unedited draft: EXML/PKDD workshop (2001), Freiburg, Germany 2001

CORPORUM: a workbench for the Semantic Web

R. H. P. Engels, B. A. Bremdal and R. Jones*
CognIT a.s
P.O. Box 610, N-1754
Halden, Norway
{rob.engels, bernt.bremdal, richard.jones}@cognit.no

July 31, 2001

Abstract

'Web semantics' has for a long time been a term without much content. The web is organizing itself, and its pages are typically added in a random and *ad hoc* fashion by everybody who feels like contributing. Typically, there has not been much concern about how to present contents in the best way, other then pure lay-out issues. This fact, combined with the fact that the representation language used at the world wide web is mainly format oriented (i.e. not depending on a complex formal logic representation mechanism), makes publishing on the WWW easy, giving it its enormous expressibility. Although widely acknowledged for its general and universal advantages, the increasing popularity of the web also shows us some major draw-backs. The developments of the information contents on the web during the last year alone, clearly marks the need for some changes. Perhaps one of the most felt problems with the web as a distributed information system is the difficulty to find and compare information which is provided on it. Many people add private, educational or organizational content to the web which is of immense diverse nature. Content on the web is growing closer to a real universal *knowledge base*, where there is only one problem relatively 'undealt' with; the problem of the interpretation of such contents.

In this paper, the authors provide a discusson on a technical solution which is aimed at helping the web to become more *semantic*. The CORPORUM tool set that is developed for this task exists of a set of programs that can fulfill a variety of tasks, either as 'stand-alone', or augmenting each other. As the aim of the semantic web is to enhance the *precision* and *recall* of search, but also enable the use of *logical reasoning*on web contents in order to answer queries. Important tasks that are dealt with by CORPORUM are related to information retrieval (find relevant documents, or support the user finding them), but also information extraction (can we built a knowledge base from web documents to answer queries?), information dissemination (summarizing strategies and information visualisation), and automated document classification strategies performed by so-called intelligent agents which are present on the world wide web on a pertinent basis. The current article discusses the CORPORUM tool set and shows how it can support generation and utilisation of semantics on the web.

*CognIT a.s is a full partner in the EU-funded project "OnToKnowledge" IST-1999-10132. The authors wish to thank aidministrator Nederland BV for the usage of their visualisation tools used to visualise the structure of figure 4 within this article. Please refer to the workshop website for the final version.

1 Two scenarios to put more semantics on the web

Generally speaking, there are two fundamentally different scenarios in which the world wide web could evolve further. Either the currently existing mass of documents available on the web can be analysed in its current 'as is' form and contents can be extracted from it, or the representation format of the world wide web is changed up front so that documents are available in a format that expresses such 'semantics' more explicit.

Each of these approaches have their own drawbacks, a fact that might be the reason that there is still an ongoing debate on what the next generation Internet should look like. The disadvantage of 'flat', mainly format based representation languages (cf. HTML, LaTeX) is that they mix information on content (the text a writer wants to disseminate) and the format in which this is done (lay out issues). Such a disadvantage is to be opposed to a rather easy to learn language, so that virtually anybody with web access can easily publish information, knowledge and opinions.

Another reason for the need for more *web semantics* is that although the web is a media for publishing content, far from all its contents are created on or for it! In most cases documentation has to be reformatted and analysed before it can be published on the web, and extracting semantic contents from such un(web)structured documents might appear not to be easy at all.

Using an explicit representation language with clear semantics, where *content* is represented explicitly, usually sets a halt to the ease of use for most average users. Using 'higher' representation languages (cf. XML/RDF, or formal languages) in a similar manner as todays web publishing tools might therefore not be the best way to go, because it is expected to thwart publishing and sharing his or her knowledge and thoughts due to its higher complexity. Additionally, *backward compatibility* of a new *semantic web* representation language should be guaranteed.

Having had this debate for a few years now, in the meanwhile consensus seems to be that a combination of the two approaches could solve most of its drawbacks. As possible solution one can imagine tool support in order to either analyse pages that are not represented in a 'semantically rich' manner, or offering graphical interfaces (editors) to people that support creating such semantic representations (semi-)automatically.

These two scenarios are both seen as important, as long as they coexist on the web. A variety of projects with semantic representation languages have shown that representing all web contents in 'higher order languages' might not be feasible unless more automated approaches become available. However, an increasing acceptance of more semantic representation language on the web can be noticed and several initiatives aim at supporting them. Several project are initiated world-wide to support the *semantic web* ([FvHKA99], [Hen00], etc.), languages, extensions on languages and query languages are defined ([BG00], [FHH$^+$00]) and tools for (semi-)automatic content extraction are implemented ([BJ99], [oMAG00], [aA00]).

Nevertheless, last years of research, be it conducted government supported or private, have shown emerging technologies that, although often predicted and already initiated for many years ago, only recently unleashed some of the power that lies in a combination of semantic analysis and distributed information systems. One of these tools is developed in the private sector during the past four years, and has grown mature enough to serve as bottom technology in a variety of products, as well as in co-operation projects on a European level (cf. the On-ToKnowledge project [FvHKA99]).

