A formal model of agent interaction based on MASQ

Razvan Dinu, Tiberiu Stratulat, and Jacques Ferber

University of Montpellier 2, LIRMM

Abstract. MASQ (Multi-Agent Systems based on Quadrants) is a generic meta-model that integrates many of the concepts issued from the research in the multi-agent systems field by defining four perspectives over agent-based interaction according to two axes: internal/external and individual/collective. The aim of this paper is to provide a formal description of MASQ by specifying clear relations between the basic elements of the four quadrants and by analysing the dynamics of a multi-agent system through its transitions and through the classical mind cycle. The proposed formal model has been designed as a theoretical framework and therefore the specifications of many components of a multi-agent system such as the internal architecture of minds, the casual laws of environments or the social laws in organizations and institutions are intentionally left generic to allow for many further concretizations.

1 Introduction

MASQ (Multi-Agent Systems based on Quadrants) [11] is a very generic meta-model for the agent-based interaction that takes its inspiration from the 4-quadrant model by Wilber [15] in social sciences and psychology. MASQ integrates many of the concepts issued from the research in the multi-agent systems field by defining four perspectives over agent-based interaction according to two axes: internal/external and individual/collective. It considers equally the concepts of actions, environments[14], organizations and institutions and integrates them into the same conceptual framework.

MASQ is built on five basic elements (Figure 1): minds in quadrant I (individual-interior), objects and bodies in quadrant II (individual-exterior), spaces in quadrant III (collective-exterior) and cultures in quadrant IV (collective-interior). In addition, a set of relations between the basic elements, that actually form the link between the four quadrants, and a set of laws that describe its dynamics are proposed.

MASQ is an intuitive map to understand the agent-based interaction. The complete scenario in MASQ consists of a mind that acts through a body in a space, where it interacts with other objects or bodies. The culture in which the minds are immersed allows them to collectively interpret the interaction and construct the institutional reality, as proposed by Searle [10]. We acknowledge that among the first proposals of modeling agents with concepts inspired from Searle are [1] and [9] in the context of institutionalized power. From this point of view, a close research effort is [12], but it is only oriented on roles and organizations. Also, in [2] a detailed description of types of norms inspired from Searle's typology of institutional rules is given.

MASQ has been successfully used both in modeling multi-agent systems for real world scenarios and for social simulations. [8] is a good example of using MASQ

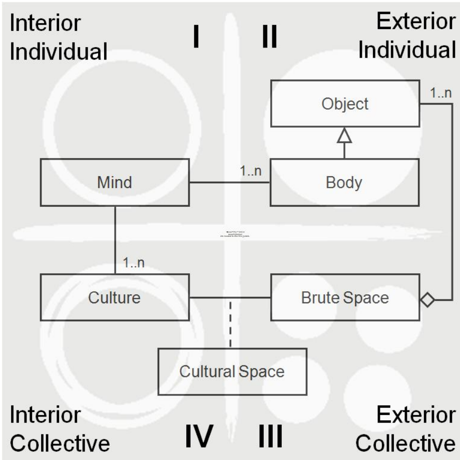

Fig. 1. MASQ meta-model

to model the logistics of a warehouse and show, using the MASQ concepts, what are the most important decisions when implementing such a system. In [4], a multi-agent framework based on MASQ, called Agent-based Business Coordination Lab (ABC Lab), is implemented as a plugin for the Repast Symphony simulation framework and it allows the simulation of trading networks in order to study the role and impact of intermediation. In both [4] and [8] the MASQ meta-model is chosen because it allows on one hand the explicit modeling of individual, economic, social and material factors and on the other hand the design of models that are both modular and extensible.

Even though MASQ has been successfully applied in MAS modeling and simulations it is still used in an ad-hoc manner and every approach makes its decisions about how the different MASQ elements should work together. For instance, the communication between the minds and their bodies is usually neglected or is left as an implementation detail for other underlying frameworks. Also, different existing frameworks, such as OperA[3] (in [4] and [8]) and AGR[6], are used for the collective quadrants but it is not clear how the elements of these frameworks map to the MASQ concepts. The aim of this paper is to make a first step in addressing this issue by proposing a formal characterization of MASQ preserving as much as possible from its generality. Having a clear formal description of MASQ will allow us to better understand the MASQ meta-model, how to construct a model based on MASQ and how to integrate, in an unambiguous way, with different existing frameworks.

In [5] a first step is made towards a formal description of MASQ by identifying a set of seven principles that must be followed when using the MASQ meta-model. Also it discusses the most important design choices that have to be made when creating a formal model based on MASQ and shows that these choices introduce new constraints that were not originally present. That is why in this paper we try to make as few choices as possible. The theoretical framework we propose is therefore very generic so that it could be further concretized with a specific environment and features. It should be noted that some parts of our formalization, especially those related to the relationship between individual and collective intentionality, deal with aspects that are still in open debate in philosophy and social sciences.

