

# Towards Verification of Multi-Agent Systems

Pablo Gruer  
UTBM site de Sévenans  
4 rue du château  
90010 Belfort Cedex, FRANCE  
pablo.gruer@utbm.fr

Vincent Hilaire      Abder Koukam

## Abstract

*This paper presents a verification approach for multi agent systems. The specification is based upon Object-Z and uses the influence/reaction model. The verification process consists in the transformation of Object-Z specifications into transition systems. This allows us to verify automatically some properties of the specification such as history invariants expressed by temporal logic formulas.*

## 1. Introduction

Although many Multi-Agent Systems (MAS for short) applications have been implemented, there is a crucial lack concerning design and development methodologies. Specification is fundamental to handle the complexity related to describing precisely the desirable behavior of MAS before their implementation phase. The specification process must fulfill two roles. The first is to provide the underlying rationale for the system under development. The second is to guide subsequent design, implementation and verification phases. A variety of specification formalisms are available in the multi-agent field. Such formalisms put the emphasis on the first role but fail to provide a basis to fulfill the second. As stated in [1], they are often abstract and unrelated to concrete computational models. We believe that one way to bridge the gap between the abstract and concrete level is to build the specification of systems using a formal specification language which on one hand enables verification of MAS specifications and on the other hand can be refined down to an implemented computer system.

The purpose of this paper is to present a formal approach based upon Object-Z (O-Z for short) for MAS specification that allows verification of properties even at high stages in the specification process.

## 2. Specification and verification approach

### 2.1. Specification framework

We now present a general framework to guide the specification of systems of reactive, situated agents, based on the I&R theory [2]. The framework presents as a set of three O-Z classes, one for agent objects, other for passive objects and the third for multi agent systems that integrate agents and passive objects. While agents possess an internal state and produce reactions according to behavioral functions that depend on their internal state, passive objects react according to laws that do not depend on the past history endured by the object. Typically, passive objects describe environmental reactions to agent actions, named influences. We present now the principles of *AgentObject* class which acts as a general framework for an agent specification. First come constant declarations and type definitions, The axiom schema allows to introduce constraints on constants and types, expressed as predicates. The state schema allows to declare the attributes of the agent, together with predicates to express attribute properties. Let us consider the general O-Z class framework for passive objects, called *EnvironmentObject*. Keeping close to the I&R theory, attribute declarations, which are made using state schemas, can be used to introduce, in their predicative part, the so called environment laws. An *EnvironmentObject*, like any *AgentObject*, includes two kinds of operations. On one side, percept operations produce environment percepts, aimed to become the content of agent perceptions. On the other hand, law operations describe how the value of environment attributes change as a result of agent actions, according to environment's laws.

Once agents and environment have been specified, the time comes to give indications about their integration into a MAS. The *MultiAgentSystem* class results from the aggregation of an environment and a set of agents.

## 2.2. From Object-Z classes to transition systems

A class specification written in O-Z can be seen as the description of an abstract machine. Objects of the class are instances capable of producing computations of the machine, as sequences of state changes caused by operations. The aim of the proposed transformation is to obtain, from the definition of the class, written in O-Z, a compact representation of the set of computations an object of the class can produce. This representation is intended to take the form of a transition system. The intended result of the transformation is a triple  $(S, \mathcal{A}, \mathcal{H})$  where  $S = \langle \mathcal{V}, \Sigma, \mathcal{T}_e, \mathcal{T}_c, \Theta \rangle$  is a transition system,  $\mathcal{A}$  and  $\mathcal{H}$  are sets of linear temporal logic (LTL) formulas called respectively axioms and history invariants. An O-Z class named  $cl$  is characterized by the following sets.  $Attr(cl)$  is a set of class attributes, declared in the *state schema*,  $Param(cl)$  is a set of operation parameters. Both are subsets of a set of identifiers such that  $Attr(cl) \cap Param(cl) = \emptyset$ .

$State(cl) \subseteq Attr(cl) \leftrightarrow Val$  is the set of class states, a subset of the finite partial functions from attributes to values. The set  $Val$  contains all possible values of attributes of any type. Finally,  $Op(cl)$  is the set of class operations. We introduce the auxiliary functions  $\pi_i : Op(cl) \rightarrow \mathbb{P} Param(cl)$  and  $\pi_o : Op(cl) \rightarrow \mathbb{P} Param(cl)$  that give respectively the set of input and output parameters of an operation.

