Knowledge Entry Maps: ontology to structure method knowledge in the IT industry

Remko Helms, Sjaak Brinkkemper (UU)
Frank de Nijs, Job van Oosterum (Centric)

institute of information and computing sciences, utrecht university
technical report UU-CS-2004-035
www.cs.uu.nl

Knowledge Entry Maps: ontology to structure method knowledge in the IT industry

Remko Helms [1], Sjaak Brinkkemper[1], Job van Oosterum[2], Frank de Nijs [2]

[1]Institute of Information and Computing Sciences, University Utrecht, The Netherlands
{R.W.Helms, S.Brinkkemper}@cs.uu.nl
[2]Centric, IJsselstein, The Netherlands
{job.van.oosterum, frank.de.nijs}@centric.nl

Abstract. By providing professionals in the IT industry with knowledge about the day-to-day routine the efficiency of processes and the quality of the output can be improved. The knowledge can be made easy accessible through a knowledge infrastructure. An important requirement for the accessibility is how the knowledge is structured in the knowledge infrastructure. This paper presents a technique for modeling knowledge structures that is applied in two case studies in the professional IT industry. The results of the case studies are compared to analyze if there exists a generic structure for knowledge about the execution of the processes.

1 Knowledge in the professional IT industry

Knowledge intensive companies in the IT industry such as software developers and service providers depend on their employees for the proper execution of their business processes. Knowledge of these processes has become a valuable asset for these companies that can give them a competitive advantage over their competitors. Therefore companies should leverage this knowledge and make it available to all employees. Knowledge about the execution of processes is referred to in this paper as method knowledge. A method can be defined as an integrated collection of procedures, techniques, product descriptions, and tools for effective, efficient and consistent support of business processes [4]. Hence method knowledge is the explicit representation of these procedures, techniques, product descriptions, and tools, i.e. knowledge about the method.

For developing a method companies can use a method engineering approach. Method engineering is the engineering discipline to design, construct, and adapt methods [5], [6], [11]. The deliverable of the method engineering approach is a documented, i.e. explicit, method. For effective leverage of this method knowledge it is important to share this knowledge within the organization [14]. Knowledge repository systems can be used for this purpose [3]. Examples here are the Powerpacks at Ernst & Young [8] and the Electronic Sales Partner at Hewlett Packard [6]. Both systems are used to make method knowledge available to consultants and sales people respectively.

Retrieving knowledge from repository systems is often based on browsing certain categories or keyword search. From other research is known that ontologies can be an effective means for indexing and retrieving knowledge [1]. As a first step in improving the sharing of method knowledge in IT organisations we conducted two case studies to discover if ontology based modelling can be used to structure method knowledge. For this purpose we introduce Knowledge Entry Maps and the use of these maps is evaluated in the case studies. Moreover, the case studies offered the opportunity to compare the Knowledge Entry Maps to find out whether there is such thing as a generic structure for method knowledge in the IT industry.

2 Research approach

The problem definition as presented in the previous section can be translated in the following problem statement: *Can ontology based modelling be used for structuring method knowledge in IT organizations*

To study this topic we use a Design Research approach, which is widely accepted in the field of Information Systems research. The philosophy behind Design Research is that new scientific knowledge can be generated by means of constructing an artifact [7], [9]. In the following we present the main steps of our research based on the Design Research methodology [7].

Awareness of problem: Awareness was raised through several discussions with professionals responsible for organizing IT knowledge in the professional IT industry. They indicated the problems of making method knowledge available to the employees in their organization (section 1).

Suggestion: This resulted in the idea to use ontology modeling to define structures for storing method knowledge in a knowledge repository system.

Development: For this purpose we developed a (tentative) method for modeling IT knowledge structures called Knowledge Entry Maps (section 3).

Evaluation: The Knowledge Entry Map modeling method has been applied in two case studies (section 4). The goal of applying the method is to study whether it is a useful aid in capturing the structure of method knowledge in IT organizations. Applying the model will reveal any possible problems and suggestions for improvement. Moreover, we compared the results of the case studies to analyze the possible existence of a generic structure for structuring method knowledge in IT organizations (section 5).

Conclusion: Based on the evaluation the conclusions were formulated (section 7).

