https://doi.org/10.31449/inf.v45i2.2967

Towards a Formal Multi-Agent Organizational Modeling Framework Based on Category Theory

Abdelghani Boudjidj
Ecole nationale Supérieure d'Informatique (ESI), BP 68M, 16270, Oued-Smar Algiers, Algeria
ICOSI Lab University, Abbes Laghrour khenchela BP 1252 El Houria 40004 Khenchela, Algeria
E-mail: a_boudjidj@esi.dz

Elkamel Merah and Mohammed El Habib Souidi
ICOSI Lab University, Abbes Laghrour khenchela BP 1252 El Houria 40004 Khenchela, Algeria
E-mail: ekmerah@gmail.com, souidi.mohammed@univ-khenchela.dz

Keywords: multi-agent systems, organizational models, category theory, formal semantics

Received: September 23, 2019

Multi-agent organizational modeling frameworks can be considered as an efficient solution regarding the distributed applications' problems such as task bundle mechanisms, supply chain management, as well as air traffic control. The main objective of this paper is to provide a solution based on a solid mathematical theory for the modeling, the analysis and the verification of artificial organizations properties, and particularly those of Multi-Agent Systems (MASs). Category theory is a mathematical formalism used to categorically study the logics of organizations in different societies. Therefore, it can be projected on artificial organizations in a categorical way. Our work is revolved around the idea of modeling Multi-Agents organization using category theory. In other words, it consists on the transformation of Agent-Group-Role (AGR) organizational model in a categorical way in order to obtain a formal semantics model describing the MAS organization. This formal model allows the analysis, the verification and also the validation of the main concepts of an organization at a high level of abstraction.

Povzetek: Predstavljena je matematična formulacija agentnih sistemov na osnovi kategorij.

1 Introduction

Multi-agent systems (MASs), in particular, organizational modeling frameworks represent an advanced technology for modern applications that are dynamic, open and distributed. The development of these applications requires the exploration of the related design and analysis methods.

MASs are useful for the modeling and development of distributed computer systems; they have emerged to solve the problems of organizations in large-scale software systems.

Their characteristics allow a better structuration of complex systems, a wide use of MASs in several areas of computer and engineering, such as e-commerce, e-learning, communication, data mining, simulation, robotics, system transport, grid computing …

An organization is a social entity that has a specific number of entities (people, a computer system or an institution). The main objective of the organization is to permit the coordination of these entities. Each member or entity recognizes its role and those of the others, in order to achieve a collective goal [1].

The organization is supposed to have structural and strategic components. These two components are linked, an organizational strategy related to characteristics such as size, innovation, versatility and geographical distribution of the organization. These characteristics imply different coordination mechanisms. Different organizational structures are trained to ensure the coordination mechanisms, and the dependence of different types of tasks [2].

Several organizational models of multi-agent systems have been proposed. They are based on the social structure and organizational concepts for the complex systems construction, in order to propose a solution to the problem of the heterogeneity of languages.

The Organizational MASs Model transformation is an adaptation (making connections between categorical concepts and units of an organization MAS such as AGR), and a way of categorically describing an artificial model, which allows us to have a mathematical model. This model allows us to perform the analysis and the formal verifications of certain properties of organizational MASs.

Many studies have addressed organizational models in MASs, such as the AGR model (Agent Group Role), representing the former Aalaadin model evolution proposed in [3].

In this model, the organization is defined as a framework of activities. The interactions are based on group agents' notions, roles and their relations. The AGR model focuses on the organization structure and both groups and roles definitions, this model does not deal with the internal architecture of the agent but it focuses on the function of this agent within an organization (its role) [3].

We can cite three extensions of AGR organizational model, the model Agents Group Role Environment [4], known as AGRE, which takes into consideration the physical and social environments placed in domains, known as spaces.

The second extension is the Agent Group Service Role (AGSR) organizational model [5], represents another conceptual and architectural framework for organizing open and dynamic multi-agent systems.

And finally, the Agent-Group-Role-Membership Function (AGRMF) [6]. In this model, the agent is an entity capable of acting and communicating. The group is equipped with a membership function managing its flexibility (access of the agent to the group) and also represents the primitive notion of the combination of the agents.

There exist other models such as, Model of Organization for multI-agent SystEms (MOISE). Furthermore, The Roadmap organizational model is more similar to agent-oriented methods, which is based on a role, a protocol and an interaction model [7].

Organizational Model for Normative Institutions (OMNI) [8], the organizational model of YAMAM [8], based on four different concepts: Agent, Competence, Role and Task.

As a first approximation, we can say that the category theory is the mathematical study of function algebras, arises from the idea of functions system between certain objects [9]. The category theory was introduced and used as a framework in many cases. In many fields of computer science and software engineering [10], it has a rich body of theory for reasoning about structures (objects and relations between objects), and it is sufficiently abstract to represent a wide range of different specification languages.

In this paper, we are interested in modeling organizational multi-agent systems (MASs), which represent the backbone of our research activity, by using category theory. Primarily focusing on organization-based methods, these methodologies are still fairly new, and mainly focused on the analysis phase. Otherwise, the design and implementation phases are either missing or redirected to agent-based methodologies, which do not offer enough tools to model organizational concepts. Therefore, there is still a gap between the analysis and the design, which must be clearly, correctly, and completely specified.

In our work we will use category theory to interpret MAS organizational models. In other words, our objective is to transform and to validate organizational concepts of AGR (semi-formal model) to the categorial formal model with category theory. The following diagram details the different stages of our approach.

After the exploration and the Analysis of the MAS organizational concepts, we have seen the usefulness in the transformation of AGR organizational model. Once our objective is fixed, we proceed to its categorical modeling. A categorical modeling aims to construct categories for AGR model with its organizational concepts (Agents Groups and Roles) and the relation between them. We study and validate the different proprieties emerging for the agent ’interactions in the categorical model. If the properties of the chosen model are not satisfactory, it is necessary to return to the formal model.

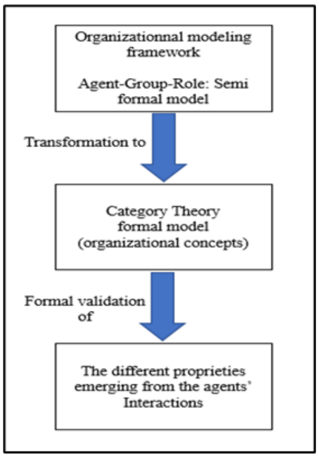
Organizationnal modeling framework
Agent-Group-Role: Semi formal model
Transformation to
Category Theory formal model (organizational concepts)
Formal validation of
The different proprieties emerging from the agents' Interactions

Figure 1: Diagram of different stages of our approach.

Firstly, we present category theory. The category is a bunch of objects. These objects are linked together with arrows known as morphisms. It can be compared to a set of objects but a category is larger than sets.

A morphism or arrow can be defined as a link between two objects. So an object A, object B and an arrow between them called path.