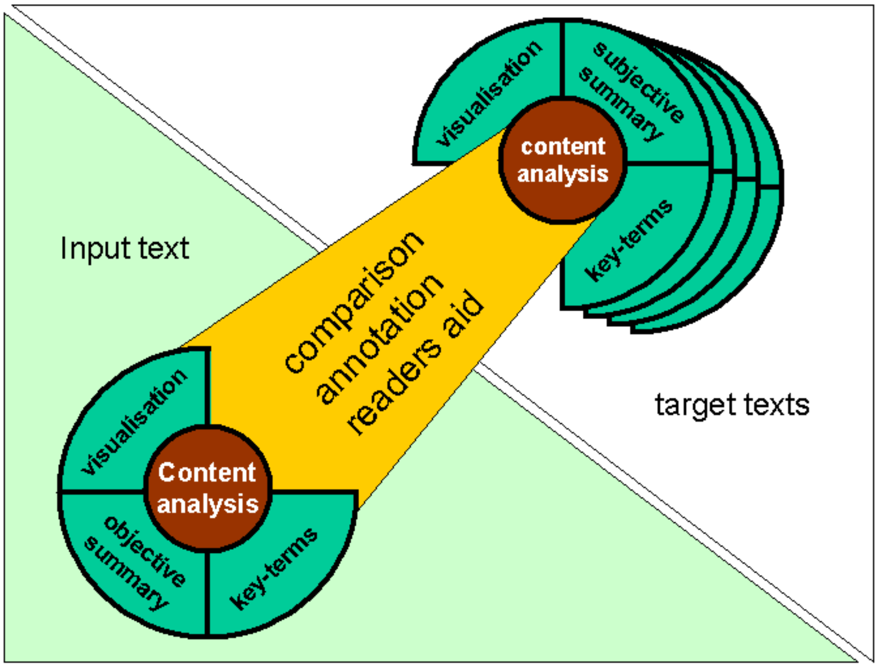


Figure 1: Core Analyser components' MiMir functionality.

2 Description of the CORPORUM system

For building up, utilising and maintaining the semantic web, there are a variety of tasks that are to be dealt with. All of these tasks find their *raison d'etre* in the fact that people need to get on top of the information overflow they get offered to them. This holds for individuals learning, organising and interacting on the web as much as for organisations that want their employees to mutually benefit of a better directed, better understandable and more clear information and knowledge sharing facility ([BJS$^+$99]).

On the theoretical side the *semantic web* is defined as the means by which this could be reached. At the technological side the CORPORUM tool set is defined as the server for semantic analyses ([BJ99], [EB00]). These analyses are performed by CORPORUMs' core component, a semantic analyser component called MiMir. Whereas MiMir is the core analysis component in the CORPORUM tool set, this very component can be used in a variety of settings due to its ability to extract contents, generate a semantic representation of the concepts (implicit as well as explicitly present in texts), relationships and roles. MiMirs' functionality is based on more formal computational linguistics. The computational linguistic paradigm (cf. figure 2 and [EB00]) takes place on three main levels: the *phonological level*, *word level*, *sentence level* and the *supra-sentential level* (very similar to the *discourse level*).

On top of this basic functionality, the analyser component has the ability to compare such representations of meaning, in order to find out how similar they are. Based on the results of this similarity analysis, the MiMir component offers advice on which documents are most pertinent to a specific analysed text, and return those parts in targeted documents most similar to a particular input text (cf. figure 1).

As soon as embedded in the CORPORUM tool set, the MiMir component is able to unleash its real strengths and serves as the 'brains' of intelli-