Finally it should be noted that the case studies have been conducted sequentially. This offered the opportunity to improve the Knowledge Entry Map before using it in the second case study. In other words there have been two iterations for further improving the method.

3 Knowledge Entry Map: structuring method knowledge

3.1 Basics of structuring knowledge

For structuring IT knowledge we introduce the concepts of knowledge fragments and knowledge structure elements.

Knowledge fragments are defined as small portions of the total method knowledge of a company. A knowledge fragment contains enough knowledge so that it can be applied independently of other knowledge fragments. Examples of such knowledge fragments in an IT organization are a checklist for making a proposal, a spreadsheet for making ROI calculations, and a standard sales presentation. In other words a knowledge fragment is an abstract representation of codified knowledge that is stored in electronic files.

Knowledge structure elements are defined as classes of knowledge fragments with content on a similar topic. For example, a checklist for making a proposal and a standard sales presentation, both knowledge fragments, could both refer to the same Product X. In that case Product X is called the knowledge structure element. Individual knowledge structure elements can be linked to other knowledge structure elements, indicating that a one element is associated with another element. Product X can for instance be associated with Product Y or with a specific Industry A to which it is sold. Linking knowledge structure elements results in a network of knowledge structure elements that we also refer to as the knowledge structure.

Graphically this can be represented as shown in figure 1. It shows a structure level and a fragment level, indicating that the structure is separated from the knowledge fragments itself.

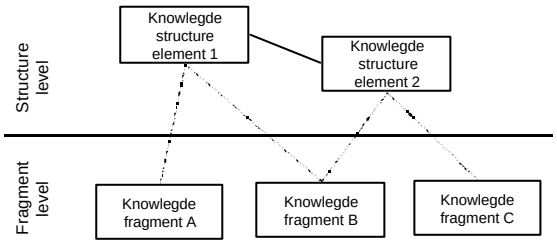

Fig. 1. Structuring knowledge fragments

The structure level shows the knowledge structure elements and the associations between the different elements. While the fragment level shows knowledge fragments that are related to each other indirectly via the knowledge structure elements. By implementing this concept in a knowledge repository system, users of the system can locate knowledge fragments by browsing the structure level and looking up related knowledge fragments.

3.2 Knowledge Entry Map

To capture the structure of method knowledge in the IT domain the concept of ontologies is used. Richter defines ontology as: "a formal, explicit specification of a shared conceptualization" [13]. We agree with Richter that database diagrams, like Entity Relationship (ER) diagrams, are appropriate for modeling ontologies [13]. Normally, ER diagrams are used to construct data models using basic graphical symbols to show the organization of and relationships between data [16]. Using the ER diagram for modeling the structure of method knowledge results in an overview of the structure of method knowledge in a specific domain. Such a model is a graphical representation of ontology because it involves an explicit specification of a shared conceptualization. Others refer to this type of overview as knowledge maps [2] or concept maps [3]. However, they do not specify a comprehensive modeling method. Therefore we introduce Knowledge Entry Map to provide a more formal modeling method for graphically representing ontologies using ER diagrams.

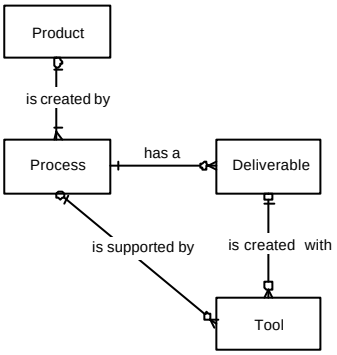

Fig. 2. Example of knowledge entry map

The basic constructs of Knowledge Entry Maps are:

1. **Concept**: A *concept* is a basic notion that classifies knowledge structure elements into classes with similar properties. Concepts are typically described using a noun, examples are Product, Process, and Deliverable.

2. **Association**: An *association* is a relationship between concepts and are typically described using a verb, examples are is created by and is supported by. In practice one will refer to an association by also mentioning the concepts, for example Product is created by Process.

3. **Rule**: A *rule* is a restriction on one or more associations. Examples are: A Process is supported by any number of Tools and A Tool supports at least one Process. Rules are defined using the cardinality concept of ER diagrams.

4. **Entry points**: An *entry point* is a concept that serves as a starting point for accessing knowledge fragments stored in the knowledge infrastructure.

