Coupled MMASS: A Formal Model for Non-deterministic Multi-agent Simulations

Flavio Soares Correa da Silva[1] and **Giuseppe Vizzari** and **Alessandro Mosca**[2]

Abstract. The Multilayered Multi-agent Situated System (MMASS) has been proposed as a general framework to build multi-agent systems in which agents are situated in an environment whose characterization can be multifaceted (each facet of the environment is named a *layer*, hence the name of this framework). Agents in the MMASS can be purely reactive and computationally lightweight, or deliberative and employ highly sophisticated reasoning mechanisms. As a consequence, the MMASS has proven to be useful to build massively multi-agent systems (in which typically each agent is computationally simple) as well as systems comprised by complex agents (in which typically we have few agents interacting with each other and with the environment). In the present article we combine a simplified version of MMASS with a specific logical system, which we suggest that can be particularly suitable to solve problems based on simulations. The proposed logical system is a variation of classical FOPL, in which logical statements are tagged with probability values so that one can reason *with* probabilities as opposed of reasoning *about* probabilities.

1 INTRODUCTION

The Multilayered Multi-agent Situated System (MMASS) has been proposed as a general framework to build multi-agent systems in which agents are *situated* in an environment whose characterization can be multifaceted (or, using the MMASS parlance, *multilayered*). Agents interact with each other and with the environment by:

- sensing and emitting fields across the different layers of the environment;
- communicating directly with each other;
- acting on the environment e.g. by picking and dropping objects; and
- moving about [11].

A distinguishing feature of the MMASS framework is that it accommodates a wide range of agent architectures, from computationally lightweight, purely reactive agents to highly demanding, sophisticated deliberative agents. The flexibility of the MMASS framework makes it useful to build multi-agent systems comprised by heterogeneous agents ranging from complex deliberative architectures to simplified and computationally tractable ones.

The MMASS framework has proven to be particularly useful to model massively multi-agent systems, which can be used to analyze system configurations based on empirical simulations (see, e.g., [3, 2]).

The flexibility of the MMASS framework stems from its agent architecture agnosticism: the MMASS framework is not related to any particular agent architecture, and this is how it can accommodate a wide range of different agent architectures. The down side of it is that, in order to employ the MMASS framework for practical problem solving, one needs to complement it with suitable agent architectures.

In the present work we introduce a specific architecture for agents to be integrated with the MMASS framework, which is quite general and seems of particular interest to model complex systems whose analysis must be based on simulations.

Our proposed architecture is paired with a slightly simplified version of the original MMASS framework. It is a variation of classical FOPL in which logical statements are tagged with probability values. As a motivating example, if we have that $\alpha \rightarrow \beta$ and that α is tagged with a probability value of 0.2, then we will be able to infer β whenever α is observed, i.e. our logical system will sometimes infer β (when α is true) and will some other times *not* infer β (when α is false), since α is a premise to infer β and it is sometimes true (with probability 0.2) and sometimes false (with probability 0.8). In the long run, and assuming that no other proof path can infer β or influence on the probability of α, β will be inferred as true about twenty percent of the times we attempt to infer it.

Notice that our proposed non-deterministic logical system is appropriate to reason *with* uncertainty, as opposed to reasoning *about* uncertainty, i.e. we are not designing a system to reason about probability values, rather we are designing a system whose inferences are subject to probabilistic truth valuations [5]. Systems to reason with uncertainty can be more useful for simulation based analysis of problem solving mechanisms (and randomness is also a recognized element in dealing with the variability of human behaviours in simulation [4]), contrasting with systems to reason about uncertainty that can be more useful for theoretical analysis of problem solving mechanisms and systems.

This paper is organized as follows: in section 2 we briefly introduce the MMASS framework and compare it with other similar work found in the literature. In section 3 we simplify the original MMASS model and put this simplified version together with our proposed non-deterministic system for deliberative agents. In section 4 we build a simple illustration, to show our proposed model in action. Finally, in section 5 we present some discussion, conclusions and proposed future work.