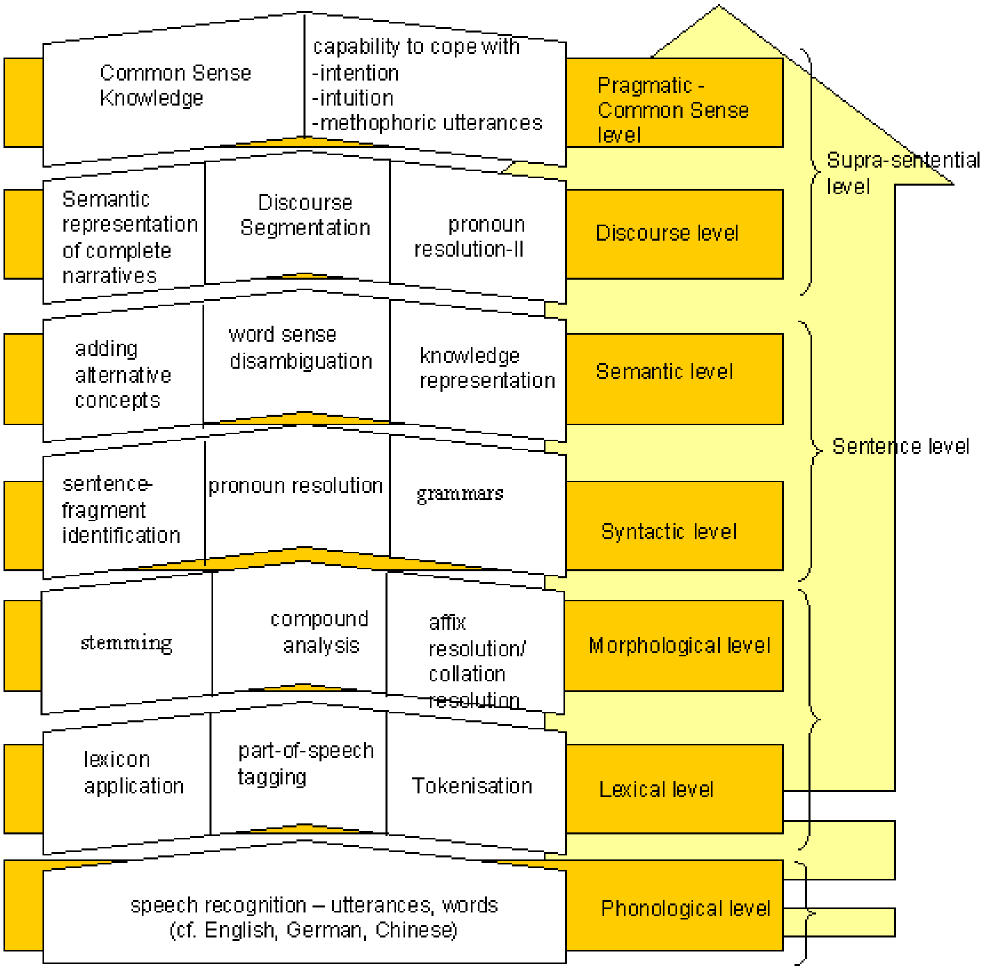
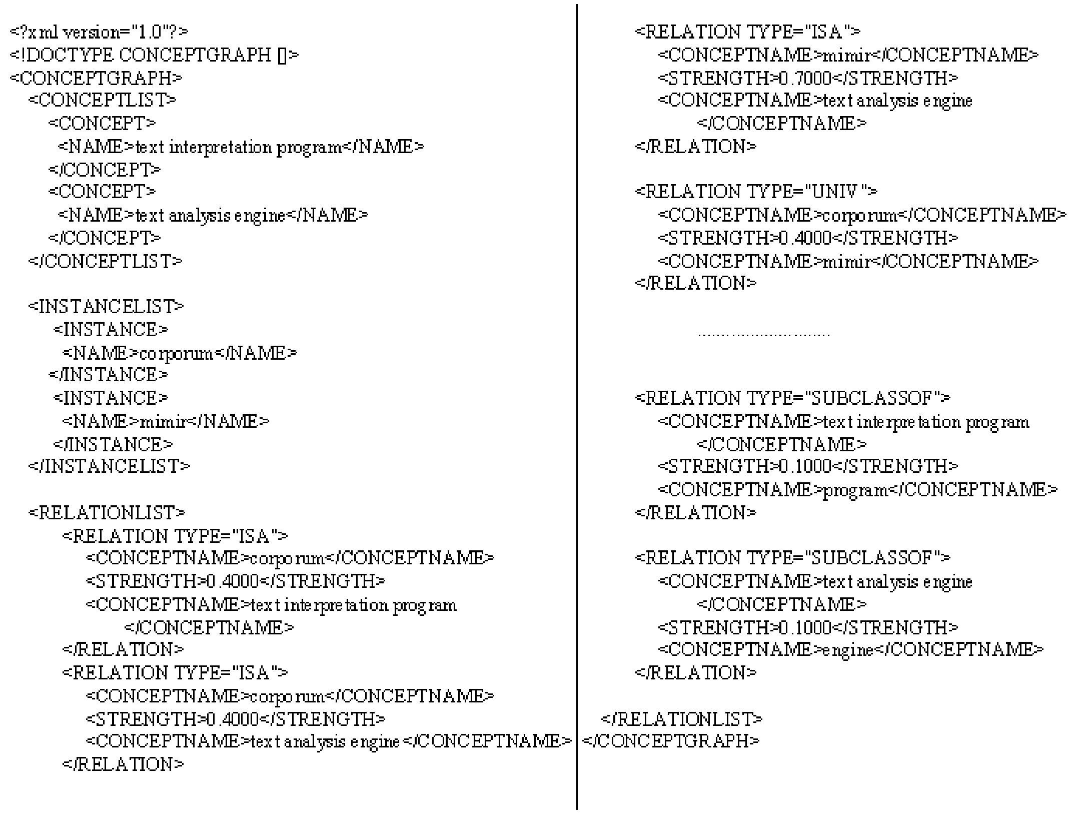

Common Sense Knowledge
capability to cope with -intention -intuition -methophoric utterances
Pragmatic - Common Sense level
Supra-sentential level
Semantic representation of complete narratives
Discourse Segmentation
pronoun resolution-II
Discourse level
adding alternative concepts
word sense disambiguation
knowledge representation
Semantic level
Sentence level
sentence-fragment identification
pronoun resolution
grammars
Syntactic level
stemming
compound analysis
affix resolution/ collation resolution
Morphological level
lexicon application
part-of-speech tagging
Tokenisation
Lexical level
speech recognition – utterances, words (cf. English, German, Chinese)
Phonological level

