

OIL: Ontology Infrastructure to Enable the Semantic Web

Dieter Fensel ¹, Ian Horrocks ², Frank van Harmelen ¹, Deborah McGuinness ³, and
Peter F. Patel-Schneider ⁴

¹ Division of Mathematics & Computer Science, Vrije Universiteit Amsterdam,
1081 HV Amsterdam, NL, dieter@cs.vu.nl, frankh@cs.vu.nl, <http://www.cs.vu.nl/~dieter>,
<http://www.cs.vu.nl/~frankh>

² Department of Computer Science, University of Manchester, UK
horrocks@cs.man.ac.uk, <http://www.cs.man.ac.uk/~horrocks/>

³ Knowledge Systems Laboratory, Stanford University, US
d1m@ksl.stanford.edu, <http://www.ksl.stanford.edu/people/d1m/>

⁴ Bell Laboratories, Murray Hill, US
pfps@research.bell-labs.com, <http://www.bell-labs.com/user/pfps/>

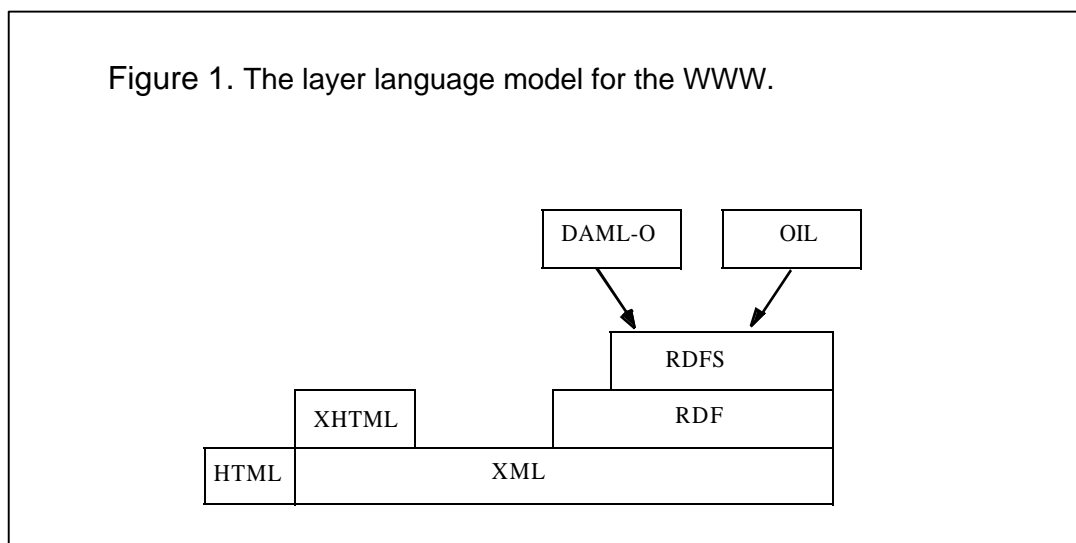
Abstract

Currently computers are changing from single isolated devices to entry points into a worldwide network of information exchange and business transactions. Therefore, support in the exchange of data, information, and knowledge is becoming the key issue in computer technology today. Ontologies provide a shared and common understanding of a domain that can be communicated between people and across application systems. Ontologies will play a major role in supporting information exchange processes in various areas. A prerequisite for such a role is the development of a joint standard for specifying and exchanging Ontologies well-integrated with existing web standards. This paper deals with precisely this necessity. We will present OIL which is a proposal for such a standard enabling the semantic web, i.e. information with machine processable semantics. It is based on existing proposals such as OKBC, XOL and RDFS, and enriches them with necessary features for expressing rich ontologies. The paper presents the motivation, underlying rationale, modeling primitives, syntax, semantics, tool environment, and applications of OIL.

1. Introduction

Ontologies will play a major role in supporting information exchange processes in various areas (cf. [Fensel, 2001]). Ontologies were developed in Artificial Intelligence to facilitate *knowledge sharing and reuse*. Since the beginning of the nineties ontologies have become a popular research topic investigated by several Artificial Intelligence research communities, including Knowledge Engineering, natural-language processing and knowledge representation. More recently, the notion of ontology is also becoming widespread in fields such as intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management. The reason ontologies are becoming so popular is in large part due to what they promise: a shared and common understanding of some domain that can be communicated between people and application systems. Because ontologies aim at consensual domain knowledge their development is often a cooperative process involving different people, possibly at different locations. People who agree to accept an ontology are said to *commit* themselves to that ontology.

Currently, we see ontologies applied to the World Wide Web creating what is called the *semantic web* [Berners-Lee, 1999]. Originally, the web grew mainly around the language HTML, that provide a standard for structuring documents that was translated by browsers in a canonical way to render documents. On the one hand, it was the simplicity of HTML that enabled the fast growth of the WWW. On the other hand, its simplicity seriously hampered more advanced web application in many domains and for many tasks. This was the reason to define XML (see Figure 1) which allows to define arbitrary domain and task specific extensions (even HTML got redefined as an XML application, see XHTML).