In sections 2 and 3 we begin with a clear formal description of the exterior quadrants and their relationship with the mind. In order to be able to go further and define the concepts of the cultural quadrant, in section 4 we propose GMS (Generic Mind based on Strings) as model of a mind and a representation language ($\mathcal{RL}$). They allow us to introduce, in section 5, the *ethos* of a group as well as some concepts inspired from Searle's work [10] such as *count-as* relation, *constitutive* and *regulative* rules and finally *institutions*.

2 Minds and environment

In this section we give clear definitions of *objects*, their state and the relations between them. We regard minds in the most generic way, as simple elements, and their structure is discussed in more details in section 4. Also, we show exactly what elements link the first two quadrants, *actions* and *data sensors*. Finally, a complete description of the environment is given and how it evolves.

To make things more clear, throughout this section we will use the example of a ping-pong game in which two agents use two rackets and a ball to play on a table.

2.1 Minds and objects

Definition 21. Let $\mathcal{M} = \{m_0, m_1, \dots\} \cup \{\mu\}$ be a set of elements m_i called *minds*. μ is called the *null* mind.

Definition 22. Let $\mathcal{O} = \{o_0, o_1, \dots\} \cup \{\omega\}$ be a set of elements o_i called *objects*. ω is called the *null* object.

Minds represent the central element of quadrant I and objects of quadrant II. We suppose that we are given two sets, not necessarily finite, of them. The importance of μ and ω will be outlined in the following sections.

In the ping-pong example we have two players, player one and player two, $\mathcal{M} = \{p_1, p_2, \mu\}$ and a few objects $\mathcal{O} = \{room, table, racket_1, racket_2, ball, \omega\}$.

Definition 23. A binary relation "$\sqsubseteq$" $\subset \mathcal{O} \times \mathcal{O}$ is a *containment* relation and $o_i \sqsubseteq o_j$ is read as "o_i is contained in o_j" if "$\sqsubseteq$" has the following properties:

1. $\forall x, y, z \in \mathcal{O}, x \neq \omega$ such that $x \sqsubseteq y \wedge x \sqsubseteq z$ then we have $y \sqsubseteq z$ or $z \sqsubseteq y$.
2. $\forall x \in \mathcal{O}, x \neq \omega\ \exists! s \in \mathcal{O}$ such that $\forall y \in \mathcal{O}$ $x \sqsubseteq y \Rightarrow s \sqsubseteq y$.
3. $\exists! \mathcal{U} \in \mathcal{O}$ such that $\mathcal{U} \sqsubseteq \mathcal{U}$; $\mathcal{U}$ is the *universe* element.

The containment relation has three properties which we will discuss now. The first one states the an object cannot belong to two branches of inclusion. As it will be more clear in the following section, after defining what a space is, this means that spaces can only include each other and not intersect.

The second property says that for any object o there is a unique smallest object s that contains it and consequently it is included in all other objects that o is included in. Admitting that there are no infinite branches of containment it can be proven that the $\mathcal{U}$ element in property 3 contains all other objects.

The "$\sqsubseteq$" *containment* is actually the connection between the exterior quadrants (II and III) of the MASQ meta-model.

In our example we have the following relations: $ball \sqsubseteq table$, $racket_1 \sqsubseteq table$, $racket_2 \sqsubseteq table$, $table \sqsubseteq room$, $room \sqsubseteq room$.

Definition 24. A binary relation "$\sim$" $\subset \mathcal{M} \times \mathcal{O}$ is an *embodiment* relation and $m_i \sim o_j$ is read as "m_i is embodied in o_j" if "$\sim$" has the following property:
$\forall m_i, m_j \in \mathcal{M}$, $\forall b \in \mathcal{O}$ we have $m_i \sim b \wedge m_j \sim b \Rightarrow m_i = m_j$

The "$\sim$" *embodiment* relation represents the connection between the individual quadrants (I and II) of the MASQ meta-model. In our example we have: $p_1 \sim racket_1$ and $p_2 \sim racket_2$.

2.2 Spaces and bodies

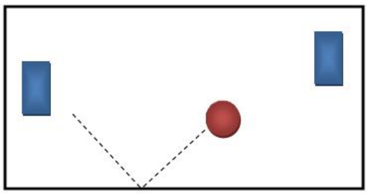

Fig. 2. Ping-pong game environment

Below we identify two particular types of objects: spaces and bodies.

Definition 25. An entity $s \in \mathcal{O}$ is a *space* if and only if there exists at least one object $o \in \mathcal{O}$ such that $o \sqsubseteq s$. Also, if $s \in \mathcal{O}$ is a *space* then $\omega \sqsubseteq s$. Let $\mathcal{O}_S \subset \mathcal{O}$ be the set of all spaces w.r.t. a containment relation "$\sqsubseteq$".

The *null object* ω introduced in definition 22 is actually a marker object in the sense that for a space $s \in \mathcal{O}_S$, $\omega \sqsubseteq s$ can be interpreted as "s can contain other objects".

