An Approach to the Design of Adaptive and Normative Software Agents

Nome do Aluno: Marx Leles Viana
Nome do Orientador: Carlos José Pereira de Lucena
Nível: Doutorado
Email dos Contatos:
Aluno: mleles@inf.puc-rio.br
Orientador: lucena@inf.puc-rio.br
Ano de Ingresso: 2012/2
Ano Previsto de Conclusão 2016/2
Data da Qualificação: 12/12/2015
Eventos Relacionados: SBES

Abstract. As a new paradigm of building and software modeling, multi-agent systems require modeling languages exploiting its benefits and characteristics. However, software engineering based agents have not been widely adopted, mainly due to lack of modeling languages that explore the use of agent-related abstractions and are responsible for refining the design models to code. In addition, most modeling languages do not define how these abstractions interact at runtime, but many software applications need to change their behavior and react to changes in their environments dynamically. We observed that most languages do not represent some important concepts such as adaptation and normative. Nevertheless, many types of applications currently need to support adaptive and normative mechanisms, including, emergency services and natural disasters (eg, floods, fires, landslides, earthquakes) homes and smart cities, construction of virtual heritage and health (eg, diseases or disorders involving the immune system such as HIV). This doctoral research proposes a novel approach to modeling and implementing adaptive normative software agents.

Keywords: — Modeling language, adaptation, norms, multi-agent systems.
1. Introduction

Agent-Oriented Software Engineering (AOSE) emerged as a new technology for building complex systems. These systems are characterized by being distributed and composed of autonomous entities that interact with each other [Wooldridge 2011]. Multi-Agent Systems (MAS) are societies in which these heterogeneous and individually designed entities (agents) work to accomplish common or independent goals [López 2003]. Thus, the use of agents for construction of such complex systems is considered a promising approach [Zambonelli 2003]. However, MAS has not been widely adopted, mainly due to lack of modeling languages that support the use of agent-related abstractions and promote the refinement of design models to code [Silva 2004]. In addition, most modeling languages do not define how these abstractions interact at runtime [Beydoun 2009], but many software applications need to change their behavior and react to changes in their environments dynamically.

In general, modeling languages should represent the structural and dynamic aspects of software agents, expressing the essential characteristics of the agent-related entities. The structural aspects incorporate the definition of the entities, their properties and their relationships. Several authors recognized the importance of modeling agents in their environment both in design time and in runtime [Silva 2004], [Bresciani 2004], [Beydoun 2009]. These authors also observed that most modeling languages do not represent some important concepts present in SMAs, for example, adaptation and norms [Beydoun 2009], [Hollander 2011]. Nevertheless, many types of applications currently need to support adaptive and normative mechanisms, including, emergency services and natural disasters (eg, floods, fires, landslides, earthquakes) [Beamon 2006] [Cerqueira 2009], [Janssen 2010], homes and smart cities [Cook 2012], [Atzori 2011], [Gubbi 2013], construction of virtual assets (eg, Second Life [Bogdanovych 2011]) and health (eg, diseases or conditions involving the immune system such as HIV [Zang 2005], [Laroum 2011]).

2. Background

This section describes the main characteristics of the concepts related to adaptation and norms. First, we will discuss what is adaptation, how it occurs and how software agents can adapt. We will also discuss what norms are and how they are understood by the agents. Then the ability of norms to adapt in a multi-agent system and the impact of norms on adaptation will be discussed.

The word adaptation means "any change in the structure or functioning of an organism that makes it better suited to its environment" [Oxford Dictionary of Science 2015]. Therefore, according to this definition, the adaptation of agents has two levels of action: structural change and behavioral change. These changes aim to make applications more adapted to their evolutionary context [Da 2011]. Generally, there are three types of adaptation [Chefrour 2005]: (i) reactive adaptation, in which the agents will change their behavior due to the execution of changes in their environment; (ii) evaluative adaptation, which aims to extend the functionality of software systems, fix bugs or to enhance their performance; and (iii) adaptation of integration, which occurs when the integration of services or components become incompatible due to other adjustments made to the system.
The general adaptation cycle of a software agent includes environmental monitoring, adaptation selection, and adaptation implementation. This cycle is explained in four different phases: Collection, Analysis, Decision and Execution. In this cycle, in the case of MAS, the system is fully autonomous. First, the agent data Collection activity gathers the information resulting from environmental monitoring. Next, the Analysis activity evaluates the context data and produces an adaptation plan for adaptation, which consists of a single plan or a list of plans. In the Decision activity the risk of each plan will be assessed and the adaptation will occur. Finally, in the Execution activity the adaptation will be performed in accordance with what was described in the chosen plan. The core of the adaptation involves the Analysis and Decision activities.