Still, XML is basically a defined way to provide a serialized syntax for tree structures. Therefore, it is just an important first step in the direction of a semantic web, where application programs have direct access to the semantics of data. An important additional step has been taken by RDF which define a syntactical convention and a simple data model for representing machine-processable semantics of data. The *Resource Description Framework (RDF)* [Lassila & Swick, 1999] is a standard for Web meta data developed by the World Wide Web Consortium (W3C).¹

Basically, RDF defines a data model based on triples: Object, Property, and Value. A step into a richer representation formalism was taken by RDF Schema (RDFS) [Brickley & Guha, 2000] which introduces basic ontological modeling primitives into the web. With RDFS, we can talk about classes, subclasses, sub-properties, domain and range restrictions of properties, etc. in a web-based context. OIL took RDFS as a starting point and enriched it to a full-fledged ontology language (see [Broekstra et al., 2000] for more details). That's includes the following aspects:

- A more intuitive choice of some of the modeling primitives and an extension to richer ways to define concepts and attributes (spoken in a nutshell, not each type must receive an explicit name).
- The definition of a formal semantics for the language
- The development of customized editors and inference engines to work with the language.

These aspects of OIL together with some applications are further discussed during the paper.

The contents of the paper is organized as follows. In Section 2, we illustrate the general role ontologies may have in improving information access for knowledge management and electronic commerce. In Section 3, we explain why the language OIL may be the right choice in enabling ontologies for web applications helping to realize what is called the semantic web. An important asset for this is the layered architecture of OIL that is explained in Section 4. Section 5 illustrates some of the modeling primitives provided by OIL. Tools for OIL are described in Section 6 and applications in Section 7. Finally, conclusions and outlook are given in Section 8.

2. Ontologies: A Revolution for Information Access and Integration

Many definitions of ontologies have been given in the last decade, but one that, in our opinion, best characterizes the essence of an ontology is based on the related definitions by [Gruber, 1993]: *An ontology is a formal, explicit specification of a shared conceptualisation.* A 'conceptualisation' refers to

¹ <http://www.w3c.org/rdf>

an abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon. 'Explicit' means that the type of concepts used and the constraints on their use are explicitly defined. 'Formal' refers to the fact that the ontology should be machine understandable. Hereby different degrees of formality are possible. Large ontologies like WordNet² provide a thesaurus for over 100,000 terms explained in natural language. On the other end of the spectrum is CYC³, that provides formal axiomatizing theories for many aspect of common sense knowledge. 'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not restricted to some individual, but accepted by a group.

The three main application areas of ontology technology are Knowledge Management, Web Commerce, and Electronic Business.⁴ We will briefly discuss these application areas.

Knowledge Management is concerned with acquiring, maintaining, and accessing knowledge of an organization. It aims to exploit an organisation's intellectual assets for greater productivity, new value, and increased competitiveness. Due to globalisation and the impact of the Internet, many organizations are increasingly geographically dispersed and organized around virtual teams. With the large number of on-line documents, several document management systems entered the market. However these systems have severe weaknesses:

- *Searching information:* Existing keyword-based search retrieves irrelevant information which uses a certain word in a different context, or it may miss information where different words about the desired content are used.
- *Extracting information:* Human browsing and reading is currently required to extract relevant information from information sources, as automatic agents lack all common sense knowledge required to extract such information from textual representations, and they fail to integrate information spread over different sources.
- *Maintaining* weakly structured text sources is a difficult and time-consuming activity when such sources become large. Keeping such collections consistent, correct, and up-to-date requires a mechanized representation of semantics and constraints that help to detect anomalies.

² <http://www.cogsci.princeton.edu/~wn>

³ <http://www.cyc.com/>

- *Automatic document generation:* Adaptive web sites which enable a dynamic reconfiguration according to user profiles or other relevant aspects would be very useful. The generation of semi-structured information presentations from semi-structured data requires a machine-accessible representation of the semantics of these information sources.

Using Ontologies, semantic annotations will allow structural and semantic definitions of documents providing completely new possibilities: Intelligent search instead of keyword matching, query answering instead of information retrieval, document exchange between departments via ontology mappings, and definition of views on documents.

Web Commerce (B2C): Electronic Commerce is becoming an important and growing business area. This is happening for two reasons. First, electronic commerce is extending existing business models. It reduces costs and extends existing distribution channels and may even introduce new distribution possibilities. Second, it enables completely new business models or gives them a much greater importance than they had before. What has up to now been a peripheral aspect of a business field may suddenly receive its own important revenue flow. Examples of business field extensions are on-line stores, examples of new business fields are shopping agents, on-line marketplaces and auction houses that make comparison shopping or meditation of shopping processes into a business with its own significant revenue flow. The advantages of on-line stores and the success story of many of them has led to a large number of such shopping pages. The new task for a customer is now to find a shop that sells the product he is looking for, getting it in the desired quality, quantity, and time, and paying as little as possible for it. Achieving these goals via browsing requires significant time and will only cover a small share of the actual offers. Very early, shopbots were developed that visit several stores, extract product information and present to the customer a instant market overview. Their functionality is provided via *wrappers* that need to be written for each on-line store. Such a wrapper uses a keyword search for finding the product information together with assumptions on regularities in the presentation format of stores and text extraction heuristics. This technology has two severe limitations:

- *Effort:* Writing a wrapper for each on-line store is a time-consuming activity and changes in the outfit of stores cause high maintenance efforts.

⁴ For the reasons of limited space we do not discuss application areas such as C2C electronic commerce and e-science.

- *Quality*: The extracted product information is limited (mostly price information), error prone and incomplete. For example, a wrapper may extract the direct product price but misses indirect costs such as shipping costs etc.

