

Working with a Knowledge Management Tool in a Domain Context

Tang-Ho Lê¹ and Luc Lamontagne²

¹ Computer Science Department
Université de Moncton, 165 Massey Ave, Moncton, (N.B.)
CANADA, E1A 3E9

² Département d'informatique et de recherche opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal (Québec)
CANADA, H3C 3J7
letangho@umoncton.ca
lamontal@iro.umontreal.ca

Abstract

In this paper, we provide an overview of our software tool to exploit and interchange procedural knowledge represented as networks of semi-structured units. First, we introduce the notion of Procedural Knowledge Hierarchy; then we present the modeling of Procedural Knowledge by our software. We claim that the “bottom-up” approach, that can be carried out by this tool, is appropriate to gather new candidate terms for the construction of a new domain ontology. An educational project, conducted with our tool, is briefly reported.

1 Introduction

Current research on the semantic web has emerged from the most pressing problems of information overload on the Web. This phenomenon is natural as the Web represents a virtual medium in which people, with different cultural and knowledge background, can disseminate documents and interchange ideas, products, etc. However exploitation of this information is complicated by its abundance. For instance, the evolution of terminologies and their semantic is a source of ambiguity. Several new words occur daily and some terms even change their meaning when used in different domains. For example, the term “home page” did not exist before and the term “navigating” now has a new meaning when used in the context of the Web which differs from its nautical origins.

The research community has been conducting for several years intensive research to devise formal methods to understand and to determine declarative knowledge based on ontologies. Many research groups are formed and contribute results to this issue; for example, several proposals were made for ontology languages and ontology builder such as DAML, Protege 2000, Ontolingua, etc.

But there is also a need to better represent procedural knowledge which is still poorly understood. To our knowledge, fewer research efforts were devoted to model

and specify knowledge being procedural in nature, and to combine this with declarative knowledge available on the web. In this paper, we report on our work to address some of these issues. First, we emphasize the essential role of procedural knowledge in the knowledge interchange, then we discuss our software tool used to describe procedural knowledge in a semi-structured form. Afterwards, we report on some experiences using this tool in the practical context of an e-learning project.

2 Procedural Knowledge

To progress toward a semantic web in a large sense, the classification of concepts into a giant hierarchy, with the terms being assigned according to the community’s consensus, is just one half of the solution. This half is based on the *knowledge about knowledge*, while the other half is, after having *access to* the right knowledge in this giant hierarchy, to know *how to perform* some tasks to *produce the thing abstracted by this knowledge*. Put in other words, we want to know how to create the product P after having the concept of P. This refers to various knowledge to be tackled. The first (categorize and classify) suggest “the ontology” that belongs to the declarative knowledge while the second (how to perform some tasks) is the procedural knowledge.

The procedural knowledge is at the core of the knowledge that we interchange daily. To interchange efficiently, first, a consensus about the elements of discourse is needed, and then, some knowledge about how to achieve or better perform our tasks is required. The world (or an application domain) can not be solely conceived as static, its evolution being partially made possible through human actions. Our knowledge encompasses the knowledge about these actions. This kind of knowledge is essentially what we want to know (know how) in addition to the “know what” in our daily learning. For instance, in the web page domain, we firstly need the definition of terms such as hyperlink, site structure, etc. but we also need to know “how to make links” and “how to navigate in the web site”.

3 Procedural Knowledge Hierarchy

In the design of a Knowledge Management (KM) system, one has to cope with the granularity problem. If we want to efficiently transfer the desired knowledge to knowledge workers, what will be its suitable volume? To provide answers to this question, we believe that one must take into account the user's cognitive capability. This cognitive limitation, which can hinder the transfer of knowledge, is difficult to determine for two reasons :

- Because the knowledge is not well defined yet. Fundamentally, is the knowledge to be transferred declarative or procedural in nature? The former concerns concept definitions or factual information while the latter is essentially actions to perform some tasks (as an algorithm). In the both cases, calling them "knowledge" might confuse the receiver.
- Because of the lack of a clear separation between information and knowledge. Consequently, instead of transferring the right level of knowledge, we may collect some information of all types. In the worse case, the receiver of this "knowledge", cannot consume it, has the impression that he/she always lacks the desired knowledge and wants to know more. In other cases, one put them in the "Knowledge Pool" to deal with later.

