Abstract

Methodology is generally a guideline for solving a problem, with specific components such as phases, tasks, methods, techniques and tools. In this paper, we will extend/adapt the ASPECS methodology by adding a new phase called Goal identification which allows identifying the Goals of the system from the beginning of the methodology.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of [name organizer]

Keywords: Agent-oriented methodology, Design Methodology, ASPECS, Goal

1. Introduction

Agent technology [1] has received an enormous deal of attention in the last few years and, as a result, the industry is beginning to get interested in using this technology to develop its own products. In spite of the different developed agent theories, languages, architectures and the successful agent-based applications, very little work for specifying (and applying) techniques to develop applications using agent technology has been done. The role of agent-oriented methodologies is to assist in all the phases of the life cycle of an agent-based application, including its management.

The concept of Goal has been used in different areas of computer science since the early days of the discipline. In Artificial Intelligence, problem solving and planning systems have used the notion of Goal to describe desirable states of the world. More recently, Goals have been used in software engineering [2, 3, 4, 5] to model early requirements [6] and non-functional requirements [7] for a software system. Identifying goals of the system is not an easy task. While goals could be explicitly stated by the stakeholders or in the various sources of information available to requirements engineers.

Requirement analysis [8] represents the initial phase in most software engineering methodologies like Tropos and ASPECS [9, 10]. Tropos [11, 12] is an agent-oriented software development methodology founded on two novel features. First, the methodology is defined in terms of the concepts of agent, Goal, and related
mentalist notions (such as actor, plans and task).

The objective of this paper is to introduce and to explain the idea of how we have extended the ASPECS methodology. ASPECS methodology use the concept of Goal just in the others phases (Agent Society Design, Implementation and Deployment) not in the system requirements analysis phase. For this reason, we think that the concept of Goal must be identified at the launch of ASPECS Process, exactly in the system requirement analysis phase.

This paper is structured as follows. Section 2 introduces the two methodologies: Tropos and ASPECS. The way for building a new ASPECS methodology is then described in section 3, while Section 4 summarizes the results of the paper and offers directions for future work.

2. Related Works

2.1. Tropos

The Tropos methodology [12] is proposed to support all analysis and design activities in the software development process, from application domain analysis down to the system implementation. In particular, Tropos rests on the idea of building a model of the system-to-be and its environment that is incrementally refined and extended, providing a common interface to various software development activities, as well as a basis for documentation and evolution of the software.

Tropos methodologies introduce the five main development phases: Early Requirements, Late Requirements, Architectural Design, Detailed Design and Implementation.

Models in Tropos are acquired as instances of a conceptual meta-model resting on the following concepts/relationships: Actor, which models an entity that has strategic Goals and intentionality within the system or the organizational setting. An actor represents a physical, social or software agent as well as a role or position. Goal, which represents actors’ strategic interests. Plan, which represents, at an abstract level, a way of doing something. The execution of plan can be a means for satisfying a Goal. And others concepts like Resource, Dependency, Capability and Belief (for more details see Ref [12]).

2.2. ASPECS: Agent-Oriented Software Process for Engineering Complex Systems

ASPECS is a step-by-step requirement to code software engineering process based on a meta-model 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 [13] and RIO [14] 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) multi agent systems.

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

- The ASPECS design process explicitly deals with the design of open, dynamic and complex systems.

- 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).

- 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.

ASPECS Models are instances of a meta-model concepts such as: **Role**, an expected behavior (a set of role tasks ordered by a plan) and a set of rights and obligations in the organization context. **Interaction**, a dynamic, not a priori known sequence of events (a specification of some occurrence that may potentially trigger effects on the system) exchanged among roles, or between roles and entities outside the agent system to be designed. **Capacity**, a specification of a transformation of a part of the designed system or its environment. **Organization**, 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. 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. The different activities of the System Requirements phase of ASPECS are represented by the SPEM diagram in Fig. 1.

The Capacity Identification activity consist in identifying the generic part of the role behavior and to distinguish it from all behaviors which could depend on the internal properties and data of the entity which will play the role.

The main objective of the Capacity Identification (CI) activity is the definition of generic role behaviors by identifying which know-how a role requires from the individual that will play it. For the scope of this paper it is important to say that the capacity is a description of what an organization (and therefore one of its composing roles) is able to do without any kind of specification on how to do it. It means that the results described by a capacity may be reached by adopting different strategies (the realization of the capacity is a concern of the Agency Domain and will be discussed later). This activity thus consists in identifying the
generic part of the role behavior and to distinguish it from all behaviors which could depend on the internal properties and data of the entity which will play the role. For example, if personal acquaintances of the entity will influence a particular choice that is required in the role behavior, i.e. choose the best partner to fulfill this task. This choice may be externalized as a capacity, because there will be necessary various ways of carrying it out and because it depends on data which are strictly personal with the entity (beliefs, acquaintances, etc). This is often a design choice; this function can be hard-coded in the role behavior and to impose on the entity which plays it to carry out it in the way described in the role. Or it may be externalized to let free each entity to carry out it in the way in which it wishes it. This depends obviously on the application and the level of generic the designer wants to give to the role.