Definition 26. An object $b \in \mathcal{O}$ is a *body* if and only if there exist a mind $m \in \mathcal{M}$ such that $m \sim b$. Let $\mathcal{O}_B \subset \mathcal{O}$ be the set of all bodies w.r.t. an embodiment relation "$\sim$".

The *null mind* μ introduced in definition 21 is actually a marker mind in the sense that for a body $b \in \mathcal{O}_B$, $\mu \sim b$ can be interpreted as "b is not yet embodied by any real mind, but it can be".

One last thing to notice is that an object $o \in \mathcal{O}$ can be a body and a space at the same time. This offers a great amount of flexibility when designing a multi-agent system and contributes to the genericity of our framework.

In our example we have $\mathcal{O}_S = \{room, table\}$ and $\mathcal{O}_B = \{racket_1, racket_2\}$.

2.3 States and Attributes

Besides its relations with other objects or minds an object is characterized by a state which is associated with a set of attributes.

Definition 27. Let $\Sigma_{\mathcal{O}}$ be a set of elements called *states*. If $O \subset \mathcal{O}$ then a function $\sigma\colon O \to \Sigma_{\mathcal{O}}$ is called a *state function on* O and associates each object $o \in O$ with its state $\sigma_o = \sigma(o)$.

The state σ_o of an object can encapsulate any type of information about the object (a set of name-value pairs, place in a topology, etc.). Type information can be included here too.

Definition 28. Let $\mathcal{A}$ be a set of elements called *attributes*. A function $\alpha\colon \Sigma_{\mathcal{O}} \to 2^{\mathcal{A}}$ is called an *attribute function* and associates each *state* with its set of *attributes*.

The role of attributes is very important as they represent the basic unit of perception for objects' states. A mind will not be able, through bodies, to perceive the whole state of an object but only a set of attributes corresponding to the state of the object.

In our example we won't go into the details of the state of all objects. We just give two examples: $\{color, position\} \subset \mathcal{A}$, $\{\text{"}color = red\text{"}\} \subset \alpha(\sigma_{ball})$, $\{\text{"}position = 50, 100\text{"}\} \subset \alpha(\sigma_{racket_2})$.

2.4 Actions

Now we continue with the part of the environment that allows mind's interaction. The first thing we define is an *action*:

Definition 29. An *action* γ is a function $\gamma\colon \Sigma_{\mathcal{O}} \to \Sigma_{\mathcal{O}}$ that modifies the state of an object. We say γ is applicable in a state σ_o if and only if $\gamma(\sigma_o) \neq \sigma_o$. And let $C_\gamma = \{\sigma_o | \sigma_o \in \Sigma_{\mathcal{O}}, \gamma(\sigma_o) \neq \sigma_o\}$ be the *context* in which γ can be applied.

The above definition of an action is very generic. We leave to the designer of a MAS to choose a more concrete description of actions and also, very important, how minds understand what actions do in order to decide which one to perform. Actions can be associated with bodies and form behaviors.

Definition 210. A set of actions $\beta_b = \{\gamma_{b0}, \gamma_{b1}, \dots\} \subset \Gamma$ is called a *behavior*, where Γ is the set of all actions. If $B \subset \mathcal{O}_B$ then a function $\beta\colon B \to 2^{\Gamma}$ is called a *behavior function on* B and associates each body $b \in B$ with its behavior $\beta_b = \beta(b)$.

The behavior of a body is actually the set of *commands* a mind can use to *influence* the environment. This fits perfectly with the *Influence/Reaction* principle which states that an agent cannot directly modify the environment but only influence it and wait for its reaction [7].

In our example we have $\Gamma = \{move_{up}, move_{down}\}$ and $\beta_{racket_1} = \beta_{racket_2} = \Gamma$.

2.5 Brute Percepts

As described until this point the environment is composed of objects with states that contain attributes and actions associated to bodies that form behaviors. Objects can be included in other objects called spaces and minds can be embodied in bodies. Below we give the definition of *brute percepts* which is the form under which a mind perceives its environment.

Definition 211. Let $\mathcal{P}_b$ be a set of elements called *brute percepts*. Let $\pi\colon \mathcal{O} \cup \sqsubseteq \cup \mathcal{M} \cup \sim \cup \Gamma \cup (\mathcal{O}_B \times \Gamma) \cup \mathcal{A} \cup (\mathcal{O} \times \mathcal{A}) \rightarrow \mathcal{P}_b$ be a *brute projection* function such that:

$\forall o \in \mathcal{O} \Rightarrow \pi(o) = o_\pi \quad \forall o \sqsubseteq s \Rightarrow \pi(o \sqsubseteq s) = (o_\pi, s_\pi)$
$\forall m \in \mathcal{M} \Rightarrow \pi(m) = m_\pi \; \forall m \sim b \Rightarrow \pi(m \sim b) = (m_\pi, b_\pi)$
$\forall \gamma \in \Gamma \Rightarrow \pi(\gamma) = \gamma_\pi \quad \forall \gamma \in \beta_b \Rightarrow \pi((b, \gamma)) = (b_\pi, \gamma_\pi)$
$\forall a \in \mathcal{A} \Rightarrow \pi(a) = a_\pi \quad \forall a \in \alpha(\sigma_o) \Rightarrow \pi((o, a)) = (o_\pi, a_\pi)$

One very important thing to notice from the definition of π, which projects the environment into the set of brute percepts, is that a mind can perceive another mind (the existence of m_π and (m_π, b_π) in $\mathcal{P}_b$). This means that a mind is capable of ”sensing” there is another mind behind an object, or in other words, a mind can make distinction between objects and bodies. But it can perceive absolutely nothing about what's inside another mind.