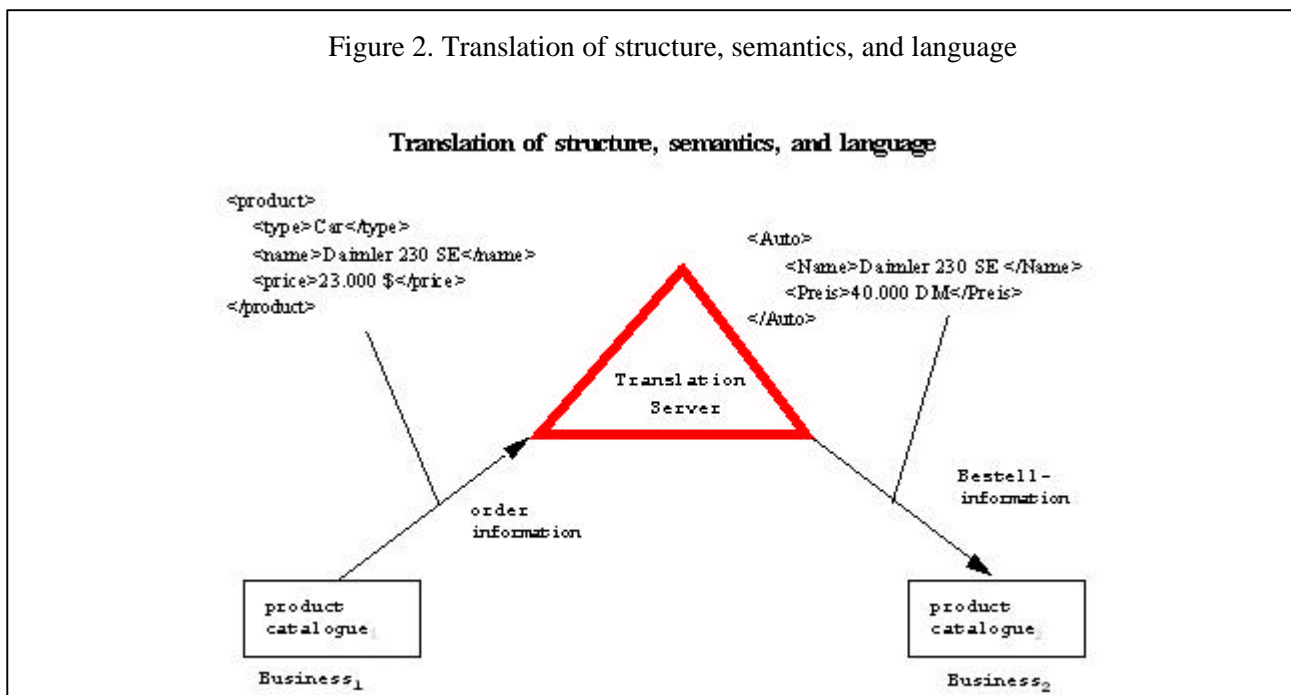
These problems are caused by the fact that most product information is provided in natural language, and automatic text recognition is still a research area with significant unsolved problems. However, the situation will drastically change in the near future when standard representation formalisms for the structure and semantics of data are available. Software agents then can *understand* the product information. Meta-on-line stores can be built with little effort and this technique will also enable complete market transparency in the various dimensions of the diverse product properties. The low-level programming of wrappers based on text extraction and format heuristics will be replaced by ontology mappings, which translate different product descriptions into each other. An ontology describes the various products and can be used to navigate and search automatically for the required information.

Electronic Business (B2B): Electronic Commerce in the business to business field (B2B) is not a new phenomena. Initiatives to support electronic data exchange in business processes between different companies existed already in the sixties. In order to exchange business transactions sender and receiver have to agree on a common standard (a protocol for transmitting the content and a language for describing the content) A number of standards arose for this purpose. One of them is the UN initiative *Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT)*. In general, the automatization of business transactions has not lived up to the expectations of its propagandists. This can be explained by some serious shortcomings of existing approach like EDIFACT: It is a rather procedural and cumbersome standard, making the programming of business transactions expensive, error prone and hard to maintain. Finally, the exchange of business data via extranets is not integrated with other document exchange processes, i.e., EDIFACT is an isolated standard. Using the infrastructure of the Internet for business exchange will significantly improve this situation. Standard browsers can be used to render business transactions and these transactions are transparently integrated into other document exchange processes in intranet and Internet environments. However, this is currently hampered by the fact that HTML do not provide a means for presenting rich syntax and semantics of data. XML, which is designed to close this gap in current Internet technology, will therefore drastically change the situation. B2B communication and data exchange can then be modeled with the same means that are available for the other data exchange processes, transaction specifications can easily be rendered by standard browsers, maintenance will be cheap. XML will provide a standard serialized syntax for defining the structure and semantics of data. Still, it does not provide standard data structures and terminologies to describe business

processes and exchanged products. Therefore, ontologies will have to play two important roles in XML-based electronic commerce:

- *Standard ontologies* have to be developed covering the various business areas. In addition to official standards, vertical marketplaces (Internet portals) may generate de facto standards. If they can attract significant shares of the on-line transactions in a business field they will factually create a standard ontology for this area. Examples are: Dublin, Common Business Library (CBL), Commerce XML (cXML), ecl@ss, Open Applications Group Integration Specification (OAGIS), Open Catalog Format (OCF), Open Financial Exchange (OFX), Real Estate Transaction Markup Language (RETM), RosettaNet and UN/SPSC.⁵
- *Ontology-based translation services* between different data structures in areas where standard ontologies do not exist or where a particular client wants to use his own terminology and needs translation service from his terminology into the standard. This translation service must cover structural and semantical as well as language differences (see figure 2).

Then, ontology-based trading will significantly extend the degree to which data exchange is automated and will create complete new business models in the participating market segments.



3. Why OIL

Effective and efficient work with ontologies must be supported by advanced tools enabling the full power of this technology. In particular, we need an advanced ontology language to express and represent ontologies. Such an ontology language must fulfill three important requirements:

- It must be highly intuitive to the human user. Given the current success of the frame-based and object-oriented modeling paradigm they should have a frame-like look and feel.
- It must have a well-defined formal semantics with established reasoning properties in terms of completeness, correctness, and efficiency.
- It must have a proper link with existing web languages like XML and RDF ensuring interoperability.

