REQUIREMENTS ENGINEERING DOMAIN KNOWLEDGE IN INFORMATION TECHNOLOGY

Rajeev Ranjan [1] & Dr.B.Mishra[2]
[1]Resource Person. Department of B.C.A, S.R.K.Goenka College, Sitamarhi
B. R. A. Bihar University, Muzaffarpur, Bihar, India
[2] University Professor & HOD Mathematics (Rtd), S.R.K.G.College, Sitamarhi,
B. R. A. Bihar University, Muzaffarpur, Bihar, India

Abstract

This research is concerned with identifying and defining domain knowledge obtained during requirements engineering, we term "Requirements Engineering Domain Knowledge". Further, we claim that, with a clear definition of this domain knowledge, and with an understanding of the way it affects information systems implementation, we can develop ways to improve development processes so that Requirements Engineering Domain Knowledge can be utilized in other contexts.

The definition of a scope for domain knowledge, the definition of its content, and the suggestion of methods for its representation. Using the results of this research, domain knowledge embedded in code can potentially be utilized in contexts other than IS development.

Since the introduction of IT in organizations, organizations have gradually become increasingly dependent on it. Business rules and knowledge about the domain, which used to be possessed by individuals, are nowadays often encoded in the organization's information systems. However, the encoded knowledge generally referred to as "domain knowledge". Since its scope and definition is not very clear, and since it is often "hidden" in the code, this knowledge is not readily accessible as a source of knowledge. Extracting domain knowledge from code requires programming knowledge as well as a deep understanding of the software and might not be practically possible even then.

The objective of this research is to develop ways that will advance our ability to explicitly represent domain knowledge gathered during the development processes. Such representation of gathered domain knowledge can be used to support explicit representation in system code so that this domain knowledge will be easier to extract.

Keywords:- Explicit, domain Knowledge

1. Introduction

An approach appealing to our research views RE mainly as a knowledge modelling process. This approach was also taken by Kavakli and Loucopoulo, who suggested that modelling of organizational change encompasses the following four concerns:

1. Understanding the current enterprise situation.
2. Knowledge about how change can take place.
3. Knowledge about the future enterprise system
4. The concern of evaluating enterprise models against the criteria of the parties Involved.

These four types of knowledge bring the RE process to four different states of knowledge respectively:

1. The As-Is knowledge state,
2. The Change knowledge state,
3. The To-Be knowledge state, and
4. The Evaluation knowledge state.

The two knowledge states, in which knowledge about the environment is established, are knowledge states 1 and 3. Domain knowledge consists of As-Is knowledge, which remains valid after the change process. It also consists of To-Be knowledge, which relates to entities that are part of the business environment. It is interesting to note that the order of transition between the states is not strictly defined, but rather the employed RE methodology defines the order in which the different knowledge states are traversed. i.e. the order of arrival at different knowledge states, As-Is knowledge state, Change knowledge state, To-Be knowledge state, and Evaluation knowledge states, is different in different RE methodologies.

While goal-oriented methods put more emphasize on the domain than behavioral and data modeling approaches, domain modeling approaches take the stand that even more emphasize should be placed on the domain rather than on the system. According to this approach goals by themselves do not make a good starting point for requirements engineering. Zave and Jackson illustrate this by considering a project to develop a computer-controlled turnstile guarding the entrance to a zoo.

2. SCOPE, GOALS, AND HYPOTHESIS

They show that domain knowledge is required to define the scope of relevant goals. If the engineers are told that the goal of the system is to deny entrance to people who have not paid the admission charge, it may be suggested that the goal has been stated too narrowly as perhaps the real goal should be to ensure the profitability of the zoo. Thus perhaps the engineers should consider other ways of improving profits, such as cutting costs. Following this line, it may be good to consider whether more money can

be made by closing the zoo and selling the land; and so on (Zave and Jackson). Basically, almost every goal is a sub goal with some higher purpose. Zave and Jackson therefore highlight the need to have a clear relationship between requirements and specifications. This relationship talks about the mediating effect of domain knowledge between specifications and requirements.

From the above it follows that specifications can be said to satisfy requirements only when incorporating domain knowledge. The domain knowledge basically guides us to the scope of relevant domain. More specifically, a specification together with relevant domain knowledge should be sufficient to guarantee that requirements are satisfied. This is formalized in Zave and Jackson :K,Sj-R Where K is a description of the problem domain, S is the specification of the solution, and R is the problem requirement. Parnas and Madey take the approach that requirements are in essence constraints imposed on the environment. They define the relation they designated as REQ, which incorporates constraints on the environmental quantities. Specifically Pamas and Meday define in the System Requirements Document environmental quantities, which are measured or controlled by the computer system. This document includes a specification on each environmental quantity as either monitored (aquantity that the system needs to measure), controlled (a quantity that the system needs to control), or both. The environment knowledge incorporated in this document can be described by a relation defined by Pamas and Madey as the NAT relation (standing for Nature):

- **Domain(NAT) is a set of vectors of time functions containing monitored values at different times**
- **Range(NAT) is a set of vectors of time-functions containing the values allowed by controlled variables**.

