Zeitschrift: Comtec : Informations- und Telekommunikationstechnologie = information and telecommunication technology

Herausgeber: Swisscom

Band: 82 (2004)

Heft: [1]: A collection of publications of Swisscom innovations

Artikel: Boosting the information age

Autor: Schubiger, Simon

DOI: https://doi.org/10.5169/seals-876889

Boosting the Information Age

SIMON SCHUBIGER **Starting with digital data, computer systems shaped the information age and are now heading towards even richer representations. Knowledge representations (KR) are structured models of accepted facts built to make a number of applications more capable of handling complex and disparate information.**

They appear most effective when the semantic distinctions that humans take for granted are crucial to the application's purpose. KR can be the first step in building semantically aware information systems to support diverse enterprise and customer activities.

Introduction

People can find patterns in data to perceive information and information can be used to enhance knowledge. While computer systems today easily handle terabytes of data and provide interfaces to search and navigate gigantic information spaces (for example WWW), knowledge is still mostly created in the human brain. Despite recent advances in Artificial Intelligence, technologies can support the activities of knowledge acquisition, but are unlikely to replace the human mind in the near future. Whereas the knowledge *construction* will remain for some time the realm of humans, knowledge *representation* is increasingly managed by computer systems. The two motivations for transferring human knowledge into computer systems are:

- Making knowledge machine processable and in turn enabling new applications (for example Semantic Web).
- Encoded knowledge can easily be replicated and consumed by many others (knowledge reuse).

Knowledge generally passes through the three steps of knowledge acquisition, representation and reuse. Domain experts structure and enter knowledge through appropriate tools in a knowledge base. Multiple applications are then built on top of this knowledge base, which interpret and present various aspects of the encoded knowledge. Through these applications knowledge is consumed and reused. For decades universities and companies have been working on knowledge representations, but only in the last years could some convergence be observed.

Two Prototype Applications for KR

As an example of KR, this article presents two prototype applications from the domain of mobile devices and ser-

Fig. 1. Digitally represented knowledge simplifies knowledge sharing and reuse as well as the creation of novel applications.

vices. The goal was to get hands-on experience with KR technologies by using them to model mobile handsets and their functionalities. An appropriate model of a mobile handset resulted in many interesting application ideas from which two were realised:

A customer care application supporting the call agent while resolving handset related problems. The application basically gives the shortest sequence of keystrokes from a given state to another state of the mobile handset. This allows a quick answer to questions such as, "I have received an SMS and would now like to store the sender's number in my address book".

A mobile phone chooser application which supports a customer searching for a new phone. By answering a number of questions, a profile of the customer is compiled and then compared to existing phones. The best match between the customer profile and the existing phones is then offered to the customer.

Both applications are based on the same knowledge model that is just interpreted differently. This already shows how knowledge can be reused by quite different applications with the obvious advantages of a shared model (for example extension, maintenance, consistency).

The underlying model is quite simple. A mobile phone is represented as a set of states with transitions in between. Additionally, common services, such as call, text messages, address book etc. are modelled. Transitions between states usually occur when the user types on the keypad and hence changes the status of the phone accordingly. Also, external events, such as an incoming call or SMS can change the state of the phone. Each state is enriched by attributes like a screenshot of the state, a list of synonyms to refer to the state (for example "SMS Editor", "Where I write SMS", "Where I can type", "The screen for Text") and other information. Transitions are described by their keystrokes. Weights are introduced at various places in order to obtain a metric for the usability of the services provided by the phone. For example, a phone with small keys receives a lower global weight for the keypad than one with large keys (expressing that the usability of large keys is better than that of small keys). The metric can be easily extended by pricing information, phone reliability, and other marketing information.

While the customer care application only navigates the model to figure out the sequence of keystrokes between two states, the "mobile chooser" application extensively uses the weights to compare a phone and its features with the customer profile. An interesting extension of the "mobile chooser" would be to find the best phone for a customer looking for a new phone that is as close as possible to his previous one. This would require calculating the similarity between phones (maybe by taking into account the customer profile) and to present the best match based on that. The result would be a phone that causes minimal migration problems for the customer.

The following sections will look more closely at two knowledge representations with a high probability of becoming important in the near future. The first is TopicMaps, a kind of super-index. The second is the Web Ontology Language, OWL, the foundation of the Semantic Web effort allowing machines to interpret knowledge stored throughout the World Wide Web.

TopicMaps

Topic Maps are a recent ISO/IEC 13250 standard aiming at capturing semantics by providing a common terminology and easy linking to resources such as documents. TopicMaps are basically directed graphs consisting of topics, associations and occurrences.

They provide a framework for defining topics of interest separate from the material being linked to the topics. A Topic Map allows the definition of:

– *Topics:* Topics can be assigned to topic "types", which group related types of topics together. For example, in the context of a company, "A" and "B" may be topics, both of type "Business Unit".