In this respect, many of the existing languages like CycL [Lenat & Guhy, 1990], KIF [Genesereth, 1991], and Ontolingua [Farquhar et al., 1997] fail. However, the Ontology Inference Layer OIL matches the criterion mentioned above. OIL⁶ (cf. [Fensel et al., 2000a]) unifies three important aspects provided by different communities: Epistemologically rich modeling primitives as provided by the Frame community, formal semantics and efficient reasoning support as provided by Description Logics, and a standard proposal for syntactical exchange notations as provided by the Web community.

- **Frame-based systems.** The central modeling primitives of predicate logic are predicates. Frame-based and object-oriented approaches take a different point of view. Their central modeling primitives are classes (i.e., frames) with certain properties called attributes. These attributes do not have a global scope but are only applicable to the classes they are defined for (they are typed) and the "same" attribute (i.e., the same attribute name) may be associated with different range and value restrictions when defined for different classes. A frame provide a certain context for modeling one aspect of a domain. Many other additional refinements of these modeling constructs have been developed and have led to the incredible success of this modeling paradigm. Many frame-based systems and languages have been developed and, renamed as object-orientation they have conquered the software engineering community. Therefore, OIL incorporates the *essential modeling primitives* of frame-

⁵ See for more info <http://www.diffuse.org/>

⁶<http://www.ontoknowledge.org/oil>.

based systems into its language. OIL is based on the notion of a concept and the definition of its superclasses and attributes. Relations can also be defined not as an attribute of a class but as an independent entity having a certain domain and range. Like classes, relations can be arranged in a hierarchy.

- **Description Logics (DL).** DLs describe knowledge in terms of concepts and role restrictions that are used to automatically derive classification taxonomies. The main effort of research in knowledge representation is in providing theories and systems for expressing structured knowledge and for accessing and reasoning with it in a principled way. In spite of the discouraging theoretical complexity of their results, there are now efficient implementations for DL languages, see for example DLP and the FaCT system, which will be explained later on. OIL inherits from Description Logic its *formal semantics* and the *efficient reasoning support* developed for these languages. In OIL, *subsumption* is decidable and with FaCT we can provide an efficient reasoner for this.
- **Web standards: XML and RDF.**⁷ Modeling primitives and their semantics are one aspect of an Ontology language. Second, we have to decide about its syntax. Given the current dominance and importance of the WWW, a syntax of an ontology exchange language must be formulated using existing web standards for information representation. First, OIL has a well-defined syntax in XML based on a DTD and a XML schema definition. Second, OIL is defined as an extension of the Resource Description Framework RDF and RDFS. In regard to ontologies, RDFS provides two important contributions: a standardized syntax for writing ontologies, and a standard set of modeling primitives like instance of and subclass of relationships. Extend this approach to a full-blown ontology language.

When agreeing on the above mentioned criteria there is not really any competitor for OIL. A simple subset of OIL is defined by XOL⁸ which may help to customize simpler versions of OIL for applications if required. We will discuss the layered approach of OIL in section 5.

4. The layered Architecture of OIL

It is unlikely that a single ontology language can fulfill all the needs of the large range of users and applications of the Semantic Web. We have therefore organised OIL as a series of ever increasing layers

⁷ For a brief introduction to RDF and XML see the tutorial in this issue.

⁸<http://smi-web.stanford.edu/projects/bio-ontology/>

of sublanguages. Each additional layer adds functionality and complexity to the previous layer. This is done such that agents (humans or machines) who can only process a lower layer can still partially understand ontologies that are expressed in any of the higher layers. A first and very important application of this principle is the relation between OIL and RDF Schema. As shown in the Figure 3, Core OIL coincides largely with RDF Schema (with the exception of the reification features of RDF Schema). This means that even simple RDF Schema agents are able to process the OIL ontologies, and pick up as much of their meaning as possible with their limited capabilities.

- **Core OIL** coincides largely with RDF Schema (with the exception of the reification features of RDF Schema). This means that even simple RDF Schema agents are able to process the OIL ontologies, and pick up as much of their meaning as possible with their limited capabilities.
- **Standard OIL** is a language intended to capture the necessary mainstream modelling primitives that both provide adequate expressive power and are well understood thereby allowing the semantics to be precisely specified and complete inference to be viable.
- **Instance OIL** includes a thorough individual integration. While the previous layer - Standard OIL - included modelling constructs that allow individual fillers to be specified in term definitions, Instance OIL includes a full-fledged database capability.
- **Heavy OIL** may include additional representational (and reasoning) capabilities. Especially a more expressive rule languages and meta-class facilities seem highly desirable. These extensions of OIL will be defined in cooperation with the DAML initiative⁹ on a rule language for the web.

