

# Towards a Risk Driven Method for Developing Law Enforcement Middleware

Gustavo Carvalho<sup>1</sup>, Rodrigo Paes<sup>1</sup>, Ricardo Choren<sup>2</sup> and Carlos Lucena<sup>1</sup>

<sup>1</sup> Departamento de Informática – Laboratório de Engenharia de Software – PUC-Rio  
Rua Marquês de São Vicente, 225 Rio de Janeiro, Brasil, 22453-900  
e-mail: {guga, rbp, lucena}@inf.puc-rio.br

<sup>2</sup> Departamento de Engenharia de Sistemas - Instituto Militar de Engenharia - IME  
Praça General Tibúrcio 80, Rio de Janeiro, Brasil, 22290-270  
e-mail: choren@de9.ime.ub.br

**Abstract.** Nowadays, multi-agent system (MAS) analysis relies mostly on the specification of individual software agents, considering unpredictability as a consequence of the interaction among them. In this paper, we propose a law enforcement method that uses risks as guidelines for open MAS infrastructure specification and development. The method contributes to a better understanding of the problem characteristics and of the causes and consequences of specific behaviors in the MAS, thus contributing to the improvement of dependability attributes. By structuring the design decisions as risks, the method also aims to help on future evolutions and reengineering of the law enforcement middleware.

**Keywords:** software agents, open systems, risks, law-enforcement, methodologies, middleware

## 1. Introduction

Software systems have become larger and more complex than ever [23]. A system is a collection of elements related in a way that allows a common objective to be accomplished [21]. Aiming to provide a new generation or an improvement of existing tools and techniques to help the development of those complex systems, software agents have been proposed as a promising paradigm [22,25].

The causes of unexpected or undesirable side effects in multi-agent systems may be very complex. They may lie in the environment where software agents inhabit, in the multi-agent system infrastructure, in any particular software agent behavior, or in some interdependencies between those entities. Because of those interdependencies, it is necessary to understand not only the agents' individual behavior or the basic infrastructure characteristics, but also the effects of the interaction between software agents and the consequences of those interactions within the environment. Furthermore, the characteristics of multi-agents systems present many other sources of uncertainty; for instance, this kind of solution could be based on complex organizations

where participants collaborate in a harmonious or competitive way to achieve their individual goals and it could also present a systemic evolution or adaptation without having a centralized control on how or when those adaptations would happen. In multi-agent systems, this uncertainty about what exactly will happen is described as emergent behavior.

Generally, achieving sufficient dependability in system development and demonstrating this achievement in a convincing and rigorous manner is of crucial importance [21]. The term dependability was proposed by Laprie [13] to cover the related system attributes of availability, reliability, safety and security. In this sense, a developer should consider and assess dependability attributes while developing any system. In an open multi-agent system the experience of emergent behavior makes it even more important to consider those attributes. Currently, no approach for developing multi-agent system infrastructures provides means to effectively gauge those characteristics.

Basically, a systematic exposition of the concepts of dependability consists of three parts: the threats to, the attributes of, and the means by which the dependability is attained [2]. Besides threats, it is important to extend this definition including beneficial consequences or opportunities that could also arise from interactions among agents.

A mechanism to enforce behavioral rule (law enforcement mechanism) aims to provide a way to map the consequences that arise from the interactions to the qualitative or even quantitative criteria presented by the dependability attributes. An enforcement mechanism intercepts particular interactions between agents to control and audit the execution flow of conversations. The enforcement approach aims to contribute to a convergence from an unstable and totally unpredictable open multi-agent system under development to a dependable, more stable and less unpredictable one.

Enforcement rules are specified as laws and norms that are associated with well-known consequences that may be subjected to any participant of the multi-agent system. The laws and consequences can be identified through a risk-driven analysis of the system environment. The use of an enforcement mechanism and of a risk driven method permits to control the failures and to promote the benefits and also contributes to tame the uncertainty presented by open multi-agent systems.

In this paper, we propose a method that considers risk analysis concepts to provide a structured way to specify, implement, monitor and maintain systems requirements and the existent challenges, opportunities, threats and limitations identified through the development of the solution. Briefly, this analysis is based on the identification, control and assessment of relationships between cause and consequences states, events and their characteristics. With this approach, we propose to evolve and develop a law enforcement infrastructure using risks as guidelines. The goal of this method is to provide a structured way to develop the law enforcement middleware, i.e., an infrastructure that contributes to the fulfillment of open multi-agent systems specifications and laws. This approach considers that the software agents are developed by different people using those specifications and laws generated with this method.

In the following section, we present a synthesis of the law enforcement approach and we discuss risks on open MAS development. After that, we present the method

for developing open multi-agent middleware in section 3. In section 4, we present the case study used to experiment the method. Finally, we present some related work and then conclude the article with some considerations and directions about future works.

