Ontology-Driven Information Systems: Challenges and Requirements

Burcu Yildiz[1] and Silvia Miksch[1,2]

[1] Institute for Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria
{yildiz, silvia}@ifs.tuwien.ac.at

[2] Department of Information and Knowledge Engineering,
Danube University Krems, Krems, Austria
silvia.miksch@donau-uni.ac.at

Abstract. The increased use of ontologies in several application fields makes it possible to observe requirements for their smooth integration within Information Systems. In this paper we analyse these requirements and propose the usage of additional semantic knowledge in the ontology to reconcile them. We think that these properties are essential to enhance the performance of ontology-driven Information Systems in general and ontology-driven Information Extraction Systems in particular.

Keywords: Ontology, Ontology-driven Information Systems, Ontology-driven Information Extraction

1 Introduction

Ontologies, being explicit specifications of conceptualisations (Gruber, 1993), can play a major role in many of todays Information Systems (ISs) as knowledge bearing artifacts. With regard to the impact an ontology can have on an IS, Guarino (Guarino, 1998) distinguishes between a temporal and a structural dimension. The *temporal dimension* describes whether an ontology is used at development time or at run-time, whereas the *structural dimension* describes in which way an ontology can affect the components of an IS (e.g., application programs, information resources, and user interfaces).

In our work we focus on the temporal dimension, more precisely on the use of ontologies at run time. Using an ontology at run time can yield two forms of IS: ontology-aware IS and ontology-driven IS. An *ontology-aware IS* is a system that is just aware of the ontology and can use it whenever needed. An *ontology-driven IS*, on the other hand, is a system where the ontology is yet another component of the system that co-operates with other components of the IS at run time (Guarino, 1998).

In this paper, we will examine what the requirements are that have to be reconciled in order to enhance their smooth integration within Information Systems (ISs) in general and Information Extraction Systems (IESs) in particular.

2 Ontologies in Information Systems

When we are going to build an IS we will have to provide the IS with some kind of domain and task knowledge. We cannot expect that the IS predicts what we want and just behaves like that.

If we want, for example, an application to compute graph drawings with as little edge crossings as possible, we have to tell the IS what a graph is, what kind of graphs we want to process (e.g. planar, non-planar), how an edge crossing is defined, etc. All this information will be, in general, implicitly coded in the systems' architecture. This implies, that other people, who may want to build similar applications cannot make use of this implicit knowledge, unless they examine the code of the application, which can be a very tedious task.

Ontologies can be used within ISs to make domain knowledge explicit, thus reusable. In addition to that ontologies can contribute to the portability of an IS, as they could be replaced by other ontologies that represent a totally different domain, enabling the IS to work largely domain-independent. However, developers of ontology-driven systems are also confronted with a large set of obstacles, because the ontology life cycle comprises many phases and systems have often to deal with more than one concurrent phase.

2.1 Obstacles on the Way

Despite the benefits ontologies can offer, it is not yet a common approach amongst IS developers to integrate and use ontologies in their systems. The main reason for that is perhaps that it still takes more time for a developer to build an ontology-driven application than a usual application.

To reduce the integration and run-time costs of ontologies, the ontology engineering process should to be automated to a large extent and ontology management services have to be provided in form of an Ontology Management Module (OMM). The general architecture of such an ontology-driven IS, where the required domain knowledge is captured in an ontology, can be seen in Figure 1.

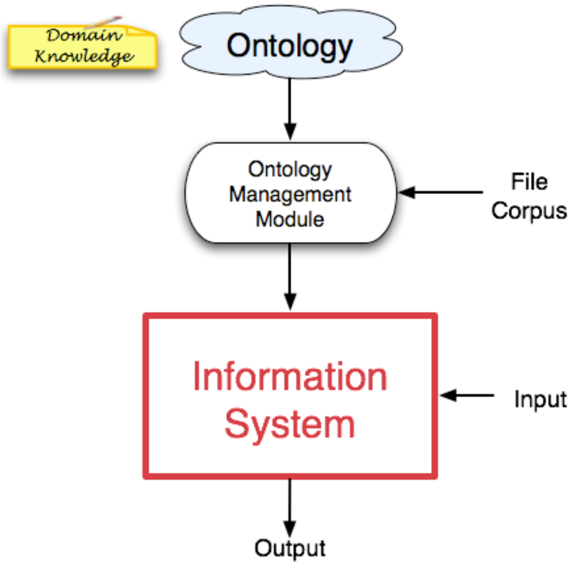

Figure 1: General architecture of an ontology-driven Information System

In the following we will take a look at the requirements to the OMM w.r.t. different phases of the ontology life-cycle. These requirements are different from the known requirements for ontology management in the context of the Semantic Web (Berners-Lee, 1999). In a scenario where an ontology is used to capture the domain knowledge needed for an IS, where the focus is on portability and scalability, the requirements that the OMM has to reconcile are different. In the following we will take a look at these requirements to an OMM w.r.t. different phases of the ontology life cycle. The main challenge is that most of these requirements have to be reconciled at the same time to provide the needed services, whereas in the Semantic Web context often only few of them are demanded.