2.6 Data sensors

Sensors operate on a state of the environment so before giving the definition of a sensor we will define what an *environmental state* is.

Definition 212. An *environmental state* is a tuple $\sigma_E =< O, \sigma, \beta, \sqsubseteq, \sim>$ where $O \subset \mathcal{O}$, σ is a *state function on* O, β is a *behavior function on* $O \cap \mathcal{O}_B$, ”$\sqsubseteq$” is a *containment* relation and ”$\sim$” is an *embodiment* relation. Let Σ_E be the set of all such tuples.

Definition 213. A *data sensor* is a function $\delta\colon \mathcal{O}_B \times \Sigma_E \rightarrow \mathcal{P}_b$ and it extracts, on behalf of a *body*, some data from an *environmental state* under the form of *brute percepts*. Let Δ be the set of all *data sensors*.

Definition 214. A set of *data sensors* $\rho_b = \{\delta_{b0}, \delta_{b1}, \dots\} \subset \Delta$ is called *responsiveness*. If $B \subset \mathcal{O}_B$ then a function $\rho\colon B \rightarrow 2^\Delta$ is called a *responsiveness function on* B and associates each body $b \in B$ with its responsiveness $\rho_b = \rho(b)$.

An important thing when introducing sensors is the principle of *locality of perception*. It is realized by the *data sensor* (that is why it is defined on $\mathcal{O}_B \times \Sigma_E$, to take into account a body too).

Definition 215. Let $M \subset \mathcal{M}$ be a set of minds. A function $\epsilon\colon M \rightarrow 2^{\mathcal{P}_b}$ is called a *brute environment function on* M and associate each mind $m \in M$ with its set of *brute percepts* of the environment $\epsilon_m = \epsilon(m)$.

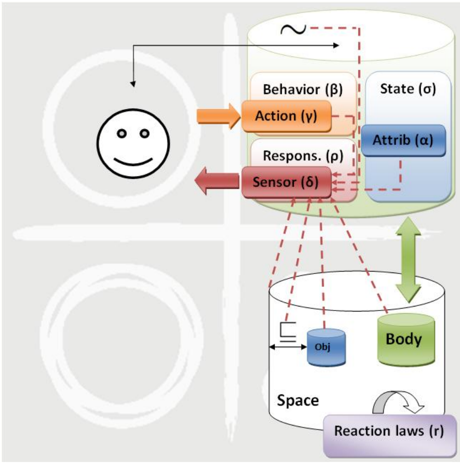

Fig. 3. Environment and relation to minds

ϵ_m represents the *brute reality* that m perceives through its *bodies* and it is constructed by the union of all brute percepts from all sensors from all bodies.

In our example the only sensors needed are *position sensors* that will be able to transmit to a mind the coordinates of the two rackets and the ball ($\rho_{racket_1} = \rho_{racket_2} = \Delta = \{position\}$

2.7 Reaction laws

Definition 216. A *reaction law* is a function $r\colon \Sigma_E \to \Sigma_E$. We say r is applicable in an *environmental state* σ_e if and only if $r(\sigma_e) \neq \sigma_e$. And let $C_r = \{\sigma_e | \sigma_e \in \Sigma_E, r(\sigma_e) \neq \sigma_e\}$ be the *context* in which r can be applied. Also, let $\mathcal{R}$ be the set of all reaction laws.

In our example we have the following reaction laws $R = \{move_{law}, reflection_{law}, colision_{law}\}$.

Definition 217. An *environment* is a tuple $E =< \sigma_E, \rho, R >$ where σ_E is an environmental state, ρ is a *responsiveness function* on the set of bodies in σ_E and $R \subset \mathcal{R}$ is a set of reaction laws.

The evolution of the environment is seen as a continuous change of its σ_E environmental state through reaction laws. As it can be seen from its definition, a reaction law can change just about everything in the environment except for the set of reaction laws.

2.8 Multi-Agent Systems

Now we are able to give a complete definition of what a multi-agent system is:

Definition 218. A *multi-agent system* is a tuple $S =< M, \epsilon, E >$ where $M \subset \mathcal{M}$ is a set of minds, ϵ is a *brute environment function on* M and E is an environment.