Work viewing RE under the domain knowledge approach dates back to 1982 with the work of Dubois, who suggested the **Entity-Relationship-Attribute-Event, (ERAE)** language. This language borrows from ideas in Semantic Networks and Logic.

The Language uses a semantic network type of graphical notation, and supports the use of the fundamental Entity Relationship constructs as well as the construct of Event. To this end, ERAE is also proposed as a possible language to help derive requirements for further validation, basing them on available requirements and temporal logic.

Today this view is mostly reflected in the Problem Frames method. As the name implies, Problem Frames structure the problem domain and describe the effects of the system on the problem domain. By emphasizing problems rather than solutions, Problem Frames can exploit the understanding of a problem class, allowing a problem owner with specific domain knowledge to drive the RE process (Hall

et al. 2005).The unique approach of problem frames is in the clear attempt to distinguish between system and environment requirements. The roles of descriptions in the framework are twofold.

1) Indicative descriptions express what is assumed to be true in the problem frame, and

2) Opative descriptions express what is desired to be true once using the system.

According to this, given domains have an indicative role, while the requirement description and machine specification have an optative role. In the context of our work, indicative descriptions about the domain are naturally the descriptions which relate to domain knowledge. In the Problem Frame framework, a problem diagram captures the characteristics of the problem domain as well as requirements that basically constrain the domain. Within a problem frame a machine domain is defmed. The machine domain is the system to be built together with its underlying hardware. In contrast to the machine domain, other domains, termed given domains, represent parts of the world that are relevant to the problem. These domains include physical events and states that are causally related. The different domains may share events and state information. These are called shared phenomena in the problem frames framework. Phenomenon a shared between two domains are observable by both, but controlled by one of them only.

Operators in the problem frame framework are the human operators. They are termed as biddable domains, meaning that they may obey stipulated procedures, but not reliably, and they may generate events spontaneously. In the problem frames framework, a requirement is defined as a condition in the problem domain that the machine domain must guarantee to qualify as a solution to the problem. The phenomena of biddable domains are events; those of causal domains are states and events (generalized as causal phenomena).The domain constructs of the domain modeling approaches are summarized in the meta model depicted in figure1.

3.Integration

In this section, our purpose is to create a meta-model of RE domain knowledge that will encompass the four RE approaches, namely the data based, the behavioural, goal-oriented, and domain based. To combine the domain knowledge elements included in the four models, we suggest creating a mapping of these constructs to those of the Enterprise Ontology proposed by Uschold et al. (1998). Uschold et al. (1998) provide a generalized ontology of domain concepts used in business. We are interested in requirements engineering domain concepts, which are only part of the ontology of business concepts.

Relationship

4-RELATIONSHIP

Construct Construct

Construct A is a Construct B Construct A Construct B

Construct B is a manifistation of Construct A Construct A — — "1Construct B

Figure 1: A Meta model of domain knowledge approaches constructs

4. Approach for Representing Requirements Engineering Domain Knowledge

In this section we provide a way for representing REDK domain knowledge constructs. Our approach is ontology based and object-oriented. The use of object constructs can facilitate the representation of REDK in Information Systems code. The approach and the outcome representation objects are described in this section.

4.1 Approach Explained

We are looking for a way to represent REDK explicitly in IS code. By representing REDK in system code, the ability to understand the domain using system code may be improved. One could examine knowledge associated with the domain using constructs commonly applied in Requirement Engineering methodologies. Another benefit stems from enabling the validation that an information system is a good representation of the application domain, as defined by Wand and Weber. More specifically, Wand and Weber suggest a set of requirements for an information system, necessary and sufficient for an information system to be a good representation of reality. These requirements all relate to a mapping between the real world defined as a triplet <S,L,E> (S is the possible state space, L is a set of system laws, and E is a set of external events) and an information system defined as <M,P,T> (M is the set of possible states, P is the information system law, and T is a set of external events). Having state variables, events, and laws explicitly represented enables validation that the requirements set by Wand and Weber are met. In our approach we use the widely applied 00 paradigm to enable the representation of REDK constructs. Further, Objects are introduced with ontological representation guidelines in order to have a coherent and well defined representation. Generally, we suggest that using the REDK meta

model constructs, explicit representation of REDK can be enabled. Taken to IS code, we will suggest a representation of REDK constructs using objects to enable direct use of the Meta model constructs in IS code.