3. How to extend the ASPECS methodology

In this section we will illustrate and explain the idea of how we extend/adapted an existing agent design methodology, ASPECS [16], in order to create a customized agent-oriented methodology meeting a specific requirement: providing designers with a methodology supporting goal oriented requirements analysis and maintaining the skills and background of all the designer that have been ASPECS user since now. Reviewing the ASPECS and Tropos methodology, we note that the roles use their capacities to participate in the fulfillment of organizational Goals. A capacity is a specification of a transformation of a part of the designed system or its environment. This transformation guarantees resulting properties if the system before the transformation satisfies a set of constraints. It may be considered as a specification of the pre- and post-conditions of a goal achievement. Whereas, the concept of Goal is a description of an objective used to pursue and represents an abstraction of a projected state of affairs to obtain. This concept is used in ASPECS methodology just and only in the Agency domain phase and is not referred from the beginning of the processus.

3.1. Contribution

The idea of refinement that is from the outputs of the Identification Capacity activity of requirements analysis phase of ASPECS methodology, identify the different Goals (global Goals and individual Goals) of the system to be. So, we will add a new activity called Goal Identification (GI) which will take the class diagram relates to the Capacity Identification activity, as an input and will produce a class diagram Goal Diagram (GD) as an output. The addition of Goal Identification activity is related to the Capacity Identification activity as shown in Fig. 2.

The new phase of analysis will eventually need an activity Goal identification aims to identified the objectives (or Goals) of the roles and organizations of the system to be. According the behavior of the organization conference (example Conference Review Process), the organization required a capacity called "be fair reviews in assignment". Through logical analysis, we can conclude that this organization achieves its global Goal named the "be fair reviews in assignment". In the following, the overall behavior of the organization is split on the different roles of the same organization. For example, the PC Member role required the capacity "review papers" is part of the overall behavior of the organization. So, from this capacity, we extract the individual Goal associated with the PC Member role, it is a Goal called "review papers". In low levels (Agent Society Design of ASPECS), is distributed to agents who play the roles of the same organization. The identification process of Goals of the system is based necessarily on the phenomenon/system description and the behavior of the studied system (system to be).

Applying the process described above, we shall have a model (Goal diagram) as a decomposition tree of Goals where the root is a global Goal and the leafs are sub-Goals that can be achieved by the agents in the form of task.

In the decomposition tree of Goals, each global Goal \( g \in G \) (where \( G \) is the set of global Goals) is decomposed into sub-Goals and plans through using two techniques of reasoning as follows:

- AND/OR decomposition: combine AND/OR decompositions of a root objective into sub-Goals which is a more refined modeling of the structure of the Goal.
- Contribution analysis: identifies the objectives that can contribute positively or negatively in the accomplishment of the Goal that must be analyzed.
3.2. Goal Diagram

The concept of Goal is related to Goal Diagram (GD) shown in Fig.3. This concept is represented by a UML class stereotyped "Goal". These Goals can be analyzed from the perspective of the role by the reasoning techniques and decomposition (contribution analysis and AND/OR decomposition) such as is defined in the Tropos methodology [12].

Fig.3 gives a diagram of Goal-related behavior of the role "PC member" has the Goal "review papers". The objective "review papers" is decomposed (using the AND-decomposition) into two sub-Goals: "assign papers to reviewers" and "collect the reviews". The Goal "assign papers to reviewers" is also broken down into two sub-Goals: "send the papers", which is operationalized as send documents by e-mail, and "select reviewers". The two sub-Goals and send the papers may select reviewers contributed positively to the satisfaction (fulfillment) of the overall Goal of the conference organization, named "Be fair in the review assignment" is described by the relation of positive contribution in the diagram of Goal (see Fig.3).

4. Conclusion

In this paper, we extend/adapt the current ASPECS methodology to support the concept of Goal from the beginning of the analysis. The new ASPECS methodology is very effective because it targets the extraction of Goals of studied system from the beginning of the specification which is used to represent how an MAS/HMAS usually reaches these goals overall (corporate), indicating how these Goals are decomposed (by plans) and distributed to agents (by missions) in a lower level of the methodology (phase agentification for example). In this work, we adapted ASPECS and extend it with some inspiring ideas coming from Tropos and others works.

References