Expert Systems with Applications 40 (2013) 1323–1332

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An approach for the integration of swarm intelligence in MAS: An engineering perspective

Vincent Hilaire [a,*], Massimo Cossentino [b], Franck Gechter [a], Sebastian Rodriguez [a,c], Abderaffiaa Koukam [a]

[a] Laboratoire Systmes et Transports, UTBM, 90010 Belfort Cedex, France
[b] ICAR Institute, National Research Council, Palermo, Italy
[c] Centro de Investigación de Teconologías Avanzadas de Tucumán, UTN, San Miguel de Tucuman, Argentina

ARTICLE INFO

Keywords:
Agents
Swarming systems
Agent-oriented software engineering

ABSTRACT

For more than 20 years, researchers have designed models in order to describe swarm intelligence and apply the resulting techniques to complex problems. However, there is still a gap between these models and current MAS methodologies. The goal of this paper is to propose a principled and methodological approach for the engineering of systems based upon swarm intelligence. The constraints are, on the one hand, to enable the analysis, design and implementation of such systems; and, on the other hand, to formally analyze and verify properties of resulting systems. The principles of the approach are based, on the one hand, on requirement driven activities that produce goals to be fulfilled by the system of interest and, on the other, hand on an ontological modeling of the problem domain. This ontological modeling conceptualizes the phenomenon one seek to imitate and thus allows it understanding. The produced ontology is refined through the methodology activities down to organizational models.

1. Introduction

For more than 20 years, researchers have designed models in order to describe swarm intelligence and apply the resulting techniques to complex problems. However, there is still a gap between these models and current MAS methodologies such as ASPECS (Cossentino, Gaud, Hilaire, Galland, & Koukam, 2010), TROPOS (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004) or GAIA (Zambonelli, Jennings, & Wooldridge, 2003).

The goal of this paper is to propose a principled and methodological approach for the engineering of systems based upon swarm intelligence. The constraints are, on the one hand, to enable the analysis, design and implementation of such systems; and, on the other hand, to formally analyze and verify properties of resulting systems.

Some works have contributed to this issue (Naturwissenschaftlichen Fakult At Ii et al., 2000; Bonabeau, Dorigo, & Theraulaz, 1999; Bernon, Camps, Gleizes, & Picard, 2005, Chapter VII) . In Bonabeau et al. (2003) the authors build a library of patterns of self-organizing behaviors. However, even if it is an interdisciplinary and interesting attempt to define a framework for engineering systems based upon swarm intelligence a lot of work has still to be done in order to define principled and methodological approaches relying on swarm intelligence. The author of Naturwissenschaftlichen Fakult At Ii et al. (2000) proposes general rules in order to engineer swarming systems and a formal framework for analysis and validation. These rules are not related to current methodologies or development platforms. In Bernon et al. (2005, Chapter VII) the authors design a complete methodology based upon the theory of Adaptive Multi-Agent Systems. This methodology allows the analysis and design of swarming systems but impose a cooperative internal medium which maybe a strong constraint for some cases.

Swarming approaches are usually inspired by biology (Parunak, 1997) or physics (Marco, Mamei, Zambonelli, & Leonardi, 2004; Reynolds, 1987). These systems usually exhibit features such as self-organization, emerging phenomena, robustness and adaptability. One of the main problems with this kind of approaches is the small number of methodologies and guidelines which help the engineering of such systems. Indeed, to be fully adopted, methodologies must be able to decompose the underlying principles of these systems with abstractions in a principled way. Some experiments were done in the domain of swarming MAS architecture reverse-engineering (Hilaire, Simonin, Koukam, & Ferber, 2005) but they usually lack a systematic support. For instance, in Parunak and Brueckner (2004) the authors give general principles in an informal way. However, it is a real problem and despite the interest risen by these architectures the claims that stripping away centralized control is enough to allow the emergence of interesting properties has never been proved (Stepney, Polack, & Turner, 2006).

In order to engineer such kind of MAS we propose to use a combined approach based, on the one hand, on requirement driven activities that produce goals to be fulfilled by the system of interest

* Corresponding author. Tel.: +33 384 583 009; fax: +33 384 583 342.
E-mail address: vincent.hilaire@utbm.fr (V. Hilaire).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.08.058

and, on the other, hand on an ontological modeling of the problem domain. The principle is to produce an ontology which conceptualizes the phenomenon one seek to imitate and thus allows it understanding. Indeed, swarming approaches (Parunak, 1997) are based on the replication of existing behaviors, that have produced in certain experimental conditions the desired emerging properties. This statement was already observed in Parunak and Brueckner (2004):

> Swarming is a discovery, not an invention. It is a naturally occurring phenomenon that we seek to imitate in engineered systems. Design principles for effective artificial swarming systems must be developed from an understanding of why swarming works in natural systems.

The produced ontology is then refined down to organizational models. The analyst can then choose from a library of organizational models the ones that will satisfy the goals issued from the requirements analysis. This combined approach is associated with formal proofs techniques in order to formally verify properties of organizational models and fulfillment of goals.