Figure 2 shows an example of a Knowledge Entry Map. For creating the diagrams one can use simple tools such as PowerPoint or Visio that contains templates for ER diagrams.

4 Using KEM to structure method knowledge – case studies

4.1 Background of case studies

The case studies are applied in two different types of companies in the professional IT industry. The first company is Baan, which is a major developer of ERP software. Around 2000 Baan was an independent company with about 5,000 employees, 8,000 customers, and 15,000 operational sites. After some downsizing it has recently been acquired by SSA Global. The Baan Research and Development unit develops the Baan products with offices in the Netherlands, India, and Germany. This department employs about 600 software engineers.

The second company is Centric, which is a provider of IT services. Its Headquarters are located in the Netherlands and there are offices in Belgium and Germany as well. In total Centric has 2,750 employees but this case study was conducted in the Managed ICT Services (MIS) division that has 1,000 employees. The MIS division is specialized in office automation products and services, such as consultancy, helpdesk services, migration services, ICT management and Outsourcing.

As researchers we got involved at the case study companies at the stage where they were investigating how they could make method knowledge better accessible for all employees in their organization. Both companies were interested in our Knowledge Entry Map to find out if it would help them to structure their knowledge. Consequently we created the KEM's for both companies in co-operation with employees of the organization. The employees received a short introduction into the KEM modeling method in order to be able to interpret the KEM and to provide feedback. As stated before, the case studies were conducted sequentially. Hence it was possible to improve the modeling method in two iterations. Afterwards we compared the two KEM's that were constructed in the two case studies to study if there is a generic structure for method knowledge in the professional IT industry.

4.2 KEM of Baan

The Knowledge Entry Map that is constructed by the team at Baan is presented in figure 3. We will discuss part of the KEM in this section for illustration purposes.

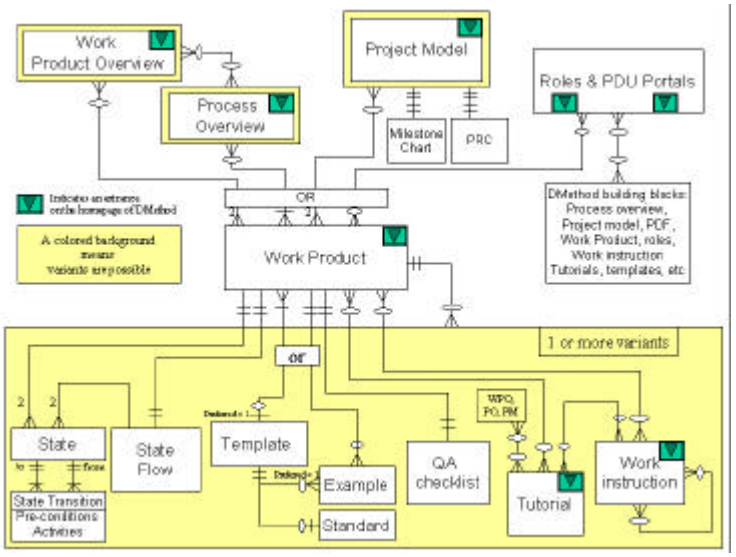

Fig. 3. Knowledge entry map of Baan

The main *concepts* in the Knowledge Entry Map at Baan are:

Project Model: Project Models involve all deliverables needed to complete a project milestone. Project models are available for Release delivery, Software development, Knowledge transfer, Integration test, and System test.

Process overview: The main phases or steps in the development process. Examples of these phases are Feasibility study, Design and System Test.

Work Products: Work Products are standards for documentation of deliverables. Examples are project plan, version definition, definition study, software unit, test design.

Work Instructions: Work instructions are detailed procedures for the completion of well-defined smaller tasks, such as performing a software unit test, or a project audit, or risk management.

Roles: A set of responsibilities that is defined within the organization. Examples are: Software Engineer, Test Coordinator, and Project Leader

Tutorials: Training materials to learn how to execute various tasks in the method.

Some examples of *rules* are:

Rule 1: A Work Product has always 2 or MORE States.

States are formally defined versions of the work products (WP), such that every employee in the R&D department can judge the status of the contents and impact for the ongoing job. Each WP has a state Initial, a state in which there is just an empty tem-plate or a copy from an old version. The Actual state refers to a WP in which the content is available to be used. Next to these two mandatory states, it is preferred to have a third state in which the contents are at the end of its lifecycle. This state is preferably called Historic.