3 Environment dynamics

This section analyzes the dynamics of a multi-agent system as it was described in the previous section. We use the classical system-transitions approach. There are three main types of transitions:

Γ-transitions. When a mind wants to perform a set of actions (send commands to its bodies).
r-transitions. When a reaction law is applied in the environment.
Δ-transitions. When a mind receives data from its bodies.

3.1 Internal evolution

If there are no active agents in an environment the only way it can change its state is by means of reaction laws. Let $S =< M, \epsilon, E >$ be a multi-agent system and r a reaction rule.

Rule 31. Reaction law

$$\frac{\begin{array}{c} S =< M, \epsilon, E >, \ \ E =< \sigma_E, \rho, R >, \ \ r \in R \\ \sigma_E \in C_r, \quad \sigma'_E = r(\sigma_E) \end{array}}{S \to < M, \epsilon, < \sigma'_E, \rho, R >>}$$

3.2 Actions and responses

After deliberating, a mind decides on a set of actions that it wants to perform through its bodies. Let m be a mind and B_m the set of its bodies. Then the system can move to a state $< M, \epsilon, E' >$, by a Γ-*transition*, where the state of the bodies of m have changed.

Rule 32. Action

$$\frac{\begin{array}{c} S =< M, \epsilon, E >, \ \ E =< \sigma_E, \rho, R >, \ \ \sigma_E =< O, \sigma, \beta, \sqsubseteq, \sim> \\ m \in M, \ \ \Gamma_m = \{b : \gamma\}, \ \ b \in B_m, \ \ \gamma \in \beta(b), \ \ \sigma(b) \in C_\gamma \\ \sigma' : O \to \Sigma_{\mathcal{O}}, \ \ \sigma'(x) = \begin{cases} \sigma(x) & \text{if } x \neq b \\ \gamma(\sigma(b)) & \text{if } x = b \end{cases} \\ \sigma'_E =< O, \sigma', \beta, \sqsubseteq, \sim> \\ E' =< \sigma'_E, \rho, R > \end{array}}{S \to < M, \epsilon, E' >}$$

The *data sensors* that are attached to *bodies* send data back to the mind and update the *brute reality* it perceives. Then the system can move to a state $< M, \epsilon', E >$, by a Δ *transition*, where the *brute reality* of m has changed.

Rule 33. Response

$$
\frac{\begin{array}{c} S =< M, \epsilon, E >, \;\; E =< \sigma_E, \rho, R >, \\ m \in M, \;\; B_m = \{b \in \mathcal{B} | m \sim b\} \\ \epsilon'_m = \bigcup_{b \in B_m} \bigcup_{\delta \in \rho(b)} \delta(b, \sigma_E) \\ \epsilon' : M \to 2^{\mathcal{P}_b}, \;\; \epsilon'(x) = \begin{cases} \epsilon(x) \text{ if } x \neq m \\ \epsilon'_m \;\;\text{ if } x = m \end{cases} \end{array}}{S \to < M, \epsilon', E >}
$$

3.3 Request/Drop body

We also have two special types of transitions: *request* and *drop* body.

Let $S =< M, \epsilon, E >$ be a multi-agent system, m be a mind and b a body. When m requests the body b the system moves to a state $< M, \epsilon, E' >$, by a *request body* transition, where the body b will be assigned to m if it isn't assigned to another mind.

Rule 34. Request body.

$$
\frac{\begin{array}{c} S =< M, \epsilon, E >, \;\; E =< \sigma_E, \rho, R >, \\ m \in M, \;\; b \in O, \;\; \mu \sim b \\ \sigma'_E =< O, \sigma, \beta, \sqsubseteq, (\sim \backslash (\mu, b)) \cup (m, b) > \end{array}}{S \to < M, \epsilon, E' >}
$$

When m drops body b then the system moves to a state $< M, \epsilon, E' >$, by a *drop body* transition, where the body b will be assigned to *null* mind if it is assigned to m.

Rule 35. Drop body.

$$
\frac{\begin{array}{c} S =< M, \epsilon, E >, \;\; E =< \sigma_E, \rho, R >, \\ m \in M, \;\; b \in O, \;\; m \sim b \\ \sigma'_E =< O, \sigma, \beta, \sqsubseteq, (\sim \backslash (m, b)) \cup (\mu, b) > \end{array}}{S \to < M, \epsilon, E' >}
$$

4 Interior of a mind

The previous two sections described the exterior quadrants from the MASQ perspective. The reasoning process of minds is responsible for Γ-transitions of a MAS. Even though MASQ makes no assumption about the internal structure of minds, we need a minimal formal model of a mind that will serve us as support to clearly explain how the perception, reasoning and acting phases happen in the mind of an agent and also to introduce the concepts of the forth quadrant: Searle's *count-as* relation, *constitutive* and *regulative* rules, and institutions. In this section we introduce a simple and generic mind model called GMS (Generic Mind based on Strings) that continues in a natural way the formal description of the environment in the previous sections by using a simple and generic representation language $\mathcal{RL}$.

