Model for Knowledge Transfer in Agent Organizations: a Case Study on Moise

Giovani P. Farias, Paulo F. G. L. Rodrigues, Diana F. Adamatti, Eder M. N. Gonçalves
Federal University of Rio Grande – FURG – Center for Computational Sciences – C3 – RS, Brazil
{giovanifarias, paulofglr}@gmail.com, {dianaadamatti, edergoncalves}@furg.br

Abstract
Knowledge transfer enables the development of complex multi-agent systems featuring agents that share knowledge to execute tasks. Environments shaped by knowledge transfer involve agents assuming specific roles, reasoning about the environment and other agents, and forming organizations that transfer knowledge through well-defined plans and strategies. This paper presents an organizational model for knowledge transfer, introducing a centralizing role, named organizer, responsible for managing relations between the system, agents, and their roles. An interaction protocol is established to guide the step-by-step communication between the organizer and other agents in the system during knowledge transfer. To model knowledge transfer within agent organizations, we employ a dynamic implementation of the Moise+ organizational model, called MoiseLight, enabling the creation of organizations at runtime. We validate the proposal by adapting the model to MoiseLight, demonstrating the organizer’s ability to facilitate knowledge transfer among agents following the specified interaction protocol.

Introduction
Multi-Agent System (MAS) consists of multiple agents interacting within an environment. Each agent performs actions to achieve specific goals, either independently or collaboratively. The agents’ actions influence the environment and contribute to the agent’s adaptation to these changes (Lesser 1999; McArthur et al. 2007; Grover et al. 2018).

Intelligent agents are autonomous entities equipped to perceive, reason about actions, interact with other agents, and make decisions. These interactions are defined based on an established standard language and the goals set for each agent. The surrounding environment and the activities influence an agent’s behavior and can undertake to achieve its goals (Demazeau 1995; McArthur et al. 2007; Premm and Kirn 2017; Tuyls and Stone 2017).

In MAS, the activities and interactions of agents are regulated by norms and social convention policies. These constraints set levels for the knowledge transfer among agents and the establishment of organizations and societies within the environment (Shoham and Tennenholtz 1995; Huhns and Stephens 1999; Weiss 1999).

The initial hypothesis of this work considers that the fundamental level of knowledge transfer involves the direct exchange of information between agents. At the intermediate level, agent organizations are tasked with devising global strategies and plans for problem resolution. Lastly, agent societies are at the top of the hierarchy, constituted by aggregating agents and organizations, incorporating social policies and norms. At each level, there is an increment in the semantic complexity of the data to be transferred (da Rocha Costa 2019). Figure 1 illustrates the proposed distribution across different levels for knowledge exchange.

Figure 1: Three levels for knowledge transfer.

To implement knowledge transfer in MAS, various techniques such as Transfer Learning (TL) and Reinforcement Learning (RL) are widely employed and can be found extensively in the literature. Knowledge transfer is crucial as it affects how well an agent can accomplish its activities and the level of complexity assigned to this actor. In an unfamiliar environment, an agent learns from another agent through exchanging knowledge/policies (Liang and Li 2020). This work is situated at the organizational level for knowledge transfer. Several authors classify different types of organizations in MAS, with the majority categorizing them as either agent-centric or organization-centric (Lemaître and Excelente 1998; Hübner 2003). The hypothesis here is that
knowledge transfer at the organizational level increases the flexibility of the organization and decreases the complexity of individual learning at the agent level.

The M\(\text{oise}^+\) organizational model has been selected to validate the proposed modeling. The model operates based on a structure formed by the relationship between the agents’ roles, employing a concept called social level, in addition to tasks and responsibilities named as the individual level (Hübner 2003). M\(\text{oise}\)Light was chosen for the case study, representing a dynamic implementation of M\(\text{oise}^+\), constructing the organization at runtime (Hübner 2022). For testing purposes, the JaCaMo platform was utilized. JaCaMo is a framework composed of the Jason agent programming platform, the environment programming in C\(\text{ArtAgO}\), and M\(\text{oise}^+\) for implementing organizational models (Boissier et al. 2020).