2 THE MMASS FRAMEWORK

The MMASS framework was fully introduced in [11], although some partial renditions of this framework had been published before that.

---

[1] University of Sao Paulo, BRAZIL, email: fcs@ime.usp.br

[2] University of Milano-Bicocca, ITALY, email: { giuseppe.vizzari, alessandro.mosca }@disco.unimib.it

It has been employed successfully in a variety of problems, such as the modeling of human interactions in Computer Supported Cooperative Work [8], simulation of crowd behavior [1] and ubiquitous computing [7].

Very briefly, a MMASS is a collection of *layers*. Each layer is comprised by an undirected graph, in which nodes represent locations in which agents can be placed (each node admits at most one agent) and edges represent accessibility between nodes. Agents can move across nodes following their accessibility relations, and in the original MMASS model agents can only exchange messages with other agents located in neighboring nodes.

In the original MMASS model, there exist *interfaces* connecting different layers. Essentially, an interface is an edge connecting nodes from two different graphs.

An agent, when located in a node, can sense the environment and perceive fields that are reaching that node, receive messages from neighboring agents, and then, based on internal states and deliberative capabilities, decide to move to a different node, act upon the environment, emit fields and send messages to agents.

A *field* is characterized by its source, initial intensity, diffusion and dispersion. A field source is the node from which it is being emitted; the initial intensity is typically represented as a non-negative real number; diffusion determines the intensity of the field as a function of the topological distance between any node and the field source; and dispersion determines the intensity of the field at the source node as a function of the time since the field was emitted.

The power of the MMASS framework stems from its flexibility, which is a direct consequence of its simplicity. In order to be applied in concrete situations, it must be complemented by specific agent models that determine agent behaviors based on sensed fields and received messages. Frequently, the specification of agent behaviors comes together with some sort of specialization of the generic MMASS framework, to build special cases of interest to particular applications.

In the present work we propose a partial specialization of the MMASS framework, thus building a special case of the generic framework that can be particularly well suited to model complex multi-agent systems whose dynamics must be analyzed based on empirical discrete-event simulations.

Our proposed partial specialization of the generic MMASS framework is built in order to couple it with a particular non-deterministic inference system, as detailed in the following section.

3 COUPLING MMASS WITH A NON-DETERMINISTIC LOGICAL SYSTEM

Our goal is to put the MMASS framework together with a specific agent model, to build a specialized framework for simulation of complex multi-agent systems. Essentially, we have built a normative theory based on which agents can infer obligations and permissions, i.e. logical statements representing actions that they are either required to perform (obligations) or allowed to perform (permissions). We intend this normative theory to work as a foundation upon which concrete theories of action for agents can be built. We also intend that the normative theory, heretofore referred to as *Coupled MMASS*, can be implemented to support runs of an agent system, i.e. concrete simulations that can be used to analyze empirically the behavior of complex systems.

Permissions and obligations connect naturally to *deontic logics* [9], a particular class of multimodal logics that have been historically used to model legal systems, and more recently have been employed to model interactions in multi-agent systems [10]. We have explored deontic logics as a formalism to model multi-agent systems for digital entertainment, and obtained rather appealing results [6].

For the purpose of the simulation of the dynamics of interactions among agents in a multi-agent system, however, deontic logics may not be the most appropriate formalism, as proof systems for even relatively simple deontic logical systems can become computationally intractable. For this reason, we have preserved the *notion* of permissions and obligations and explored simpler logical systems, that can be less computationally demanding and therefore more useful for simulations of practical systems.

We have extended the language of classical FOPL with *deontic tags* as follows. We have added to the alphabet of FOPL the tags **O** (standing for *obligation*) and **P:**φ (standing for *permission*), in which φ is a real number in the interval $[0, 1]$.