We explain now the translation of a simple O-Z class named  $cl$ , towards a model  $(S^{cl}, \mathcal{A}^{cl}, \mathcal{H}^{cl})$ , with  $S^{cl} = \langle \mathcal{V}^{cl}, \Sigma^{cl}, \mathcal{T}_e^{cl}, \mathcal{T}_c^{cl}, \Theta^{cl} \rangle$ . We state that:

$$\mathcal{V}_p^{cl} = Attr(cl), \mathcal{V}_i^{cl} = \bigcup_{o \in Op(cl)} \pi_i(o), \mathcal{V}_o^{cl} = \bigcup_{o \in Op(cl)} \pi_o(o)$$

so that, for the *Game* class we have  $\mathcal{V}_p^{Game} = \{heap, turn\}$ ,  $\mathcal{V}_i^{Game} = \{take?\}$  and  $\mathcal{V}_o^{Game} = \emptyset$ .

State schema of the form  $[declaration\ part \mid Ax_1; \dots; Ax_n]$  declares attributes and, optionally, states a list of axioms relative to declared attributes. Axioms  $Ax_1, \dots, Ax_n$  are first order predicates on the attributes defined in the declaration part. We state that:

$$\mathcal{A}^{cl} = \{Ax_1, \dots, Ax_n\}$$

The general form of the initial state schema is  $INIT \hat{=} [Pr_1; \dots; Pr_m]$  where the  $Pr_i (i \in [1..m])$  are first order predicates on the class attributes. We state that:

$$\Theta^{cl} = Pr_1 \wedge \dots \wedge Pr_m$$

The class operations  $OperationName \hat{=} [declaration\ part \mid predicate\ list]$  are the portion of the specification which is translated towards the transitions of  $S^{cl}$ . In many cases, an operation gives rise to a set

of elementary transitions. Let  $\mathcal{T}^o = \{\tau_1, \dots, \tau_i\}$  be the set of elementary transitions resulting from operation  $o \in Op(cl)$ . We distinguish the following cases:

- if  $\mathcal{T}^o$  is a singleton (i.e.,  $\mathcal{T}^o = \{\tau\}$ ), then  $\tau \in \mathcal{T}_e^{cl}$ .
- if  $\mathcal{T}^o = \{\tau_1, \dots, \tau_n\}$ , with  $n > 1$  then let  $\tau_c = [\tau_1; \dots; \tau_n] \in \mathcal{T}_c^{cl}$ , provided that  $\tau_c$  is coherent.

Given the richness of the O-Z language, a detailed presentation of all the translation rules from operations into transitions is out of the scope of this work. Interested readers may see [3] for more details.

## 3. Conclusion

In this paper, we have presented a formal specification approach for MAS of situated reactive agents, based upon the I&R theory. The language used by the specification framework can describe reactive and functional aspects of MAS. It is structured as a class hierarchy so one can inherit from these classes to produce its own specification. The used specification language allows verification of specifications, by translating the later towards a transition system, and using verification environments such as STeP. Moreover, it enables incremental and modular validation of a specification through its decomposition, which is particularly interesting when using model checking approaches. There have been several attempts to specify MAS, by Wooldridge [5] and Luck & d'Inverno [4] for example. However our approach differs in two ways : first, we use a specification language which express MAS reactive and functional aspects ; and second, our approach is structured by a framework and specifications can be refined to an implementation.

## References

- [1] M. d'Inverno, M. Fisher, A. Lomuscio, M. Luck, M. de Rijke, M. Ryan, and M. Wooldridge. Formalisms for multi-agent systems. In *First UK Workshop on Foundations of Multi-Agent Systems*, 1996.
- [2] J. Ferber and J.-P. Müller. Influences and reaction: a model of situated multiagent systems. In *ICMAS'96*, december 1996.
- [3] P. Gruer, V. Hilaire, and A. Koukam. Formal Specification and Verification of Multi-Agent Systems. Technical report, submitted to ATAL'2000.
- [4] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In A. P. Press, editor, *Proceedings of the First International Conference on Multi-Agent Systems*, pages 254–260, 1995.
- [5] M. Wooldridge. *The Logical Modelling of Computational Multi-Agent Systems*. Phd thesis, Manchester Metropolitan University, 1992.