A Unified Model for Physical and Social Environments

José-Antonio Báez-Barranco, Tiberiu Stratulat, and Jacques Ferber

LIRMM
161 rue Ada, 34392 Montpellier Cedex 5, France
{baez,stratulat,ferber}@lirmm.fr

Abstract. The AGRE model by Ferber *et al.* is based on an interesting generalization of both physical and social environments. In this paper we revisit the AGRE model and extend it with richer social concepts such as powers, norms and a dependency relationship similar to the *count as* operator introduced by Searle to describe the construction of social reality. Our main contribution consists in the fact that we attribute to the environment the main role in describing and controlling the (social) interaction. [1]

1 Introduction

In the area of multi-agent systems (MAS), Castelfranchi claimed [1] that *social order*, which is a (social) metaphor for the problem of coordinating the agents or organizing the interactions among them while preserving their autonomy, could be obtained by using social concepts such as norms and social control. Norms are rules describing the expected ideal behavior of an agent or of a group of agents. Social control means that the agents themselves observe the behavior of the other agents, check if they are norm compliant and act consequently. Recently many research works [2–6] proposed models that integrate social and organizational concepts in MAS and sugested tools to implement the social metaphor. However, most of them propose *ad-hoc* solutions of how social concepts are constructed and then manipulated. For instance, there is not always very clear if the social knowledge (i.e. about the obligation to do an action, the power of doing an act, the membership to an institution, etc.) is shared among agents or is represented somehow externally and independently of them.

In previous works we also studied how to integrate in MAS the organizational concepts of group and role [7] and proposed the AGRE model [8]. The AGRE model is based on the idea that the environment could be used to represent not only the physical part of the interaction but also its social aspect. The agents interact only with the environment which will react according to agent's influences [9] and to the rules of change defined at both physical and social levels of interaction.

In this paper we present the AGREEN model which is a revisited extension of the AGRE model. Its main goal is to provide a much simpler and unified way of representing (physical and social) environments. The originality of our work consists in the fact that it attributes to the environment the main role in describing and controlling the (social) interaction. This is the major difference when compared with some related works that use social concepts [3, 4, 6].

[1] This work has been supported by France Telecom R&D.

The AGREEN model is based on a clear separation between what an agent tries to do and the effects obtained as independent consequences of its acts on the environments. The architect of an agent concentrates only on the internal structure of the agent, that is on the design of the decision making mechanisms that help the agent to know what to do next. The architect of the system describes the environment as a set of rules governing the interactions, completely ignoring how the agents are constructed. The main benefit of this separation is that it guarantees the autonomy of the agents and the non-intrusive control of their behavior.

When describing the effects of the agent's influences on the environment, there is also a clear distinction between what an agent can do, as capacities, and what an agent is supposed to do, as deontic constraints (such as obligations, permissions or interdictions). If the description of the capacities of an agent is compulsory, the description of deontic constraints is optional.

Another advantage obtained from the separation agent-environment is that the semantics of an action could be given respecting both aspects: internal (agent's point of view) and external (environment's point of view). The external semantics of an action can be further expressed at both physical and social levels. This is a major step forward in giving agent communication languages a public perspective and a social semantics, since they are requested by the agent community [10].

In the rest of the article we revisit the AGRE model and show that it could be simplified and enhanced with richer social concepts such as powers, norms and a dependency relationship which is similar to the *count as* operator introduced by Searle to describe the construction of social reality [11]. Finally, we propose a more general abstract architecture that integrates the new concepts and illustrate two implementations for MadKit and Jade platforms.

2 Social reality and AGRE

In this section we describe some social concepts such as norms and social reality that were announced in the original paper of AGRE but which deserve much more attention.

