Ontology Based Proactive Design and Patterns towards the Adaptability of Knowledge Management Systems

Yinglin Wang[1] and Zheying Zhang[2]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
ylwang@sjtu.edu.cn
[2] Department of Computer Sciences, University of Tampere, FIN-33014, Finland
Zheying.Zhang@cs.uta.fi

Abstract. Knowledge management in large enterprises is composed of complex processes through which knowledge items are created, collected, evaluated and interconnected before they can be reused later. The challenge is how to cope with the different requirements of knowledge management systems, especially over times. The paper proposes a four level abstraction framework for modeling the commonalities and differences of knowledge management systems to cope with the possible changes in the future. Based on an ontology schema, the meta-meta model is given via combining the OSM model with task models and the commonalities from variability analysis. Then, the knowledge organization pattern, state diagram and task patterns are analyzed. The requirements and future changes are proactively modeled using patterns or variability models, so that the adaptability can be achieved through configuration by end-users in the deployment and runtime phases.

Keywords: knowledge management system, software reuse, ontology.

1 Introduction

Knowledge management systems are supposed to help people in the processes of the creation, capture, evaluation, retrieval and reuse of knowledge [1]. Just like Enterprise Information Systems (EIS) dealing with general data, knowledge management systems deal with knowledge items across all functional levels and management hierarchies and provide services coordinating organizational business processes [2]. Due to the increasing size and complexity of organizations and the business processes, meeting high stakeholders' expectations in a timely and a cost effective manner in an ever-changing and highly open dynamic turbulence environment is an essential goal of all information systems development [3],[4]. However, due to the intrinsic contradiction between the volatile requirements and the rigid software structure, trying to build adaptable systems is a great challenge. To overcome the difficulties, the abstraction strategy and patterns of generalized tasks in the collaborative knowledge management processes should be studied.

Aiming at building adaptable system architectures which facilitate runtime configuration of KM systems in accordance with the changing organizational needs

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 616–621, 2009.

and stakeholders' expectations, in this paper we propose a proactive adaptable knowledge management framework, which is based on ontologies and combines the object system model(OSM) with the task schema model. The task model of lifecycle collaborative knowledge management and the way of combining the unstructured and structured knowledge is discussed.

2 The Methodology

In order to achieve the objective, this study spans mainly two phases of KM system development, e.g. the requirements phase and the design phase, and covers four levels of abstraction, as shown in Fig. 1. Definition of the four levels of abstraction follows the architecture of the ISO IRDS (Information Resources Dictionary Standard (ISO 1990)) [5].

- The application level includes application data and program execution. It is a concrete presentation of an application. An example of information at this level would be a MS Word file, or a design case saved in a knowledge management system.
- The model level includes any intermediate specifications of the business processes and work flows. An example would be the definition of different file types, process templates, and functions templates that are used in a knowledge management system.
- The meta-model level specifies the languages in which specifications on the model level are expressed. The generic task types such as selection, identify can be used to represent the elements in the model level. It may also contain the specification of possible generalizations of the model level.
- Finally, the meta-meta-model level specifies how the meta-model level objects and concepts can be described and interlinked. It contains an abstract aggregated conceptualization of the related application domain. In our study, the ontology of EIS presents the concepts and their relations which are aggregations of the elements of the meta-model.

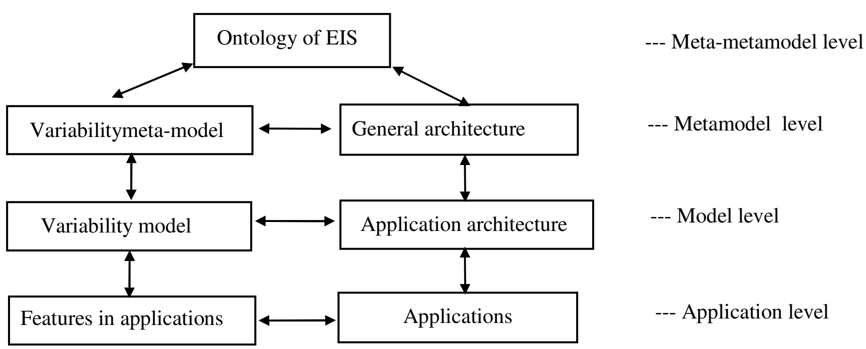

Fig. 1. Four levels of abstraction of system implementation

The bi-directional arrows illustrate the instantiation and abstraction relationships between models on the consequent abstraction levels. Horizontal arrows present the possibility to map requirements on to the system architectural design. The mapping occurs on every abstraction level.

3 The Meta-meta-Model of Adaptable KM Systems

It is reasonable to expect that 60 to 70 percent of a software application's functionality is similar to the functionality in other software applications, and that 40 to 60 percent of its code, and 60 percent of its design, is reusable in other applications [6]. KM systems have commonality to represent, record, control and compute the objects (e.g., organizations, people, machines and equipments, facilities, products, events and knowledge, etc.) and their interactive behaviors in each application function. The commonalities between different KM systems could be generalized as a reference framework (Fig. 2) and be reused. To model the commonalities, we combines the previous research results of object system models (OSM), task schema model described in the domain theory [7], and other required elements together.