An object can be considered as primitive in this theory. It has no properties and does not have an internal structure. An object can be compared with an atom (it is like a point without properties). A morphism R is also a primitive, it has no properties except that each arrow has a beginning and an end. In fact, the reason of having objects is to mark the ends of the arrows (morphisms). In other words, if we have objects *a* and *b*, we can have zero or more arrows linking between them [11].

For each pair of objects, we need to specify the arrows that link between them with different names, some objects are not connected with arrows, other are connected with an arrow, and other objects are connected with an infinite number of arrows. The number of arrows can be an incalculable linking from two objects to the others. There exists a morphism of identity for each object in a category.

Our contribution will be reflected by the introduction of the composition with the aim of obtaining a composition table that composes the morphisms, and provides different categories [11].

Category theory is general and aims to unify mathematical modeling languages that provides many universal building principles [12]. This framework offers a structure for formalizing large specifications and provides primitive compositions in algebraic languages and specification languages of temporal logic [13].

There exist different types of categories such as:

- Set Category, in which the objects are considered as sets. The arrows or morphisms linking between sets A and B represent all functions from A to B [14].

- Discrete Category, is a category where morphisms are only morphisms of identity. For example, we suppose that X and Y are different objects in category C, the morphism of X to X can only be the morphism of identity of X, and the morphism of X to Y does not exist, which means :$(X, X) = \{id_x\}$ for all X objects, and mor$(X, Y) = \emptyset$ for all $X \neq Y$ objects [15].

- Path Category, before presenting this Path we need to have basic knowledge about directed graphs. A directed graph G is a set O of objects called vertices or nodes, and a set A of ordered pairs of vertices are called arrows or directed edges [16]. Each arrow diagram or directed graph can be interpreted as a category named Path, whose morphisms are sequences (Paths) of arrows.

Other types of examples that we often see in mathematics are the categories of structured sets. In other words, sets with another "structure" and functions which "preserve" them, where these notions are determined independently such as: groups and group homomorphisms, vector spaces and linear maps, graphs and homomorphisms of graphs, the real numbers $\mathbb{R}$ and the continuous functions $\mathbb{R} \to \mathbb{R}$, the open subsets $U \subseteq \mathbb{R}$ and the continuous functions $f: U \to V \subseteq \mathbb{R}$ defined on them, topological spaces and continuous maps, differential manifolds and smooth maps, the natural numbers $\mathbb{N}$ and all recursive functions $\mathbb{N} \to \mathbb{N}$, or as in the example of continuous functions, we can take partial recursive functions defined on the subsets $U \subseteq \mathbb{N}$.

The paper is organized as follows: in the first section, we will discuss the use of Category Theory (CT) in relation to the multi-agent systems. In the second section, we will describe the problem, in which we detail the different concept forming AGR organizational model, and also the concept of category theory. The third section is devoted to the AGR model interpretation via the utilization of category theory. In the fourth section, we will present a case study for the validation of the categorical model of the AGR organization. In the final section of this paper, we will conclude by brief notes reflecting the importance of the proposed work.

2 Related work

Several research activities are based on categories theory and their mathematical concepts at a very abstract level, in agent-oriented or organization-oriented multi-agent systems.

In a recent work [17], they studied human organization using the category theory in a philosophical way. They also represented the human society concepts via the category theory concepts, and put links between the two concepts. According to this work, the categories theory with a high level of abstraction allows the formalization of social (collective) models within the framework of social theory in order to explore their interaction, to express the organizational concepts of MASs and software components with the same categorical terms and also their interactions. Category theory helps in organizational models' validation. Once the MAS organizational categorical model is generated, it is possible to analyze and study some properties of organizations.

In [18], they proposed a high-level agent-oriented modeling. The authors have categorically detailed the structure of a Belief Desire Intention (BDI) agent, through the modeling of these concepts. After a design and modeling of an Autonomous Reactive System (ARS) by a multi-agent system, the Category Theory will be applied to formalize the autonomous behavior of MAS and also that of the ARS. The mapping of ARS and MAS via category theory allows the formalization of the autonomous behavior of these lasts with the aim of proving that the two categorical representations mapped from ARS and MAS are isomorphic. This step will guarantee that the autonomous behaviors of ARS and MAS are the same. This work presented the possibility of coupling MASs and CT.

In [19], the authors introduced a MAS categorical generic model, leading to the MAS category. In this category, objects are agents of different types and the morphisms represent all kinds of relations between agents, called communication and general cooperation arrows. This general structure of communication and cooperation is represented by a corresponding arrow diagram, called Basic Diagram of MAS. The first formulation of the idea is based on the categorical modeling of relations such as the category Path(X, R), a natural description of the basic diagram of a MAS in categorical notions arises.

In [20], they introduced a formal verification of a concurrent system based on category theory. In other words, they managed the consistency between design and implementation in the phase of a concurrent system's development.

As a first step, they designed the concurrent system using communicating sequential processes (CSP). The next step focused on the implementation of the concurrent system with a process-oriented programming language known as Erasmus.

They categorically verified the implementation against the design of the concurrent system as a last step.

In [21], the authors proposed a formal approach, named Reactive Autonomic Systems Framework (RASF), based on category theory, to face the challenges such as group behavior that emerges complex and unexpected, which be in need of a formal specification and verification, they focus on the formal specification of substitutability property for the fault-tolerance, their approach was illustrated via Mars-world case study.

They used RASF to model the Reactive Autonomic Systems (RAS). They built a RAS meta-model with categorical specification as a first step. Secondly, they transformed it to be applicable in the case of exploring the Mars-world. Finally, they implemented these RAS models through the Multi-Agent Systems (MAS) using JADEX, where they have five types of agents (robots) as Manager, Supervisor, Sentry, Producer and Carry agents. They have proved in this work that category theory is largely used to capture constructs and their relations in a system, in a formal way and in a single categorical presentation.

In [22], the contribution is reflected through the construction of a transformation system for Multi-Agent System (hereafter, MAS) modeled and based on category theory as a common linguistic formal support. They introduced the Category MAS of all Multi-Agent Systems, the objects represent agents and morphisms represent the relations between the agents. The relations between agents can change according to the MAS dynamism. Therefore, it is justifiable to define the Category MAS of All MASs, where the objects are Multi Agent Systems and the morphisms are MAS's morphisms. They associate a base diagram to each MAS representing the complete relational structure. The Double Pushout Method (hereafter, DPO), a concept widely developed in the field of algebraic graph transformations, they proposed a MAS semantic to the DPO approach. They proved in this work that there is a strong need for formalization of MAS, and their goal was to develop a toolbox for MAS modeling using category theory notions.

3 Problem description

The use of organizations in multi-agent systems is an attractive research activity requiring a formal framework to mathematically manage it, and proceed to the validation of the different properties.

Several models have been realized in order to reflect the importance of organization in multi-agent systems, and to lead to efficient solutions regarding the complex problems. New concepts have been used in MASs according to the proposed model such as AGR, which we are going to detail.

The category theory provides many mathematical aspects and concepts at a very abstract level. This fact will allow us to represent or transform the concepts related to organizational MAS. This formalization will lead us to a categorical model for the AGR model (formal model).