This work presents a model for knowledge transfer at the organizational level of agents. Objectives include defining the information to be transferred, the required data structures, the authorized agents to carry out these interactions, and their corresponding protocols. The primary contribution to the field of MAS lies in the computational modeling of diverse scenarios for knowledge transfer within the organizational dimension, serving as the overarching motivation for this work.

**Related Works**

Several works have explored knowledge transfer in MAS. In this section, our focus will be on those that utilize organizational models for knowledge transfer, aligning with the scope of our current study.

The work by (Dignum and Dignum 2005) addresses issues arising from agents executing different roles in socially regulated environments. The authors investigate how relationships between agent roles occur and in what manner this relationship influences the coordination mechanisms of a MAS.

According to (Maia and Sichman 2018), organizations and the environment in MAS restrict agents’ behavior by limiting their available actions to promote cooperation. Based on this principle, the authors propose a representation of autonomy in MAS using the M\(\text{oise}^+\) organizational model in JaCaMo. It mirrors the behavior of human organizations where agents have different degrees of autonomy.

The works of (Silva 2019) and (Liang and Li 2020) address the AdHoc Advising framework, an essential tool for learning in situations where no single experienced agent is transmitting all knowledge. In this context, agents seek guidance from others to learn policies and how to solve tasks, with agents understanding correctly when and how to transfer knowledge, individually or as a team.

In (Baldoni et al. 2021), an extension of the organizational model in JaCaMo is presented, incorporating the concept of exceptions. The authors propose a mechanism for managing exceptions in organizational structures within MAS, relying on organizational concepts such as responsibilities, goals, and norms. In a multi-agent organization, an exception is considered an event that impedes an agent from fulfilling one of its responsibilities, failing to achieve a goal. The authors highlight a way for agents to report exceptions during the organization’s execution.

The works of (Maia and Sichman 2018), (Machado 2020), and (Baldoni et al. 2021) employ organizational models in the JaCaMo framework, addressing different aspects in their respective approaches. While these works are closer to the context of this research, they conceptually and practically differ from the study’s concept of knowledge transfer at the organizational level. Most literature focuses on knowledge transfer solely at the agent level. The absence of academic productions with the same scope in the conducted systematic review underscores the significant contribution of this work to the field of MAS.

**Theoretical Background**

**Multi-agent systems**, since their emergence in the 1980s, have been considered as societies of agents, i.e., a group of agents interacting to coordinate their behavior and achieve collective objectives. Therefore, most research in this area must consider individual agents and societies (Ferber, Gutknecht, and Michel 2003).

(Lesser 1999) defines MAS as computational environments where agents interact or work cooperatively to satisfy their objectives. These environments can be heterogeneous or homogeneous, considering an agent in a system as a problem-solving entity with a certain degree of autonomy. According to the author, an agent’s autonomy is directly related to its ability to make decisions for task execution. An agent must decide which activities to perform, how and when to execute them, what to communicate with other agents, and how to absorb information from these interactions.

**Knowledge transfer** is a method for learning a new task in a new environment. In MAS, when an agent is unfamiliar with its environment and seeks to learn to enhance its rewards, it benefits from exchanging knowledge with another more experienced agent (Liang and Li 2020).

**Multi-agent organizations** are used for developing autonomous, distributed, and non-centralized systems. The main characteristics of organizational models are the functional decomposition of the organization’s goals and a system of norms. Norms define the scope of responsibilities that a member agent of the organization obeys, guiding the agent on which actions to execute to help the organization achieve its objectives (Baldoni et al. 2021).

M\(\text{oise}\) is an organizational model proposed by (Hannoun et al. 2000), centered on the agent figure for task execution. M\(\text{oise}\) is structured into three levels: the individual level, where responsibilities for each agent are defined; the collective level, where these agents are aggregated into different structures; and finally, the social level, where global structuring and interconnections of agent structures occur.

M\(\text{oise}^+\) is an extension of the model by (Hannoun et al. 2000), structured into three means for organizational constraints: structure (agent roles), functioning (global plans), and norms (organization obligations). M\(\text{oise}^+\) has two main specifications: organizational specification and organizational entity. A group of agents adopts an organizational
specification to generate an organizational entity, aiming to achieve a specific goal (Hübner 2003).

The Structural Specification (SS) of Moise+ is related to roles and the connections made (between roles, groups, and their hierarchies), aiming to limit agent autonomy through these role connections.