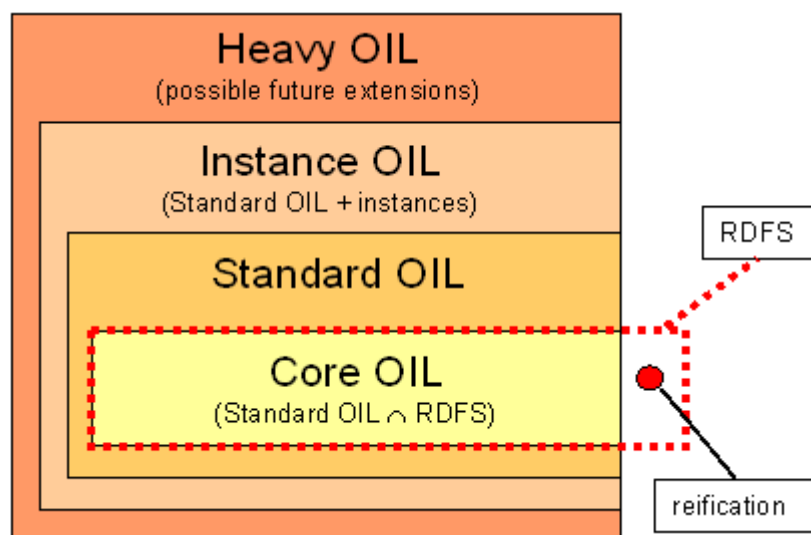


Figure 3. The layered language model of OIL

The layered architecture of OIL has three main advantages:

- First, an application is not forced to work with a language that offers significant more expressiveness and complexity that is actually needed.
- Second, application that can only process a lower level of complexity are still able to catch some of the aspects of an ontology.
- Third, an application that is aware of a higher level of complexity can still also understand ontologies expressed in a simpler ontology language.

Defining an ontology language as an extension of RDF-Schema means that every RDF-Schema ontology is a valid ontology in the new language (i.e., an OIL processor will also understand RDF Schema). However, the other direction is also available: defining an OIL extension as close as possible to RDF Schema allows maximal reuse of existing RDF Schema-based applications and tools. However, since the ontology language usually contains new aspects (and therefore new vocabulary, which an RDF Schema processor does not know), 100% compatibility is not possible. Let's give an example. The following OIL expression defines *herbivore* as a class, that is a sub-class of *animal* and disjoint to all *carnivores*.

```
<rdfs:Class rdf:ID="herbivore">
  <rdf:type
    rdf:resource="http://www.ontoknowledge.org/oil/RDFS-schema/#DefinedClass"/>
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    <oil:NOT>
      <oil:hasOperand rdf:resource="#carnivore"/>
    </oil:NOT>
  </rdfs:subClassOf>
</rdfs:Class>
```

An application limited to pure RDFS is still able to capture some aspects of this definition.

```
<rdfs:Class rdf:ID="herbivore">
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    ...
  </rdfs:subClassOf>
</rdfs:Class>
```

It encounters that *herbivore* is a subclass of *animal* and a subclass of a second class, which it cannot understand properly. This seems to be a useful way to preserve complicated semantics for simpler applications.

⁹ <http://www.daml.org>

5. An Illustration of the OIL Modeling Primitives

An OIL ontology is itself annotated with meta-data starting such things as title, creator, creation-date, etc. OIL follows the W3C Dublin Core Standard on bibliographical meta-data for this purpose.

The core of any ontology language is its hierarchy of class-declarations, stating for example that DeskJet printers are a sub-class of printers. Classes can be declared "defined", which indicates that the stated properties are not only necessary but also sufficient conditions for membership of the classes. Instead of using single types in expressions, classes can also be combined in logical expressions indicating intersection, union, and complement of classes.

Slots (relations between classes) can be declared, together with logical axioms stating whether they are functional (i.e., having at most one value), transitive, symmetric, and which (if any) slots are each other inverse. Range restrictions can be stated as part of a slot-declaration, as well as the number of distinct values that a slot is allowed to have. Slots can be further restricted by value-type or has-value restrictions. A value-type restriction demands that every value of the property must be of the stated type(s). Has-value restrictions require that the slot have at least values from the stated type(s).

A crucial aspect of OIL is its formal semantics [Horrocks et al., 2000]. An OIL ontology is given a formal semantics by mapping each class into a set of objects and each slot into a set of pairs of objects. This mapping must obey the constraints specified by the definitions of the classes and slots. We omit the details of this formal semantics, but it is crucial that it exists, and can be consulted whenever necessary to resolve disputes about the meaning of language constructions, and as an ultimate reference point for OIL applications.

Below, we give a very simple example of an OIL ontology.¹⁰ It only illustrates the most basic constructs of OIL.

```
class-def Product
slot-def Price
  domain Product
slot-def ManufacturedBy
  domain Product
class-def PrintingAndDigitalImagingProduct
  subclass-of Product
class-def HPPProduct
  subclass-of Product
```

¹⁰ It is a simplified example provided by Interprice (<http://www.interprice.com>).

```

    slot-constraint ManufacturedBy
      has-value "Hewlett Packard"
class-def Printer
  subclass-of PrintingAndDigitalImagingProduct
slot-def PrinterTechnology
  domain Printer
slot-def Printing Speed
  domain Printer
slot-def PrintingResolution
  domain Printer
class-def PrinterForPersonalUse
  subclass-of Printer
class-def HPPrinter
  subclass-of HPPProduct and Printer
class-def LaserJetPrinter
  subclass-of Printer
  slot-constraint PrintingTechnology
    has-value "Laser Jet"
class-def HPLaserJetPrinter
  subclass-of LaserJetPrinter and HPPProduct
class-def HPLaserJet1100Series
  subclass-of HPLaserJetPrinter and PrinterForPersonalUse
  slot-constraint PrintingSpeed
    has-value "8 ppm"
  slot-constraint PrintingResolution
    has-value "600 dpi"
class-def HPLaserJet1100se
  subclass-of HPLaserJet1100Series
  slot-constraint Price
    has-value "$479"
class-def HPLaserJet1100xi
  subclass-of HPLaserJet1100Series
  slot-constraint Price
    has-value "$399"

```

This defines a number of classes and organises them in a class-hierarchy (e.g. HPPProduct is a subclass of Product). Various properties ("slots") are defined, together with the classes to which they apply (e.g. a Price is a property of any Product, but a PrintingResolution can only be stated for a Printer (an indirect subclass of Product). For certain classes, these properties have restricted values (e.g. the Price of any HPLaserJet1100se is restricted to be \$479). In OIL, classes can also be combined using logical expressions, for example: an HPPrinter is both an HPPProduct and a Printer (and consequently inherits the properties from both these classes).

