f is isomorphic to A. This can be specified by the predicate: $S_1(x,y,f)= |pushout(x\underset{f}{\overset{f}{\rightrightarrows}} y)|=|x|$ with shape graph arity $ar(S_1)= x\underset{f}{\overset{f}{\rightrightarrows}} y$, defined as such $ar(x)=x$, $ar(y)=y$, $ar(f)=f$. This shape graph will be mapped to the components of the graph $\mathcal{G}^B$, from Fig. 3, by a set of diagrams as follows: diagram μ^β defined as follows $\mu^\beta(x)=R$, $\mu^\beta(y)=Ag$, $\mu^\beta(f)=\beta$; diagrams μ_i^β defined as follows $\mu_i^\beta(x)=R_i$, $\mu_i^\beta(y)=Ag_i$, $\mu_i^\beta(f)=\beta_i$, $1\leq i\leq n$; diagram μ^α defined as follows $\mu^\alpha(x)=R$, $\mu^\alpha(y)=Ac$, $\mu^\alpha(f)=\alpha$; diagrams μ_i^α defined as follows $\mu_i^\alpha(x)=R_i$, $\mu_i^\alpha y)=Ac_i$, $\mu_i^\alpha(f)=\alpha_i$, $1\leq i\leq n$; diagram μ^δ defined as follows $\mu^\delta(x)=Ag$, $\mu^\delta(y)=Ac$, $\mu^\delta(f)=\delta$ and diagrams μ_i^δ defined as follows $\mu_i^\delta(x)=Ag_i$, $\mu_i^\delta y)=Ac_i$, $\mu_i^\delta(f)=\delta_i$, $1\leq i\leq n$;

The conditions that $\beta=\delta\circ\alpha$ and respectively $\beta_i=\delta_i\circ\alpha_i$ can be specified by the predicate:

$S_2(x,y,z,f,g,h)=\forall r\in x,\ f(x)=h(g(x))$, with shape graph arity $ar(S_2)=x\xrightarrow{f}y,\ x\xrightarrow{g}z\xleftarrow{h}y$, defined as follows: $ar(x)=x$, $ar(y)=y$, $ar(z)=z$, $ar(f)=f$, $ar(g)=g$, $ar(f)=f$, $ar(g)=g$, $ar(h)=h$. This shape graph will be mapped to the graph components by the diagram ω defined as follows $\omega(x)=R$, $\omega(y)=Ac$, $\omega(z)=Ag$, $\omega(f)=\alpha$, $\omega(g)=\beta$, $\omega(h)=\delta$, and by the diagrams ω_i, defined as follows $\omega_i(x)=R_i$, $\omega_i(y)=Ac_i$, $\omega_i(z)=Ag_i$, $\omega_i(f)=\alpha_i$, $\omega_i(g)=\beta_i$, $\omega_i(h)=\delta_i$, $1\le i\le n$.

Therefore, in this metamodel $\Pi=\{Q_i|1\le i\le 8\}\cup\{S_1,S_2\}$, and $\boldsymbol{\mathcal{D}}(\Theta)=\{\eta_1:ar(Q_i)\to\boldsymbol{\mathcal{G}}\ |1\le i\le 8\}\cup\{\mu^{\alpha}:ar(S_1)\to\boldsymbol{\mathcal{G}},\ \mu^{\beta}:ar(S_1)\to\boldsymbol{\mathcal{G}},\ \mu^{\delta}:ar(S_1)\to\boldsymbol{\mathcal{G}},\ \omega:ar(S_2)\to\boldsymbol{\mathcal{G}}\}\cup\{\mu_i^{\alpha}:ar(S_1)\to\boldsymbol{\mathcal{G}}\ |\ 1\le i\le n\}\cup\{\mu_i^{\beta}:ar(S_1)\to\boldsymbol{\mathcal{G}}\ |\ 1\le i\le n\}\cup\{\mu_i^{\delta}:ar(S_1)\to\boldsymbol{\mathcal{G}}\ |\ 1\le i\le 8\}\cup\{\mu_i^{\alpha}:ar(S_i)\to\boldsymbol{\mathcal{G}}\ |\ 1\le i\le n\}\cup\{\omega_i:ar(S_2)\to\boldsymbol{\mathcal{G}}\ |\ 1\le i\le n\}$. Depending on the requirements of the model, other constraints can be added, such as imposing a fixed or limited number of agents or actions.