Classical FOPL sentences can be tagged with zero or one deontic tags. Operationally, when an **O** tag is found, the corresponding sentence is forcefully true, i.e. **O** tags can be ignored in the course of a proof. When a **P** tag is found, a random number generator is trigged to generate a random value in the interval $[0, 1]$ based on a uniform probability distribution. If the generated value is larger than φ, the tagged sentence is evaluated as false, otherwise it is true.

Intuitively, obligations are necessarily true, and permissions can be either true or false, according to a probability estimate.

Using this logical system, well formed sentences are sometimes labeled as theorems and some other times labeled as non-theorems for the same logical theory. Logical proofs are non-deterministic, depending on the interdependence of well formed sentences and the behavior of permitted sentences, which is determined by the values φ.

We have envisaged this non-deterministic formal system to model the actions of and interactions among agents in multi-agent systems. We now couple this system with MMASS.

Before putting the non-deterministic FOPL extended with permissions and obligations and MMASS, we simplify the original MMASS in three ways, to make it more specifically applicable for simulation models.

1. We add an explicit notion of *time*, based on the simplest possible model of time, namely discrete linear time with one starting point. Since we aim at simulations and animations, we add a uniform and very simple representation of time in the model. Time is represented as clock ticks, which are synchronized for all entities and layers, i.e. we have a universal clock accessible to any element of the model. A clock tick is a natural number.
2. We enforce that all layers have the same topological structure. In other words, we have a single *layer structure* i.e. an undirected graph - that is applied to all layers in an MMASS model. The edges of the layer structure are tagged with real, non-negative values, with the addition of the symbol $\perp$ that are representations of costs to move between nodes and for fields to propagate. The tags can vary across layers.

   Edge tags are abstractions of some appropriate notion of distance. For example, one of the layers can represent geometrical distances between nodes, and tags can represent these distances using some uniform distance unit measure. This is the reason why we assume that the tags are non-negative real numbers. Notice that we leave on purpose the possibility of having zero as a legal tag. This can be used to collapse nodes in a layer without having to change the layer topology. The symbol $\perp$ represents instead an infinite distance, meaning that agents cannot move from one of the connected sites to the other, and also fields do not diffuse across the edge.

3. We assume that one layer represents the physical layer, on which agents can move, and that the movements of an agent imply on travel costs that are calculated independently for each layer. Agents move across the layer topology, and therefore
   (a) When an agent moves from one node to another, it does so in *all* layers; and
   (b) It makes no sense for an agent to migrate between layers, since agents do not move from the physical layer. Therefore, we do not need to have explicit interfaces between layers.

This basic structure for agents' environment supports the representation of composite situations, characterized by different aspects that may influence agents' behaviours. In particular, Figure 1 depicts a three–layered environment comprising (i) a basic grid whose edges represent airline costs of movement between sites, (ii) an additional layer in which edges represent the actual possibility to move from a site to another (e.g. a map of a building including walkable space, passages and walls) that could be used to effectively constraint agents' movements, but also (iii) a layer connecting "homogeneous" sites (e.g. indoor or outdoor sites) with edges of null cost, creating areas in which fields diffuse uniformly, for instance to support agents in distinguishing the different areas.

For each layer we also have the definition of a finite collection of *fields*. A field is determined by four elements:

1. A source node p_0, i.e. the identification of a node in the layer topology;
2. A starting moment t_0, i.e. a natural number;
3. An initial intensity f_0, i.e. a real, non-negative number;
4. A field equation, i.e. any mathematical relation that univocally determines the intensity f of the field at any node p and at any moment $t \geq t_0$, based on the topological distance between p and p_0. As a simplifying assumption for Coupled MMASS, we assume that fields are represented as real non-negative values that depend primarily on time interval and topological distance between nodes.

Fields and edge tags are the available resources in this framework to model *context*.

We now add entities to our model. Entities can be of two basic sorts:

1. Objects: an object is any inanimate entity, i.e. an entity that does not perform actions, does not emit fields and does not move by itself. We can have an arbitrary number of objects sitting on any node.
2. Agents: an agent is an entity that performs actions including actions that interact with objects emits fields and moves across the edges of the layer topology. We can have at most one agent on a node at any given moment.