Also, category theory will allow us to examine the AGR in an abstract way by formalizing the system as collections of objects (categories) and morphisms with the aim of reasoning about these objects and their relations or interactions (morphisms). This point is very suitable for agent-based systems like organizational MAS. The formal obtained model will lead itself to formal verifications of interesting properties of organizational SMA.

Several studies have presented the possibility of using CT in relation to MAS, and consequently, the modeling of the AGR model (Organization Oriented).

The problem studied is to reformulate AGR organization with CT and to validate this approach. An instance of market organization is used in order to clarify the use of AGR model, to find an appropriate supplier among the set of participating suppliers to the market. Specifically, this case study will be used to test our AGR formalization where groups and roles will interact and communicate with each other. The agents engaged and involved in the market will proceed to negotiation in order to appoint the supplier. Properties will be presented in this example such as the flexibility of agents, negotiation between agents playing the roles, as well as an agent playing one or more roles.

3.1 An agent in agent group role (AGR) model

This model represents the agent as an autonomous entity that can communicate with other agents, and can play one or more roles in one or more groups. No constraint is placed on the internal architecture of the agent to allow each designer to choose the adequate agent according to the processed problem. A simple semantic structure formula of Agent $(A)$is a tuple:

$A: < Id, N, PLs >$

Id: The Agent Identity.

N: Agent Name.

PLs: roles played by the agent.

3.2 Group in agent group role (AGR) model

The group represents a set of agents that shares characteristics, such as usual MAS, where each agent can be a member of one or more groups. The group is used to divide the organization where each group has an activity; and any agent can start a group, a semantic structure formula of a group (G) represented as:

$G < Gn, r >$

Gn:group name,

r: resident roles in the group.

Figure 2 represents a class diagram of AGR.

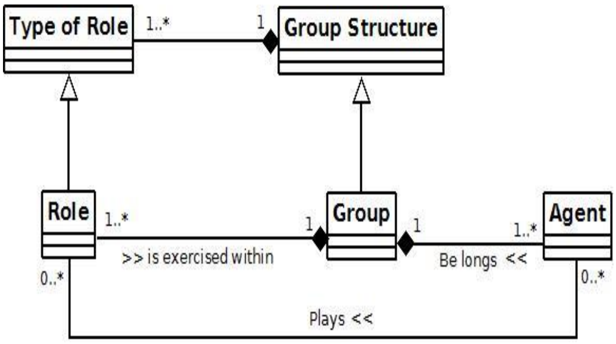

Figure 2: AGR meta-model.

3.3 Role in agent group role (AGR) model

The Role can be considered as the abstract representation of the function of an agent within a particular group. A Role is local in a group where an agent can have one or more roles. A Role is a set of tasks, and in turn a task is a set of actions. The Role (R) is given by a semantic structure as follows:

$R: < Rn, Gn >$

Rn: role name

Gn: group name where role reside.

3.4 Category theory concepts

3.4.1 A category

Includes the following data [14],

- *Objects*: A, B, C, etc.
- *Morphisms* (Arrows): $f, g, h,$ etc.
- Domain (dom) and Codomaine (cod): For each arrow f, we give objects: $dom\ (f)$, $cod(f)$ called domain and codomaine of f, we write:

$f: A \to B$ to indicate that$A = dom(f)$ and $B = cod(f)$.

- Composition($\circ$): From the arrows $f: A \to B$ and $g: A \to C$, that is to say with: $cod(f) = dom(g)$, we have a given arrow: $g \circ f$: A → C.
- Identity: For each object A there is a given arrow $Id_A: A \to A$, called identity arrow of A.

These components are required to comply with the following laws:

Associativity: $h \circ (g \circ f) = (h \circ g) \circ f$, for all $f: A \to B$, $g: B \to C$, $h: C \to D$.

Unit: $f \circ Id_A = f = Id_B \circ f$, for all $f: A \to B$.

3.4.2 Isomorphism

In any category C, an arrow $f: A \to B$ is called an isomorphism if there is an arrow $g: B \to A$ in Csuch that $g \circ f = Id_A$ and $f \circ g = Id_B$.

Since inverses are unique (proof), we write $g = f^{-1}$. We say that A is isomorphic to B, written $A \approx B$, if there exists an isomorphism between them [9].

3.4.3 Functor

Definition of a functor [9]:

$F: C \to D$ between categories C and D is a mapping of objects to objects and arrows to arrows, in such a way that:

(a) $F(f: A \to B) = F(f): F(A) \to F(B)$,

(b) $F(Id_A) = Id_{F(A)}$,

(c) $F(g \circ f) = \mathrm{F}(g) \circ F(f)$.

That is, F preserves domains and codomains, identity arrows, and composition.

A functor $F: C \to D$ thus gives a sort of picture —perhaps distorted—of Cin D.

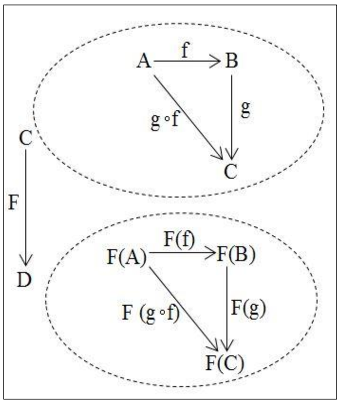

Figure 3: Ffunctor from category $C$to category D.

3.5 Tool supporting the approach

Category theory is known for its ability to organize the key abstractions that make up most areas of mathematics, and it becomes useful for writing elegant and maintainable code through categorical ideas. Haskell is a programming language used to stimulate the construction of category theory (www.haskell.org).

4 Interpretation of agent group role in category theory

In this AGR model, the organization represents the AGR itself including all the existing groups. A group represents a set of agents and roles. The AGR model gives us the possibility to choose the type of appropriate agent (reactive, interactive, cooperative ...) according to the system or the addressed problem.

Our work focuses on a formal transformation of a specific organizational model. Therefore, we have based our work on reactive agents. An agent is represented by a set of goals, skills, and characteristics. The goals that the agent works for their accomplishment are the performance of a set of actions and also the play of one or more roles. Agents may have new goals after a system update causing new actions to be performed and roles to be played.

On the one hand, the role is a sequence of tasks to be performed by one or more agents (playing the same role). On the other hand, a task is a sequence of actions to be executed in a discrete-time.

The units of the AGR organization (agent groups) can communicate with each other via messages (send and receive). The exchange of messages can provoke new goals and events.

The use of category theory allows us to focus on the morphisms or relations between objects instead of focusing on the internal representations of objects. In the AGR, as it is mentioned in section problem description, each group contains a set of roles played by the belonging agents. The agents belonging to the same group cooperate with each other in order to perform different tasks.

In a group of agents, there exist two different types of communication: local communication and global communication. Local communication occurs only within a group, and worker agents communicate with each other to cooperate. Global communication occurs between agents of different groups.

In this section, we will present the categorical modeling of AGR. We will examine the agent structure and role. Moreover, we will represent the main concepts such as Actions, tasks, role, groups, agent, and their relations via category theory. Then a return on the global system, which is the set of groups, the wholes represents the AGR with the help of constructions of category theory.