The Functional Specification (FS) focuses on how global objectives (plans, missions, and global goals) can be achieved by agents in the MAS. A global goal is the world state sought in a MAS, distinguishing itself from a local goal specific to a single agent. The functional specification describes how a MAS achieves its global goals. It is essential to mention the independence between the functional and structural specifications, as there is no need to reference structural specification in functional specification. It allows the MAS to change its structural specification (roles and groups) without altering the functional specification and vice versa (Hübner 2003).

There is also the so-called Deontic Specification (DS), which links the previous two (indicating the responsibilities of each role in executing global plans, forming a relationship between structural specification and functional specification) and defines which missions a role is allowed or obliged to perform. An organizational specification (composed of structural specification, functional specification, and deontic specification) does not include agents, as roles represent them. Therefore, in a MAS, this specification must be instantiated by a set of agents, forming an organizational entity (Hübner 2003; Boissier et al. 2020).

Methodology

This Section presents the proposal for knowledge transfer within and among agents’ organizations, describing its computational modeling and interaction protocol.

Organizational Models

This work uses organizational models developed in Moise+ to validate knowledge transfer in agent’s organizations. We observe four potential cases for handling knowledge transfer in organizational models, depicted in the quadrants of the Cartesian plane in Figure 2, with increasing complexity levels along each axis.

For knowledge transfer at the organizational level, we organize the Moise+ model as follows:

- different domains and their respective resources are part of the environment.
- organizations have Moise+ global goals, strategies, global plans, and groups to which each agent belongs. Global goals are the organization’s final objectives. Strategies are schemes necessary for an organization to achieve its goals. Global plans encompass objectives and actions necessary for the organization to perform environmental tasks.
- the organizer role is responsible for knowledge transfer between roles, whether they are present within the same organization or in different organizations, and for communication to accomplish this transfer. Each organization must have at least one organizer capable of managing and executing knowledge transfers. An agent can have other roles, including the organizer one.

The model developed for knowledge transfer in agents’ organizations is extensible to other tools beyond Moise+, becoming a general model for the organizational dimension in MAS. The organizer role is crucial for the entire knowledge transfer process, and its existence is mandatory, along with the definition of the associated communication protocol. To illustrate the concepts addressed in the modeling, we use the example of Soccer Team (Hübner, Sichman, and Boissier 2002), adapting it to the context of this work. Agents assume roles such as goalkeeper, defender, midfielder, and attacker, all of which are player sub-roles, each with its own local goals. Additionally, there is the coach sub-role, which, in this proposal, corresponds to the organizer role in the model.

The Organizer Role

In the proposed model, the organizer role is essential for facilitating knowledge transfer within and among agent organizations. The agent assigned this role acts as the centralizing manager in these relationships within the system, capable of observing and reasoning about the organization in which it is embedded, thus creating a representation of it.

The organizer controls the knowledge transfer between different roles. In transfers between organizations, the data flow occurs through communication among the organizers of each organization. In the context of Soccer Team example, the coach sub-role assumes the organizer role, responsible for determining which agents will engage in knowledge transfer. The organizer obtains methods for these agents to execute a new strategy to achieve the organization’s goal.
Table 1 represents the organizer behavior, based on the (Hannoun et al. 2000) model, described as follows:

- $m_1$ represents the mission transfer where: $g_1$ is the goal to be achieved; $p_i$ are plans to follow; $a_i$ are actions to be executed, and $r_i$ are resources available for the role; $O$ is an obligation for the role, and $Any$ is the set of all resources.
- goal $g_1$ of the organizer is to transfer knowledge, defining how the process is initiated and concluded.
- the set of plans $\{p_1, p_2, p_3\}$ includes: $p_1$ (call for transfer) defining which roles will participate in the process; $p_2$ (proposal for transfer) responsible for communication between organizers from different organizations; and $p_3$ (transfer knowledge) represents the actual knowledge transfer.
- the actions set $\{a_1, a_2\}$ includes: $a_1$ (send a message) enabling the organizer to communicate with other roles; and $a_2$ (update role) responsible for updating one of the roles involved in knowledge transfer. When communicating with another organizer, the agent can accept or reject the transfer proposal through the transmitted message content.
- $r_1, r_2,$ and $r_3$ are resources available to the organizer to perform its actions: observed data ($r_1$) is obtained through the observation and reasoning capabilities about the organization – exclusive to the organizer role; received data ($r_2$) consists of data received from other agents – with other roles — by the organizer, namely, the plans, social schemes, and exchanged messages; sent data ($r_3$) contains the roles updated by the organizer in the organization.