For the transfer of knowledge, some research results already available from Intelligent Tutoring Systems (ITS) are applicable. In such systems, a knowledge object (or knowledge unit) corresponds roughly to a teaching subject. Several related subjects form a lesson (what we call a Knowledge Network - KN) that one can teach in a lecture (during about one hour). Moreover, it has been determined that the teaching of procedural knowledge will be more efficient if accompanied by demonstrations (Lê, Gauthier, and Frasson 1998) which can be realized by animation files (e.g. by Macromedia Flash) or video films.

Even in the context of ontological negotiation as proposed in (van Elst and Abecker, 2002), one must first frame out the underlying knowledge. To do this, a cognitive analysis is necessary. From the light of ITS, we can design a KN as a task hierarchy consisting of just a small number of levels (from one to three levels is ideal). That is, the KN is named as a global task, and its content can be detailed with some Knowledge Units as primitive tasks which are arranged as a sequence or as a hierarchy. While the ontology construction is based on the entities of the external world, the procedural knowledge relies on our epistemological states of mind. This distinction leads to these two important consequences. First, the ontology construction can apply the principles of *object oriented design* with class definition (a template to classify instances) in which the emphasis is put on the view of

entities (expressed by nouns) and the relation between super and sub-classes as generalization and specialization. However, the *actions* are designed as relations between two different class entities; for example, a "Car" being *driven* by a "Person" (Noy and McGuinness 2001). Second, for procedure knowledge, the importance relies on the description of actions or the achievement of tasks. In order to make understandable this knowledge, *pedagogical principles* can be applied, e.g. one can specify the prerequisite (condition) for each knowledge progression step. Thus, as a result, one produces also a hierarchy of *domain knowledge concepts* which are a mix of entity concepts and procedural concepts. The latter can be expressed by verbs and by nouns derived from verbs (not like the entity nouns); for example, "*navigation* in the web site".

4 Procedural Knowledge Management Tool

The design of a KM system refers essentially to the modeling of knowledge so that it is understandable to the users, or at least, to provide them with the conditions for an adequate awareness. As for many KM systems, we make use of frames to model knowledge. To clarify the terminology, we use the term "Knowledge Unit" (KU) to designate some knowledge about an *entity* or about a *process*. Thus, we distinguish two kinds of units: "Static" KU and "How-to" KU. A static KU contains concept definitions, labels, facts and information related to the underlying domain. The names of these KU form (or are derived from) the domain ontology. How-to units are task-oriented. They contain the procedures to follow through (actions or tasks) and refers to static KU when necessary. The description of a How-to KU, in natural language, focuses on a limited scope, and its attributes specify the underlying context, a situation or some conditions. This context allows users to become aware of the conditions to understand it. A How-to KU can also give some references (links) to other available documents. Knowledge units always relate to other KUs, linked together as Knowledge Networks (KN). Such networks provide a complete view of the knowledge related to a specific topic. For additional information on the structure of the knowledge units and networks, we refer the reader to [Le and Nguyen 2001].

In the current version, in addition to the subtask links, the KN contains three other kinds of links: The *workflow links* reflect the order between KUs in an activity, a project or an organization. The *pedagogical links* connect prerequisite KU for understanding the actual KU. The required knowledge is called prerequisite knowledge which is usually some Static KUs that provide users with explanations on the domain ontology (e.g. concept definitions). The *logical links* provide information on the logical relationships between KUs (e.g. a KU being

deduced from another one). Logical links are provided to support the description of reasoning schemes. A global KU index is maintained by the system. Existing KUs can be reused by integrating them into newly created KNs.