The following section contains definitions of the different concepts allowing the formalization of the groups in an AGR organization.

4.1 The role

A role is represented by a sequence of tasks to be sequentially executed. In turn, a task is represented by a set of actions sequentially executed in a discrete-time, presented in a Task category. So, to represent the tasks and their big category TASK, we will first define the category ACTIONS.

4.1.1 Category ACTIONS

ACTIONS is a category that contains all the required actions to perform in order to build a task, objects, and morphisms in that category. These proprieties are defined as follows:

- Objects: are a set of executable actions designated by $A1$, $A2$, ...
- Morphisms: morphisms of identity

ACTIONS is a discrete category that contains identity morphisms. Actions are defined as an abstraction of agents' reaction to environmental events. The following figure shows the ACTIONS category.

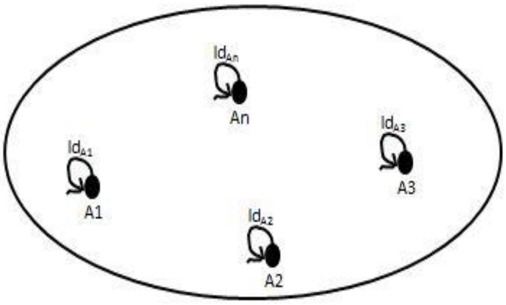

Figure 4: ACTIONS Category.

4.1.2 Category Task

Task is a category representing the sequence of actions where objects are of type $A1$, $A2$ ... and morphisms are the morphisms of object identities, and morphisms of type Before, this type of morphism ensures the order of execution of the actions.

An object named A_{null} means a null action. This type is used to capture exceptions, and it only has the identity morphism. The order of executions of the actions is interesting in this category and by obligation the morphisms of length one, will be accepted ensuring the partial order of execution of the actions. The Figure 5 shows an example of a Task category

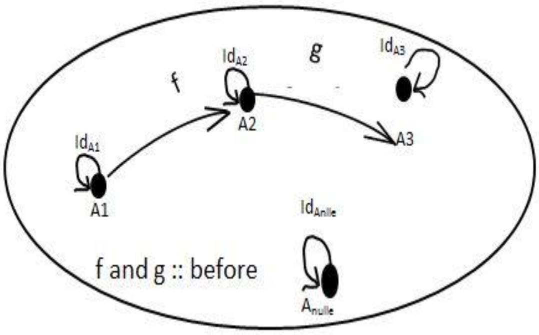

Figure 5: ACTIONS Category.

Where:

Objects: $A1$, $A2$ and $A3$, represent the execution of tasks $A1$, $A2$ and $A3$ respectively.

Morphism: morphisms of identities of objects (Id).

Morphisms between objects of type $(::)$ *before*, the morphism before assures us that the action $A1$ executed during t_1 before $A2$ which is executed in t_2, and vice versa.

The task category is represented by a special category called *Path* which accepts morphisms (arrows), of length one.

In the following figure, we will showcase the non-displayed path of length equal to 2, which is the composition of the two morphisms f and g. As we have noted before, the category theory is based on the composition of the morphisms between the objects and the composition of the functors between the categories.

Morphism before $:: g \circ f = h: A1 \rightarrow A3$. The length of this morphism is equal to 2, it will not be displayed.

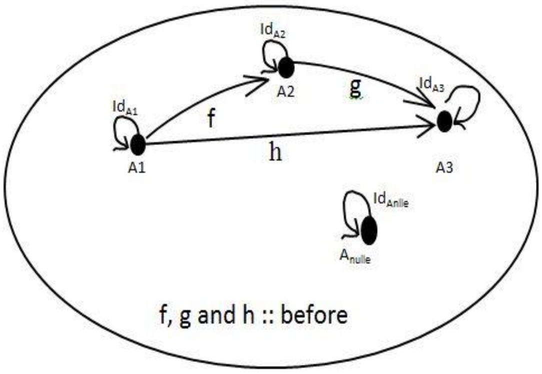

Figure 6: Task Category without the rule application of the PATH category

The morphism h represents the composition of the two morphisms f and g satisfying the laws of associativity and unity. Therefore, the validity of the Task Category is proven.

4.1.3 Functor Sequence_A

This functor maps objects and morphisms from the category ACTIONS to a category as follows:

- Objects: actions in the ACTIONS category maps to the Action objects in the Task Category.
- Morphisms: all the morphisms of identities of the objects $A1$, $A2$, ..., of the category ACTIONS towards the objects $A1$, $A2$, ..., of the category Task.

The figure 7 shows the categories ACTIONS, Tasks of type Task, and functors of type *Sequence_A.*

The preceding diagram presents two categories of the type Task, which were injected from the *Category ACTIONS* by the functors F and G of type *Sequence_A*,

F Functor:

Objects

$F(A1) = task1.A1$

$F(A2) = task1.A2$

$F(A3) = task1.A3$

Morphisms

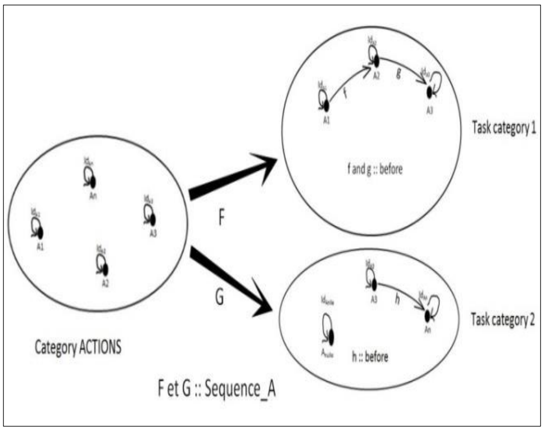

Figure 7: Diagram shows an example of two categories of tasks extracted from the ACTIONS category by two functors.

$F(Id_{A1}) = Id_{F(A1)}$
$F(Id_{A2}) = Id_{F(A2)}$
$F(Id_{A3}) = Id_{F(A3)}$
G Functor:
Objects
$G(A3) = task2.A3$
$G(AN) = task2.AN$
Morphisms
$G(Id_{A3}) = Id_{G(A1)}$
$G(Id_{AN}) = Id_{G(AN)}$

4.1.4 Category TASK

The *Category TASK* represents a large category where, the objects are: Tasks of the categories of the type Task x, $(x = 1\ or\ 2\ ...)$.
Morphisms are: the morphisms of identities.

4.1.5 Functor collapses_T

Functor that collapses a category of type Task to a T object of the big *Category TASK*:

All the objects of a category of the type Task x, will be collapsed towards an object T in the Category TASK, and all the morphisms will be also collapsed towards a single morphism of identity. The object T of Category TASK will represent a black hole collapsing a whole category of Task x type into a single point.

The figure 8 shows an example of *Task* type categories that would melt into T objects of the large *Category TASK*.

