

Providing Knowledge Management Support to Communities of Practice through Agent-oriented Analysis

Renata S. S. Guizzardi

(Computer Science Department, University of Twente
P.O. Box 217 – 7500 AE Enschede – The Netherlands
souza@cs.utwente.nl)

Anna Perini

(ITC-irst
via Sommarive 18, I-38050, Trento-Povo, Italy
perini@itc.it)

Virginia Dignum

(Institute of Information and Computing Sciences, Utrecht University
P.O. Box 80089 – 3508 TB Utrecht – The Netherlands
virginia@cs.uu.nl)

Abstract: This paper concerns the need for methodological support in providing Knowledge Management (KM) IT solutions. Due to the distributed nature of knowledge, the support of KM often requires complex, distributed IT systems, which are inherently difficult to design. We propose an agent-oriented methodology based on *Tropos* for the analysis and design of KM systems that offers appropriate abstractions for modeling and designing the characteristics of the organizational setting of the system. The method is illustrated using a fictitious scenario where a newcomer in a knowledge organization decides to join an existing Community of Practice (CoP) in order to share knowledge and adjust to his new working environment.

Keywords: communities of practice, agent-oriented analysis, IT support

Categories: D.2.1, H.3.4, H.5.3, I.2.11, K.4.3

1 Introduction

Research in Knowledge Management (KM) has evolved substantially in the past 30 years, coming from a centralized view of knowledge management processes to a distributed view, grounded in organizational and cognitive sciences studies that point out the social, distributed, subjective and inter-subjective nature of knowledge [1]. Many corporate-wide knowledge management initiatives, however, are still based on centralized repositories and portals, which assume standardized vocabularies, languages, and classification schemes. These initiatives often fail due to trust and motivation issues [7], e.g. users resist on sharing knowledge, since they do not know who is going to access it and what is going to be done with it. Workers in the organizations have their own natural way to share knowledge. They usually gather in groups, based on similar interests, personal affinity and trust. These groups are commonly known as Communities of Practice (CoPs) [9]. The potential of CoPs in

KM has recently drawn the attention of both businesses and educational organizations. It is important to note that such communities cannot be forcibly created, but they may be fostered by creating the conditions, both social and technological, for a community to emerge [3,5].

Deciding which IT could enable CoP creation and management becomes a critical issue, which requires a deep understanding of the organization, of the common and the individual goals of its members. For example, consider the scenario where a company's newcomer wants to become a member of a CoP. The design of a CoP support system must take in account the different perspectives of the newcomer, the CoP and the organization's management. The analysis should start from questions such as: how can the newcomer's integration be facilitated by allowing him/her to become an active member of the CoP? Which assistance do community leaders need to help new members adjust to the community? This analysis enables understanding the rationale behind the newcomer's learning process and integration into the organization, and facilitates the elicitation of requirements of an IT infrastructure that could support them.

The agent paradigm offers an effective way to model and analyze complex systems composed of multiple and distinct components. An organization can be seen as a set of agents whose interactions are regulated by mechanisms of social order and are created to achieve common goals. In [6], an agent-oriented (AO) approach to elicit distributed KM needs has been presented. Along this line, we propose to use the *Tropos* agent-oriented methodology [2] to model the requirements of Information Technology (IT) to support KM processes within a CoP. This analysis may be the input to a system design phase, leading to the implementation of an adequate IT.

This paper is organized as follows: section 2 summarizes our approach; section 3 presents a fictitious scenario that we use to illustrate the proposed methodology; section 4 brings the actual analysis of the scenario; and, finally, some conclusions are presented in section 5.

2 Approach to the Requirement Elicitation Process

Our approach prescribes a methodology based on AO analysis, which considers individual and social goals in order to elicit the requirements for an appropriate IT solution to support KM in CoPs.

As depicted in Fig. 1, the proposed methodology starts with a *Problem Description* phase, in which specifics about the domain are collected in different ways (e.g. interviews with the target personnel of the organization, and active observation of the employee's activities, e.g. through ethnographical analysis). The next step is to carry out the analysis itself, for which the *Tropos* methodology [2] is applied. The analysis allows to point out the rationale behind stakeholders' needs of IT tools for enabling KM processes (*Early Requirements* phase). Finally, we are able to elicit the requirements for an IT solution and to trace them back to the fulfillment of the social and individual goals previously analyzed (*Late Requirements* phase). The elicited requirements are the basis for the architectural and detailed design phases. Notice that Fig. 1 shows that this chain of activities may be performed several times, in an iterative software development process.

