Del 6: CORPORUM-OntoExtract
Ontology Extraction Tool

Robert Engels,
CogniT a.s
On-To-Knowledge Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-1999-10132. The partners in this project are: Vrije Universiteit Amsterdam (VU) (co-ordinator), NL; the University of Karlsruhe, Germany; Schweizerische Lebensversicherungs- und Rentenanstalt / Swiss Life, Switzerland; British Telecommunications plc, UK; CognIT a.s, Norway; EnerSearch AB, Sweden; AIdministrator Nederland BV, NL.

Vrije Universiteit Amsterdam (VU)
Faculty of Sciences, Division of Mathematics and Computer Science
De Boelelaan 1081a
1081 HV Amsterdam, the Netherlands
Fax and Answering machine: +31-(0)20-872 27 22
Mobil phone: +31-(0)6-51850619
Contactperson: Dieter Fensel
E-mail: dieter@cs.vu.nl

University of Karlsruhe
Institute AIFB
Kaiserstr. 12
D-76128 Karlsruhe, Germany
Tel: +49-721-608392
Fax: +49-721-693717
Contactperson: R. Studer
E-mail: studer@aifb.uni-karlsruhe.de

Schweizerische Lebensversicherungs- und Rentenanstalt / Swiss Life
Swiss Life Information Systems Research Group
General Guisan-Quai 40
8022 Zürich, Switzerland
Tel: (41 1) 284 4061, Fax: (41 1)284 6913
Contactperson: Ulrich Reimer
E-mail: Ulrich.Reimer@swisslife.ch

British Telecommunications plc
BT Adastral Park
Martlesham Heath
IP5 3RE Ipswich, UK
Tel: (44 1473)605536, Fax: (44 1473)642459
Contactperson: John Davies
E-mail: John.nj.Davies@bt.com

CognIT a.s
Busterudgt 1,
N-1754 Halden, Norway
Tel: +47 69 1770 44, Fax: +47 669 006 12
Contactperson: Bernt. A.Bremdal
E-mail: bernt@cognit.no

EnerSearch AB
SE 205 09 Malmö, Sweden
Tel: +46 40 25 58 25; Fax: +46 40 611 51 84
Contactperson: Hans Ottosson
E-mail: hans.ottosson@enersearch.se

AIdministrator Nederland BV
Julianaplein 14B
3817CS Amersfoort, NL
Tel: (31-33)4659987, Fax: (31-33)4659987
Contactperson: Jos van der Meer
E-mail: Jos.van.der.Meer@aidministrator.nl
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>4</td>
</tr>
<tr>
<td>2. CORPORA-ONTOEXTRACT</td>
<td>6</td>
</tr>
<tr>
<td>3. EXTRACTED KNOWLEDGE TYPES</td>
<td>ERROR! BOOKMARK NOT DEFINED.</td>
</tr>
<tr>
<td>4. TECHNICAL ARCHITECTURE</td>
<td>16</td>
</tr>
<tr>
<td>5. PROGRAMMERS GUIDE</td>
<td>18</td>
</tr>
<tr>
<td>6. DISCUSSION ON USE OF CORPORA TECHNOLOGY IN ONTOEXTRACT</td>
<td>20</td>
</tr>
</tbody>
</table>
1. Introduction

Known as the knowledge acquisition problem, many “reasoning” tasks tend to require large investments in building up the necessary knowledge bases and require complex knowledge representation languages.

Already starting in the early days of the OnToKnowledge project, there were two issues that became a major point of discussion, representation formalisms and information extraction issues. The DAML-OIL language is finally developed as representation language (cf. OTK-project deliverable 1: Ontology Language). The next issue identified as important was the need for reasoning with knowledge represented in this language and of course filling up knowledge bases in the language defined. RQL (cf. OTK-project deliverable 9: Query Language Definition) was defined and implemented on top of a RDF server (SeSame).

Filling out the bones with flesh, the choice for an information retrieval and extraction tool was made. CORPORUM (cf. OTK-project deliverable 5: State-of-the-Art in Information Extraction) was taken for its capability to crawl the internet and local file systems for documents, analysing relevance and content on the way, while carefully maintaining a knowledge base with information that was encountered.

Now, having identified the need for support in setting up knowledge bases and representing information available in a specific domain, there were two types of information to be extracted (structured vs. unstructured), whereas the evaluations that where performed pointed into the directory of an additional piece of software that was required: an Ontology Editor (cf. OTK-project deliverable 3: OntoEdit). The final bottom layer system architecture of the OnToKnowledge workbench was now defined: CORPORUM-OntoExtract and CORPORUM-OntoWrapper takes care of (unstructured and structured, respectively), textual information sources at the bottom, retrieving documents from specified domains at the Internet or Intranet. The extracted information is so pumped into the RDF-DB (Sesame), where it can be edited with the OntoEdit tool. Finally, the RQL reasoning engine allows for the querying of this database and delivers results to a user through a smart user-interface (cf. OTK-project deliverable 11: RDF-Ferret).

At the end the complete architecture of the OnToKnowledge Workbench components cover all the aspects that need to be covered in a “semantic web” (cf. Figure 1: OTK system architecture).