Figure 2: Grammatical analysis of texts.

gent agents gathering intelligence of all sorts on the web, be it medical knowledge for a specific new pro-leukine based medicine, a student wanting to collect material for a course or business intelligence about potential market opportunities or treats. For the intelligent agent scenario, a web server component, a database server, mission schedulers and a client server component are included. Another component in the CORPORUM tool set is the Summariser, which is capable of making summaries of texts based on a MiMir supported analysis of where the real information contents reside in the document. Alternatively such summaries can be made interest-driven, by using an interest profile (in the form of a natural language text) and generate summaries according to these.

Reflecting on the above, it can be said that three main scenario's for application of MiMir are most pertinent: a) *extraction* of information from texts for building knowledge bases, b) *retrieval* of information from other sources (search scenarios) and c) strategies to compact, visualise and disseminate information to people (dissemination and navigation). With the *semantic web* philosophy of an explicitly represented semantics as a given (RDF/OIL), the scenarios b) and c) become less important for the current discussion and we will therefore only pro-



```
<?xml version="1.0"?>
<!DOCTYPE CONCEPTGRAPH []>
<CONCEPTGRAPH>
  <CONCEPTLIST>
    <CONCEPT>
     <NAME>text interpretation program</NAME>
    </CONCEPT>
    <CONCEPT>
     <NAME>text analysis engine</NAME>
    </CONCEPT>
  </CONCEPTLIST>

  <INSTANCELIST>
     <INSTANCE>
      <NAME>corporum</NAME>
    </INSTANCE>
     <INSTANCE>
      <NAME>mimir</NAME>
     </INSTANCE>
  </INSTANCELIST>

  <RELATIONLIST>
      <RELATION TYPE="ISA">
         <CONCEPTNAME>corporum</CONCEPTNAME>
         <STRENGTH>0.4000</STRENGTH>
         <CONCEPTNAME>text interpretation program
               </CONCEPTNAME>
      </RELATION>
      <RELATION TYPE="ISA">
         <CONCEPTNAME>corporum</CONCEPTNAME>
         <STRENGTH>0.4000</STRENGTH>
         <CONCEPTNAME>text analysis engine</CONCEPTNAME>
      </RELATION>

      <RELATION TYPE="ISA">
         <CONCEPTNAME>mimir</CONCEPTNAME>
         <STRENGTH>0.7000</STRENGTH>
         <CONCEPTNAME>text analysis engine
               </CONCEPTNAME>
      </RELATION>

      <RELATION TYPE="UNIV">
         <CONCEPTNAME>corporum</CONCEPTNAME>
         <STRENGTH>0.4000</STRENGTH>
         <CONCEPTNAME>mimir</CONCEPTNAME>
      </RELATION>

               ............................


      <RELATION TYPE="SUBCLASSOF">
         <CONCEPTNAME>text interpretation program
               </CONCEPTNAME>
         <STRENGTH>0.1000</STRENGTH>
         <CONCEPTNAME>program</CONCEPTNAME>
      </RELATION>

      <RELATION TYPE="SUBCLASSOF">
         <CONCEPTNAME>text analysis engine
               </CONCEPTNAME>
         <STRENGTH>0.1000</STRENGTH>
         <CONCEPTNAME>engine</CONCEPTNAME>
      </RELATION>

  </RELATIONLIST>
</CONCEPTGRAPH>
```

Figure 3: An XML export based on a SemStruc.

vide short examples of them. Focus of this text will be on the generation of explicit knowledge and information from a specific text, so that it can be used for building knowledge bases and question answering (cf. RDF query language and tools [KCPA00]). Eventually generated knowledge bases contain results of semantical analysis of (web) documents and techniques to "mine" the underlying concepts and relations.

2.1 Making content explicit

For the question answering scenarios, but even for visualisation of contents for easy graphical interpretation, the content of the texts found on f.e. the web should be made explicit. There are several ways of performing content analysis, all having their own definition of *meaning*. An often found approach to content analysis is the statistical approach. In such approaches, words are not regarded as representing real-world artifacts of specific sorts, but are merely seen as patterns with statistical properties (frequencies and co-occurance frequencies). Typically an advantage of such an approach is that information retrieval can be made relatively language independent (pattern matching is universal), and is implemented rather computationally efficient. Instead of using pure statistical methods, Vector Space Models possibly combined with neural net technology or genetic algorithms, are also used. A mayor drawback of such approaches is the fact that elements in word-vectors typically have to be exact matches, causing certain word forms not to be recognized as being similar, even if they principally are (cf. plural and singular forms of words, different tenses of verbs, etc.). Whereas the problem with different suffixes (as in plu-