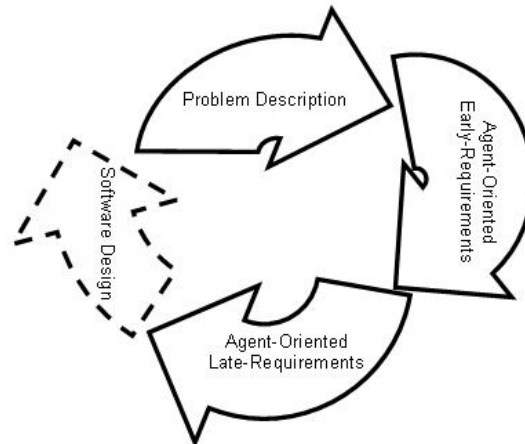


Fig. 1. A Cyclic Process to develop IT Support to CoPs based on AO analysis

The *Tropos* methodology supports an agent-oriented software development approach which consists of five main phases devoted, respectively, to early-requirements and late-requirements specification, to system architecture and detailed design and to implementation [2]. Basic artifacts of this process are the conceptual models which are built and refined according to a transformational approach. The modeling language adopted in *Tropos* provides basic constructs such as those of actor, goal, plan, softgoal, and resource. An actor can represent a stakeholder in a given domain, a role or a set of roles played by an agent in a given organizational setting. A goal is a condition or state of affairs in the world that an actor would like to achieve. A plan (or task) specifies a particular way of doing something, a particular course of action. A softgoal is used to represent how a state of affair should be reached. A resource is an (physical or informational) entity.

Dependency links between pairs of actors allow to model the fact that one actor depends on another in order to achieve a goal, execute a plan, or acquire a resource. The former actor is called the *dependor*, while the latter is called the *dependee*. The object (goal, plan resource) around which the dependency centers is called *dependum*. If the dependee fails to deliver the dependum, the dependor would be adversely affected in its ability to achieve its goals. In this sense, the dependor becomes vulnerable due to its dependency links. This type of information can be graphically depicted according to a diagrammatic notation derived from the *i** framework [10], using two types of diagrams: the *actor diagram*, a graph whose nodes represent actors (circles) and whose arcs represent dependencies (a couple of arrows linked by its dependum), and the *goal diagram*, a balloon including a goal analysis conducted from the point of view of a specific actor.

3 Problem Description: Fictitious Scenario

In this paper, we use a fictitious scenario to illustrate the proposed methodology. Although not a real case study, this scenario was carefully tailored, taking into consideration available literature [3,5,9]. Here follows a brief scenario description.

“Luca starts working in BHI Software Company. He is a programmer with 10 years of experience. As a newcomer at BHI, he needs to adjust to the organization’s work practices. This involves adapting to the work style of his working team and immediate supervisor, and also includes learning about the company’s policies and management directives. Aiming at providing its workers with a rich environment for knowledge sharing, BHI Management fosters the development of Communities of Practice (CoPs) across the organization. These communities are self-organizing groups whose members share interests and goals, or perform similar tasks within the organization. They are not necessarily from the same working team or division, and their members are dispersed across the 10 branches of BHI. BHI Management legitimates and supports the CoPs’ activities, granting incentives for those that stand out, contributing to the organization as a whole and providing technological infrastructures and tools. It’s important to note that the CoP leaders have an important role in motivating Luca, as well as the other members, on sharing knowledge. Especially when business processes are tight, answering some questions or making one's knowledge available will get very little priority. Besides, it may not be very clear for Luca what he will get in return for his willingness to contribute to the CoP members. External incentives, provided by the CoP, play an important role here. On the other hand, having appropriate Information System tools may also strongly contribute to his will towards knowledge sharing.”