Social reality The work of Searle on the construction of social reality [11] is becoming very influencing on the research in agent based systems [2, 3]. The main idea is that a social institution, even that it has no physical support, has its own (social) reality and is constructed by mutual convention on how to interpret what happens in the physical reality. Searle makes the distinction between *brute* facts and *institutional* facts. A brute fact represents something true in the physical reality (i.e. a piece of paper with a €10 sign marked on it). An institutional fact is a fact that is considered to be true only locally to an institution (i.e. money such as a ten euro banknote).

Searle considers also that an institution is defined mainly in terms of two types of rules: constitutive and normative. Constitutive rules show how to construct the social reality by giving an interpretation to brute facts or other social facts through the use of the *count as* operator. For instance a piece of paper with €10 special printings on it counts as a ten euro banknote in the money institution. Jones and Sergot [2] give a formalization of *count as*, and present the concept of institutionalized power as being the (social) capacity to act in an institution. Normative rules describe ideal situations or behaviors from the point of view of an institution.

The AGRE model In [8], Ferber *et al.* propose an extension of the AGR model [7] and consider an organization as being a special kind of environment. Social actions are associated with an organization, i.e. playing a role, entering and leaving a group, communicating inside a group, etc. The main ideas presented in this work concern 1) the use of both social and physical environments to describe the interaction among agents; 2) the concept of *space* which is a generalization of the concepts of physical area and social group, introduced to partition the environment; and 3) the concept of *mode*, which is a generalization of the concepts of physical body and social role, used to describe the agent's capacities to influence [9] physical and respectively social environments.

However, the AGRE model presents some inconveniences. First, the generalized concepts of space, mode and institution show very well the relationship that should exist between an agent and an environment, but they remain abstract and unused. Moreover, like in AGR, there is no explicit description of the expected behavior of the agents, i.e. a role is simply a label with no other semantics. Normally we should be able to associate to a role powers and deontic constraints such as obligations, permissions or interdictions. Finally, AGRE in its original form did not take into account the ideas on social reality by Searle. What misses in AGRE is something similar to the *count as* relationship that links together physical and social environments or more generally any two environments.

3 The AGREEN Model

In this section we propose to improve and generalize the concepts introduced initially in AGRE, that is, we propose (i) to use only the generalized concepts of space, mode and capacity, (ii) to better explain the generalization of physical and social properties by a unified concept, (iii) to better explain the role of the environment from the point of view of behavior control, (iv) to give more details on the role of modes as capacities to act in an environment, and (v) to try to generalize the relationship existing between physical and social environments.

Space The aim of a space is to describe how the environment change as a consequence of the agents' influences. Its role is to simulate the physical and the social environment in a multi-agent system. A space normally exists only at the execution time and it is characterized by a name, an initialization type and an actual type. A space is composed by objects and could be linked to other spaces. Objects could be of three types: ordinary objects, modes and recursively other subspaces.

Definition 1 (Space). *A space S is a tuple $<Id_S, IST, CST, S^*, M^*, O^*>$ where: Id_S is the space's Id that uniquely identifies it, IST is a space type used at the initialization phase, CST is the current type of the space, S^* is the set of subspaces contained in S, M^* is the set of modes contained in S, and O^* is the set of objects existing at a certain time in S.*