6. OIL Tools

Effective and efficient work with ontologies must be supported by advanced tools enabling the full power of this technology. OIL is an **Ontology languages** to express and represent ontologies. However, we need

also further tool support to make the language alive. In particular, OIL has rather strong tool support in the following areas: **Ontology Editors** to build new ontologies; **Ontology-based annotation tools** to link unstructured and semi-structured information sources with ontologies; **reasoning with Ontologies** enabling advanced query answering service, support ontology creation, and help to map between different ontologies.

6.1 *Ontology Editors*

Ontology editors help human knowledge engineers to build ontologies. Ontology editors support the definition of concept hierarchies, the definition attributes for concepts, and the definition of axioms and constraints. They must provide graphical interfaces and must conform to existing standards in web-based software development. They enable inspecting, browsing, codifying and modifying ontologies and supports in this way the ontology development and maintenance task. Currently, two editors for OIL are available and a third one is under development:

- **OntoEdit¹¹**: OntoEdit is an Ontology Engineering Environment developed at the Knowledge Management Group of University of Karlsruhe, Institute AIFB. OntoEdit is a tool which enables inspecting, browsing, codifying and modifying ontologies and supports in this way the ontology development and maintenance task (see Figure 4). Currently OntoEdit supports the following representation languages: Frame-Logic, OIL, RDFS, and XML. It is linked with the SilRi inference engine and the FaCT reasoner. It is commercialized by Ontoprise¹².
- **OILed¹³**: A freely available and customized editor for OIL is implemented by the University of Manchester and sponsored by the Vrije Universiteit Amsterdam and Interprice. The intention behind OilEd is to provide a simple, freeware editor that demonstrates the use of, and stimulates interest in, OIL. OilEd is not intended as a full ontology development environment - it will not actively support the development of large-scale ontologies, the migration and integration of ontologies, versioning, argumentation and many other activities that are involved in ontology construction. Rather, it is the "NotePad" of ontology editors, offering just enough functionality to allow users to build ontologies and to demonstrate how we can use the FaCT reasoner to check those ontologies for consistency.

¹¹ <http://ontoserver.aifb.uni-karlsruhe.de/ontoedit/>

¹² <http://www.ontoprise.de>

¹³ <http://img.cs.man.ac.uk/oil/>

- **Protégé¹⁴**: Protégé (cf. [Grosso et al., 1999]) allows domain experts to build knowledge-based systems by creating and modifying reusable ontologies and problem-solving methods. Protégé generates domain-specific knowledge-acquisition tools and applications from ontologies. Protégé has been used in more than 30 countries. It is an ontology editor which you can use to define classes and class hierarchy, slots and slot-value restrictions, relationships between classes and properties of these relationships. The instances tab is a knowledge-acquisition tool which you can use to acquire instances of the classes defined in the ontology. Protégé is built at the University of Stanford. Currently it only supports RDF, work on extending Protégé to OIL is currently starting.

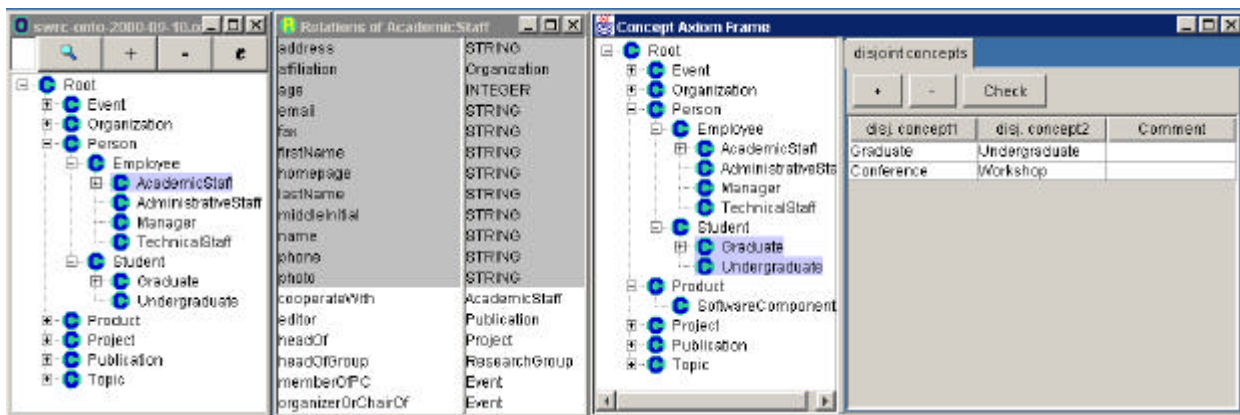


Figure 4. A screen shot of OntoEdit

6.2 Ontology-based annotation tools

Ontologies can be used to describe large instance population. In the case of OIL there are currently two tools that provide help for such a process. First, an XML DTD and an XML schema definition can be derived from an ontology in OIL. Second, an RDF and RDFS definition for instances can be derived from OIL. Both provide means to express large volumens of semi-structured information as instance information in OIL. More details can be found in [Broekstra et al., 2000], [Klein et al., 2000], and [Erdmann & Studer, to appear].