4 Analysis

This section presents some results of our analysis of the scenario previously described, where the main stakeholders are the BHI organization, CoP and workers participating in the CoP. We model them as actors, according to the *Tropos* methodology. Our main objective is to analyze the goals of the scenario’s actors (*Early Requirements*), and to elicit the requirements for a system to support KM in this context (*Late Requirements*).

4.1 Analyzing goal dependencies between the Organization and the CoP

Figure 2 depicts the initial *Early Requirements* model where the dependency relationships between the Organization and the CoP are represented.

We start our analysis from the **Organization** actor, which has an initial softgoal of having the organization’s team working well. This softgoal leads the organization to CoPs fostering. The **Organization** relies on the **CoP** for creating and sharing knowledge, for developing new capabilities, and for strengthening trust among the employees. The **CoP** expects legitimization from the **Organization** (being legitimized goal dependency), being recognized as an important entity within the organization and receiving incentives. Inside the dotted rectangle, we analyze the structure of the **CoP**. Here, we show two members of the **CoP** (Member *i* and

Member i+1). All members of the CoP may assume two different roles concerning KM: *knowledge seeker* and *knowledge provider*. This is illustrated in Fig. 2 by the Seeker and Provider roles. In this example, Member i assumes the role of Provider, while Member i+1 plays the role of Seeker. The diagram shows the mutual dependency between these actors. The Seeker depends on the Provider for finding needed knowledge, and for developing required capability. On the other hand, the Provider depends on the Seeker for gaining visibility, and for receiving recognition.

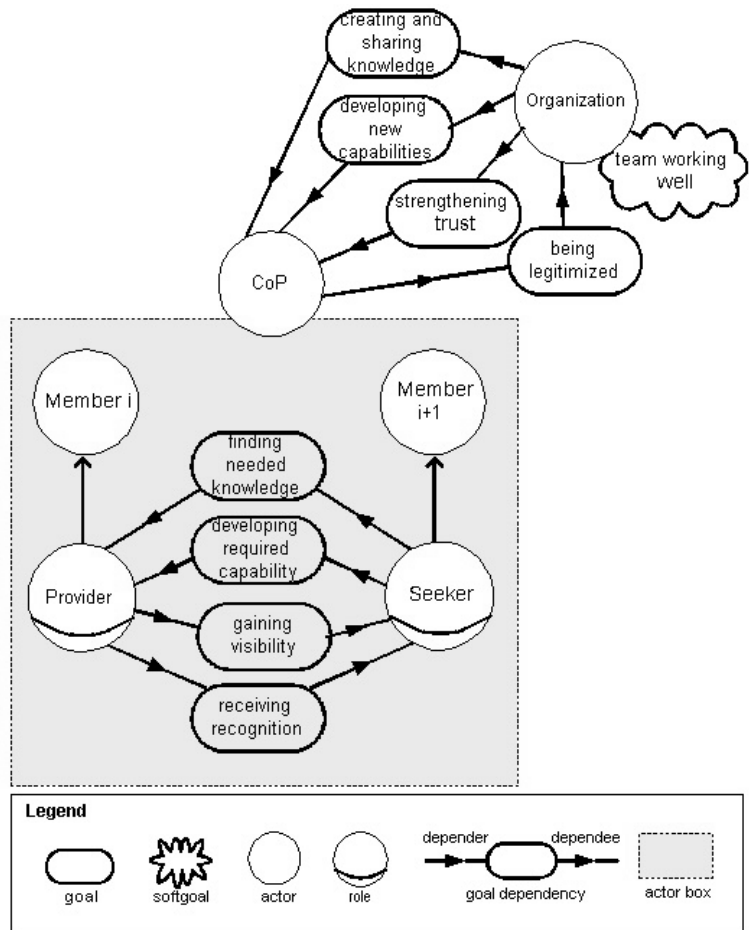


Fig. 2. Actor Diagram showing the dependency between the Organization and the CoP actors

Taking the Seeker and the Provider, we note that there are goal dependencies in both directions. This mutual dependency characterizes a “*sustainable relationship*”, i.e. a relationship in which two actors depend on each other to achieve one or more of

their own goals. Sustainable relationships indicate that there is some kind of balance between the two actors, thus helping them to achieve personal goals.