Regarding the concept of norms, Webster online dictionary [Merriam-Webster 2015] offers three definitions: (i) an authoritarian pattern; (ii) a principle of right actions associated with the members of a group that is used to guide, control or regulate proper and acceptable behavior; and (iii) a weighting. A norm is treated as: (a) a standard set of development or achievement, usually derived from the average or median performance of a large group; (b) a standard understood as the typical behavior of a social group; and (c) a general or usual practice. These definitions are meanings of the term norm that are used in various areas in normative research, including deontic logic, law theory, sociology, social psychology and social philosophy, decision theory and game theory [Verhagen 2000].

When applied to multi-agent systems, norms are understood as mechanisms to regulate the behavior of agents, representing the way in which agents understand the responsibilities of other agents [López 2003]. Thus, the agents work under the belief that other agents will behave according to the prescribed norms, but above all, norms are mechanisms that enable agents to require the other agents to behave in a certain way [Santos Neto 2010]. Thus, the use of norms is a necessity in multi-agent systems in which the members are autonomous, but not self-sufficient, and, therefore, cooperation is required but cannot be assured without the specific introduction of specific mechanisms that support it.

Knowing that norms are mechanisms that society has to influence the behavior of agents [Bogdanovych 2011], norms can be used to achieve different purposes, ranging from the construction of a simple agreement between agents to more complex cases involving the legal system [López 2002]. The norms may persist for different periods of time, for example, while the agent remains in the society, or just for a short period of time until the social objective has been achieved [Lopez 2002]. Therefore, different aspects can be used to characterize norms. First, norms are always created to be met by a particular set of agents in order to achieve specific social goals. Second, norms are not always applied, and their activation conditions depend on the context of the agents involved in a specific interaction. In addition, the type of element that the norm regulates must be known; if it is an action of the agent, or a state of the environment. Finally, in some cases, norms may provide a set of sanctions to be imposed when agents fulfill or violate certain norms [López 2002], [López 2003], [Santos Neto 2010].

An aspect of a norm that needs to be emphasized in the MAS area is its dynamic nature and its tendency to undergo changes over time [Neumann 2008]. An examination
of human society produces clear examples of this phenomenon: the norms of a generation are rarely identical to the norms of the next. This leads us to the concept of adaptive norms. Adaptive norms refer to the process by which the norms of a system change over time [Hollander 2010]. In theory, this is done through the use of social learning processes, such as copying (imitation) and socializing. Once a new norm is created, it has the potential to emerge in certain circumstances and, if it happens, the norm will compete or replace other existing norms, which are internalized in the same context. This type of adaptation occurs at the macro level. Alternatively, a norm can be modified when it is acquired by an agent [Hollander 2010]. During the transmission phase, the polarization of the receiving agent may result in subtle changes. This is the main concern in systems where the norm may not be copied directly. The changes that occur during transmission and internalization of norms are adaptations at the micro level.

3. Modeling Languages

Although some modeling languages involve adaptation abstractions and norms implicitly in the agent design time [Bresciani 2004], [Zambonelli 2003], these approaches do not explicitly show how these abstractions influence both the design time and runtime behavior of the agents and their reactions to environmental changes. Due to this limitation, these languages cannot be used to model many essential aspects in the representation of agents such as those aspects involving adaptation, norms and the interactions between them. For example, certain adaptations may require that the norms are adaptive, that is, that they change over time the applicability, or even their existence. On the other hand, norms can also have an impact on adaptation, since some adaptations must meet some normative criteria (e.g., obligations, permissions, prohibitions).

In recent years, various modeling languages have been proposed for MASs [Silva 2008], [Zambonelli 2003] [Beydoun 2009] [Cervenka 2007] [Bernon 2002]. However, there is still the need for a modeling language that describes concepts related to adaptation and norms and: (i) supports these concepts as first-class citizen abstractions; (ii) supports these concepts through an explicit description of a MAS metamodel; (iii) can be used to model the structural and dynamic aspects often described in MAS for these concepts; and (iv) promote the refinement of these models from design to code.