¹⁴ <http://www.smi.stanford.edu/projects/protege/>

6.3 Reasoning with Ontologies: Instance and Schema Inferences

Inference engines for ontologies can be used to reason about instances and schema definition of an ontology or over ontology schemas, for example, automatically derive the right position of a new concept in a given concept hierarchy. Such reasoners help to build ontologies and to use them for advanced information access and navigation. OIL makes use of the **FaCT (Fast Classification of Terminologies)**¹⁵ system in order to provide reasoning support for ontology design, integration and verification. FaCT is a Description Logic classifier that can also be used for consistency checking in modal and other similar logics. FaCT's most interesting features are its expressive logic (in particular the *SHIQ* reasoner), its optimized tableaux implementation (which has now become the standard for DL systems), and its CORBA based client-server architecture. FaCT's optimizations are specifically aimed at improving the system's performance when classifying realistic ontologies, and this results in performance improvements of several orders of magnitude when compared with older DL systems. This performance improvement is often so great that it is impossible to measure precisely as unoptimised systems are virtually non-terminating with ontologies that FaCT is easily able to deal with [Horrocks & Patel-Schneider, 1999]. Taking a large medical terminology ontology developed in the GALEN project [Rector et al., 1993] as an example, FaCT is able to check the consistency of all 2,740 classes and determine the complete class hierarchy in about 60 seconds of (450MHz Pentium III) CPU time. FaCT can be accessed via a Corba interface.

7. Applications of OIL

In the beginning we sketched three application areas for ontologies: knowledge management, B2C web commerce, and B2B electronic business. Not surprisingly, we find applications of OIL in all of these three areas. In *On-To-Knowledge*¹⁶ [Fensel et al., 2000b], OIL is extended to a full-fledged environment for knowledge management in large intranets and websites. Unstructured and semi-structured data will be automatically annotated, and agent-based user interface techniques and visualization tools help the user to navigate and query the information space. Here, On-To-Knowledge continues a line of research that was set up with SHOE [Luke et al., 1996] and Ontobroker [Fensel et al., 2000c]: using ontologies to model and annotate the semantics of information resources in a machine-processable manner. On-To-Knowledge

¹⁵ <http://www.cs.man.ac.uk/~horrocks/FaCT/>

¹⁶ On-to-knowledge is an European IST project, see <http://www.ontoknowledge.org> for more details.

is carrying out three industrial case studies to evaluate the tool environment for ontology-based knowledge management.

- **Swiss Life¹⁷: organizational memory.** Swiss Life [Reimer et al., 1998] implement an intranet-based front end to an organizational memory with OIL. The starting point is the existing intranet information system, called ZIS. ZIS has considerable drawbacks. Its great flexibility allows for its dynamic evolution according to the actual needs, but this also makes it very hard to find certain information. Search engines only help marginally. Clearly, formalized knowledge is connected with weakly structured background knowledge here. Experience shows that this is extremely bothersome and error-prone to maintain. The only way out is to apply content-based so that we no longer have a mere collection of web pages but a full-fledged information system that can rightly be called an organizational memory.
- **British Telecom: call centers.** Call Centers are an increasingly important mechanism for customer contact in many industries. What is required in the future is a new philosophy in customer interaction design. Every transaction should emphasize the uniqueness of both the customer and the customer service person. To do this one needs effective knowledge management¹⁸. This includes knowledge about the customer but also knowledge about the customer service person, so that the customer is directed to the right person to answer their query. This knowledge must also be used in a meaningful and timely way. Some of BT's own Call Centers will be targeted to identify opportunities for effective knowledge management. More specifically, call centre agents tend to use a variety of electronic sources for information when interacting with customers, including their own specialized systems, customer databases, the organization's intranet and, perhaps most importantly, case bases of best practice. OIL is used to provide an intuitive front-end tool to these heterogeneous information sources, to ensure that the performance of the best agents is transferred to the others.
- **EnerSearch: virtual enterprise.** EnerSearch is a virtual organization researching new IT-based business strategies and customer services in deregulated energy markets (e.g., [Ygge & Akkermans, 1999])¹⁹. Essentially, EnerSearch is a knowledge creation company, knowledge that must be transferred to its shareholders and other interested parties. Its website is one of the mechanisms for

¹⁷ <http://www.swisslife.ch>

¹⁸ <http://www.bt.com/innovations/>.

¹⁹ see further <http://www.enersearch.se>. Enersearch research affiliates and shareholders are spread over many countries: its shareholding companies include IBM (US), Sydkraft (Sweden), ABB (Sweden/Switzerland), PreussenElektra (Germany), Iberdrola (Spain), ECN (Netherlands), and Electricidade do Portugal

this. However, it is rather hard to find information on certain topics – the current search engine supports free text search rather than content-based search. Therefore, EnerSearch applies the OIL toolkit to enhance knowledge transfer to (1) researchers in the EnerSearch virtual organization in different disciplines and countries, and (2) specialists from shareholding companies interested in getting up-to-date information about R&D results on IT in Energy.

In this context, CognIT²⁰ extended their information extraction tool Corporum to generate OIL ontologies from semi-structured or unstructured natural language documents. Important concepts and their relationships are extracted from these documents and used to build up initial OIL ontologies. Further field studies on using ontologies and their OIL-based representations are executed at Boing and DaimlerChrysler.

In addition to knowledge management, OIL is used for advanced access of product data in B2C electronic commerce by Interprice²¹ and its usefulness in B2B electronic commerce is evaluated by ContentEurope²², number one on content management for B2B EC in Europe.