A comprehensive interaction protocol was established for the model to understand better the communication dynamics between the agent with the organizer role and other roles in one or more organizations (Figure 4).

**Interaction Protocol**

In the model developed in this work, communication is facilitated through the agent organizer. The process starts with the method “call for transfer”, where the content to be transmitted, through the message exchange among the organizer agents, is the role that an agent is currently performing. This role comprises plans assigned to it, along with their respective goals and social schemes. In the transmission process, agents are assumed to have the necessary capabilities to undertake any role.

![Interaction Protocol](image)

The organizer agent determines the roles that will participate in knowledge transfer and will be assigned to new agents during the interaction flow in the protocol. Following a successful interaction, the organizer receives a message containing information about the role, its plans, and social schemes. For instance, in Figure 3, we observe the transfer of role $Y$ to another agent that was previously performing role $X$. In the event of an error, a fail message is transmitted to the organizer. The organizer’s thread continues to run throughout the process.

The description of the fundamental internal communication structure within a single organization of agents, including the representation of its threads and interactions, is shown in Figure 3. In scenarios involving two or more organizations, the organizer roles are annotated with the respective organization names (e.g., Organization A and Organization B). The interaction protocol is defined as follows:

- in the knowledge transfer protocol, the agent assigned the organizer role within Organization A initiates the process by transmitting a message containing a knowledge transfer proposal to an organizer agent situated in Organization B. Subsequently, organizer B replies to the proposal, either rejecting it with a refusal, providing an error message (indicating a lack of understanding), or presenting a counterproposal.
- upon receiving a counterproposal, organizer A proceeds to send a message indicating its decision, whether it is an acceptance or a refusal, to organizer B.
- following the same pattern established for the model of a
single organization, internally, organizer B initiates a call for transfer to a role Y. The selected role responds with information or reports a failure.

- in the event of success in the internal communication between organizer B and role Y, organizer B transmits the content of the role to organizer A.
- finally, if organizer A receives the data sent by organizer B, it transmits an information or fail message to an internal role in its organization, named role X. In the event of successful communication, the agent assigned with role X begins to perform role Y.

All these response mechanisms adhere to the principles of the Contract-Net Protocol. As explained earlier, Figure 4 illustrates the interaction protocol for two distinct agents’ organizations.

**Figure 4: Knowledge transfer in two different agents’ organizations.**

**Case Study: MoiseLight**

After contextualizing four possible organizational models for knowledge transfer (Figure 2), the decision was made to use the JaCaMo tool for the implementations. JaCaMo, an acronym for Jason, CArtAgO, and Moise, an open-source solution for the development of integrated MAS across three dimensions: agent, environment, and organization (Boissier et al. 2020). Jason encompasses agents and their relationships, including their assignment to Moise+ groups. The groups of agents and the roles of the model are within the Moise dimension. CArtAgO is the environment that includes different domains and their resources. Figure 5 represents the Soccer Team model in JaCaMo dimensions.

**Figure 5: Soccer Team in JaCaMo dimensions.**

Jason is an extension of the AgentSpeak(L) language implemented in Java. It offers various functionalities for multi-agent development, such as communication between agents based on speech acts, annotations in beliefs with data on the source of information, annotations in plan labels, and the ability to run distributed MAS on a network. Agents in Jason execute plans in the environment through their plan library. The agent perceives the environment through changes in its belief base (Bordini and Hübner 2005; Maia 2018; Hübner, Bordini, and Vieira 2022).

CArtAgO is a platform for developing environments in MAS. Programming in CArtAgO is done through artifacts, integrated with Jason. From the agent’s perspective, these artifacts represent resources available in the environment, which agents use and share to perform their actions (Ricci et al. 2009).