Rule 2: A Work Product can have zero or 1 Template, but preferred is 1

In case the WP is a document, a template is mandatory. If it is not a document (for example a change request, which is a record in a database), a template may be omitted.

At Baan they build a knowledge repository for managing method knowledge called DMethod. It is built using html pages where the homepage shows the main entries as indicated in the KEM. By accessing for instance the processes section you get a html page that shows a list of all processes (all instances). When accessing a process, i.e. an instance, a description of that process is shown including the relation to specific instances of work products that are related to that process.

4.3 KEM of Centric

The Knowledge Entry Map that is constructed by the team at Centric is presented in figure 4. We will discuss part of the KEM in this section for illustration purposes.

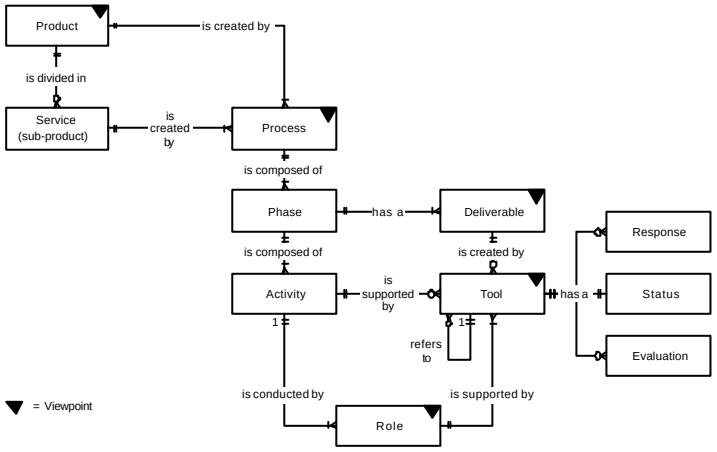

Fig. 4. Knowledge entry map of Centric

The main *concepts* in the Knowledge Entry Map at Centric are:

Product: The services that are sold to the customer, examples are ICT management and Helpdesk services. Centric has defined standard services that are altered for specific customers.

Process: A defined set of activities to produce a specified output for a particular customer or market. An example of a process is the Execution process that is part of the ICT management product.

Phase: The defined stages or steps in a process. For example Implementation preparation, Implementation and Exploitation, which are the phases in the Execution process.

Deliverable: The standardized end products of a phase, examples are project plan, proposal, and service level agreement.

Tool: An aid that is used to create a deliverable. Examples are a ROI spreadsheet, a template proposal, and an example project plan.

Role: A set of responsibilities that is defined within the organization. Examples are: account manager, product manager, and project leader.

Some examples of *rules* are:

Rule 1: A Deliverable is created by ZERO or MORE Tools.

Although standard Deliverables have been defined there is not a Tool for every Deliverable available in the method. An example of a deliverable might be a data model. When there is no tool attached to it, the user can select any tool he likes.

Rule 2: A Tool is supported by EXACTLY one Role.

For every Tool is should be known which Role is using the tool. In the case of Centric it is not possible to add a tool to the method base without specifying the Role that uses it. This rules makes it possible to make an overview of all the Tools that can be used by a single Role.

At Centric they have developed a prototype of a knowledge repository based on Microsoft Sharepoint Portal server. However, it was not possible to use the KEM, i.e. ontology, as a basis for browsing the knowledge stored in the repository. At the moment they are evaluating if there is a commercially-of-the-shelf system that is capable of supporting ontologies.

5 Discussion

5.1 Capability of KEM

The case studies revealed that both companies were able to map the structure of their method knowledge with KEM. Only two extensions were required in order to match the modeling requirements of the companies. We consider it to be minor

changes because it only required an extension of the current constructs instead of introducing completely new constructs. However, the changes can be a potential threat with respect to easy interpretation of the model because it increases the complexity of KEM. In the following the extensions to KEM are presented:

- **Variants of a concept**. The different product lines of Baan R&D used different development platforms, which gave necessity to accommodate variants of templates, and therefore of work products and instructions. In the design phase conventional 4GL development used Entity-Relationship modeling, whereas object-oriented approaches needed Class diagrams, resulting in a Work product for a Functional Design in two variants. Variants were indicated in the model by adding a colored background to a specific concept and the specific conditions were described in the rules.
- **Condition based associations**. At Baan we encountered the situation that an *example* (A) only needed to be associated with a *work product* (B) if there was no *template* (C) associated with a *work product* (B) and vice versa. In other words the existence of an association between two concepts A and B was dependent on the non-existence of an association between concepts B and C. This resulted in the introduction of the 'OR' condition on two associations. Placing a box on the involved associations in the KEM graphically indicates the type of logic condition that applies to the associations.