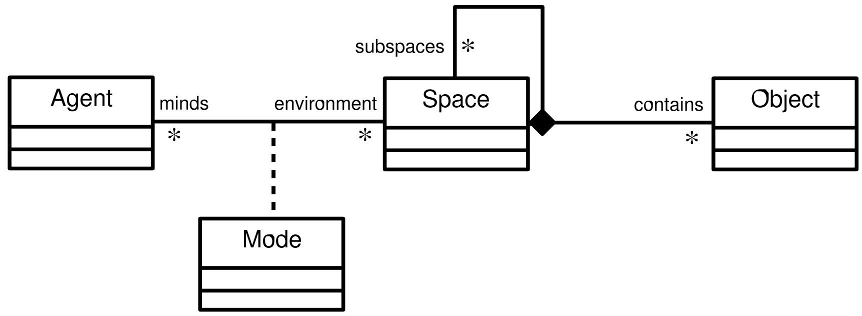

Fig. 1. Simplified UML representation of AGREEN

The type of a space is a concept similar to that of a class in object-oriented programming and contains the description of common properties and behaviors of identical instances, that is concrete spaces. A space type defines how its instances can be composed of subspaces, modes and other objects. This description is realized by giving to each kind of component a list of types whose instances are accepted in the instances of the space type. However, the changes of a space are more dramatic than those of objects, since we let a space change dynamically its type at the execution time, and possibly it will no more correspond to its initial form. Therefore, the initialization space type IST of a space is used only at the space's creation time. It helps the concrete space to build its internal structure and initialize its attributes with default values. At the initialization time, the current space type CST is identical with IST, but it can evolve after that and become different.

Definition 2 (SpaceType). *A space type ST is a tuple $<ST^*, MT^*, OT^*, DR^*, C_S^*>$ where: ST^* is the set of space types that could be instantiated as subspaces in the instances of ST, MT^* is the set of mode types that could be instantiated in the instances of ST, OT^* is the set of objects types that could be instantiated in the instances of ST, DR^* is the set of dependency rules that link instances of ST to other spaces, and C_S^* is a set of environmental constraints for the environmental control.*

The dependency rules should be seen as another set of rules of change since they show how an environment modifies its properties according to some external influences that are produced this time by other environments. A dependency rule is a sort of link between two environments that introduces constraints of various natures: causal (some external events produced in other environments are the cause of a local event), logical (a local property is the logical consequence of some external properties), social (like *count as*). For instance, a physical environment space is a space with an empty set of social dependency rules.

Object The aim of objects is to partially encapsulate the internal state of the environment and the laws that govern its change. An object could exist only at the execution time. It is characterized by being of a certain object type and a current state.

Definition 3 (Object). *An object* O *is a tuple* $<Id_O, OT, T^*_O>$ *where:* Id_O *is the object's name that uniquely identifies it;* OT *is the object type; and* T^*_O *is the set of attributes of the object.*

The type of an object is a concept similar to that of a class in object-oriented programming and contains the description of common properties and behaviors of identical instances, that is concrete objects. An object type describes the possible states of its instances and how their states evolve under agents' influences.

Definition 4 (ObjectType). *An object type* OT *is a tuple* $<TD^*_O, D>$ *where:* TD^*_O *is the set of attribute declarations of the instances of this object type; and* D *is a description of how the instances of this object type evolve under the agents' influences.*

Mode There are mainly two reasons for which we introduced the concept of mode: 1) to allow the space to individually attribute capacities to agents; 2) to allow the space to specify the expected behavior by using social deontic constraints. We propose to use the term *capacity* to describe the unifying concept of physical capacity and social power. A capacity is associated to a mode and a space and defines the possibility of its owner to modify the space at the execution time. A mode is characterized by being of a certain mode type and a set of attributes. More precisely, the role of modes is to allow the space to attribute - individually - capacities to agents.

Definition 5 (Mode). *A mode* M *is a tuple* $<Id_M, MT, A, T^*_M, O^*>$ *where:* Id_M *is the mode's name that uniquely identifies it,* MT *is the mode type,* A *is the owner's agent identifier,* T^*_M *is a set of attributes, and* O^* *is a set of deontic constraints.*

Definition 6 (ModeType). *A mode type* MT *is a tuple* $<P^*, TD^*, C^*_M, N^*>$ *where:* P^* *is a set of capacities (or powers) that the instances of this mode type will offer to their owner,* TD^*_M *is a set of attribute declarations,* C^*_M *is a set of conditions that should be fulfilled by the agent demanding to obtain a mode in an space or to release it, and* N^* *is a set of norms that describe the conditions of apparition of a deontic constraint that applies to a mode only internally to an institution.*

The types related to modes play the same role as classes in object-oriented programming. They are an abstract description of the internal structures of the mode and of the operations that could be executed on them to change their state. The set of conditions C^*_M should be verified on the agent at the creation of its mode or rechecked later to see if the agent still posses the necessary conditions to continue to interact with the environment. The set of capacities P^* and the set of norms N^* are transferred to the mode at the creation time, then they possibly dynamically change.