An agent can tune to a layer and sense the fields that are hitting it at the node where it is. Fields are composed based on specified composition laws. Fields interact only with other fields at the same layer. An agent can also communicate with other agents. We simplify and relax a little the conditions imposed in the original MMASS, and allow that agents communicate independently of location (in the original MMASS, agents could only interact including communicate with other agents at neighboring nodes). An agent can interact with objects which are at the same node where it is, performing operations allowed for each object (open a box, kick a ball, pick an object, drop an object, etc.). An agent can emit fields the source node is necessarily the node where the agent is, the starting moment is the moment in which the agent decides to start emitting the field, the initial intensity can be arbitrarily selected by the agent, and of course the field type must be defined in the layer in which the agent is emitting the field. Finally, an agent can move to a neighboring node following the layer topology.

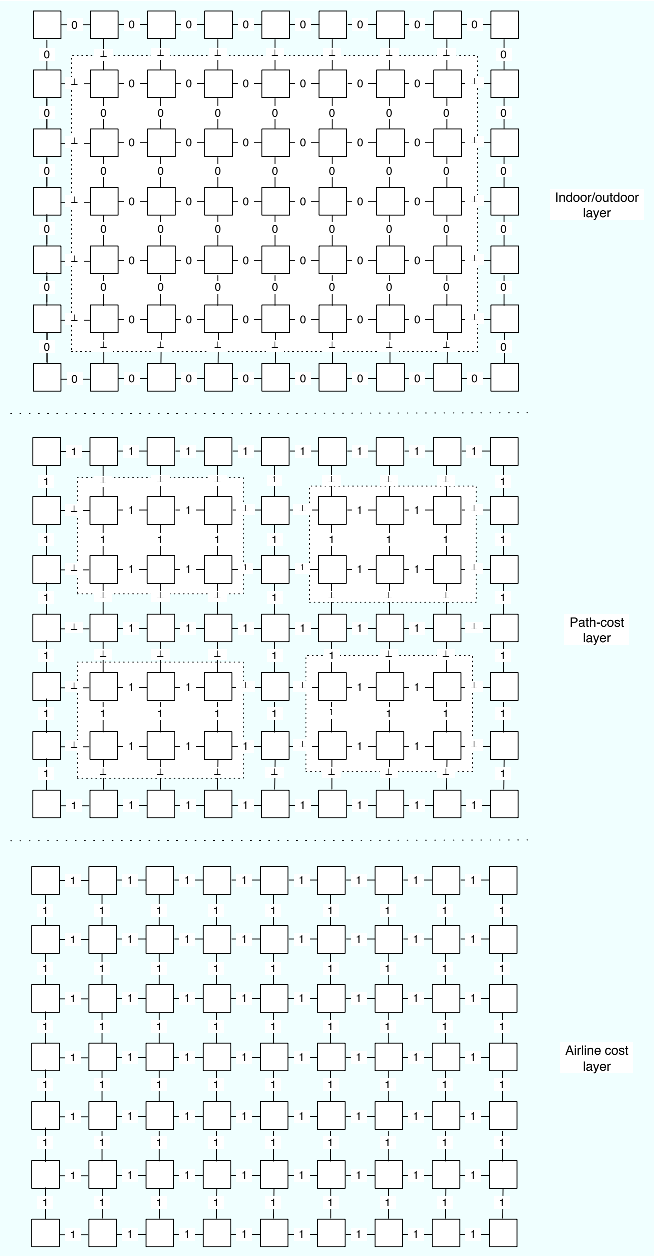

Figure 1. A sample multilayered structure of a coupled MMASS environment.