4.1 Reactive agents

Reactive agents are agents that have no memory and act guided by a set of rules. A reactive agent's mind will work directly with the set of brute percepts $\epsilon_m \subset \mathcal{P}_b$. Formally, a reactive mind can be associated with a function $react\colon 2^{\mathcal{P}_b} \to 2^{\mathcal{O}_B \times \Gamma}$ that maps brute percepts into actions on bodies.

4.2 Symbols and attitudes

On the other hand, cognitive agents will interpret the brute percepts they receive from their bodies by using *symbols* and *attitudes*.

Definition 41. Let $\mathcal{S}$ be a set of elements called *symbols*. They are virtual representations that exist only in agents' minds. Symbols are the basic elements for constructing institutional realities.

As example of symbols we can take a *goal*, a *fault* or an *off-side* in a football match. They do not exist in the brute reality, they are symbols associated with specific situations or things. Most of the symbols will be created by interpreting brute reality and this will be explained in more details in section 5.3.

Definition 42. Let Λ be a set of elements called *attitudes*. An attitude can be thought of as the position an agent has towards something. An agent can have attitudes over brute facts, over institutional facts and even over other attitudes.

As example of attitudes we have: knowledge (K), belief (B), desire(D), goal (G), intention (I), acceptance(A), etc.

A special set of attitudes is $\Lambda_D = \{O, P, I\} \subset \Lambda$ which is called the set of *deontic attitudes* (obligation, permission, prohibition). Its role will be highlighted in section 5.1.

4.3 Representation language

Before we propose our model of a mind we need to specify how the brute percepts, on one side, and symbols on the other side come to co-exist inside a mind and also how attitudes fit in. For this purpose we define a language ($\mathcal{RL}$) in which we can describe both the exterior world (the perceived data through sensors), the institutional reality and the agent's attitudes over them.

The set of atomic symbols for the $\mathcal{RL}$ language is composed of brute percepts (o_π, m_π, etc.), symbols, a set of key words and a few additional atoms.

Definition 43. Let $\mathcal{K} = \{included, embodied, action,$
$attribute\}$ be a set of key words.

Let $Values = \mathcal{O}_\pi \cup \mathcal{M}_\pi \cup \Gamma_\pi \cup \mathcal{A}_\pi \cup \mathcal{S}$ be a set whose elements are called *values*. In other words, all symbols $s \in \mathcal{S}$ and all brute percepts $o_\pi, m_\pi, \gamma_\pi, a_\pi \in \mathcal{P}_b$ are considered values.

Let $Variables = \{x_0, x_1, x_2, \dots\}$ be a set of elements called variables.

The set of atomic symbols for our language is represented by $Atoms = Values \cup Variables \cup \mathcal{K} \cup \Lambda \cup \{(,), \wedge, \to\}$.

Definition 44. The $\mathcal{RL}$ representation language is defined by the following BNFs:

$\varphi \quad ::= \varphi_b \mid \varphi \wedge \varphi \mid (\varphi) \to (\varphi) \mid \lambda(\varphi) \mid \lambda m_\pi(\varphi) \mid \lambda g(\varphi).$

$\varphi_b \quad ::= s \mid \varphi_o \mid \varphi_m \mid \varphi_\gamma \mid \varphi_a \mid$
$\quad included(\varphi_o, \varphi_o) \mid embodied(\varphi_m, \varphi_o) \mid$
$\quad action(\varphi_o, \varphi_\gamma) \mid attribute(\varphi_o, \varphi_a).$

$\varphi_o \quad ::= o_\pi \mid x.$

$\varphi_m \quad ::= m_\pi \mid x.$

$\varphi_\gamma \quad ::= \gamma_\pi \mid x.$

$\varphi_a \quad ::= a_\pi \mid x.$

where $s, g \in \mathcal{S}$ are symbols, $\lambda \in \mathit{\Lambda}$ is an attitude symbol, x is a variable, $o_\pi \in \mathcal{O}$ is an object and so on. φ_b is called a *basic string* and contains a *value*, a *variable* or relation (*included*, *embodied*, etc.) between two *values* or *variables*.

Both the atomic symbols $\wedge$ and $\to$ are used to create a new well formed string from two well formed strings and their use will be outlined in the following sections.

We will denote by $\mathcal{RL}$ the set of all valid strings in the previously defined language.

4.4 Types of strings in $\mathcal{RL}$

Now we will outline the most important types of strings in RL language.

A string of the form $\lambda(\varphi)$, $\lambda m_\pi(\varphi)$ or $\lambda g(\varphi)$ is called an *attitude string* and let $\mathcal{RL}_\Lambda$ be the set of all such strings. A string which is not an attitude string is an *information string*. One particular type of information string is a string that contains no symbols, attitudes or variables and it's called a *brute fact string*.

Brute fact strings correspond to an objective description of the environment. Example of brute fact strings: $ball$, $room$, $included(ball, room)$, $attribute(ball, red)$, etc.

On the other hand, *attitude strings* correspond to a subjective description, from a mind's perspective. Examples of attitude strings: $K(included(ball, room))$, $B(attribute(ball, red))$, $A(attribute(ball, blue))$, etc.