The role of capacity rules is to implement the necessity and impossibility properties of the interaction. When an agent influences the environment, the capacity rule triggered on its mode is immediately executed by the environment. A capacity rule should be seen as additional preconditions on the influences. As shown before, a mode gives to its agent the possibility to act (or not) in an environment. The capacity rules are mainly employed to externally control the behavior of the agents, since they have an impact only on their bodies while preserving their autonomy.

The norms reflect the deontic aspect of interaction. They are used for social control. As shown in the previous section the deontic aspect refers only to social interaction. A deontic property describes the social obligation or interdiction to do something or to arrive in a certain state of affairs. We note here that the deontic constraints that we think of, don't represent general deontic properties as in standard deontic logic, but directed deontic properties. Like capacities, a deontic constraint is always connected to a mode. Deontic properties are mainly employed to externally influence the behavior of the agents. A physical mode, i.e. a body, is only a mode with an empty set of norms.

4 Conclusion

The AGREEN model described in this paper is a revisited extension of the AGRE model. Its main goal is to provide a much simpler and unified way of representing (physical and social) environments. The model is based on 1) a clear separation between what an agent tries to do and the effects obtained as independent consequences of its acts on the environments and 2) a clear distinction between what an agent can do, as capacities, and what an agent is supposed to do, as deontic constraints. When modeling the interaction, the only difference between physical and social environments is that physical environments don't posses deontic constraints and social dependency relationships. We note here that the social dependency relationship, which is similar to the *count as* operator, is not formalized in this paper and will be the subject of future works. Finally, we implemented the various institutional concepts introduced for AGREEN as a service on the agent platforms MadKit [12] and Jade [13].

References

1. Castelfranchi, C.: Engineering social order. In: ESAW'00. Volume 1972 of LNCS. (2000)
2. Jones, A.J.I., Sergot, M.: A formal characterisation of institutionalised power. Journal of the Interest Group in Pure and Applied Logics **4**(3) (1996) 429–445
3. Boella, G., van der Torre, L.: Structuring organizations by means of roles using the agent metaphor. In: WOA'04. (2004)
4. Vázquez-Salceda, J., Dignum, V., Dignum, F.: Organizing multiagent systems. Autonomous Agents and Multi-Agent Systems **11**(3) (2005) 307–360
5. Stratulat, T., Clérin-Debart, F., Enjalbert, P.: Norms and time in agent-based systems. In: Proceedings of ICAIL'01, USA (2001)
6. Artikis, A., Pitt, J., Sergot, M.: Animated specifications of computational societies. In: AAMAS '02, New York, NY, USA, ACM Press (2002) 1053–1061
7. Ferber, J., Gutknecht, O.: Alaadin: a meta-model for the analysis and design of organizations in multi-agent systems. In: ICMAS'98, IEEE Computer Society (1998) 128–135
8. Ferber, J., Michel, F., Báez-Barranco, J.A.: AGRE : Integrating environments with organizations. In: E4MAS'05. Volume 3374 of LNCS., Springer (2005) 48–56
9. Ferber, J., Müller, J.P.: Influences and reaction : a model of situated multi-agent systems. In: ICMAS'96, The AAAI Press (1996) 72–79
10. Singh, M.P.: Agent communication languages: Rethinking the principles. Computer **31**(12) (1998) 40–47
11. Searle, J.R.: The Construction of Social Reality. The Free Press (1995)
12. The MadKit Project (a Multi-Agent Development Kit). http://www.madkit.org (1998)
13. JADE (Java Agent DEvelopment Framework). http://jade.tilab.com (2000)