It seems a sound principle as the ontology is an understanding of the modeled system and so describes at least partially how it works. The interacting entities the designer wants to replicate and their possible actions are part of this ontology. In order to introduce a systematic activity, some guidelines are provided in order to identify organizations and roles from this ontology. This activity is integrated in an existing methodology, namely ASPECS. The modified methodology is then able to decompose such systems in roles, interactions and organizations which are the key concepts of the ASPECS analysis phase and which are identified, in the original methodology, by means of use cases.

This kind of approach is advocated by the authors of Parunak and Brueckner (2004) who state "we ought to be able to reverse-engineer the underlying mechanisms of swarming systems for use in synthetic systems".

The paper is organized as follows: Section 2 presents the background related to the ASPECS methodology and the specific architecture that illustrates the presented approach. Section 3 presents the ontological identification approach. Section 4 is dedicated to related works and Section 5 concludes.

2. Background

2.1. ASPECS

ASPECS is a step-by-step requirements to code software engineering process based on a metamodel which defines the main concepts for the proposed MAS and Holonic MAS analysis, design and development. It integrates design models and philosophies from both object- and agent-oriented software engineering (OOSE and AOSE) and is largely inspired by the PASSI (Cossentino, 2005) and RIO (Hilaire, Gruer, Koukam, & Simonin, 2008) approaches. The target scope of ASPECS can be found in complex systems and especially hierarchical complex systems. The main vocation of ASPECS is towards the development of societies of holonic (as well as not-holonic) multiagent systems.

The ideas underpinning the ASPECS design process can be described as follows:

1. The ASPECS design process explicitly deals with the design of open, dynamic and complex systems.
2. The adoption of an organizational approach. Functionalities to be realized are assigned to organizations. An organization is defined by a collection of roles that take part in systematic institutionalized patterns of interactions with other roles in a common context. A role is defined as an expected behavior (a set of role tasks ordered by a plan) and a set of rights and obligations in the organization context. The goal of each Role is to contribute to the fulfillment of (a part of) the requirements of the organization within which it is defined. A role can be instantiated either as a Common Role or Boundary Role. A Common Role is a role located inside the designed system and interacting with either Common or Boundary Roles. A Boundary Role is a role located at the boundary between the system and its outside and it is responsible for interactions happening at this border (i.e. GUI, Database, etc.).
3. Domain related ontological knowledge is used as a tool for enhancing the quality of design. This has been already adopted in some previous methodologies (Iglesias, Garijo, Gonzalez, & Velasco, 1998) but it is lacking in most modern approaches. We think that in dealing with intelligent agents it is particularly important to explicitly catch an ontological model of the problem and solution domains; this allows an easy application of several AI techniques as well as the adoption of semantic-based communications among agents.
4. Three main levels of abstractions, called models according to the model-driven engineering terminology, are considered. Concepts of the problem domain are used to model system requirements in terms of organizations and interacting roles; concepts of the agency domain are the result of a set of transformations from the previous domain and are used to depict an agent-oriented solution; concepts of the solution domain are again the result of some transformations and are devoted to design a platform-specific solution at the code level.

The different activities of the System Requirements phase of ASPECS are represented by the SPEM diagram of Fig. 1. ASPECS software process is driven by requirements. Thus the first activity, Domain Requirement Description, of the first phase of ASPECS, System Requirements, deals functional and non functional requirements. Functional requirements describe the functions the software has to exhibit (Software Engineering Body of Knowledge, 2004) or the behavior of the system in terms of interactions perceived by the user. Non functional requirements are sometimes known as constraints or quality requirements (Software Engineering Body of Knowledge, 2004). The global objective of the Domain Requirements Description (DRD) activity is gathering needs and expectations of application stakeholders and providing a complete description of the behavior of the application to be developed. In the proposed approach, these requirements should be described by using the specific language of the application domain and an user perspective. This is usually done by adopting use case diagrams for the description of functional requirements; besides, conventional text annotations are applied to use cases documentation for describing non-functional requirements.

The global objective of the Problem Ontology Description (POD) is to provide an overview of the problem domain. Stakeholders naturally express requirements in their own terms and with implicit knowledge of their own works (Sommerville, 2004). Therefore the aim of this activity is deepening the understanding of the problem by complementing the usual requirements description in terms of use cases with a description of the concepts that compose the problem domain. It describes concepts used in the specific language of the application domain and users. Results of this activity can sometime imply modifications in uses cases. The design of the domain ontology occurs very earlier in our process and this has a direct consequence in the organization and capacity identification activities. Problem Ontology is modeled by using a class diagram where concepts, predicates and actions are identified by specific stereotypes. The ontology is inspired from the FIPA proposal (FIPA, 2001). The main point is that actions are distinguished concepts associated to the concept that act and the concept that is manipulated. This specific type of ontology is described in the next subsection.