Our tool (KMT) can be used as an environment in which we are mostly interested on how new terms, emerging from the worker's creative knowledge, are integrated within a knowledge hierarchy. This is of interest because an ontology reflects this hierarchy of knowledge [Chandrasekaran, Josephson, and Benjamins 1998]. In other words, we would like to intervene upon the emergence of a new term. This task can be seen as the "bottom-up" approach to construct an ontology. This approach involves the capturing of the knowledge in evolution to keep trace of the domain progresses.

In our KMT environment, the content of the How-to units is considered as a corpus and each new significant term with its explicit meaning is recorded in a Static KU. When the number of KN reach a manageable size, a grouping task is necessary. We call it the sub-domain promotion (it seems like the promotion of an index level in the B-tree indexing). The knowledge engineer should name a sub-domain by a generic category term which conceptually covers all the underneath KNs. These category terms progressively form the domain knowledge hierarchy. The method proposed in (Biebow and Szulman 2000) can help in this task. The knowledge hierarchy is then updated in the domain attribute for these KNs.

5 Preliminary Experiments

During the summer of 2001, we conducted a project¹ to improve the self-learning abilities of forty workers. One of the problem we had to face was to coordinate the authoring of knowledge resources by contributors residing in three different cities. The KMT was used to set up the content of a distance learning course that teaches Web page creation using the FrontPage software (17 lessons, 65 KUs and 35 Flash files for demonstration). The distributed version of the KMT was also used for knowledge interchange between the participants in our collaborative work. At the beginning, the team leader had created a template KN including some units representing the tasks to be accomplished by each member. During their work, the members gradually refined their own KU with some subtask units and new findings. The updated knowledge networks were made available to the project leader and all members. The KN continually grew up until a final state was reached. So the sequence of changes brought to the KN was informative for following the evolution of the project's knowledge. Mutual review and comparison of the member's experience also stimulated

¹ This project was financially supported by the ministry of Human Resources Development Canada.

the knowledge creation among them. In addition, the leader could provide them with some internet resource links to redirect them about some relevant public documents. At this point, we feel that the result of the project has been sustained by the use of this tool.

8 Conclusion

Languages, tools and methods for building domain ontologies are increasingly perceived as means to achieve the semantic web. However, research efforts are still required to devise adequate formalisms for the management of procedural knowledge assets. In this paper, we provided an overview of our software tool to exploit and interchange procedural knowledge represented as networks of semi-structured units. We claim that the "bottom-up" approach, that can be carried out with this tool, is appropriate to gather new terms for the construction of a new domain ontology which will be candidates for the discussion/selection by the underlying community. An educational project, conducted with our tool, is briefly reported. For future work, we intend to include some high-level categories which would remain stable for most domains and upon which new declarative knowledge could be integrated to the knowledge networks. A concept hierarchy inspired from the university structure (with faculties, schools, departments) is being considered as a candidate model.

References

- Biebow, B. and Szulman, S. 2000 Une approche terminologique pour categoriser les concepts d'une ontologie, in *Ingénierie des connaissances : Évolutions récentes et nouveaux défis*, Eyrolles, Paris, France.
- Chandrasekaran, B., Josephson, J. R. and Benjamins V. R. 1998 The Ontology of Tasks and Methods, in *Proceedings of KAW'98*, Gaines, B. and Musen, M. (eds.), Banff, Canada.
- Noy, N. and McGuinness, D. L. 2001 *Ontology Development 101: A Guide to Creating Your First Ontology*. 2001, Technical Report: SMI-2001-0880, Stanford Medical Informatics.
- Lê, T.H. and Nguyen, T.H. 2001 Toward an Universal Knowledge Management: Using Ontology Approach with methods and tools in *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI-2001)*, 79-83, Orlando, USA.
- Lê, T.H., Gauthier, G. and Frasson, C. 1998 The Process of Planning for an Intelligent Tutoring System, in *Proceeding of The Fourth World Congress On Expert Systems*, 709-714, Mexico City, Mexico.
- van Elst, L. and Abecker, A. 2002. Domain Ontology Agents in Distributed Organizational Memories, in *Knowledge Management and Organizational Memories*, Kluwer Academic Publishers, USA.