A model of the sketch B is a functor $\mathfrak{B}:\boldsymbol{\mathcal{G}}\to Set$, which satisfies all the constraints specified by $\boldsymbol{\mathcal{D}}^B(\Theta)$, where: $\mathfrak{B}(W)$ is a set of possible worlds; $\mathfrak{B}(R)$ is a set of arcs that define the accessibility relation on the set of worlds $\mathfrak{B}(W)$; each $\mathfrak{B}(R_i)$, $1\le i\le n$, is a set of arcs that define the accessibility relation on the set of worlds of agents of type Ag_i; $\mathfrak{B}(Ag)$ is the set of agents involved in the model; each $\mathfrak{B}(Ag_i)$, $1\le i\le n$, is the subset of agents of type Ag_i; $\mathfrak{B}(Ac)$ is the total number of actions; each $\mathfrak{B}(Ac_i)$, $1\le i\le n$, is the subset of actions of type Ac_i. By the functor $\mathfrak{B}:\boldsymbol{\mathcal{G}}\to Set$, we understand its image in the Set category, which represents a behavioural model of the multiagent system. We denote the set of these models by $Mod(\boldsymbol{\mathcal{S}}^B, Set)$.

3.3 Aggregation of the Two Models

As we saw in the previous sections, both the static and the behavioural dimensions can be specified, at the metamodel level, through appropriate categorical sketches.

The static dimension of the categorical model is represented, at the metamodel level, by a categorical sketch $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}},\boldsymbol{\mathcal{D}}(\Theta))$, which specifies the set of possible states of the system that we denoted with $Mod(\boldsymbol{\mathcal{S}}, Set)$. The transition from one state to another of the model is done by the actions of the agents involved in this model. If the initial state of a model is $\mathfrak{I}_0,\in Mod(\boldsymbol{\mathcal{S}}, Set)$ then the set of possible states $\mathfrak{I}$ of the behavioural model becomes:

$\mathfrak{I}=\{\mathfrak{I}_k\in Mod(\boldsymbol{\mathcal{S}}, Set),\ k\ge 1\ |\ \mathfrak{I}_k$ is the result of successive actions of the agents on $\mathfrak{I}_0\}$.

The behavioural dimension of a multiagent system is represented, at the metamodel level, by the categorical sketch $\boldsymbol{\mathcal{S}}^B=(\boldsymbol{\mathcal{G}}^B,\boldsymbol{\mathcal{D}}^B(\Theta))$, as we saw in the previous section.

Now we can aggregate the two models to specify the categorical model of a multiagent system. A categorical metamodel for multiagent systems is a tuple $MM=(\boldsymbol{\mathcal{S}},\boldsymbol{\mathcal{S}}^B)$, where $\boldsymbol{\mathcal{S}}$ is a categorical sketch that represents a metamodel for the static dimension of the multiagent system, and $\boldsymbol{\mathcal{S}}^B$ is a categorical sketch that represents a metamodel for the behavioural dimension of the multiagent system.

A categorical model of a multiagent system is a tuple: $\boldsymbol{\mathcal{M}}=(\mathfrak{I},\mathfrak{B},\mathfrak{I}_0,\boldsymbol{\mathcal{P}},\pi)$, where $\mathfrak{I}\subseteq Mod(\boldsymbol{\mathcal{S}},Set)$, is a set of static models of the sketch $\boldsymbol{\mathcal{S}}$; $\mathfrak{B}\subseteq Mod(\boldsymbol{\mathcal{S}}^B,Set)$, is a behavioural model of the $\boldsymbol{\mathcal{S}}^B$ sketch, with the property that $\mathfrak{B}(W)=\mathfrak{I}$; $\mathfrak{I}_0\in\mathfrak{I}$, is the initial state of the system; $\boldsymbol{\mathcal{P}}$ is a set of atomic propositions, which characterizes the state of the system, and $\pi:\mathfrak{I}\to 2^{\boldsymbol{\mathcal{P}}}$, is an evaluation application, which associates to each state the set of valid atomic propositions in that state.

In this paper, we present only the general idea that is the basis of the specification of the logical language that specifies the behaviour of the agents without going into details. Each agent acts with the aim of achieving an objective. The objectives of an agent are specified by logical formulas, which it can satisfy through the actions it is able to perform. The agent pursues the satisfaction of some objective formulas that are not satisfied in the current world but can be satisfied after performing some actions, that is, in the following worlds.

For each type of Ag_i agent, there is a subset $\mathcal{P}_i$ of atomic formulas that such an agent can satisfy through its actions. Therefore, the set of logical formulas that a type of Agi agent can satisfy will all be well-formed formulas, starting from the set of atomic formulas that it can satisfy using logical operators. In a similar way, the logical formulas that can be satisfied by a group of agents can be specified.