The framework in Fig. 2 can be regarded as the specification of the meta-meta model level of EIS. It specifies and implements the most abstract concepts, relations

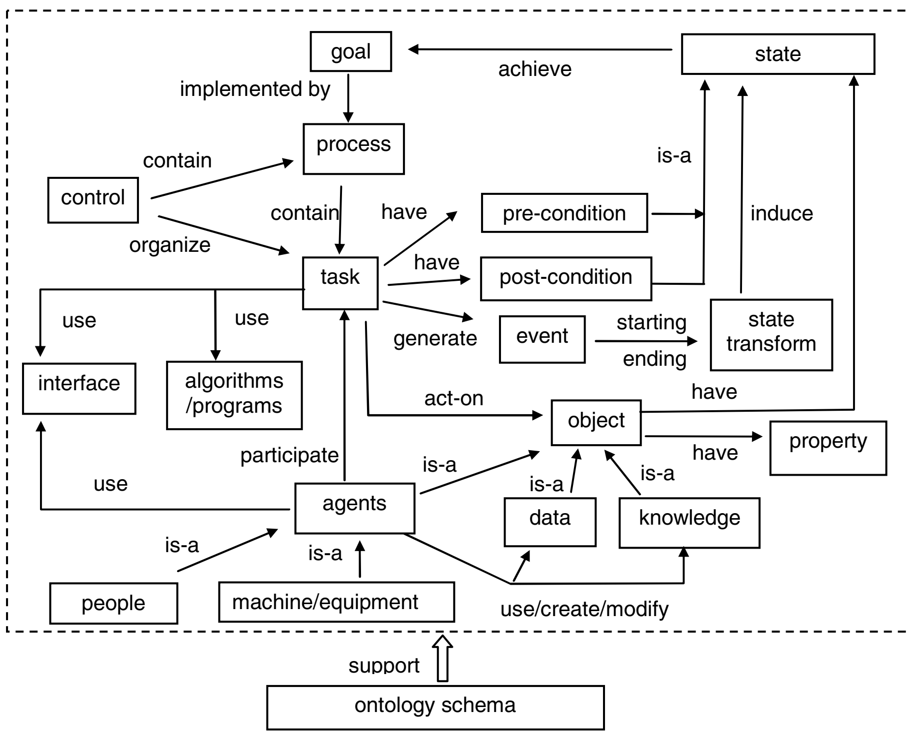

Fig. 2. The meta-meta model used for adaptable KM systems

and functions for every kind of EISs. These abstract elements can be used as a good starting point for reifying the specific domains and applications. As Fig. 2 shows, any operations of enterprises are related to one or several goals. Each goal can be implemented by one or more processes, and each process will contain several tasks which can be organized in certain sequential or concurrent relationships. The pre-conditions should be met before the execution of each action which is executed by authorized agents. At the same time, the actions will trigger certain events and may change the state of key objects. These concepts and their relations reflect a common pattern of EIS, including KM systems as the specialization. They can be further refined at the meta-model and model levels through specialization. Moreover, other common elements obtained through variability analysis phase can be added to enrich the framework. The ontology schema in Fig. 2 is an infrastructure to describe the concepts and relationships of the domain.

4 Patterns in Collaborative KM Processes

A body of formally represented knowledge is based on a *conceptualization or ontology*: the concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them [8]. In fact, every KM system is committed to some ontology and an ontology may be changed in the future.

Hence, when the ontology is represented explicitly, the system can be easily adapted if the change of ontology is required in the future. To represent domain ontologies, we should use a most general ontology language, or ontology schema.

4.1 Ontology Schema and Patterns for Knowledge Representation

The ontology schema includes five kinds of elements: set of classes (concepts), set of enumeration types, set of relations between concepts, set of relations between instances of concepts, and set of axioms. For details of the ontology schema, please refer to the paper [9]. Different with other ontology language, we introduce static properties in the class (concept) representation.

Definition 1: A static property is a property that has the same value for all the objects of the class. A property is normal if it is not static.

For instance, in a mechanical design scenario, a class named "bearing for shaft" may contain a static property whose value is a text description like: "A bearing is a device to allow constrained relative motion between two or more parts, typically rotation or linear movement". And it may have a static property whose value is a figure which is used to describe the semantic meaning of each parameter that will be used in the description of the instances.

We represent knowledge in a four level of abstractions in Fig. 3. At the meta-meta levels, knowledge is an aggregation of all kinds of knowledge in applications, and as the most abstract type it reflects the intension of knowledge and the relationships with other elements. At the meta-model level, more specific types of knowledge emerges, reflect some constructs of knowledge, such as the domain concepts, rules or formulas, cases, and processes to conduct some tasks. At the model level, the format of each type at the meta-model is given and used directly for the coding of the system.