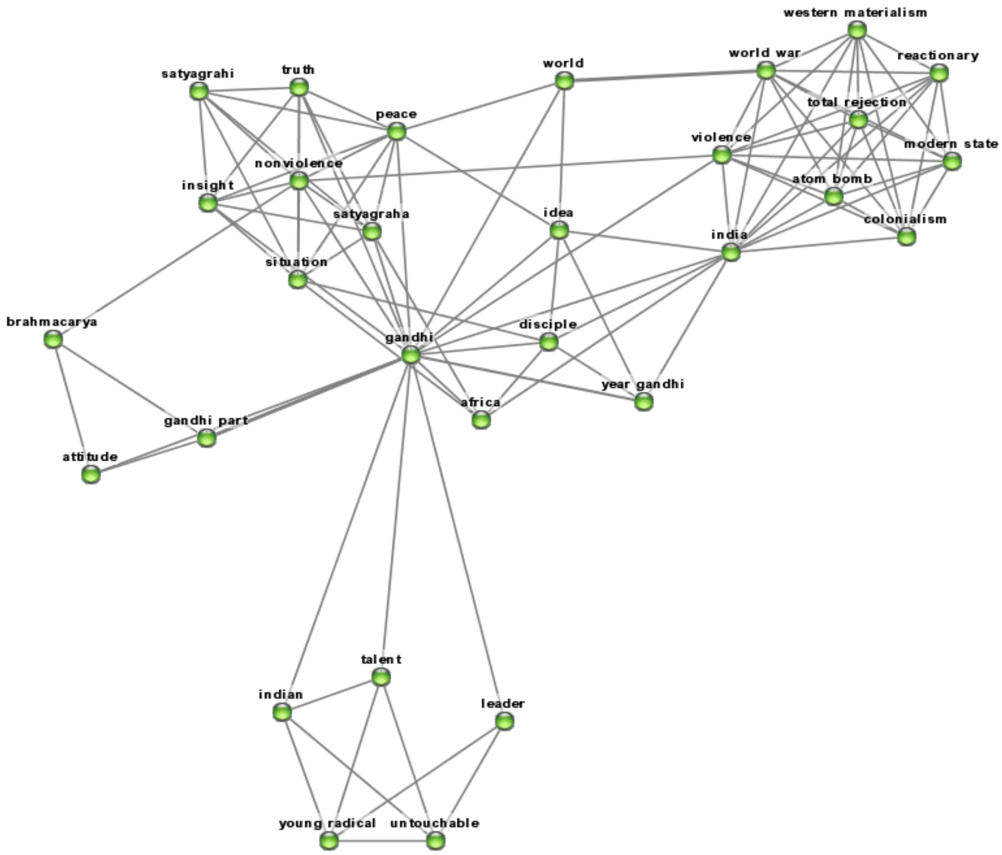


Figure 4: A visualised SemStruc generated from a CogLet (simplified version)

ral/singular and verb-tenses) has some solutions, there is less clarity on how to deal with synonyms, antonyms, etc.

On the other hand, there are pure formal grammar approaches, aiming to grasp meaning and semantics in a more formal sense. The definition of content as used in the *semantic web* defines meaning as an *explicit representation of the intention of a texts' author*. A natural way to represent such an explicit representation could be a (graphical) structure containing all concepts that play a role in a certain discourse, including intended concepts and relations between those (cf. three related concepts "Luther", "Martin" and "King" vs. a single 'intended' concept "Martin Luther King" bearing all semantic information in a single artifact. The concept of MLK could then be related to for example the concept of "civil rights leader" through grammatical sentence analysis). On top of this basic structure, concepts can be classified (f.e. 'instance', 'concepts', 'numbers' and 'names'). Currently MiMir is able to grasp the difference between specific types of relations that hold as well as a categorisation of the concepts it deals with. This capability makes MiMir not only suitable for typical Information Retrieval tasks, but also supports knowledge building for the semantic web and provides the information needed for *question answering*.

Linguistic Text Analysis

As discussed above, the basic analysis of a text as performed by MiMir is based on a tokeniser, a Part-Of-Speech tagger, stemming algorithms, Named Entity recognition facilities as well as a proprietary algorithm for generating concepts out of single words (cf. figure 2). From the information that is gathered during these analyses, a CogLet representation is generated which puts all information in relation and defines the context in which information should be interpreted.

The information residing in a single CogLet can now be used to export semantic structures (so-called *SemStrucs*). SemStrucs can be represented in XML format, which could be fed into visualisation algorithms. CogLets also contain the necessary information to analyse web pages and augment them with a Resource Description Framework (RDF) part describing document meta data (according to Dublin Core) and a lightweight ontology based on the analysed natural language text contained in the document.

Semantic Structures in XML

Information contained in a CogLet can also be exported into an XML format, so that it can be used as semantic annotation on a web site or in a knowledge base. XML has been chosen because it is regarded as the next step upward from standard HTML annotations. Only a subset of the information in the CogLet is used for the XML generation, while containing enough information for the generation of graphics.

Figure 3 provides an example of such an XML representation. Relations in such visualisations are not only typed, but also annotated with a calculated heuristic strength. The XML represented information in figure 3 could be used as input for the graph visualiser.