Considering the minor changes we are convinced that KEM can also successfully applied at other companies for structuring the method knowledge, especially software developers and service providers.

5.2 Generic structure of method knowledge

This section discusses a comparison of the Knowledge Entry Maps to find out if there is something as a generic structure for method knowledge in the professional IT industry. Comparing the KEM's shows that there are similarities because they contain to a large extent similar concepts and associations. For example both contain concepts such as Process (i.e. Process versus Process overview) and Deliverable (Deliverable versus Work Product). These similarities can be caused by the fact that the case studies were conducted sequentially. However, the influence of the outcome of the first case study on the second case study is low because the business requirements of each company have been the source for constructing the KEM.

Despite the similarities that might support the possible existence of a generic structure, there are also some major differences:

- At Baan they use a generic description for their projects. Consequently all projects have the same phases, milestones and work products. Therefore a key concept in their KEM is the Project Model. Centric on the other hand uses a different approach for different Products. Consequently the phases in a project depend on the type of product. Therefore a key concept in their KEM is the Product.
- Studying the associations in both KEM's reveals that for Centric most of the associations are mandatory while for Baan most of the associations are optional. This results in a more rigid structure for Centric and a more flexible structure at

Baan. Consequently at Baan they can add an instance of for example a Tutorial and it is not required to link it to any other instance of a concept.

The fundamental difference between a project oriented KEM at Baan and a product oriented KEM at Centric does not support the idea of a generic structure for the whole IT industry. On the other hand it shows that the way in which a company organizes its business, i.e. project vs. product oriented, influences how it structures its method knowledge. But it goes to far to say that we have found two generic structures because it is supported by only two case studies. Further research would be required to validate if we have found a generic structure for project oriented and a generic structure for product oriented companies.

5.3 Usage of KEM

Applying the KEM in the case studies also resulted in some additional findings. First of all it was experienced that other people involved in the project found it rather easy to learn the basics of KEM. A possible explanation is that people in the IT industry are familiar with data modeling and therefore find it rather easy to master KEM. On the other hand this knowledge of data modeling is a disadvantage, because a KEM is *not* a data model, but a meta-level model of knowledge structures. So after the initial (easy) introduction of KEM we had to discuss this difference in more detail with the team members so that they would fully understand the KEM.

Secondly, it was experienced that it can be hard to identify concepts in the organization, because the method does not prescribe how to elicit this information from the organization. In both case studies information was gathered using known methods for business analysis, using interviews and complimentary models such as value chain models and process models. Although these are not focused on identifying concepts it provided satisfactory input for creating the KEM.

6 Related work

Using ontologies for structuring knowledge is not a new field of research. Zhang *et al*. show that ontologies can be used for structuring archaeological knowledge [16]. Another example uses ontologies for structuring 'relevant human resource topics', where the KAON OI-modeller is used to make a graphical representation of the ontology [11]. A third example is OntoWeb, which is a semantic web community portal [14]. In that case they use ontology to structure information that comes from different web communities. In the studies the ontologies are used to enable ontology based browsing for finding knowledge on the Internet [11], [14]. It enables the user to browse the ontology and to find knowledge related to a specific concept. This option is offered next to full text search and attribute search, giving the user several options to find the knowledge that he needs.

It is also interesting to see that one of the studies also mentions the use of workflows for publishing information [14]. The workflows are used for reviewing the information before it is incorporated in the system. This is also a topic in method

engineering where new parts of the method have to be approved before they are made available to others in the organization. Another study shows how an ontology browser can be used on top of a document management system [9]. This is relevant because much of the method knowledge is stored in documents. Therefore document management functionality is also an important system requirement at the case study companies.