– *Associations between topics:* Topics can be linked by topic associations. For example, the "B" topic may be linked to the "company" topic by the "is a business unit of" association. Association types (for example "is a business unit

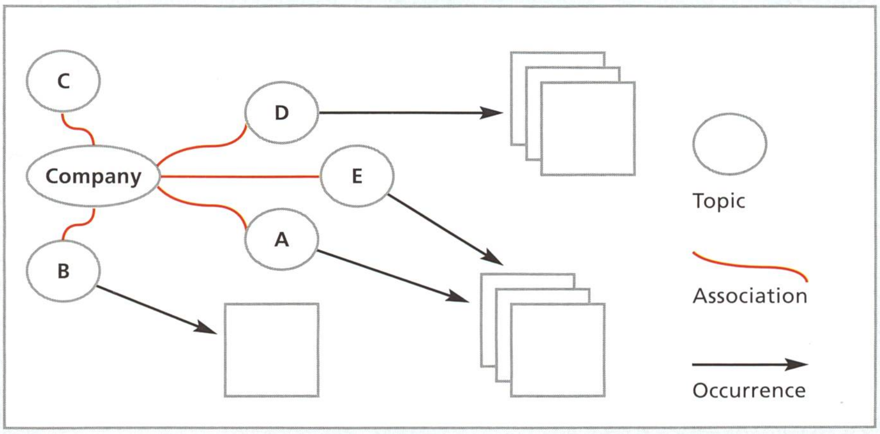


Fig. 2. The three key elements of TopicMaps.

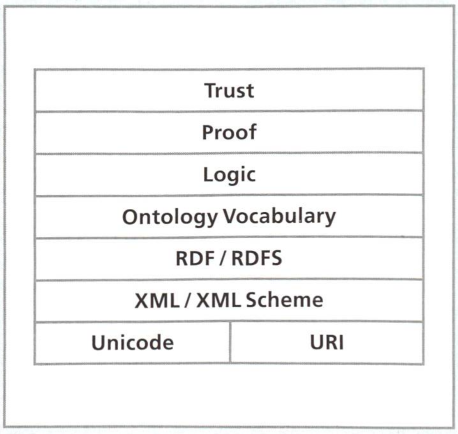
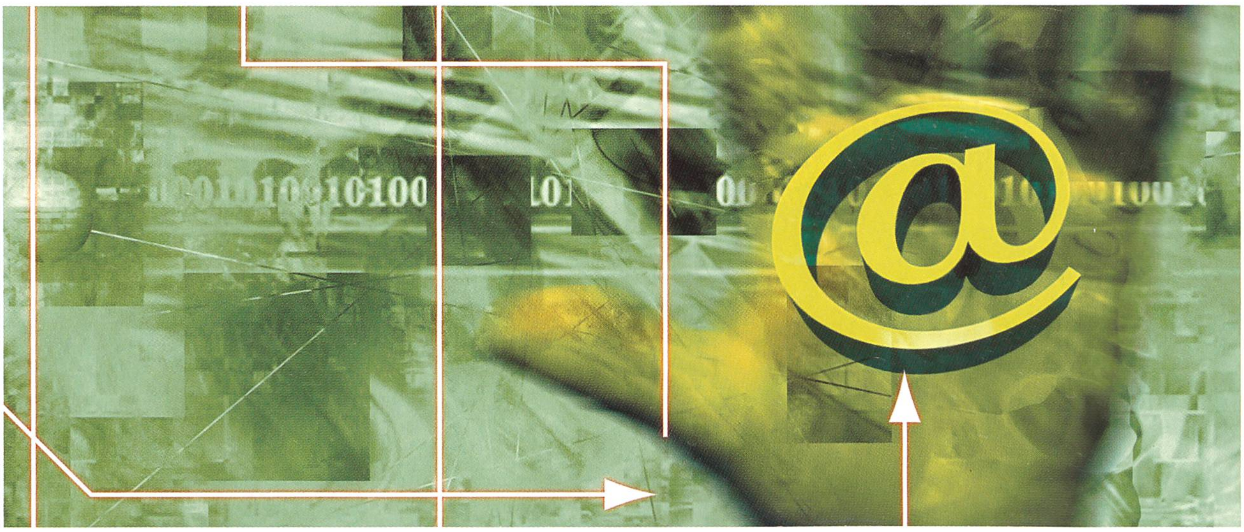


Fig. 3. The semantic Web effort builds upon the technology stack shown.



of") are themselves topics.

– *Occurrences:* Topics can be linked to parts of the underlying information resource being described. For example, the "A" topic could be linked to a document describing a project of Business Unit A. An occurrence role can also be provided to describe the type of information resource being linked ("marketing", "architecture", etc). Occurrences roles are also topics in their own right.

– *Facets:* The underlying information resources can be described by arbitrary name/value pairs (which themselves can both be "topics"). Facets allow information resources to be filtered based on their properties, much as is possible with standard metadata properties.

