A Meta-model for Intelligent Adaptive Multi-Agent Systems in Open Environments

Thomas Juan
Department of Computer Science and Software Engineering, The University of Melbourne
Victoria, 3010, Australia
tlj@cs.mu.oz.au

Leon Sterling
Department of Computer Science and Software Engineering, The University of Melbourne
Victoria, 3010, Australia
leon@cs.mu.oz.au

ABSTRACT
In this paper, we introduce the ROADMAP meta-model, designed to describe intelligent adaptive multi-agent systems in open environments. The meta-model captures our understanding of the properties of such systems and our perspective on organizational concepts such as roles. It defines agent constructs such as agents and roles, and their inter-relationships such as aggregation, in a way that best facilitates the development of intelligent adaptive systems in open environments.

We intend to use the meta-model for knowledge sharing. We expect developers of AOSE methodologies, tools, programming languages and frameworks to benefit from understanding the design and structure of the ROADMAP meta-model. By adopting the meta-model, the resulting methodologies, tools and languages should inherit its desirable characteristics and better support the development of intelligent adaptive systems in open environments.

Categories and Subject Descriptors
D.2.10 [Software Engineering]: Design – Methodologies
I.2.11 [Computing Methodologies]: Artificial Intelligence – Multiagent systems

General Terms
Design, Verification

Keywords
Meta-model, agent-oriented software engineering, software quality goals, intelligent adaptive systems, open environments.

1. INTRODUCTION
In the last decade, the focus of agent research has shifted from single agent systems to multi-agent systems. Multi-agent research is concerned with creating and using artificial organizations of agents [1, 2, 5]. Significant research effort is devoted to exploring different meanings for organizational concepts such as roles. Few attempts have been made to formalize the theory of agent organization as meta-models [2]. However, these attempts did not discuss nor address issues related to intelligent adaptive systems in open environments.

An intelligent system performs tasks that require particularly large amounts of knowledge and reasoning with the knowledge. For example, for an intelligent search engine to return the most relevant webpages, it must reason about the nature of the user and the context of the search. It must also understand the content of the webpages it indexed to determine what makes a page relevant. An adaptive system senses changes in its usage and its environment, and alters its behavior at runtime for better results. For example, a business system may change its architecture to mirror changes in the human organization. Machine learning could be used by the system for self-optimization. In an open environment, we cannot trust all agents to act according to the system-wide goals. Instead, we must create mechanisms to ensure correct system-wide behaviour even when individual agents misbehave.

Intelligence, adaptiveness and open environments create difficult challenges for system development. Traditional quality goals such as correctness cannot be assumed to hold at runtime after deployment. Adaptation and the open nature of the environment may cause system performance, reliability, security and maintainability to be lost or change significantly at runtime. Further, as we rely more on multi-agent systems, new and imprecise quality goals such as privacy, politeness and good taste for suggestions begin to emerge. Whether these quality goals are achieved is very open to the user’s interpretation.

2. THE ROADMAP META-MODEL
This section introduces the ROADMAP meta-model, which describes multi-agent systems with organization constructs. We focus on high-level issues rather than precise definitions of the concepts. The core elements of the meta-model are shown in Figure 1. All these entities are realized at runtime.

Following the analogy that putting people together does not form efficient organizations unless sufficient processes, regulations, infrastructure and organization goals are also in place, we propose to model multi-agent systems by two hierarchies at runtime. The role hierarchy represents a high-level specification of requirements, capturing organizational structures, regulations, processes, goals, responsibilities and various permissions for the agents in the system. The agent hierarchy provides concrete implementation of system functionalities. Fine-grain control over agents is achieved by constraining the services they provide with
protocols from roles they take. The recursive nature of both hierarchies ensures system scalability.

![Diagram of the ROADMAP meta-model at runtime]

**Figure 1. The core ROADMAP meta-model at runtime**

An important part of the roles is the definition of goals agents should pursue, and evaluation functions to measure their performances. For example, a goal of a role may be maximizing privacy, according to an attached evaluation function that defines the meaning of privacy for agents taking this role. Official communications and messages in the organization should go through roles. This ensures agent behaviours can be validated at runtime for correctness and other quality attributes, by invoking the evaluation functions.

A knowledge component is a modular unit of knowledge. A knowledge component can aggregate other knowledge components, allowing knowledge in the system to scale. The presence of a knowledge component in the meta-model independent of agent services fulfills the principles of abstraction of knowledge from functionalities, and the modularity of knowledge. Roles can include knowledge components, allowing the distribution of knowledge in the system to be represented and modified. This also allows knowledge sharing and reuse in the organization to be reasoned about and controlled. If an agent possesses illegal knowledge, the agent may be removed from the role and from the organization. To allow heterogeneous systems, the meta-model does not prescribe implementation architectures or technologies for agents and knowledge components.

During development time, each runtime entity depicted in Figure 1 will have a corresponding class from which the runtime entities are instantiated. They are straightforward and not discussed here. An interesting inclusion is the environment class. We recognize the importance of describing and analyzing agent environments. Yet at runtime the environment is usually accessed through existing APIs. Therefore so far we have not included a runtime environment entity. We intend to clarify this issue in future.

### 3. EXAMPLE APPLICATIONS

#### 3.1 The ROADMAP Methodology

The ROADMAP methodology [3] is designed to support the development of complex open systems. Figure 2 shows the models of the ROADMAP methodology. The methodology closely implements the meta-model; for each runtime entity outlined in the ROADMAP meta-model, there is a corresponding development-time model in the methodology. The methodology complements the models with an iterative development process.

![Diagram of the structure of ROADMAP models]

**Figure 2. The structure of ROADMAP models**

By adopting the meta-model, the ROADMAP methodology inherits its desirable characteristics and is suitable for developing intelligent adaptive systems in open environments. This example shows the applicability of sharing knowledge via the meta-model.

### 3.2 Custom AOSE Methodologies by Reusing AOSE Features

In related work, we described an approach for assembling project specific methodologies from modular AOSE features [4]. The ROADMAP meta-model can be used to ensure semantic consistency between AOSE features that originated from different methodologies. The ROADMAP roles provide an abstract, uniform and architecture-independent construct to describe functionalities, agent interaction and various quality attributes in the system.

### 4. REFERENCES