Summarising it shows that structuring knowledge using ontologies is researched by others with similar intentions, namely to make knowledge better accessible in knowledge repositories. What makes our research unique is the application in the domain of method engineering and professional IT organizations.

7 Conclusion

In this paper we showed how ontologies can be used to model the structure of method knowledge in professional IT organizations using Knowledge Entry Maps that are based on ER diagrams. The case studies showed that only small changes to the existing constructs of the modeling method were required in order to be able to model the specifics of the structures of method knowledge at both case studies. But the case studies also revealed that further research is needed to develop a technique for identifying concepts in the organization.

Comparing the knowledge structure did not provide evidence that there is a generic structure for method knowledge in professional IT organizations. Further research is needed to validate if we have found a generic structure for project-oriented organizations and a generic structure for product-oriented organizations.

Another direction for future research is the development of a system for sharing method knowledge in the organization. Our experience in the case studies points in the direction of document management systems including workflow functionality and an ontology browser. This idea is supported by the work that we discussed in the Related Work section. Because document management systems with workflow functionality are commercially available we think of an ontology browser as addition to these available systems.

8 Literature

1. Aitken, S., Reid, S., Evaluation of an Ontology-based Information Retrieval Tool. In proceedings of the ECAI-200 Workshop on Ontologies and PSMs. Berlin, Germany, 2000.
2. Awad, E.M., Ghaziri H.M., Knowledge Management. International edition, Pearson/Prentice Hall, 2004.
3. Becerra-Fernandez, I., Gonzalez, A., Sabherwal, R., Knowledge Management – Challenges, Solutions and Technologies. Pearson/Prentice Hall, 2004.

4. Brinkkemper, S., Lyytinen, K., Welke, R.J. (Eds.), Method Engineering: Prin-ciples of Method Construction and Tool Support. Chapman and Hall. 1996.
5. Brinkkemper, S., Method Engineering: Engineering of Information Systems Development Methods and Tools. Journal of Information and Software Technology, Vol. 38, Nr. 4, pp. 275-280, 1996.
6. Davenport, T., Prusak, L., Working Knowledge – How organizations manage what they now. Harvard Business School Press, 1998.
7. Design Research pages on IS World, http://www.isworld.org/Researchdesign/drisISworld.htm, July 8, 2004.
8. Dixon, N.M., Common Knowledge: How companies thrive by sharing what they know, Harvard Business School Press , 2000.
9. Guba, E. and Lincoln, Y., Competing Paradigms in Qualitative Research, in The Handbook of Qualitative Research by N. Denzin and Y. Lincoln, Thousand Oaks, CA, Sage**:** 105-117, 1994.
10. Kim, H.-L., Kim, H.-G., Park, K.-M., Ontalk: Ontology-Based Personal Document Management System, In: Proceeding of 13th Interational WWW Conference, http://www2004.org/ proceedings/index.htm, July 12, 2004.
11. Kumar, K., Welke, R.J., Methodology Engineering: A Proposal for Situation-Specific Methodology Construction. In: W.W. Cotterman, J.A. Senn (Eds.), Challenges and Strategies for Research in Systems Development. Wiley, 1992.
12. Maedche, M., Web Information Tracking Using Ontologies. In: Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management. Lecture Notes in Computer Science 2569, pp. 201-212, Springer Verlag, 2002.
13. Richter, M.M., http://sern.ucalgary.ca/courses/SENG/609.34/S2003/ slides/09.%20Ontologies.pdf, July 6, 2004.
14. Slooten, C. van, Brinkkemper, S., A Method Engineering Approach to In-formation Systems Development. In: Information Systems Development Process. Elsevier Science Publishers (A-30), pp. 167-186, September 1993.
15. Spyns, P., Oberle, D., Volz, R., *et al*. OntoWeb - A Semantic Web Community Portal. In: Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management. Lecture Notes in Computer Science 2569, pp. 189-200, Springer Verlag, 2002.
16. Stair, R.M., Reynolds, G.W., Fundamentals of Information Systems. 2nd edition, Thomson, 2003.
17. Zhang, C., Cao, C. Gu, F., *et al*., A Domain-Specific Formal Ontology for Archaeological Knowledge Sharing and Reusing. In: Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management. Lecture Notes in Computer Science 2569, pp. 213-225, Springer Verlag, 2002.