Topic associations allow powerful automated processing where the right semantics are defined and understood. For example, an application that would understand that the "is part of" association type was transitive would know that: if Topic X (for example "Ostermundigen") "is part of" Topic Y (for example "Kanton Bern") and Topic Y "is part of" Topic Z (for example "Schweiz"), then Topic X "is part of" Topic Z.

Topics can be involved in multiple associations. For example, "Ostermundigen" can be associated with the "urban area" topic and/or the "suburb" topic. "Bern" can be associated with the "city" topic, as could the "Zürich" and "Genf" topics.

Figure 2 depicts the three key elements of TopicMaps. TopicMaps are mostly useful where a large collection of resources (for example documents) has to be classified and where the knowledge of the classification itself is crucial for finding and managing documents.

OWL

The Web Ontology Language (OWL) is being designed by the W3C Web Ontology Working Group in order to provide a language that can be used for applications needing to understand the semantics of information instead of just interpreting the human-optimised presentation shown for example in a Web browser. The OWL language can be used to explicitly represent term vocabularies and the relationship between terms in these vocabularies. The expressive power is significantly greater than XML, RDF, or RDF-S, enabling greater machine-readable content. OWL is not restricted to Web applications. For example, the previously presented mobile handset model is OWL based.

The semantic Web effort builds upon the technology stack shown in figure 3. The lower layers are realised by proven technologies such as XML and RDF and are used to represent or reference data items. OWL provides the vocabulary and logic levels. In combination with a reasoner, a program that interprets a model and proves or falsifies a hypothesis, the proof level is realised. Finally a trust level can be built on top of that.

The interpretation of an OWL ontology involves a collection of objects or instances, known as the domain. These instances can be organised into classes (for example "Locations"). In particular, these classes can be described in terms of the properties of the individuals that make up the class (for example "Latitude", "Longitude"). Most uses of an ontology depend ultimately upon the ability to reason about individuals from the domain. In order to do this in a useful manner a mechanism to describe the classes that individuals belong to and the properties that they inherit by virtue of class membership is present (for example "Country" is a specialisation of "Geographic Area"). It is easy to assert specific properties about individuals, but much of the power of ontologies comes from class-based reasoning (for example "Country" can also be related to "Political entity"). Taxonomic relationship can come about either through the direct assertion of the subclass relationship, or through some inference process based on the intentional properties of classes.

Descriptions or definitions of particular classes can be given in terms of other classes and properties in the ontology (using the OWL operators). The semantics of OWL then provide a formal description as to when individuals are instances of classes – a reasoner can be used to infer additional information or relationships between classes, in particular inferring taxonomic relations, equivalences or inconsistencies.

The power of OWL comes into play when simple navigation can no longer yield the required results. In combination with a reasoner new knowledge can be inferred adding great value to the encoded knowledge.

Conclusion

Even though both TopicMaps and OWL aim at representing knowledge, they have their individual fields of application. TopicMaps are best applied in classic knowledge management scenarios where existing and new knowledge has to be structured and enhanced by a powerful index. While TopicMaps are clearly inferior in their expressive power compared to OWL, they have the great benefit of relying on a few simple concepts. One does not have to be an expert in logic programming to implement a TopicMap. But if more sophisticated reasoning about the knowledge itself is required, OWL is a good choice. For example, OWL allows automatic consistency checking of a model, which is a great advantage for every non-trivial model.

Since both technologies are very young (the OWL standard is not finalised yet), there is very little tool support available yet. Our own experience is that appropriate tool support is crucial for the success of any project involving knowledge representation or management. If knowledge transfer (into a knowledge base and out of it) is awkward, the transfers will simply not happen and the project will clearly fail. It is questionable whether standard tools can solve this problem entirely. We obtained the best results with customised solutions tailored to specific problem domains.

This is also the case for the prototype presented in the example above. Custom applications were built for the knowledge presentation, namely the "mobile chooser" and the customer care application. The handset knowledge was entered manually through a standard knowledge base editor. Entering handset information through a standard editor was tedious. Again, a customised editor could save great amounts of time here. KR is a solution for representation and access but no miracles can be expected on the presentation side. In that sense, KR is similar to other "back-end" technologies like databases.

Nevertheless it can be said that KR technologies are today ready for application. In today's fast evolving business environments knowledge is simply too valuable an asset to leave it buried in a few experts' heads. Using standards such as TopicMaps or OWL ensures that the value of knowledge is preserved and simplifies reuse. KR technologies in combination with tools may greatly help to move knowledge into computer systems which will result in increased productivity, shorter response times, and higher customer satisfaction among other advantages. ■

Simon Schubiger, Dr. sc. inf., Senior Engineer,
Swisscom Innovations, Berne, simon.schubiger@swisscom.com

Acknowledgement

The research work and report on Knowledge Representation was commissioned by Swisscom Mobile. It was part of the Swisscom Innovations' Programme «Software and Security Technologies».

Pointers

TopicMaps: www.topicmaps.org
OWL: www.w3.org/TR/owl-features