4.2 Analyzing the relationship between the Newcomer and the CoP

Figure 3 shows a Tropos Goal Diagram which models the Newcomer’s perspective. The Newcomer’s most general goal is the working well softgoal, i.e. he aims at doing his work efficiently, while also feeling good about himself and about the organization as a whole. In order to accomplish this, he aims at contributing with his competence and contributing with personal knowledge, gained in previous personal and professional experiences. Going Analyzing the contributing with personal knowledge goal, we note that two other goals contribute negatively towards it (not overworking and keeping control of his assets goals). These are common problems already noted by the KM community [3,7]. Issues of trust (keeping control of his assets goal) and motivation (not overworking goal) often lead to dissatisfaction towards the traditional centralized KM systems.

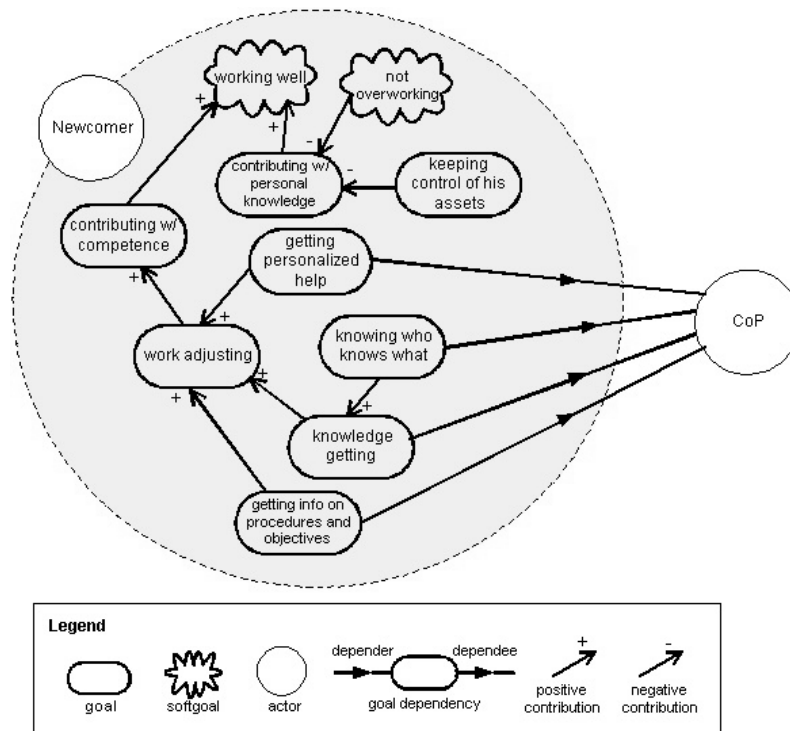


Fig. 3. A Goal Diagram showing the Newcomer’s perspective

Let us now analyze the contributing with competence goal a bit further. In order to fully and most effectively contribute with his acquired competence, the Newcomer must adjust to his work environment (work adjusting goal). In order to do so, the Newcomer needs new knowledge about his work and about the

organization as a whole (knowledge getting and getting info on procedures an objectives goals), depending on the CoP for this. Besides, the Newcomer depends on the CoP for getting personalized help when having a problem to solve, and to find out who has a specific piece of knowledge (knowing who knows what goal).

4.3 Delegating goals to the KARE System

Figure 4 depicts an excerpt of a *Late Requirements* model showing that the CoP delegates some of the goals of the newcomer to a system named KARE (Knowledgeable Agent for Recommendations).