Example functor collapses_T (H):
Objects
$H(task1.A1) = T1$
$H(task1.A2) = T1$
$H(task1.A3) = T1$
Morphisms
$H(Id_{A1}) = Id_{H(T1)}$
$H(Id_{A2}) = Id_{H(T1)}$
$H(Id_{A3}) = Id_{H(T1)}$
$H(f) = Id_{H(T1)}$
$H(g) = Id_{H(T1)}$

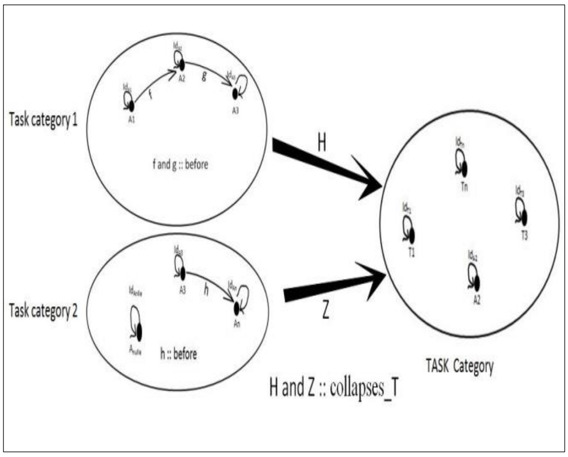

Figure 8: Collapse of *Task x* Categories in the major *Category TASK*.

4.1.6 The Category Role

The *Role* represents a sequence of *Tasks* to be executed sequentially from the large *Category TASK*. We can have several Categories *Role* from the *Category TASK*.

4.1.7 Functor Sequence_T

This Functor maps
Objects: T from *Category TASK* to *Role T* objects.
Morphisms: identity morphisms *Id* of *Category TASK* to identity morphisms *Id Role*.

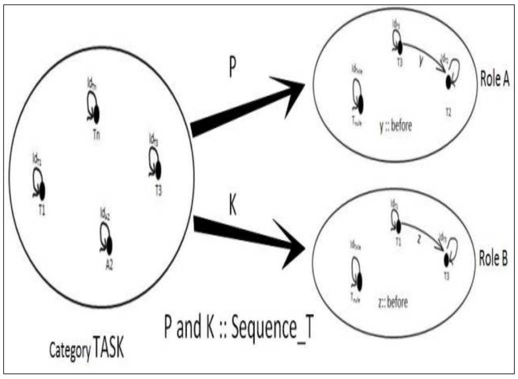

Figure 9: Extraction of *Roles* from the broad *Category TASK*.

4.2 Category Group

Category Group is a category that contains Roles residing in a Group x, with their relations,

4.2.1 Functor Collapse_G

It represents a functor collapsing a category of type *Role* x, in an object *R* of the *Category Group* x,
Objects: The *Role categories* to *R objects* of the *Category Group*.
Morphisms: the morphisms of identities of the *Role categories* towards the morphisms of identity of the objects of the *Category Group*.

All objects of a category of type *Role* x, would collapse to an *object* R in the *Category Group* x, and all morphisms would also collapse to a single identity morphism of the same object, the *object* R of *Groupe* x will also represent a black hole that will collapse an entire category of type *Role* x $(x = A \text{ or } B, \dots)$, into a single point that represents the object.

Figure 10 shows an example of *Role* type categories that would melt into R *objects* in the *Category Group* x.

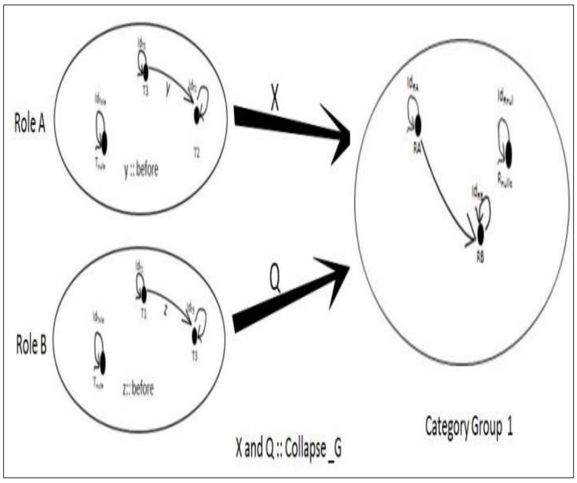

Figure 10: Collapse of Roles x categories in a Category Group x.

From the existing categories *Role*, we can have several categories *Group*. Every action depends on the *Roles* belonging to the *Groups*. This ensures that a *Role x* is resident in one or more *Groups*.

The diagram (figure 11) shows the categories: *ACTIONS*, *Task*, *TASK*, *Role* and *Group*

And the Functors: *Sequence_A*, *collapses_T*, *Sequence_T* and *collapse_G*.

4.3 Category Agents

In this work, the agent is a communicating entity capable of playing one or more Roles in one or more Groups, Category Agents is a category where:

The Agents are represented by the objects. Furthermore, the morphisms only reflect the identities of each object (Agent).

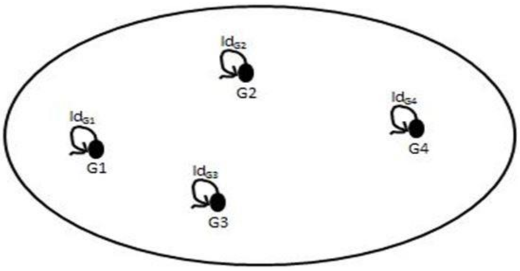

Figure 12: Category Agents.

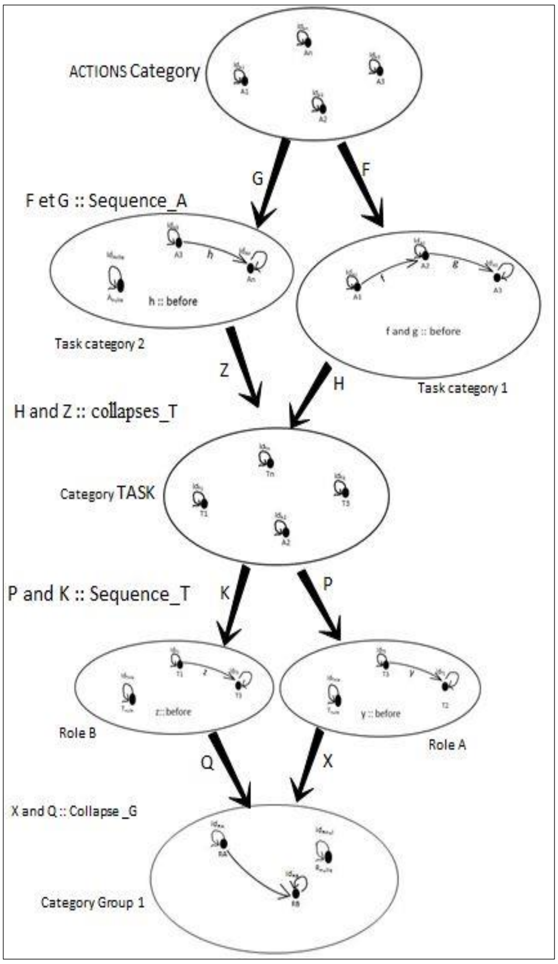

Figure 11: Representation of the categories studied and the relations between them (Functors).