2.1.1 Ontology Generation

Ontologies, used as part of ISs are in general not that large. So, it should not be hard to generate an ontology for a particular task specification at hand. Yet, it could be hard for someone who is not familiar with the particular ontology representation language or the domain. Further, changes in the task specification would require the adaptation of the ontology if not the generation of a new ontology from scratch. Therefore, automated approaches for ontology generation are preferred.

No matter how they are built, it is necessary to mark the components of the ontology with semantic knowledge regarding the level of confidence (property: *confidence_level*), which indicates how sure the ontology developer or an automated learning algorithm is about the existence of the component in the conceptualisation.

2.1.2 Ontology Integration

Ontologies can be generated using different representation languages, which are based on different knowledge representation paradigms (e.g. description logics, frame logics, etc.). To provide scalable and portable ISs, the OMM should be based on an abstract ontology model that can integrate, if not all, most of the ontological knowledge represented in different languages. This would make the system also more flexible to new-coming standards. Further, depending on the task an IS has to perform, the OMM might also have to provide reasoning support for its abstract ontology model.

2.1.3 Ontology Management

An ontology used in conjunction with an IS should not be considered a static artifact, because the changes in the task specification or the domain have to be reflected on the ontology as well. To automate such necessary adaptations, the OMM should provide data-driven change detection. This can be achieved by supplying the OMM with a file corpus of relevant documents to the domain. Enriching components of the ontology with additional semantic knowledge indicating their estimated behaviour over time, would further ease this process. In their proposed extended ontology model, Tamma and Bench-Capon (Tamma & Bench-Capon, 2002), propose an attribute (property: *value change frequency*) that indicates whether an ontological component is allowed to change its value over time or not.

Apart from the *value_change_frequency* property indicating the components' behaviour over time, certain additional components are needed to make an IS adaptive, that is, to make them sensitive to changes in the domain.

- *Source-link components:* represent links between the ontological structures in the ontology and their respective occurences in the file corpus. If documents are added to or removed from the file corpus, these links can be used to detect which components in the ontology are affected by the change.
- *Change components:* represent actual changes in the ontology. Every addition, deletion or edition can be represented in form of additional change instances, with appropriate properties about the kind of change, the date of change, etc. These change components also allow to keep track of the evolution of the ontology over time.

2.2 Ontology-Driven Information Extraction Systems

After examining the requirements for an Ontology Management Module (OMM) within ISs (compare Figure 1), we will now take a look at the more specific case, where the IS is an Information Extraction System (IES). The general architecture of such a system is depicted in Figure 2.

Information Extraction (IE) is defined as a form of natural language processing in which certain types of information must be recognised and extracted from text (Riloff, 1999). It is an important and popular research field of the current time; for it tries to extract relevant information from the overwhelming amount of data we are facing today. The question what actually 'relevant information' is comes to ones mind immediately. Unfortunately, the answer cannot be given so easily because it depends highly on the current task and domain. And it is even harder to communicate the answer to a computer. Ontologies can be used in that context to provide a specification of relevant information by representing parts of the domain.

An IES utilises extraction patterns (rules), with which it can decide whether a part of a document is relevant or not. There are two approaches to rule generation: the *knowledge engineering* approach and the *automatically training* approach. In the knowledge engineering approach, a knowledge engineer generates the rules for the information extraction process by hand, using his domain and task knowledge. In the automatically training approach, where the aim is to decrease human intervention, a set of documents needs to be annotated manually, whereas the annotations represent relevant parts and can be utilised by the system to learn patterns in order to extract relevant information also from unseen documents (Kushmerick & Thomas, 2002). An ontology can be used in conjunction with both approaches as an artifact, representing a shared conceptualisation of the domain to which the knowledge engineer can commit to while generating extraction rules and the annotator can commit to while annotating the file corpus.

Because our focus is on facilitating portable and scalable IES, we will concentrate on the case where the rule generation process is fully automated (compare Figure 2), that is where the rules are generated automatically using a given ontology.