Visualising Semantic Structures

As mentioned before, one of the strengths of SemStructs is that they can be used for visualisation interests and contents. This capability allows for usage in visual browsers and navigators based on larger document sets, and to offer people an at-a-glance overview over the information they have access to.

Figure 4 shows a simplified[1] structure created from a SemStruc generated by a CogLet and visualised with CCAviewer[2]. The structure shows the semantic clusters around the person "Ghandi". There are three main clusters recognisable, one dealing with Ghandi's roles (`<young radical>`, `<leader>` and `<talent>`), one dealing with his philosophy (`<satyagraha>`, `<non-violence>` and `<insight>`) and one dealing with the violent world Ghandi fought against (`<colonialism>`, `<violence>`, `<total rejection>`, `<western materialism>`).

Pictures that are thus automatically generated from natural language texts provide an at-a-glance overview over a piece of information. Such pictures can then be used in order to augment executive summaries and readers aids, but they are also used as visual interfaces to databases (preferably in corporate settings). As such they augment knowledge management systems, where they provide a visual entrance to pieces of information pertinent to specific interest groups within an enterprise.

As an example of the expressive power of the SemStrucs, one might take some time to analyse figure 4 and try to imagine what the original text is about, and which 'discourses' the original document contained.

Augmenting web sites with RDF

Within the OnToKnowledge project, RDF with extensions are used as representation language for the *semantic web* (cf. OTK: [FvHKA99], OIL: [HFB$^+$99]). The CORPORUM$_{OntoExtract}$ component is directed to the generation of a

[1] As SemStrucs represented in XML/RDF/OIL are formal representations, they will easily grow too large for inclusion in a paper. Hence the very short 'CORPORUM' text example. The visualisation of SemStrucs *condenses* texts, and could therefore be based on a larger-sized text (about Ghandi).

[2] The CCA viewer is a product by Aidministrator Nederland BV. It uses CogLet generated SemStrucs to generate pictures based on so-called augmented Spring Embedder technology (cf. figure 4). CCA stands for Central Concept Area, referring to the information created by the SemStrucs.

'light-weight ontology' based on linguistic analysis by CORPORUM in combination with the information that SemStrucs can provide. This means that formal taxonomic relationships that hold in the discourse at hand are disclosed and made explicit as a set of RDF tuples. Additionally, traditional web pages are augmented with Dublin Core meta data, also generated automatically by the CORPORUM$_{OntoExtract}$ component[3].

Figure 5 provides an example of such automatically generated DC and ontologic knowledge. The attentive reader will notice that there are two constructs declared that are not used in the example, i.e. <isRelated> and <hasSomeProperty>. These two constructs are defined in the OIL language. Whereas a typical ontology often represents a taxonomy (the ontology in the example is no exception on this), <isRelated> refers to cross-taxonomic links that can hold within a domain and, if represented, can make a difference in finding needed information based on context descriptions. As an example one can imagine two CCA concepts like <oil-rig> and <ship>. Such concepts are not typically 'close' in a traditional ontology, where they are not found as sub-classes of vehicles (<oil-rigs> are not typically means of transportation), neither as sub-class of a concept like <building>, <floating device>, etc. Nevertheless, people working in the oil industry typically regard the two concepts as highly related, not in the least due to their natural 'symbiosis' in everyday 'life on the rig'. CORPORUM is however able to capture such *cross-taxonomic* links and represent them using the <isRelated> structure.

The other construct (<hasSomeProperty>) is the most general, universal relation type reflecting *part-whole* relations within a taxonomy. It is currently used in CORPORUM$_{OntoExtract}$ to define not further specified *part-whole* relationships between a <concept> and a <specifier> of that concept. However, in some cases there is knowledge available from the Knowledge Base that allows us to further refine the type of properties are actually present. In such cases, CORPORUM$_{OntoExtract}$ will query the KB in order to find out how it can enhance its knowledge representation. At current this process extends Ontology generation in RDF from being single text based linguistic analysis into an augmentation process where previously generated ontologic knowledge (containing "background" knowledge about the domain) is taken into consideration as much as possible. After having augmenting the ontology generated by CORPORUM$_{OntoExtract}$ thus, the complete RDF(S) representation is send to the RDF repository maintained for ontology storage (OntoKnowledge - Sesame at the moment).