4.3.1 Functor plays_role

This functor allows the Agent to play one or more Roles in a Group or Groups. It maps the objects from the Category Agents to the objects of the Category Group.

Morphisms: It maps the morphisms' identity "Id " of agents to the morphisms of identities of the roles in the group category.

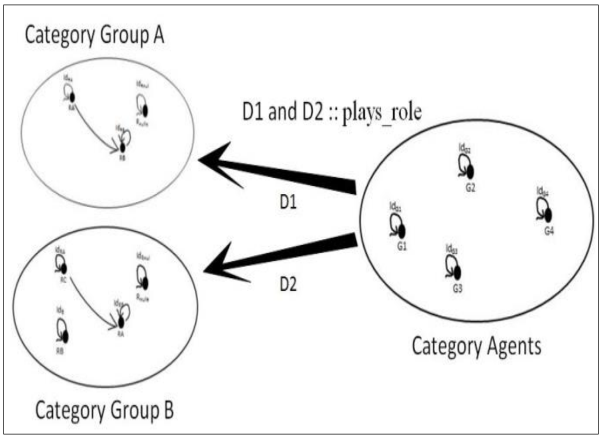

Figure 13: Relationship between Agents category and Groups categories.

The figure 13 shows the Agents category, the plays_role morphism, and the Group category.

We have presented all the categories that represent the AGR model: ACTIONS, Task, TASK, Role, Group and Agents, as well as the relations between them by the functors, Seqence_A, Collapses_T, Sequence_T, Collapse_G and plays_rôle. In Figure 14, an example of 3 Roles, Role A, Role B and Role C residing in two different Groups, Group A and Group B, and 4 Agents A1, A2, A3 and A4 playing these roles within the meaning of these groups. The example of Figure 14 shows us the important flexibility given to Roles and Agents by this categorical representation. A Role belongs to one or more groups. An Agent plays one or more Roles in one or more Groups representing the initial definition of AGR model.

5 Validation

We will validate now our interpretation of AGR organizational modeling framework in terms of category theory, through a case study of a market organization instance [3]. A set of agents represent the different elements forming the *market* structure. The Figure 15 shows the three groups in the *Market* organization, a *Group of Customers*, a *Service Provider Group*, and a *Contract Group*.

Group of Service Providers: in this group it resides the role *Broker* and *service*, the *broker* role is the representative of the *suppliers*.

Customer Group: Gathers the customers whose *customer* role we find, and to find a suitable *supplier*, *customers* interact with the *Service Provider Group* via *Broker*.

Group contract: in this group it resides the roles *seller* and *buyer*, its structure is made to bring together the agents who engaged and involved to move to the negotiation.

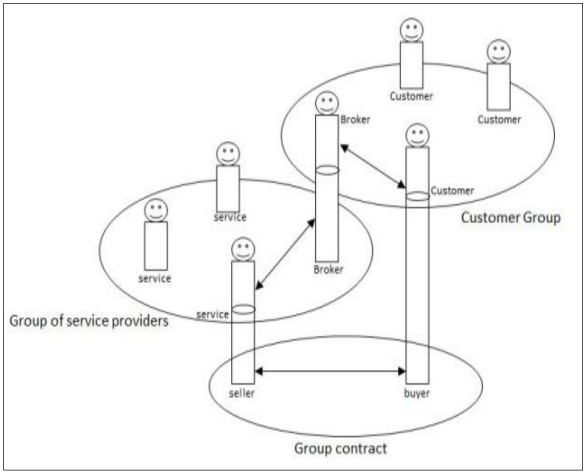

Figure 15: A market organization instance, [3].

In our categorical AGR model an agent can belong to both the client group and the service provider group (message passing). So, it is possible to play multiple roles in multiple groups by an agent and perform tasks. As shown in the last example, an agent will play the Broker role in two different groups.

The figure 16 shows a projection of the Market organization instantiated from the AGR categorical model, including the three groups with their roles.

The category Agents includes all the agents of our organization with the supposition that there exist 7 Agents. Three Agents will play the Role *Customer* in the *Category Customer Group* via the functor:

D1 :: *plays_role*, which maps the Agents with the Roles. Also, it linked one of the Agents with the *Broker* Role in the same category.

Another functor *D1* :: *plays_role*, links other Agents with *service* Roles in the *Provider Services Group Category*, and also maps the same Agent that plays the *Broker* role in the Category *Customer Group* to the *broker* Role of the Group category served customer he plays the same role in two different groups.

In the *Category Customer Group*, the *customers* are represented by the objects. This category contains the role, *broker* and also the roles *customer*. The morphisms between the role *broker* and the other roles *customer* represent the link between them, and among these links only one *customer* has a response from the *Broker* via a morphism back (an Isomorphic relation), the same thing in the group category served by providers links are generated via morphisms. The *contract group category* will bring together the Agents playing the roles of *seller* and *buyer* via the functor *D3* :: *plays_role.*

In this example we have not detailed the categories *ACTIONS* and *TASK*, we have assumed that the actions and tasks presented via each *Role* are predefined.

Each category has an identity morphism that ensures the update (id_update) of the category in case of change, the appearance of a *New Role* for example or one of the Agents does not want to play a *Role* and vice versa. This categorical transformation ensures the flexibility of Agents between Groups, which is a very important property in multi-agent organizational systems.

5.1 Comparison with the existing approaches

The table 1 contains the related works using category theory and multi-agent systems. Through the latter, a comparison study between these approaches was made according to a set of criteria such as the orientation of the multi-agent system (agent-oriented or organization-oriented), the type of agent used, the dynamism of the system, and the stability when changing the role (if one exists) of Agent.

5.2 Evaluation

The table illustrates the differences between the proposed approaches, for example, the approach of Olga Ormandjieva line 4, which is an agent-oriented based work, does not deal with the organization concept, which is the base of our approach. Otherwise, some approaches are based on the use of the organization especially in the MASs.

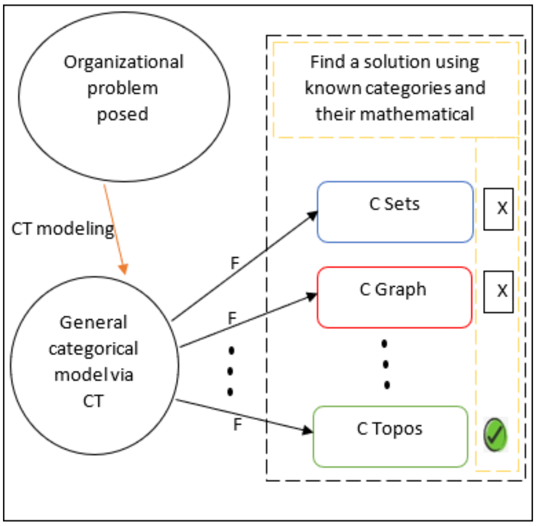
Organizational problem posed
CT modeling
General categorical model via CT
F
F
F
Find a solution using known categories and their mathematical
C Sets
C Graph
C Topos
X
X
✓

Figure 17: Using of Categorical model.