At a time t an agent can sense the fields that are hitting it and receive messages from other agents, trigger the appropriate internal mechanisms to reason about its state, goals, history, etc. and decide a course of actions, which can include messages sent to other agents, interactions with inanimate objects, emission of fields and displacement. As a consequence, at time $t+1$ the world must be updated: fields are updated, the internal states of agents are updated, messages reach their recipients, the effects of interactions with objects are registered, and displacements occur. If some sort of conflict occurs regarding displacements, e.g. if two agents decide to go to the same node at time t, some sort of *conflict resolution* is triggered. A very simple form of conflict resolution can be as follows: one layer is elected to be the referee (e.g. the one representing geometrical relations), and in case of conflict the agent with shortest label (i.e. the one that is closer to the target node) wins the race, and the other agent(s) stay where it(they) is(are). If there is a draw (e.g. if two agents are equally distant from the target node) then one agent is arbitrarily chosen to win the race.

In very few words, it is a dynamic model of agent interactions together with the structuring of space provided by the layer topology and contextual information provided by the fields emitted in each layer. Some standard tricks can be used to provide agents with useful information, e.g. we can build lamp posts, which are agents that never move and continually emit a special sort of field. Based on lamp posts, the other agents can locate themselves geographically.

Operationally, we have a new FOPL theory built for each time t. Based on this theory, the agents reason and decide what actions they shall be obliged and permitted to take. Permitted actions are followed by their corresponding probabilities. Once these actions are determined, they actually take place, thus changing the environment and updating the status of every agent for time $t + 1$ - formally, this is accomplished by updating the FOPL theory of time t so that it becomes the correct theory for time $t + 1$.

We believe that the Coupled MMASS has great potential for applications related to digital entertainment (games, feature movies), ubiquitous computing, and simulations in general.

In the next section we illustrate through a simple example the Coupled MMASS in action.

4 THE COUPLED MMASS IN ACTION

In this section we briefly illustrate the operations of the Coupled MMASS, through a simple example. The example is a simple computer game, in which the computer plays against itself and we can watch the evolution of a match.

We have a five-by-five board, and two teams of two agents each, depicted as **x** and **o** as presented in Figure 2.

The goal of the **x** team is to reach the rightmost column of the board before any agent from the **o** team reaches the leftmost column of the board, and vice-versa. The movements of each agent, however, are constrained by the following rules:

1. Any agent is *permitted* to emit a field at any moment. An emitted field lasts for one clock tick and reaches the whole board. Given that each agent is *permitted* to emit a field, we must also define the probability that each agent actually does so. In our example, we fix it as a probability of 0.2.
2. If an agent of the opposite team is emitting a field, then an agent is *obliged* to stay at the same position for one clock tick.
3. If no agent from the opposite team is emitting a field and the position immediately ahead of an agent is vacant, then the agent is *permitted* (with probability 0.5) to move to that position.
4. If no agent from the opposite team is emitting a field and the position immediately ahead of an agent is occupied, then the agent is *permitted* (with probability 0.5) to move one position sideways, always to its right, and only if the position at its right is vacant. The right of an **x** agent is one position below it, and the right of an **o** agent is one position above it.

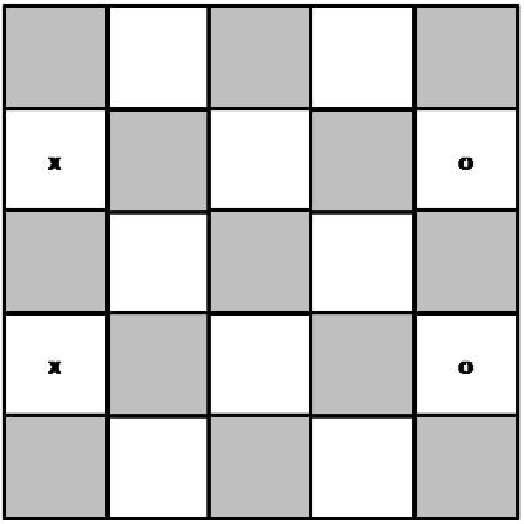

Figure 2. Initial configuration of five-by-five board.