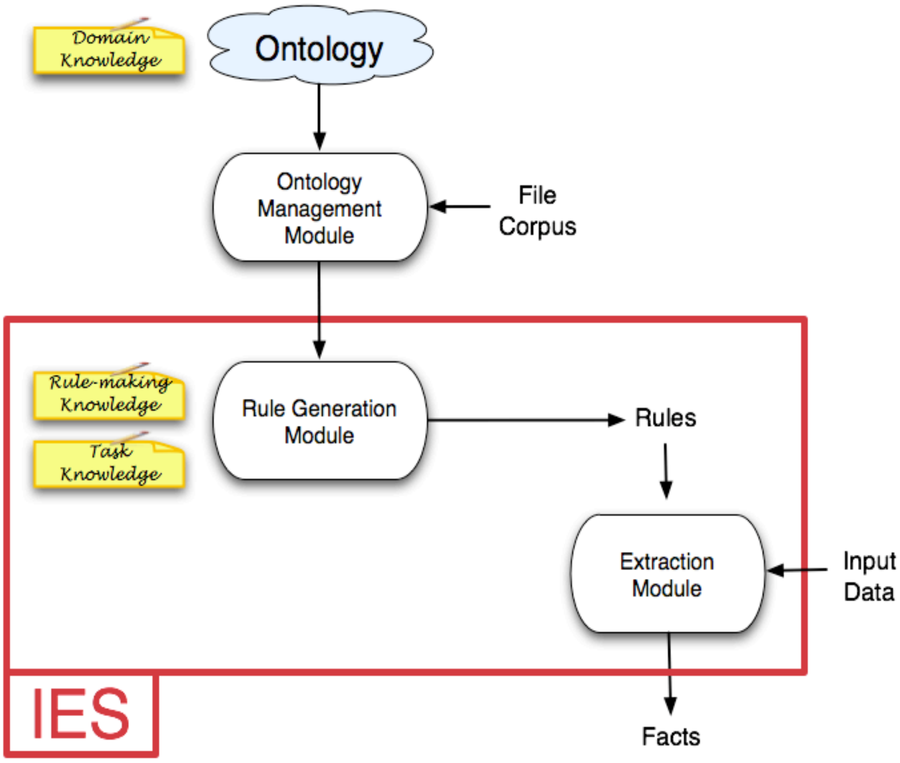

Figure 2: General architecture of an ontology-driven Information Extraction System

In the following we will take a look at the additional requirements to the ontology w.r.t. to the IE task.

2.3 Requirements w.r.t. Information Extraction Systems

During the development of an ontology-driven IES, we encountered several requirements for the smooth integration of ontologies within such systems. These requirements should be considered as additions to the ones pointed out for the use of ontologies in ISs in general (ref. Section 2.1).

The components in the input ontology should contain a few additional properties, which are essential for the Rule Generation Module (RGM) to produce accurate rules.

- *Quality Properties:* We mentioned before (ref. Section 2.1.1) that it is important for ISs to have knowledge about the confidence level of the components in the ontology (property: *confidence_level*). In the case of IESs this is absolutely mandatory, because those levels are needed in order to compute the confidence levels of the rules themselves. These computed levels are used to choose among rules, when more than one rule can be applied to a certain part of text.
- *Value Constraint Properties:* Value constraints are used to restrict property values such as the data type or cardinality. It is already possible to state this kind of knowledge in ontology representation languages such as OWL (Grigoris & Harmelen, 2004). If there are value constraints on the components of a conceptualisation, they should be implemented in the ontology, because the more constraints known the more fine-granulated rules can be generated, which in turn enhances the performance of an IES.
- *Temporal Properties:* In many settings the components of an ontology have to be marked with temporal values such as the transaction time (property: *transaction_time*), or valid time (property: *valid_time_begin* and *valid_time_end*) of the component. These properties are especially useful in connection with changing ontologies where out-of-date components are not deleted from the ontology but marked as such. In a common

scenario where the IES wants to extract information from new and relatively old data alike, a completely up-to-date ontology would not serve the purpose.

2.4 How to Implement Additional Semantic Knowledge

To include additional semantic knowledge into an ontology there are two ways: first, to extend an existing ontology representation language with the needed modeling primitives; second, to use already defined modeling primitives to add the semantic knowledge in form of additional properties to components.

One always has to think thoroughly before deciding to take the first way, because any addition to a language increases beside its expressive power also its complexity. The increase in complexity cannot be evaluated a priori, so it is not clear whether the benefits would justify the additional costs caused. Furthermore, often such additional knowledge is needed only by a particular group and not by the whole community, hindering the wide acceptance of the extension.