The work of Abderrahim. S, is oriented organization which has studied the link between category theory and human society, the type of agent used is generalized, while our work is based on the model of artificial organization AGR with a specific type of agents.

After the generation of the categorical model, the category theory allows us to switch to several known modeling modes such as graphs or sets. This very important link represented by a functor (F) allows us to switch between the mathematical representation, which give us the possibility to reformulate the studied problem via graphs, sets, topos, and this allows us to use the characteristics and properties of each domain to solve the starting problem (Figure 17).

So, if we categorically model a problem linked to an organization, we can solve it with the sets. If we will not arrive at a solution with these, we can change our mathematical tools and use the topos for example. For that, we just need to go back to our categorial model and change the functor to the topos. General categorical modeling gives us more mathematical choices for problem solving.

6 Conclusion

In this paper, we have proposed a formal semantics of AGR organizational modeling framework in terms of category theory, with its concepts and structures based on Agents, Groups and Roles. This formal semantics allows the analysis, the verification and the validation of the emergent properties.

The main objective is to use category theory to formalize the structures of the AGR as a toolbox in the

Table 1: Comparison the existing approaches in related work.

Authors	Titled work	Agent-centered MAS	Organization-centered MAS	Reactive Agent	Cognitive Agent	Dynamism	Stability
Pfalzgraf, J. 2005	On categorical and logical modeling in multiagent systems. Anticipative Predictive Models in Systems Science. [8]	✓	X	Not mentioned		X	X
Pfalzgraf, J. and T. Soboll,. 2009	On a general notion of transformation for multiagent systems and its implementation. [9]	✓	X	Not mentioned		✓	X
Kuang, H., et al. 2010.	Formal Specification of Substitutability Property for Fault-Tolerance in Reactive Autonomic Systems. in SoMeT. [10]	✓	X	X	✓	✓	✓
Olga Ormandjieva et al.,2015	Modeling multi-agent systems with category theory. [11]	✓	X	X	✓	X	X
Abderrahim, S. and R. Maamri, Dec,2018	A Category-theoretic Approach to Organization-based Modeling of Multi Agent Systems on the Basis of Collective Phenomena and Organizations in Human Societies. [12]	X	✓	Not mentioned		X	✓
Our work	Towards a formal multi-agent organizational modeling framework based on category theory.	X	✓	✓	X	✓	X

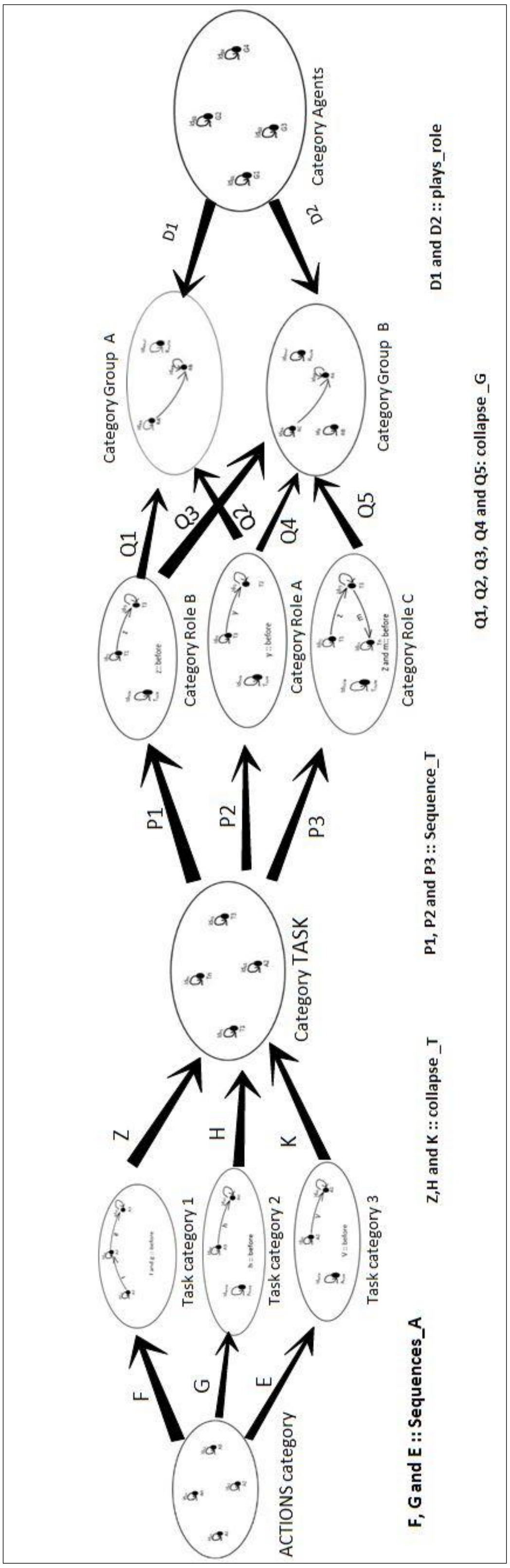

Figure 14: AGR model with Category Theory (categorical representation).