## **2. Law Enforcement Approach and Risks in MAS Development**

The explicit identification of organizational rules is an important issue in the context of open systems. With the arrival of new, previously unknown, and possibly self-interested agents, the overall organization must be able to enforce its internal coherency, despite the dynamic and untrustworthy environment [25]. Trying to regulate and to limit part of the unpredictable behavior that open systems can present, we propose to apply a law-enforcement approach [19].

In this sense, specification artifacts with rules could be elaborated to guide the development of these system components through the convergence into a desired behavior. At runtime, it is possible to enforce those rules. Rules could be seen as a boundary that stipulates the level of freedom that any unit has to act and they are used to maintain the acceptable levels of dependability. These rules or laws are applied on software agents' conversations and describe permissions, prohibitions and obligations that software agents must fulfill.

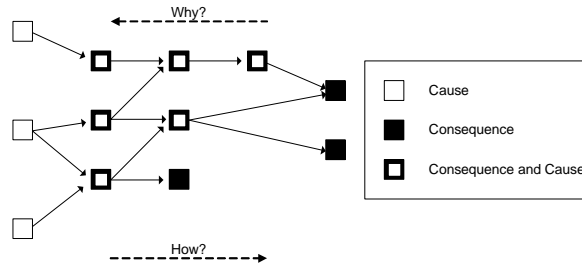
Rules specify restrictions and conditions concerning interaction protocols. A protocol is a definition of a set of legal actions that can be performed during agents' conversations. It specifies valid rules and the acceptable behavior concerning interactions that agents could participate and it also defines and restricts the context where the information exchanged must be interpreted [7]. A protocol is composed of transitions. A transition structures the evolution of the process from a previous state to the next one. It also defines the types of participants and represents the valid actions that can be performed by them.

Norms prescribe how the agents ought to behave, specify how they are permitted to behave and what their rights are [12]. A norm is composed by obligations, permissions and prohibitions. Each kind of norm has activation or deactivation conditions and consequences.

Dependable software includes extra, often redundant, code to perform the necessary checking for exceptional states and to recover from system faults. Because of this additional design, implementation and validation efforts, increasing the dependability of a system can significantly increase development costs [21]. In this sense, a criteria to establish an importance order using a risk analysis method should be applied to help the assessment of alternatives and the gauging of how much of these efforts are really necessary.

Risk is a possible future event that, if it occurs, will lead to an undesirable outcome or to unpredictable consequences and characteristics [14]. A dependability attribute can be modeled as risks, represented as a sequence or a chain of cause and consequence states. This sequence is helpful to understand the origins of consequences and how those characteristics evolve during the execution of the multi-agent system. By this arrangement it is possible to infer and analyze how we can predict and alter the consequences, understanding their causes and the links between intermediary states.

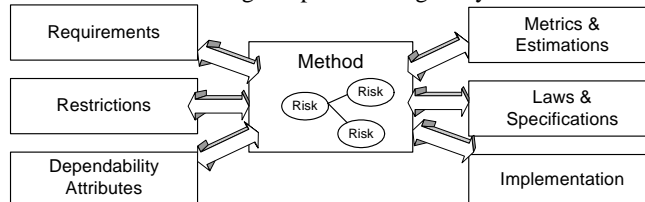
The techniques used to understand and analyze the chain of causes and consequences, navigate from both directions of a chain (Figure 1). Looking the chain from the consequences through the past, a technique tries to find out the causes and the origins of the problems or opportunities. This kind of technique is concerned with answering why the consequence could happen. In the other side, another approach could be concerned with answering how some causes would influence the system behavior and dependable attributes, this analysis begins with identifying the root causes and tries to derivate the consequences through the evaluations of future states in the chain.



**Figure 1 - Chain of cause and consequences instance**

### 3. Risk Driven Method for Developing Law Enforcement Middleware

The risk driven method for developing the open multi-agent system infrastructure described here intends to guide the law enforcement approach basing the specification, implementation and monitoring of open multi-agent system on a risk analysis.



**Figure 2 - Risk Driven Method for Developing Open MAS Overview**

Analyzing carefully the requirements, restrictions, existing components and their relationships we have a base to specify the solution. Furthermore, we aim to implement a dependable multi-agent system. In this sense, dependability attributes are mapped to requirements, but due to their importance they have to be specially considered in this process. All the design decisions and important considerations are mapped to risks. Laws are elaborated and specified through the analysis of the risks. Consequences of not observing laws are described and implemented too. During the specification phase, metrics and estimations are identified. Metrics information is collected during the execution phase and it is compared to the estimations. Reengi-