4.2 Enabling the Representation of REDK -Ontology

While our analysis enabled the conceptualization of different knowledge elements, their representation still requires guidelines. For example, while the Service construct may be conceptually understood, its clear representation remains challenging. We use ontology to guide us in representing knowledge elements in a clear and concise way. Ontology, or metaphysics, the philosophy of existence, is the branch of philosophy that deals with modeling the existence of things in the world . Different ontology's are used within the field of Ontology. The different ontologies take different philosophical positions based on a set of beliefs about the existence of certain entities in reality. In other words, each ontology makes different assumptions about what is perceived to exist in reality and how it behaves. An ontology provides a set of constructs for describing the world. This ontology helped analyze concepts of IS (Wand and Webe). It was also used to analyze the meaning of constructs used in different conceptual modeling approaches. Specifically, Wand (1989)used ontology to analyze object-oriented concepts. Much work has also been conducted in analyzing modeling grammars. Finally, there has also been research in which ontology was used to help translate models between different representation-grammars.

4.3 Representation of REDK

4.3.1 General Outcomes of Ontological Foundations

We now turn to the task of representing the REDK in IS code, guided by ontological principles. In the context of this task we seek representation for concepts which are conceptually clear, but for which representation is not clearly set.

Conclusion

Domain knowledge is a concept commonly applied and referred to in the Information Systems development literature. However, a clear understanding of the content of domain knowledge has not yet been established. The term "Domain knowledge" has different meanings in different fields of research. When narrowed to the field of IS, domain knowledge has been described as "an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area". This is most often considered as the application area for which we develop software systems. This makes domain knowledge highly related to RE. Specifically, obtaining proper domain knowledge is recognized in software development as a critical factor for accomplishing the goal of RE, namely identifying complete, consistent, and accurate needs. In other words, during the requirements

engineering process domain knowledge must be acquired. In this process, knowledge about the purpose of the information system and about the domain of discourse is established using different types of models. This knowledge shapes the implementation of the information system, and is eventually manifested in software code.

The purpose of this research was to advance our understanding of the meaning of the term domain knowledge, to define its constructs, and to enable its representation based on ontological guidelines. Such an understanding can help us relate the processes taking place during Requirements Engineering to the actual system code. As well, it can enable the use of domain knowledge gathered during RE in subsequent phases of the system lifecycle.

In this research we have examined the term domain knowledge in different phases of the system development processes. Given the objective of RE, it is the process most intimately related to the accumulation of domain knowledge We have therefore conducted an analysis of domain knowledge discussion and use within the RE literature. We have found that different requirements engineering approaches emphasize the modelling of different aspects associated with both the domain and the system.

Reference

1.Abbot, R. 3., Knowledge abstraction, Communications of the ACM, ACM Press, NY, 1987, 664-671.

2.Abrahamsson, P., Salo ,O., Rankainen, J., Warsta, 3., Agile Software Development Methods – Review and Analysis, VTT Electronics, 2002

3. Alavi, M ., Leidner, D. E., Review: Knowledge Management and Knowledge Management Systems:Conceptual Foundations and Research Issues, MIS Quarterly, 25:1, 2001, 107-136

4. Albano, A., Bergamini, R., Ghelli, G., Orsini, R., An Object Data Model with Roles, in: R. Agrawal, S. Baker, D. Bell (Eds.), Proceedings of the 19th International Conference on Very Large Databases, Morgan Kauflnann, Dublin, 1993, 39-51 Alford, M. W., Lawson,

5.T, Software Requirements Engineering Methodology (Development). Angeles, P., Dictionary of Philosophy, Harper Perennial, NY Anton A., Goal-Based Requirements Analysis. Proc. Second IEEE International Conference on Requirements Engineering (ICRE'96), Colorado Springs.

6. Appleby, D., Vandekopple, J.J., Progranmting Languages Paradigm and Practice, McGraw-Hill

7. Avison, D.E., Fitzgerald, G., Information Systems Development: Methodologies, Techniques and Tools.

About Author

Mr. Rajeev Ranjan received his Master's degrees in Computer Application in the years 2005 from IGNOU Patna. Currently she is working as Resource Person in Deptt. Of B.C.A. at S.R.K Goenka College, Sitamarhi Bihar (India). He has published 4 research papers in various National/ International journals. Her areas of research are Explicit Domain Knowledge Representation in Information Technology.