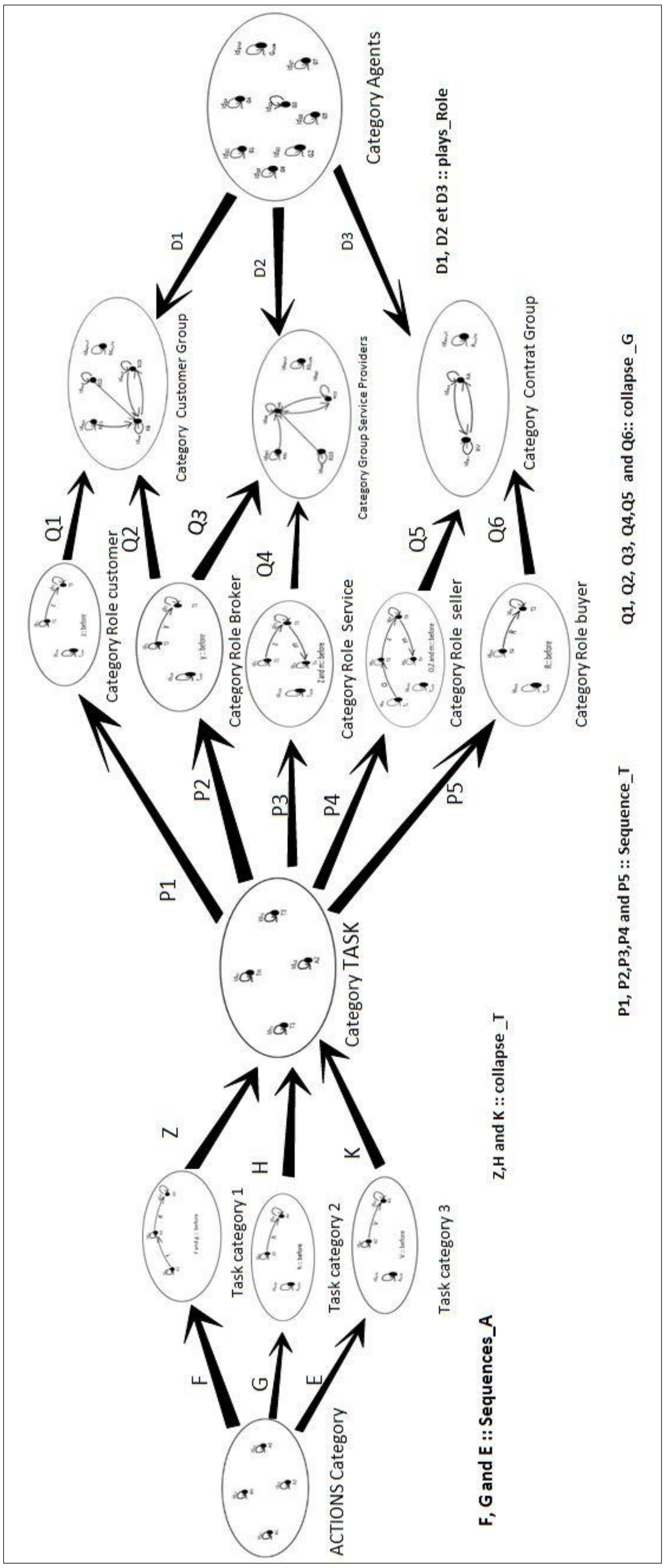

Figure16: Transforming the Market Organization Using the AGR Categorical Model.

field of MAS organization. The category theory allows us to formally achieve this goal in order to represent the different concepts in the same categorical framework. This representation is effectuated via the use of objects, morphisms and their compositions, and also functors between categories of the framework.

As shown in section 3, the different concepts of AGR can be expressed and combined. Our categorical interpretation allows us to check some properties of

MAS, such as flexibility of agents, stability of the groups, as well as the emergence of new roles. The validation of these properties will depend on the final result of the modeling and the processed problem.

In future work, we will focus on the modeling of AGR extensions, such as AGRE, AGRS and AGRMF, as well as other organizational MAS models. Moreover, we will try to establish a relation and a combination between agent-oriented modeling and organization-oriented modeling. Also, we plan to automate the transformation of AGR concepts in concepts of category theory by using Model-Driven Engineering (MDE) and Model Transformation.

References

[1] March, J.G. and R.I. Sutton, Crossroads—organizational performance as a dependent variable. Organization science, 1997. 8(6): p. 698-706. https://doi.org/10.1287/orsc.8.6.698

[2] Donaldson, L., The contingency theory of organizations. 2001: Sage. DOI: 10.1016/B978-0-08-097086-8.73110-2

[3] Ferber, J. and A. Gutknecht. *Un méta-modèle organisationnel pour l'analyse, la conception et l'execution de systèmes multiagents*. In Proceedings of Third International Conference on Multi-Agent Systems ICMAS. 1998. Un meta-modèle organisationnel pour l'analyse ... - LIRMM

[4] Ferber, J., F. Michel, and J. Báez. AGRE: Integrating environments with organizations. in International Workshop on Environments for Multi-Agent Systems. 2004. Springer. DOI: 10.1007/978-3-540-32259-7_2

[5] Mansour, S. and J. Ferber. Agent Groupe Rôle et Service : Un modèle organisationnel pour les systèmes multi-agents ouverts. In Actes du colloque JFSMA. 2007. Agent Groupe Rôle et Service - MaDKit

[6] Souidi, M.E.H., et al., Multi-agent cooperation pursuit based on an extension of AALAADIN organisational model. 2016. 28(6): p. 1075-1088. https://doi.org/10.1080/0952813x.2015.1056241

[7] Vázquez-Salceda, J., V. Dignum, and F. Dignum, Organizing multiagent systems. Autonomous Agents Multi-Agent Systems, 2005. 11(3): p. 307-360. https://doi.org/10.1007/s10458-005-1673-9

[8] Marion, N., Étude de modèles d'organisation sociale pour les environnements virtuels de formation. ftp://ftp.idsa.prd.fr/local/caps/DEPOTS/BIBLIO2006/rapbiblio_marion_nicolas.pdf

[9] Awodey, S., Category theory. 2010: Oxford University Press. Category Theory (Oxford Logic Guides, 52): Awodey, Steve

[10] Fiadeiro, J.L., Categories for software engineering. 2005: Springer Science & Business Media. Categories for Software Engineering - Research - Royal ...

[11] Milewski, B., Category theory for programmers. 2018: Bartosz Milewski. Category Theory for Programmers: The Preface | Bartosz

[12] Lane, S.M., Categories for the Working Mathematician. 1998. Graduate Texts in Mathematics, 1998. Categories for the Working Mathematician - Springer

[13] Wirsing, M., Algebraic specification. Handbook of Theoretical Computer Science (J. van Leeuwen, ed.). 1990. DOI: 10.1016/B978-0-444-88074-1.50018-4

[14] Awodey, S., Category theory. 2006: Oxford University Press. USA. DOI: 10.1093/acprof:oso/9780195307221.001.0001 Published ...

[15] Easterbrook, S., An introduction to Category Theory for Software Engineers. 1999, Department of Computer Science, University of Toronto. An introduction to Category Theory for Software Engineers*

[16] Pfalzgraf, J. and T. Soboll, On a General Notion of Transformationfor Multiagent Systems. Integrated Design Process Technology, IDPT- printed in the United States of America, 2007. K42: Building a Complete Operating System - cs.uni-salzburg.at

[17] Abderrahim, S. and R. Maamri, A Category-theoretic Approach to Organization-based Modeling of Multi Agent Systems on the Basis of Collective Phenomena and Organizations in Human Societies. Informatica, 2018. 42(4). https://doi.org/10.31449/inf.v42i4.1282

[18] Ormandjieva, O., et al., Modelling multi-agent systems with category theory. Procedia Computer Science, 2015. 52: p. 538-545. https://doi.org/10.1016/j.procs.2015.05.031

[19] Pfalzgraf, J., On categorical and logical modeling in multiagent systems. Anticipative Predictive Models in Systems Science, 2005. 1: p. 93-98. On Categorical and Logical Modeling in Multiagent Systems ...

[20] Zhu, M., P. Grogono, and O. Ormandjieva., Using category theory to verify implementation against design in concurrent systems. Procedia Computer Science, 2015. 52: p. 530-537. https://doi.org/10.1016/j.procs.2015.05.030

[21] Kuang, H., et al. Formal Specification of Substitutability Property for Fault-Tolerance in Reactive Autonomic Systems. in SoMeT. 2010. DOI:10.3233/978-1-60750-628-7-357

[22] Pfalzgraf, J. and T. Soboll, On a general notion of transformation for multiagent systems and its implementation. Electronic Communications of the EASST, 2009. 12. DOI:10.14279/tuj.eceasst.12.259.267