4 Conclusions and Future Work

The most important conclusion of the paper is the fact that the categorical sketch is an appropriate mathematical mechanism for specifying models for multiagent systems, at the metamodel level. This metamodel can be the basis of the implementation of a domain-specific modelling tool endowed with a diagrammatic language.

Constraints on models can be specified by universal properties in category theory, which can be implemented in the form of generic algorithms that work on all models specified by categorical sketches.

I mention the fact that in this paper I did not deal with the problems related to parallelism and synchronization. Of course, the agents' actions can take place in parallel with the sharing of common resources, but this will be the subject of another paper.

In the semantics based on the Kripke model, the accessible worlds are fixed. In a reactive Kripke model [19], the evaluation of logical operators can cause the reconfiguration of the model in which the formula is evaluated. In our model, graph transformations can be used to model this reconfiguration, but this will be the subject of another paper.

References

[1] S. Franklin, A. Graesser, "Is it an agent or just a program?" In: Proceedings of the 3rd International Workshop Agent Theories, Architectures, and Languages. Springer, New York (1996)
[2] M. Wooldridge, "An Introduction to MultiAgent Systems", 2nd Edition, John Wiley & Sons Ltd 2009.
[3] Uwe Wolter, Zinovy Diskin, "The Next Hundred Diagrammatic Specification Techniques, A Gentle Introduction to Generalized Sketches", 02 September 2015 : https://www.researchgate.net/publication /253963677.
[4] Zinovy Diskin, "Tom Maibaum- Category Theory and Model-Driven Engineering: From Formal Semantics to Design Patterns and Beyond", ACCAT 2012

[5] Michael Barr, Charles Wells, “Category Theory For Computing Science”, Reprints in Theory and Applications of Categories, No. 22, 2012.
[6] Robin Milner, “The Space and Motion of Communicating Agents”, Cambridge University Press, 2009. ISBN 978-0-521-73833-0
[7] Hartmut Ehrig, Claudia Ermel, Ulrike Golas, Frank Hermann, “Graph and Model Transformation General Framework and Applications”, Springer-Verlag Berlin Heidelberg 2015
[8] Olivier Gasquet, Andreas Herzig, Bilal Said, François Schwarzentruber, “Kripke’s Worlds, An Introduction to Modal Logics via Tableaux”, Springer Basel AG 2014.
[9] J. Luo, JJ. Meyer, M. Knobbout, “Reasoning About Opportunistic Propensity in Multi-agent Systems.” In: Sukthankar, G., Rodriguez-Aguilar, J. (eds) Autonomous Agents and Multiagent Systems. AAMAS 2017. Lecture Notes in Computer Science(), vol 10642. Springer, Cham. https://doi.org/10.1007/978-3-319-71682-4_13
[10] G. Campbell, “Algebraic graph transformation: A crash course”, Department of Computer Science, University of York, UK, Tech. Rep., 2018. [Online]. Available: https://cdn.gjcampbell.co.uk/2018/Graph-Transformation.pdf.
[11] G. Campbell, B. Courtehoute and D. Plump, “Linear-time graph algorithms in GP2”, Department of Computer Science, University of York, UK, Submitted for publication, 2019. [Online]. Available: https://cdn.gjcampbell.co.uk/2019/Linear-Time-GP2-Preprint.pdf.
[12] David I. Spivak, “Category Theory for the Sciences”, The MIT Press Cambridge, Massachusetts London, England, 2014 Massachusetts Institute of Technology
[13] J. Ferber, O. Gutknecht and F. Michel, „From Agents to Organizations: An Organizational View of Multi-Agent Systems”, In Proc. 4th International Workshop on Agent-Oriented Software Engineering (AOSE), volume 2935 of LNCS, pages 214–230. Springer, 2003.
[14] Virginia Dignum, “Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models”, Information Science Reference, 2009 by IGI Global.
[15] M. Huth, M. Ryan, “Logic in Computer Science, Modelling and Reasoning about Systems”, Published in the United States of America by Cambridge University Press, New York , 2004.
[16] Y. Shoham, K. Leyton-Brown, “Multiagent systems_ algorithmic, game-theoretic, and logical foundations” -Cambridge University Press, 2009.
[17] Nino B. Cocchiarella and Max A. Freund, “Modal Logic, An Introduction to Its Syntax and Semantics”, Published by Oxford University Press, Inc. 2008
[18] D.C. Crăciunean, D. Karagiannis, “A categorical model of process cosimulation”, Journal of Advanced Computer Science and Applications(IJACSA), 10(2), (2019).
[19] D. Gabbay. “Reactive Kripke semantics”, In Proceedings of CompLog 2004, W. Carnielli, ed., pp. 7–20. Centre of Logic and Computation, University of Lisbon, 2004.