Tamma and Bench-Capon (Tamma & Bench-Capon, 2002) agree to some extent with the objection that this kind of additional knowledge does not represent ontological knowledge and therefore its presence in an ontology is questionable. But they also state that ontologies must contain additional information in particular settings, especially when complex and accurate services are demanded (e.g., multi-agent systems).

We argue in a similar way, and think that in the context of IESs it is much easier to build an ontology and to implement some additional knowledge in it, than to build and refine extracting rules by hand in order to increase the performance of the system.

3 Related Work

One can observe that in many cases additional knowledge about components in the ontology is needed to perform the task at hand more accurately. Often researchers use an abstract ontology model to integrate knowledge in existing ontologies and to enrich them with their proposed additional knowledge.

For the case of ontology learning from text documents, Cimiano and Völker (Cimiano & Völker, 2005) argued in a similar way and proposed their Probabilitstic Ontology Model (POM). In this model they save the results of their learning system by attaching a probability (confidence level) to them. Doing this, they aim to enhance the interaction with the user by presenting her the learned structures ranked according to their confidence level or by presenting her only results above a certain confidence threshold. Furthermore, their POM also contains links of the structures to corresponding documents from which they were derived; allowing the user to understand the context of a particular structure and allowing the system do react to changes in the document corpus. We think that both of these additions to the components of an ontology are essential for the use of ontologies in ISs.

Tamma and Bench-Capon (Tamma & Bench-Capon, 2002) motivated an extended ontology model to characterise precisely the concepts' properties and expected ambiguities, including which properties are prototypical of a concept and which are exceptional, the expected behaviour of properties over time and the degree of applicability of properties to subconcepts. The authors claim that this enriched semantics can prove useful to describe what agents know in a multi-agent system. Because or focus is on IESs, not all of the proposed meta-properties are of interest for us. We use only the property describing the components' expected behaviour over time, for it can help during the ontology management phase when the ontology has to be adapted to changes in the domain.

4 Conclusion and Future Work

Originally, ontologies have been proposed to be used as the backbone of the Semantic Web. Consequently, most of the research that has been done in the field of ontology engineering had the application field of the Semantic Web in mind. But often, Information Systems (ISs) do not share the characteristics of the Semantic Web. So we can say that other requirements are demanded from ontologies when used in different settings. These requirements have to be analysed before a decision about using an ontology in an IS can be made.

Hence, the main contribution of our work is an in-depth analysis of the changed requirements regarding Ontology Engineering (OE) when used in ISs in general and Information Extraction Systems (IES) in particular.

We have seen what kind of obstacles are on the way to integrate ontologies in ISs and IESs. Therefore, every developer has to analyse whether the benefits justify the additional costs, which can arise because of the ontology usage.

We pointed out requirements that have to be reconciled in order to foster the wide acceptance of ontology-driven IS development. Those requirements can be summarised shortly as follows:

- Abstract ontology model (with clear formal semantics) for representing additional semantic knowledge and for ontology integration. In some cases sound and complete reasoners for such a model may be required.
- Evolutional properties to indicate the expected behaviour of particular components over time.
- Quality properties to indicate the level of confidence in particular ontology components
- Temporal properties to mark the transaction times and valid times of particular components.

The main area for future work is probably the development of a comprehensive framework for dealing with ontologies, so that programmers can start right away with the development of ontology-driven ISs. Such a framework should cover functionalities to handle the main issues during the ontology life cycle like ontology generation and management (evolution and versioning).

References

Berners-Lee, T. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor.* Harper San Francisco, 1999.

Cimiano, P. & Völker, J. (2005). *Text2Onto – a Framework for Ontology Learning and Data-driven Change Discovery.* In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'2005), 227-238.

Grigoris, A. & van Harmelen, F. (2004). *A Semantic Web Primer.* Cambridge: The MIT Press.

Gruber, T.R. (1993*). A Translation Approach to Portable Ontology Specifications.* Knowledge Acquisition, 5(2), 199-220.

Guarino, N. (1998). *Formal Ontology and Information Systems.* In Proceedings of the First International Conference on Formal Ontologies in Information Systems (FOIS), 3-15.

Kushmerick, N. & Thomas, B. (2002). *Adaptive Information Extraction: Core Technologies for Information Agents.* In Intelligent Information Agents: The AgentLink Perspective. Berlin/Heidelberg: Springer, 79-103.

Riloff, E. (2002). *Information Extraction as a Stepping Stone Toward Story Understanding.* Understanding Language Understanding: Computational models of Reading, 435-460.

Tamma, V. & Bench-Capon, T. (2002). *An Ontology Model to Facilitate Knowledge-Sharing in Multi-agent Systems.* The Knowledge Engineering Review, 17(1), 41-60.