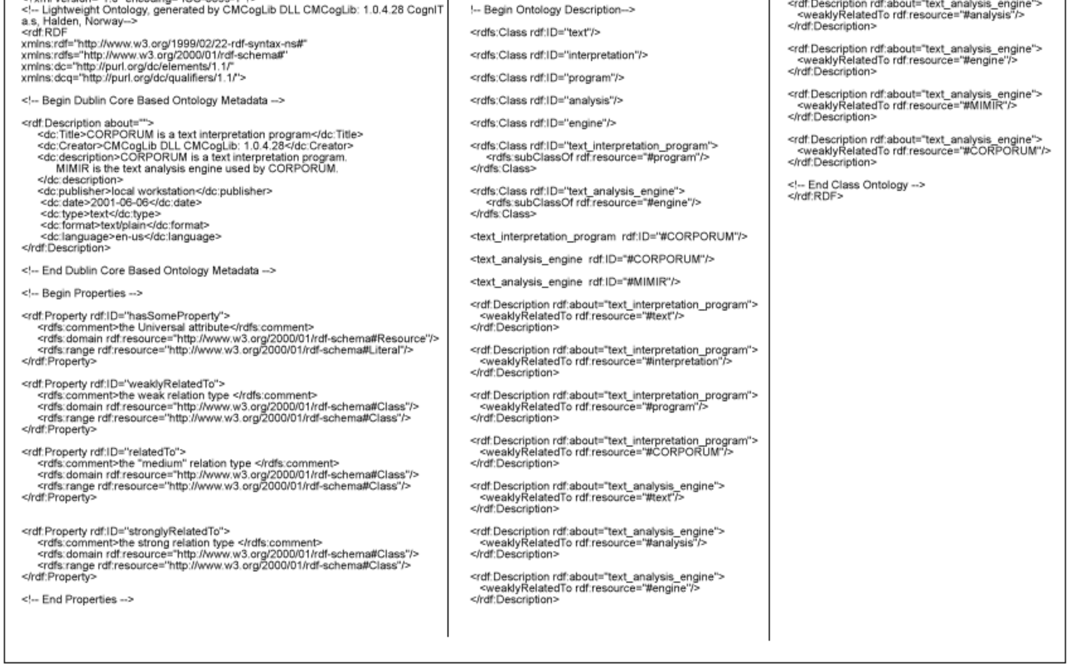
Future Developments

While used in a variety of commercial products, ranging from *Intelligence Portals*, *Intelligent Crawler Systems* to *Summarising Tools* and *Visualising Components*, the CORPORUM tool set is subject to continuous improvement. Currently the system is able to deal with English, German and Norwegian texts, whereas more of the European languages (French, Spanish, Dutch) are expected to be added soon.

The MiMir component is also subject to continuous improvement, so that the CogLet generated SemStrucs get an even richer 'meaning' representation model. At the same time the functionality of the core MiMir component is enhanced in such a way that it can serve many more tasks (think of enhanced summarising, including smoothing, discourse recognition, as well as a more flexible Natural Language based readers aid.).

An issue that is currently dealt with is directed to scenarios where one wants to 'answer questions'. In such cases

[3] DC meta data includes information about author, key concepts, summary of the content of a document, its URI, etc. Dublin Core meta data is described at: http://purl.oclc.org/dc.



```
<!-- Lightweight Ontology, generated by CMCogLib DLL CMCogLib: 1.0.4.28 CognIT
a.s, Halden, Norway-->
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcq="http://purl.org/dc/qualifiers/1.1/">

<!-- Begin Dublin Core Based Ontology Metadata -->

<rdf:Description about="">
    <dc:Title>CORPORUM is a text interpretation program</dc:Title>
    <dc:Creator>CMCogLib DLL CMCogLib: 1.0.4.28</dc:Creator>
    <dc:description>CORPORUM is a text interpretation program.
        MIMIR is the text analysis engine used by CORPORUM.
    </dc:description>
    <dc:publisher>local workstation</dc:publisher>
     <dc:date>2001-06-06</dc:date>
     <dc:type>text</dc:type>
     <dc:format>text/plain</dc:format>
     <dc:language>en-us</dc:language>
</rdf:Description>

<!-- End Dublin Core Based Ontology Metadata -->

<!-- Begin Properties -->

<rdf:Property rdf:ID="hasSomeProperty">
    <rdfs:comment>the Universal attribute</rdfs:comment>
    <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:ID="weaklyRelatedTo">
    <rdfs:comment>the weak relation type </rdfs:comment>
    <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Property>

<rdf:Property rdf:ID="relatedTo">
    <rdfs:comment>the "medium" relation type </rdfs:comment>
    <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Property>


<rdf:Property rdf:ID="stronglyRelatedTo">
    <rdfs:comment>the strong relation type </rdfs:comment>
    <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Property>

<!-- End Properties -->

!-- Begin Ontology Description-->

<rdfs:Class rdf:ID="text"/>

<rdfs:Class rdf:ID="interpretation"/>

<rdfs:Class rdf:ID="program"/>

<rdfs:Class rdf:ID="analysis"/>

<rdfs:Class rdf:ID="engine"/>

<rdfs:Class rdf:ID="text_interpretation_program">
    <rdfs:subClassOf rdf:resource="#program"/>
</rdfs:Class>

<rdfs:Class rdf:ID="text_analysis_engine">
    <rdfs:subClassOf rdf:resource="#engine"/>
</rdfs:Class>

<text_interpretation_program  rdf:ID="#CORPORUM"/>

<text_analysis_engine  rdf:ID="#CORPORUM"/>

<text_analysis_engine  rdf:ID="#MIMIR"/>

<rdf:Description rdf:about="text_interpretation_program">
  <weaklyRelatedTo rdf:resource="#text"/>
</rdf:Description>

<rdf:Description rdf:about="text_interpretation_program">
  <weaklyRelatedTo rdf:resource="#interpretation"/>
</rdf:Description>

<rdf:Description rdf:about="text_interpretation_program">
  <weaklyRelatedTo rdf:resource="#program"/>
</rdf:Description>

<rdf:Description rdf:about="text_interpretation_program">
  <weaklyRelatedTo rdf:resource="#CORPORUM"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#text"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#analysis"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#engine"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#analysis"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#engine"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#MIMIR"/>
</rdf:Description>

<rdf:Description rdf:about="text_analysis_engine">
  <weaklyRelatedTo rdf:resource="#CORPORUM"/>
</rdf:Description>

<!-- End Class Ontology -->
</rdf:RDF>
```