4. Proposed Approach

Based on these aspects, the objective of this doctoral research is to create an approach that supports the abstractions of adaptive and normative agents (i.e., agents capable of reasoning about a norm addressed to them and adapting). This approach essentially involves: (i) a metamodel for representing MASs; (ii) a generative process based on models for the implementation of the metamodel; and (iii) the introduction of techniques for assessing the metamodel and the generative process.

A metamodel defines a language for specifying models describing the semantics of a set of abstractions and defining how these abstractions are instantiated [UML 2015]. For each abstraction, the metamodel should describe its semantics, the meta-relationship with other abstractions and the graphical representation of the abstraction.
The modeling languages should describe the graphical representation of the new abstractions, their semantics and the relationships between them [Silva, 2004].

A modeling language for MASs should include structural diagrams to model the structural aspects of a system. The set of structural diagrams need to be able to model: (i) the entities usually defined in a MAS, (ii) the properties of these entities, associating properties with specific entities, and (iii) the relationship between the entities. The modeling languages proposed in the literature do not model explicitly adaptations and norms and therefore do not define the relationships between agents and those entities.

The development of appropriate approaches to implement agent-based systems is a key issue that needs to be addressed when agent technology is used in the development of a software system. In order to implement the multi-agent systems developed using a specific modeling language, it is necessary to transform the MAS design models into code. These models are models composed of high-level agent-related abstractions. To turn these models into code, the agent-related abstractions must be mapped into abstractions defined in a certain programming language.

To demonstrate the need for representing adaptation and norms, two application scenarios will be described. The first scenario relates to emergency services and natural disasters. The purpose in this case is to evacuate people from hazardous areas where civilians may be in danger and firemen aim at rescuing these civilians [Cerqueira 2009]. A second application scenario involves the immune system of the human body and its interaction organisms that cause diseases [Zhang 2005].

In the context of evacuation of people in hazardous areas, the ideal outcome would be for firemen to rescue and provide care to the injured. For this purpose, the fire department has a limited number of resources that are regulated by the firemen chief. This regulation is made by means of norms that restrict the behavior of the firemen so that the rescue is undertaken in a coordinated manner and with the best use of resources. For example, these resources can be aircraft, ground vehicles and excavation equipment. Furthermore, in terms of adaptation, the environmental conditions may change during a particular rescue due to weather instability and landslides, among other factors. Thus, there is need for firemen to be able to adapt to different conditions to carry out the rescue.

In relation to the second scenario, it is necessary to represent the interaction dynamics between the immune system and the organisms that cause the diseases. The body reacts to daily attacks of bacteria, viruses and other microbes, through the immune system. Any of these invading organisms can rapidly evolve and adapt, trying to avoid detection and neutralization by the immune system. As a result, various immune system defense mechanisms have also evolved to recognize and neutralize these organisms. By acquiring immunity, the body creates an immune memory following an initial response to a specific agent, which allows it to respond more effectively to new attacks by the same agent. In addition, the elements of the immune system must follow certain norms and strategies to combat the invaders.

To facilitate the use of agents capable of representing adaptation and norms, we will be developed a tool to help build these models, enabling the refinement of design models into code.
5. Expected Contribution

This doctoral research assumes that introducing abstractions related to adaptation and norms in MAS representation languages results in a new metamodel that supports these abstractions, a generative process for implementing these metamodels, and the use of the metamodel and the generative process in different application domains. The main contributions of this research include:

• A metamodel to define the interactions that can occur between the entities present in MASs with adaptation and norm-related abstractions;
• A modeling language that incorporates adaptation and norms, which are particular characteristics of MASs that are not addressed in UML and are not satisfactorily addressed in the proposals available in the literature;
• Design guidelines to help designers to use these new concepts in the proposed language;
• A grammar of the language, formalizing the syntax of the structural and dynamic aspects, and visual diagrams;
• A generative process as well as a tool to refine the structural and dynamic diagrams of the modeling language and generate code in a specific programming language.

6. Evaluation

The approach proposed in this study will be evaluated in various ways and may include:

• Case studies (e.g., emergency services due to natural disasters and diseases of the immune system).
• Experiments involving factors such as the ease of use of the proposed representation and tool.
• A study of the expressiveness of the proposed modeling language in comparison to other MAS representations proposed in the literature.
• A formalization of the models to support the verification of relevant properties related to adaptation and norms.

References


López, Fabiola López; Luck, Michael; D’Inverno, Mark. Constraining Autonomy through Norms, 2002.