The layer topology for the corresponding Coupled MMASS model for this simple board game coincides with the board itself, i.e. each position is a node, and we have edges connecting each node with its neighbors above, below, to the right and to the left.

We have one physical layer, which corresponds to where the agents are, and one field layer, which is used to represent the existing fields at any clock tick.

We have built a simple (and rather roughly finished) *PROLOG* program to run simulations of this game. The game depicts a radically simple world in which agents behave according to a few general laws and regulations. Despite its simplicity, it can be surprisingly entertaining to watch the non-deterministic simulations.

Our goal with this simple example is to show the potential of the Coupled MMASS framework to build models for the applications referred to in the previous sections.

5 DISCUSSION AND FUTURE WORK

In this article we have introduced a variation of the MMASS framework, coined the Coupled MMASS, in which a simplified version of the original MMASS framework is integrated with a variation of FOPL with non-deterministic proof rules characterizing permissions and obligations of agents to behave according to laws and regulations.

We envisage that the Coupled MMASS framework can be useful to model massively multi-agent systems in which the behavior of agents is governed by complex rules, thus producing complex system behavior that can be best analyzed through computer-based simulations.

Our future work concerning the Coupled MMASS shall focus on three issues:

1. We shall adjust some fine grained details about the framework (e.g. the detailed characterization of proof rules for this framework) and study more carefully some remaining formal issues (e.g. a model theoretic semantics for the framework).
2. We shall work out efficient implementations for the Coupled

MMASS, so that it can be effectively employed on practical applications.

3. We shall implement solutions for practical applications, e.g. related to the fields referred to throughout the present article.

ACKNOWLEDGEMENTS

The first author has received financial support from the Brazilian funding agencies CNPq and FAPESP.

REFERENCES

[1] S. Bandini, S. Manzoni, and G. Vizzari, 'Situated cellular agents: A model to simulate crowding dynamics', *IEICE Transactions on Simulation and Systems: Special Issue on Cellular Automata*, **E87-D**(3), 669–676, (2004).

[2] Stefania Bandini, Mizar Luca Federici, and Giuseppe Vizzari, 'Situated cellular agents approach to crowd modeling and simulation', *Cybernetics and Systems*, **38**(7), 729–753, (2007).

[3] Stefania Bandini, Sara Manzoni, and Giuseppe Vizzari, 'Multi–agent approach to localization problems: the case of multilayered multi agent situated system', *Web Intelligence and Agent Systems*, **2**(3), 155–166, (2004).

[4] Michael Batty, *Advanced Spatial Analysis: The CASA Book of GIS*, chapter Agent-based pedestrian modelling, 81–106, Esri Press, 2003.

[5] F.S. Correa da Silva, *On Reasoning With and Reasoning About Uncertainty in Artificial Intelligence*, European Summer Meeting of the Association of Symbolic Logic, Spain, 1996.

[6] F.S. Correa da Silva and W.W. Vasconcelos, *Rule Schemata for Game Artificial Intelligence*, 2006.

[7] M.P. Locatelli and G. Vizzari, 'Awareness in collaborative ubiquitous environments: The multilayered multi-agent situated system approach', *TAAS*, **2**(4), (2007).

[8] C. Simone and S. Bandini, 'Integrating awareness in cooperative applications through the reaction-diffusion metaphor', *Computer Supported Cooperative Work*, **11**(3-4), 495–530, (2002).

[9] A. Valente, *Legal Knowledge Engineering: A Modeling Approach*, IOS Press, The Netherlands, 1995.

[10] J. Vazquez-Salceda, H. Aldewereld, and F. Dignum, *Implementing Norms in Multi-agent Systems*, 2004.

[11] G. Vizzari, *Dynamic Interaction Spaces and Situated Multi-agent Systems: from a Multi-layered Model to a Distributed Architecture*, PhD Thesis, University of Milano-Bicocca, 2005.