8 Conclusions and Future Work

OIL has several advantages: it is properly grounded in web languages such as XML schema and RDFS, it offers different levels of complexity and at least the inner layers enable efficient reasoning support based on FaCT. OIL has a well-defined formal semantics that is a baseline requirement for languages for the semantic web. In concern to its modeling primitives, OIL is not just another new language but reflects certain consensus in areas such as Description Logic and frame-based systems. This could only be achieved by including a large group of scientist in the development of OIL (see acknowledgement). Therefore, OIL is also a significant source of inspiration for the ontology language called DAML+OIL²³ developed in the DAML initiative. Current planning is to start soon a W3C working group on the semantic web taking DAML+OIL as a starting point.

²⁰ <http://www.cognit.no>

²¹ <http://www.interprice.com>

²² <http://www.contenteurope.com>

²³ <http://www.cs.man.ac.uk/~horrocks/DAML-OIL/>

Defining a proper language for the semantic web is an important step into its direction. Developing new tools, architectures, and applications is the real challenge that will follow.

Acknowledgement

We thank Hans Akkermans, Sean Bechhofer, Jeen Broekstra, Stefan Decker, Ying Ding, Michael Erdmann, Carole Goble, Michel Klein, Deborah McGuinness, Alexander Mädche, Enrico Motta, Borys Omelayenko, Peter F. Patel-Schneider, Steffen Staab, Guus Schreiber, Lynn Stein, Heiner Stuckenschmidt, and Rudi Studer, all of whom were involved in the development of OIL.

References

- [Berners-Lee, 1999] T. Berners-Lee: *Waving the Web*, Orion Business Books, London, 1999.
- [Brickley & Guha, 2000] D. Brickley and R.V.Guha, Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation, World Wide Web Consortium, 2000, www.w3.org/TR/rdf-schema (current 6 Dec. 2000).
- [Broekstra et al., 2000] J. Broekstra, M. Klein, D. Fensel, S. Decker, and I. Horrocks: Adding formal semantics to the Web: building on top of RDF Schema. In *Proceedings of the Workshop Semantic Web: Models, Architectures and Management*, Lisbon, September 21, 2000, following the Fourth European Conference on Research and Advanced Technology for Digital Libraries (ECDL'2000).
- [Erdmann & Studer, to appear] M. Erdmann and R. Studer: How to Structure and Access XML Documents With Ontologies, to appear in *Data and Knowledge Engineering*.
- [Farquhar et al., 1997] A. Farquhar, R. Fikes, and J. Rice, The Ontolingua Server: A Tool for Collaborative Ontology Construction, *International Journal of Human-Computer Studies*, 46:707-728, 1997.
- [Fensel, 2001] D. Fensel: *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerc*, Springer-Verlag, Berlin, 2001.
- [Fensel et al., 2000a] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein: OIL in a nutshell. In *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag, October 2000.

- [Fensel et al., 2000b] D. Fensel, F. van Harmelen, M. Klein, H. Akkermans, J. Broekstra, C. Fluit, J. van der Meer, H.-P. Schnurr, R. Studer, J. Hughes, U. Krohn, J. Davies, R. Engels, B. Bremdal, F. Ygge, U. Reimer, and I. Horrocks: On-To-Knowledge: Ontology-based Tools for Knowledge Management. In Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference, Madrid, Spain, October 2000.
- [Fensel et al., 2000c] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, R. Studer and A. Witt: Lessons Learned from Applying AI to the Web, *Journal of Cooperative Information Systems*, 9(4), 2000.
- [Genesereth, 1991] M. R. Genesereth: Knowledge Interchange Format. In *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning (KR-91)*, J. Allen et al., (eds), Morgan Kaufman Publishers, 1991, pp 238-249. See also <http://logic.stanford.edu/kif/kif.html>.
- [Grosso et al., 1999] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu, & M. A. Musen. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). In Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management, Banff, Alberta, Canada, October 16-21, 1999.
- [Gruber, 1993] T. R. Gruber: A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, 5:199—220, 1993.
- [Horrocks et al., 2000] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta: The Ontology Inference Layer OIL, technical report, Vrije Universiteit Amsterdam, NL. <http://www.ontoknowledge.org/oil>.
- [Horrocks & Patel-Schneider, 1999] I. Horrocks and P. F. Patel-Schneider: Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.
- [Klein et al., 2000] M. Klein, D. Fensel, F. van Harmelen, and I. Horrocks: The Relation between Ontologies and Schema-Languages: Translating OIL-Specifications to XML-Schema In: Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence ECAI-00, Berlin, Germany August 20-25, 2000.
- [Lassila & Swick, 1999] O. Lassila and R. Swick: *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, World Wide Web Consortium, 1999, www.w3.org/TR/REC-rdf-syntax (current 6 Dec. 2000).
- [Lenat & Guhy, 1990] D. B. Lenat and R. V. Guha: Building large knowledge-based systems. Representation and inference in the Cyc project, Addison-Wesley, Reading, Massachusetts, 1990.

- [**Luke et al., 1996**] S. Luke, L. Spector, and D. Rager: Ontology-Based Knowledge Discovery on the World-Wide Web. In *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI96)*, 1996.
- [**Reimer et al., 1998**] U. Reimer et al. (Eds.): *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management (PAKM'98)*, Basel, Switzerland, October 1998.
- [**Rector et al., 1993**] A. L. Rector, W A Nowlan, and A Glowinski: Goals for concept representation in the GALEN project. In *Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care (SCAMC'93)*, pages 414–418, Washington DC, USA, 1993.
- [**Ygge & Akkermans, 1999**] F. Ygge and J.M. Akkermans: Decentralized Markets versus Central Control - A Comparative Study, *Journal of Artificial Intelligence Research*, **11** (1999) 301-333.