4.5 Generic Mind based on Strings

The model we propose is called *Generic Mind based on Strings* (GMS) and has four modules:

Data is a set of strings in $\mathcal{RL}$ language. For a mind m it is denoted by $Data_m$ and the fact that a string $\varphi \in Data_m$ is read "φ exists in the mind m". It is a key module of the model as it actually represents the link between the following three modules.

Interpret module is responsible for receiving the brute percepts from bodies and insert them into $Data_m$ under the form of *brute fact strings*.

Reasoning/Learning module handles all the reasoning and learning performed by a mind. Its input is the current set of strings in $Data_m$ and its output are changes to $Data_m$ (adding or removing of strings).

Action module is responsible for sending actions to bodies by analyzing the current set of strings in $Data_m$.

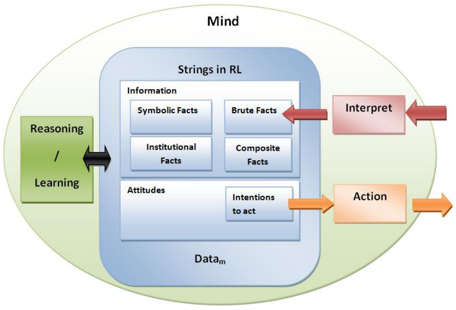

Fig. 4. Generic Mind based on Strings

The main idea of GMS is to have a module for each of the three phases of the classical mind cycle and to link them through a common set of *data* under the form of strings in $\mathcal{RL}$. The language $\mathcal{RL}$ is very important as we will make use of different forms of strings in order to introduce the concepts in the cultural quadrant.

There are three different types of strings that haven't been discussed so far. *Symbolic fact strings* contain symbols and no attitudes (i.e. $goal$, $fault \wedge penalty$), *institutional fact strings* will be defined in section 5.1 and *composite fact strings* represent any string which is an *information string* but is not of any of the other three types.

5 Collective interior

In the previous section we have chosen a very generic model of a mind in order to keep the MASQ generality and we have shown how the classical mind cycle happens in the MASQ perspective. In this section we will use the $\mathcal{RL}$ representation language, whose main concepts are *symbols*, *attitudes* and *strings*, to provide a clear definitions of the concepts in the forth quadrant: *constitutive* and *regulative* rules, *groups* and finally *institutions*.

5.1 Count-as rules

Count-as rules, in the MASQ meta-model, provide the basis for common interpretation and common behavior for a group of agents. We will introduce a special attitude called *count-as attitude* and it will be denoted by " $\Rightarrow$ " $\in \Lambda$.

Definition 51. An attitude of the from $\Rightarrow ((\varphi_c) \rightarrow (\varphi))$ is called a *count-as rule*, where $\varphi_c \in \mathcal{RL}$ is called the *context* in which the rule can be applied and $\varphi \in \mathcal{RL}$ is called the *content* of the rule.

Count-as rules represent the core element of norms and institutions. Before going any further we need to detail what it means for a mind m to apply a *count-as rule*.

Definition 52. For a mind m a count-as rule r of the form $\Rightarrow ((\varphi_c) \rightarrow (\varphi))$ is *applied* if and only if whenever an instance $\varphi'_c = instance_C(\varphi_c) \in Data_m$ then we have $\varphi' = instance_C(\varphi) \in Data_m$. Otherwise it is said that that rule is *not applied*.

So, in other words, the above definition says that whenever an instance of the context (φ_c) exists so must the corresponding instance of the content (φ) of the rule (by corresponding we mean using the same $C = \{x0 : v0, x1 : v1, \dots\}$).

One important thing to remember is that "$\rightarrow$" does not mean "implication" as in logic. It is only a special symbol in the $\mathcal{RL}$ language.

Depending on the form of φ in definition 51 we can identify two types of count-as rules: *constitutive rules* as Searle defines them and *regulative rules* also called *norms*.

Constitutive rules

Constitutive rules allow the creation of *institutional facts* by interpreting the brute reality or other institutional facts.

Definition 53. A count-as rule in which φ has the form $\varphi = (\varphi_x) \rightarrow (\varphi_y)$ is called a *constitutive rule*. In this case, $\Rightarrow ((\varphi_c) \rightarrow ((\varphi_x) \rightarrow (\varphi_y)))$ corresponds to "X counts as Y in context C" in Searle's definition.

Constitutive rules are the basic elements for construction of an institutional reality. One important thing is the definition of an *institutional fact*.

Definition 54. A string of the form $(\varphi_x) \rightarrow (\varphi_y)$ is called an *institutional fact*.

Regulative rules

Regulative rules (*norms*), are used in multi-agent systems to regulate agents' behavior.

Definition 55. A count-as rule in which φ is a *deontic attitude* ($\varphi = \lambda(\varphi_d)$ and $\lambda \in \Lambda_D$) is called a *regulative rule* (*norm*).