MoiseLight is an evolving implementation of the Moise+ model, where the organization is dynamically built at runtime (Hübner 2022). In MoiseLight, an agent creates the organization, schemes, groups, and attributes roles to other agents. The artifact related to the domain in CArtAgO is directly created in the code header of this agent, along with the group in Moise+. The agent receives its role through the adoptRole method. Subsequently, the agent establishes scheme $s_1$, defines the obligations in the model, attributes role to other agents, and, upon accomplishing all goals, dismantles the scheme and the groups, concluding the operation.

The objectives defined in MoiseLight are $g_0$, $g_1$, and $g_2$, all belonging to the agent that created the organization. Notably, $g_2$ is contingent on the achievement of $g_1$. In the proposed model, $g_2$ is the obligatory goal of the attacker role, which will be transferred to another agent without a role. In this work, the organizer role was created, and following the established interaction protocol (Figure 3), it enables knowledge transfer between different agents in the system. The attacker role is initially assigned to agent $ag_1$, while agent $ag_2$ starts without a role at the beginning of the program execution (Figure 6).

Figure 3 highlights the steps involved in knowledge transfer within the organization. In step (1), the organizer speci-
fies the role that will partake in the knowledge transfer using the call for transfer method. In the code snippet of the organizer, presented in Listing 1, the agent selects agent ag2 for the knowledge transfer process and sends a message containing a string and a condition.

```csharp
// call agent ag2 for transfer
.print("call agent for transfer")
.send(ag2,tell,callfortransfer(accept,true))
```

Listing 1: Organizer chooses ag2 for knowledge transfer.

In step (2), the plan call for transfer, defined in the plan library, is utilized. In this context, the response from agent ag2 to the organizer is transmitted as inform or cancel. To accomplish this, a string is configured, incorporating the terms accept or reject, along with the condition of true or false. Listing 2 provides the code snippets for the two potential responses from agent ag2 to the organizer.

```csharp
callfortransfer(X,C): X="accept" & C==true <- .send(organizer,tell,
\rightarrow callfortransfer(inform,C))
.print("Answer: ",X)

callfortransfer(X,C): X="reject" & C==false <- .send(organizer,tell,
\rightarrow callfortransfer(cancel,C))
.print("Answer: ",X)
```

Listing 2: Method for knowledge transfer call.

In step (3), the organizer transfers knowledge to agent ag2, who assumes the attacker role and consequently the goal g2, identical to the role and goal of ag1. Listing 3 provides a code snippet from the organizer that executes sending the new role to the agent.

```csharp
// make agent ag2 play as attacker
.send(ag2,achieve,
\rightarrow adopt_role(attacker,grp1,true))
```

Listing 3: Organizer sends the attacker role to agent ag2.

At the end of the execution, based on the Soccer Team example, the organizer assumes the coach sub-role, while agents ag2 and ag1 take on the attacker roles. Figure 6 illustrates the roles throughout the program execution, from the beginning to the end of the knowledge transfer process.

**Conclusion**

This work presented an organizational model for knowledge transfer in multi-agent organizations. Based on concepts in the MAS literature, the dimensions defined as agents, organizations, and societies were explored, considering the different complexities in handling knowledge transfer. These complexities are related to the number of organizations and roles in the model.

The research found that the existence of at least one centralizing agent in the organization, assuming the role of organizer, is necessary to observe the organization as a whole and perform processes for knowledge transfer among agents and their respective roles. A communication/interaction protocol was modeled to define how the organizer agent communicates with other agents in the system through the call for transfer method. The tests conducted in MoiseLight followed the guidelines established in this protocol.

With the development of the initial diagrams, discussions regarding the model, and progress in research, the choice was made to focus on a single organization of agents. In this context, knowledge transfer takes place among agents facilitated by the organizer. Furthermore, the model also encompasses knowledge transfer between organizations, with the specification of the interaction and communication protocol among different organizations.

MoiseLight was suitable for visualizing the proposed model, allowing for dynamic organization construction at runtime, starting from a central agent. The metrics of the organizer role aligned with the implementation and proved functional for validating the knowledge transfer process.

The organizational model developed is adaptable to other organizational models besides Moise+, making it an essential contribution to the field of MAS, along with the developed interaction protocol and the classification of different levels for knowledge transfer.

**Acknowledgements**

This research was supported by Edital FAPERGS/CNPq 07/2022 – PDJ.
References