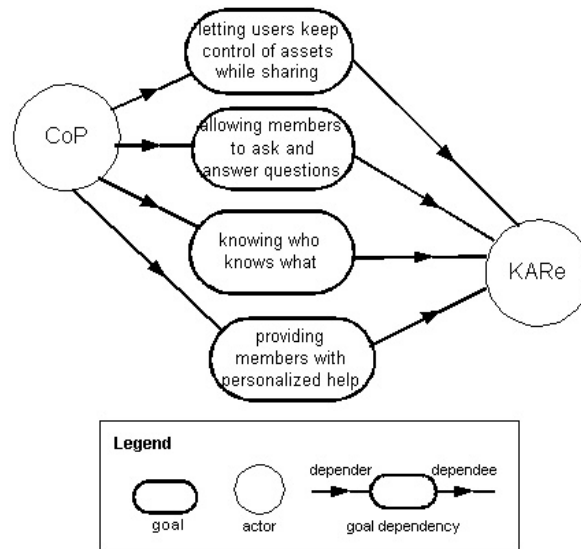


Fig.4. Actor Diagram showing the delegation of goals from the CoP to KARE

In order to fulfill some of the goals of the Newcomer (refer to Fig. 3), the CoP relies on the KARE system. The CoP delegates to KARE the goals of a) letting users keep control of their knowledge assets. The KARE system should allow each user to keep his assets in his own PCs, while making them available to other community members; b) allowing members to ask and answer questions through messages exchange. This feature is important because some of the newcomer's questions may not be answered by reading artifacts. Sometimes, it could be necessary to communicate with CoP members for building the solution to a specific problem. KARE should mediate this interaction, by finding the best colleague to answer to a specific knowledge request; c) informing who knows what; and d) providing members with personalized help, by consider their personal characteristics when providing knowledge. Therefore, these four goals become KARE's main requirements. By analyzing the four goals from the point of view of the system actors, we can identify more detailed requirements which can motivate the choice of specific type of technology, such as a peer-to-peer platform. Supposing that the stakeholders

(i.e. the organization's managers, the newcomer and the CoP members) are satisfied with such requirements, we can then proceed to the software design phase.

5 Conclusions and Directions for the Future

The interest in AO approaches for the analysis and the design of software systems has recently increased, especially in relation with the development of complex distributed systems. Basically, these approaches offer appropriate abstractions for modeling and designing properties of the organizational setting where the system-to-be will be deployed as well as for designing critical properties of the system, such as the interactions among system components. This paper presented an AO approach to support domain analysis for finding a KM solution within an organizational context which exploits the *Tropos* methodology to elicit requirements and model knowledge management domains. The approach has been illustrated using a fictitious scenario.

As related work, we mention OperA [4], a model for agent organizations that integrates both the individual and the organizations' perspectives. An OperA model can be thought of as a kind of abstract protocol that governs how member agents should act according to social requirements. However, OperA does not include the powerful mechanisms for requirement elicitation, provided by *Tropos*. An area for further research is therefore to investigate the possibilities to integrate the agent-oriented analysis presented here and the OperA framework. Further work remains to be done in the KARE System's design, for which we will apply AORML (Agent-Object-Relationship Modeling Language) [8], which provides an extension of UML to model agent-oriented information systems.

References

1. Bonifacio et al. (2000) A Distributed Intelligent Paradigm for Knowledge Management. [online] at: <http://edamok.itc.it/>
2. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., and Perini, A. (2004) Tropos: An Agent-Oriented Software Development Methodology. *Journal of Autonomous Agent and Multi-Agent Systems*, 8(3):203 – 236, May 2004.
3. Dignum, V., and van Eeden, P. (2003) Seducing, Engaging and Supporting communities at Achmea, at: <http://www.cs.uu.nl/~virginia/#Publications>
4. Dignum, V. (2004) A Model for Organizational Interaction: Based on Agents, Founded in Logic. PhD thesis, Utrecht University.
5. Gongla, P., and Rizzuto, C. R. (2001) Evolving communities of practice: IBM Global Services experience. In *IBM Systems Journal*, vol. 40, n. 4, 2001.
6. Perini, A., Bresciani, P., Yu, E., and Molani, A. (2003) Intentional Analysis for Distributed Knowledge Management. In *Agent-Mediated Knowledge Management*, L. van Elst, V. Dignum and A. Abecker eds., Springer LNAI 2926.
7. Pumareja, D., Bondarouk, T., and Sikkel, K. (2003). Supporting Knowledge Sharing Isn't Easy - Lessons Learnt from a Case Study. Information Resource Management Association Int. Conf. (IRMA'03), Philadelphia, May 2003.
8. Wagner, G. (2003) The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior. *Information Systems*, 28:5.

9. Wenger, E. (1998) *Communities of Practice: learning, meaning and identity*. New York: Cambridge University Press.
10. Yu, E. (1995) *Modeling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Department of Computer Science.