Figure 5: An excerpt of CORPORUM$_{OntoExtract}$ generated RDF annotation including Dublin Core meta data.

more separated information should be available that f.e. can make the difference between a *concept* (i.e. `<car manufacturer>`) and an *instance* thereof (i.e. 'Renault'). The question what the difference between *instances* and *concepts* actually is is not always straightforwardly answered, as can be learned from ongoing discussions at the academic level on this topic. Therefore further development of CORPORUM$_{OntoExtract}$ within the OnToKnowledge project will be directed towards the (semi-?) automatic generation of RDF represented 'semantic' knowledge, which is to be used by reasoning and query engines developed within the very same project. More specifically, the algorithms defining the `<isRelated>` relationships will be refined in order to more precisely reflect the specifiers of concepts holding in specific domains (i.e. instead of currently stating that a specific instance`<car_01>` has a property `<isRelated>` with value `<red>`, it might be able to refine the `<isRelated>` property with a subrelation `<hasColour>` with the same value.

The CORPORUM tool set as such tends to grow with the functionality of its core component as well as with the imagination of and familiarity with Knowledge Management scenarios by key 'Knowledge Managers' in the large enterprises we cooperate with. It is our experience that in many situations there is a larger problem in making people understand the potential on a human and organisational level of semantic tools, as that there is showing the technical principles behind it. One can discuss why this is the case, the main reason possible being that larger enterprises tend to have capable people working with what we would call 'Knowledge Management', although the enterprise as a whole does not always seem to realise enough the benefits of actually integrating/implementing solutions developed by such KM departments at a company width scale. An acceptance of the industry of the possibilities of the semantic web should be boosted by the availability of tools to support it. Although currently only tested in smaller, controlled environments, the tool set discussed in this paper seems to address many of the issues raised in this paper.

References

[aA00] Knowledge Management Group at AIFB. Ontology Engineering Environment OntoEdit. Technical report, Angewandte Informatik und Formale Beschreibungsverfahren, University of Karlsruhe, D, http://ontoserver.aifb.uni-karlsruhe.de, 2000.

[BG00] D. Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification 1.0. Technical report, W3C Consortium, http://www.w3.org/TR/rdf-schema/, 2000.

[BJ99] B. Bremdal and F. Johansen. CORPORUM; Technology and Applications. Technical report, CognIT a.s, Halden, Norway, http://www.cognit.no/, 1999.

[BJS+99] B. A. Bremdal, F. Johansen, Ch. Spaggiari, R. Engels, and R. Jones. Creating a Learning Organisation through Content Based Document Management. Technical report, CognIT a.s, Halden, Norway, http://www.cognit.no/, 1999.

[EB00] R.H.P. Engels and B.A. Bremdal. Information Extraction. Technical report, OnToKnowledge Consortium, http://www.ontoknowledge.org /del.shtml, 2000.

[FHH+00] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein. OIL in a nutshell. In R. Dieng et al., editor, *Knowledge Acquisition, Modeling and Management: Proceedings of the European Knowledge Acquisition Conference (EKAW-2000).* Springer, Berlin, Heidelberg, New York, 2000.

[FvHKA99] D. Fensel, F. van Harmelen, M. Klein, and H. Akkermans. On-To-Knoweldge: Ontology-based Tools for Knowledge Management. In *Proceedings of the Ebusiness and Ecommerce Conference*, Madrid, Spain, 1999.

[Hen00] J. Hendler. DARPA Agent Markup Language. Technical report, Defense Advanced Research Projects Agency (DARPA), http://www.daml.org/, 2000.

[HFB+99] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, and R. Studer. The Ontology Inference Layer OIL. Technical report, OnToKnowledge EU-IST-10132 Project Deliverable No. OTK-D1, http://www.ontoknowledge.org, 1999.

[KCPA00] G. Karvounarakis, V. Christophides, D. Plexousasikis, and S. Alexaki. Querying Community Web Portals. Technical report, ICS-FORTH, Heraklion, Greece, http://www.ics.forth.gr/proj/ isst/RDF/RQL/rql.(html, pdf, ps, dvi), 2000.

[oMAG00] University of Manchester, Free University Amsterdam, and Interprice GmbH. OilEd. Technical report, University of Manchester, UK, http://img.cs.man.ac.uk/oil/, 2000.