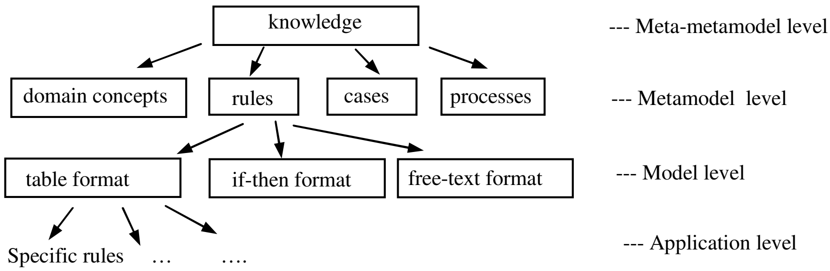

Fig. 3. The four level of knowledge organization

The specific rules and their structure, e.g., the parameters (i.e., properties) and the values, are configured at the application level by end-users. For instance, the structure (the parameters) of a rule for an object or related objects (e.g., a kind of mechanical part) can be defined in the table format as a class in the domain ontology, and then it is used to represent the permitted relationships of the parameters through a set of instances.

4.2 Task Patterns of Collaborative Knowledge Management

The goal of KM systems can be fulfilled by a series of activation and execution of tasks. In KM systems, tasks can be categorized as types of knowledge collection, knowledge evaluation, knowledge retrieval, knowledge activity statistics, etc. For represent the correctness or useful of knowledge items, several states from personal information to enterprise knowledge are used. The knowledge processing tasks will be organized to evaluate the states of the knowledge items.

The states of the knowledge items in the process of knowledge management can be generally classified into two kinds: "personal", or "enterprise". Knowledge items of state "enterprise" can be further classified into states of "quasi-knowledge", "knowledge", "rejected", "obsolete", and "processing". The state "processing" can be classified further into "evaluating", "modifying", "deleting", etc. The state transition diagram of a knowledge item is shown in Fig. 4, which is generalized from different applications and thus can be reused as patterns in building knowledge management systems. Moreover, task patterns can be modeled on the state diagram by specifying the preconditions, post-conditions and action sequences. As the abstraction of knowledge and the tasks, the generalized patterns can be reused to cope with many

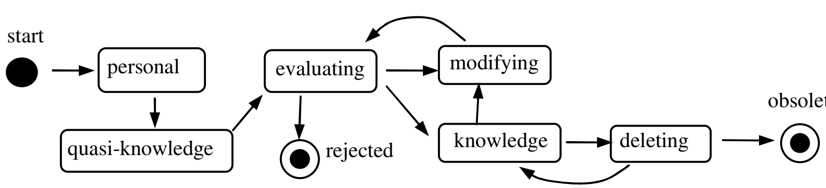

Fig. 4. The state diagram of a knowledge item in its lifecycle

specific cases that have the same features, while the differences are modeled as variants for configurations by end users.

5 Discussions

Through the four level of abstraction proposed for knowledge management systems, some of the commonalities and variability were found and they can be reused for handling changes in the future. For instance, the changes of domain ontology and the processes (such as adding new classes, modifying exist classes, redefining processes for an kind of knowledge) can be done directly by users without any coding if no new functions are required. The meta-meta model provide the infrastructure that is not likely to change. At the meta-model level, some generic tasks, such as identify, select and search an object, are modeled. The generic tasks can be used to construct the compound tasks, such as evaluation of knowledge items, at the model level. Because of the limited space, the task patterns could not be given in detail, which will be disscussed further in our next paper.

Acknowledgments. Thank the National High Technology Research and Development Program of China (Grant No. 2009AA04Z106), the Natural Science Funds of China (Grant No. 60773088) and the National Basic Research Program (Grant No. 2003CB317005) for financially supporting this research.

References

1. Birkenshaw, J., Sheehan, T.: Managing the Knowledge Life Cycle. MIT Sloan Management Review 44(1), 75–83 (2002)
2. O'Brien, J.A., Marakas, G.M.: Enterprise Information Systems. McGraw-Hill Education, New York (2007)
3. Liu, Y., Zhang, Z.: Stakeholder-Centered Requirements Elicitation: A View from User Research. In: 7th International Conference on Perspectives in Business Information Research 2008, Gdansk, Porland, pp. 25–26 (2008)
4. Perrone, V., Bolchini, D., Paolini, P.: A Stakeholders Centered Approach for Conceptual Modeling of Communication-Intensive Applications. In: SIGDOC 2005: Proceedings of the 23rd annual international conference on Design of communication, pp. 25–33. ACM, New York (2005)
5. ISO (1990) ISO-IEC 10027. Information Technology – Information Resource Dictionary System (IRDS) –Framework, ISO/IEC International Standard (1990)
6. McClure, C.: Software Reuse: A Standards-Based Guide. IEEE Computer Society, Los Alamitos (2001)
7. Sutcliffe, A.: The Domain Theory: Patterns for Knowledge and Software Reuse. Lawrence Erlbaum Associates Publishers, New Jersey (2002)
8. Genesereth, M.R., Nilsson, N.J.: Logical Foundations of Artificial Intelligence. Morgan Kaufmann, San Mateo (1987)
9. Wang, Y., Guo, J., Hu, T., Wang, J.: An Ontology-Based Framework for Building Adaptable Knowledge Management Systems. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 655–660. Springer, Heidelberg (2007)