5.2 Groups

When talking about groups there are two aspects that need to be considered. On one hand a group, in its classical sense, can only exist in the third quadrant, when minds interact through bodies and get to form a group. On the other hand we need to be able to tell what characterizes a group, besides the set of bodies of agents inside of it. Tuomela calls this the *ethos* of a group as in [13] : "*The ethos of group g in its strict sense is defined as the set of constitutive goals, values, beliefs, standards, norms, practices, and/or traditions that give motivating reasons for action*".

Definition 56. A set $e \subset \mathcal{RL}_\Lambda$ of *attitude strings* is called *ethos*. If $\lambda(\varphi) \in e$, where $\varphi \in \mathcal{RL}$ and e is the ethos of some group g, we say "group g has attitude λ over *information* or *attitude* φ" and write $\lambda_g(\varphi)$.

A mind m can become member of a group g with ethos e only if it *adopts* e. There are two ways a mind can adopt an ethos: a) $\forall\lambda(\varphi) \in e$ we have $\lambda(\varphi) \in Data_m$; b) $\forall\lambda(\varphi) \in e$ we have $\lambda g(\varphi) \in Data_m$.

The former corresponds to what Tuomela calls *I-mode groups* and the latter to *WE-mode groups*. Discussion of the difference between the two modes is not within the scope of this article.

5.3 Institutions

Inspired by Searle [10] who defines an institution as *"a set of count-as rules"* and Tuomela who says *"we can view institutions in terms of the ethos of a social group [...] ethos normatively directs the functioning of the group members, some of the norms being constitutive"* [13] we define an institution as follows:

Definition 57. An *institution* is an ethos that contains only *count-as rules*.

Due to space restrictions we will not develop on the use of groups and institutions in the MASQ model.

6 Conclusions

This paper is the first to give a clear formal description of the MASQ meta-model which was, until now, introduced and applied only in an informal manner.

Clear definitions have been given to the elements belonging to the exterior quadrants and we have shown how they are linked to the first quadrant through actions and data sensors. A simple and generic mind model has been used as a support to explain the classical perception, reasoning, action cycle in the MASQ perspective. Based on a simple representation language we have provided formal definitions for Searle's count-as relationship, constitutive and regulative rules, groups and institutions which are key elements of the cultural quadrant.

Having a formal description of MASQ will allow future works to create more specialized MAS models based on MASQ and also to analyze how the MASQ concepts relate to other existing MAS frameworks, especially in the collective quadrants.

References

1. A. Artikis, J. Pitt, and M. Sergot. Animated specifications of computational societies. In *AAMAS '02*, pages 1053–1061, New York, NY, USA, 2002. ACM Press.
2. G. Boella and L. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *IN PROCS. OF KR'04*, pages 255–265. AAAI Press, 2004.
3. V. Dignum. *A Model for Organizational Interaction: based on Agents, founded in Logic*. PhD thesis, Universiteit Utrecht, 2004.
4. V. Dignum, J. Tranier, and F. Dignum. Simulation of intermediation using rich cognitive agents. *Simulation Modelling Practice and Theory*, 18(10):1526 – 1536, 2010.
5. R. Dinu, T. Stratulat, and J. Ferber. A formal approach to MASQ (extended abstract). *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 1443–1444, May 2010.
6. J. Ferber and O. Gutknecht. Alaadin: a meta-model for the analysis and design of organizations in multi-agent systems. In *ICMAS'98*, pages 128–135. IEEE Computer Society, 1998.
7. J. Ferber and J.-P. Müller. Influences and reactions: A model of situated multi-agent systems. In *Proceedings of ICMAS'96*, pages 72–79. The AAAI Press, 1996.
8. M. Hiel, H. Aldewereld, and F. Dignum. Modeling warehouse logistics using agent organizations. In C. Guttmann, F. Dignum, and M. Georgeff, editors, *Collaborative Agents - Research and Development*, volume 6066 of *Lecture Notes in Computer Science*, pages 14–30. Springer Berlin / Heidelberg, 2011.

9. A. J. I. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logics*, 4(3):429–445, 1996.
10. J. R. Searle. *The Construction of Social Reality*. The Free Press, 1995.
11. T. Stratulat, J. Ferber, and J. Tranier. MASQ: towards an integral approach to interaction. In *AAMAS '09*, pages 813–820, 2009.
12. N. A. M. Tinnemeier, M. Dastani, and J.-J. C. Meyer. Roles and norms for programming agent organizations. In *AAMAS '09*, pages 121–128, 2009.
13. R. Tuomela. *The Philosophy of Sociality: The Shared Point of View*. Oxford University Press, 2007.
14. D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems : State-of-the-art and research challenges. In D. Weyns, H. Parunak, and F. Michel, editors, *Environments for Mutiagent Systems*, volume 3477 of *LNAI*, pages 1–47. Springer, 2005.
15. K. Wilber. An integral theory of consciousness. *Journal of Consciousness Studies*, 4(1):71–92, 1997.