This deliverable report discusses the CORPORUM-OntoExtract tool, its architecture and its use.
The first year-and-a-half of the OTK project has led to a variety of newly developed functionality based on the original CORPORUM system. Most of these developments where not originally specified in the project plan, but arose after discussions and/or needs of the case partners. The developments that were made (and finally led to the current OntoBuild architecture with CORPORUM-OntoExtract and CORPORUM-OntoWrapper) are:

- **CORPORUM – CCAexport**
  tool for extracting semantic structures in XML format. This tool was initialised by the project partners to let CORPORUM provide input for visualisation tools. For this first extended functionality CORPORUM was augmented with an export function for XML (currently there are three formats available). The XML representations could be visualised using CCAviewer software (developed by Administrator BV, NL), or by using freely available Java Applets\(^1\).

- **CORPORUM – Islands**
  Functionality for enabling context related feedback in search engines. CORPORUM is able to specify words within their context, such that a users of a search engine, scrolling a specific topic, gets domain-specific, related keywords as alternatives to the currently used keywords.

- **CORPORUM – OntoExtract**
  first, web-based version of a CORPORUM version able to extract ontologies and represent them in XML/RDF/OIL.

\(^1\) such as SUN’s graph viewer, available from [http://javaboutique.internet.com/Graph/index.html](http://javaboutique.internet.com/Graph/index.html)
2. **CORPORUM-OntoExtract:**

*Information Extraction from unstructured documents*

In order to understand the ideas beyond CORPORUM-OntoExtract we have to take a step back and have a look at the CORPORUM system as a whole (cf. OTK-project deliverable 5: State-of-the-Art in Information Extraction).

CORPORUM supports both personal and enterprise wide document and information management - that is management by content. The CORPORUM system is founded on CognIT’s Mimír technology developed in Norwegian research labs. This technology focuses on meaningful content rather than odd data or standardised document parameters. CognIT’s mission is to capture the content with respect to the interest of the individual rather than address the document itself. There are three essential aspects of this.

- CORPORUM interprets text in the sense that it builds ontologies that reflect world concepts as the user of the system sees and expresses this. This ontology constitutes a model of a person’s interest or concern.
- The interest model is applied as a knowledge base in order to determine contextual and thematic correspondence with documents presented before it.

The interest model and the text interpretation process drive an information extraction process that characterises hits in terms of relevance and in terms of content. This information can be stored in a persistent database for future reference.

![Figure 2: Corporum core competence areas (cf. OTK- deliverable 5, page 75 ff.).](image)

The CORPORUM software consists of a linguistic component, taking care of tasks as lexical analysis (tokenisation, part-of-speech tagging and possibly lexicon application), morphological analysis (collation, inflection) and analysis at the syntactical level (grammar application). At the semantic level, CORPORUM performs word sense disambiguation by describing the context in which a particular word is being used. Doing so is naturally closely related to knowledge representation issues. CORPORUM is able to augment “meaning” structures with concepts that are invented from the text. The core of the CORPORUM system (the MiMír engine) is also able to extract the information most
pertinent to a specific text for summary creation, extract the so called Core Concept Area from a text and represents results according to ranking which is based on its interestingness towards a specific contextual theme set by a user. On top of that, the CORPORUM system is able to generate explanations, which will allow a user to make an informed guess on which documents to look at and which to ignore. CORPORUM can point to exactly those parts of the targeted documents that are most pertinent to a specific user’s interest. Figure 2 shows the general architecture of the CORPORUM software.

General CORPORUM- Business Intelligence Portal architecture

The overall architecture of the CORPORUM system is depicted in Figure 3. It consists of four basic software components each coming with a well-defined API. This implies that CORPORUM can be reconfigured compared to the depiction given here. It also implies that each of the components can be reused in a different setting. This is especially important for the part that contains the MIMIR based engine. This is the heart of the CORPORUM system, but it may operate on an independent basis as long as its API is

report on deliverable 6: software tool for ontology extraction

- 7 -
observed. Because of this it is possible to include a subset of the components shown in various other configurations. One example could be an ERP system that needs contextual indexing of different types of documents. Figure 3 shows the several components that the architecture consists of. The user may have access to CORPORUM from any web-browser hooked up on the net. Access to CORPORUM is given through a regular ASP functionality that communicates with a Web Data Server. The Web Data Server handles the interface to the database. CORPORUM is suited with a standard relational database. However, any database can be applied. A change to this part will not have any effect on the main architecture since both the Web Data Server and the Data Server are designed according to a pure object-oriented standard. All database calls from CORPORUM components are made on an abstract level. The servers interpret this to SQL-calls or similar of the type required for any given database. The Data Server feeds analysis results from the CORPORUM kernel so that it can be maintained in the database. The kernel consists of several lesser components. The most important of these are the ones that accommodate the MÍMÍR technology. This consists of several algorithms that drive the analysis and information extraction functions. Several of these algorithms apply linguistic rules and information contained in separate files. In order to handle multiple languages CORPORUM will contain several sets of these files.

The CMWebHandler contains both crawler capabilities as well as document processing functions. CORPORUM can be equipped with a set of such handlers in order to treat different types of document formats beyond standard HTML. The CMWebHandler receives search instructions from the kernel component.

CORPORUM- OntoExtract has been built upon the same architecture, reusing some of the components and functionality of it (CMWebHandler, keyword extraction, cluster analysis etc.) in Figure 3.

Ontologies

CORPORUM applies ontologies in order to establish whether two entities communicate. In order to establish “real communication” both the speaker and the listener must share an ontology. If an author wants a reader to understand what he writes he must attempt to find terms that can serve two basic purposes:

- Express his ideas
- Trigger an understanding of the reader

Only when this is achieved the message can get across. CORPORUM applies these principles in its treatment of the document text to be analysed.

The MÍMÍR approach

And Odin hungered for the wisdom that the well could yield. "Nay", said the giant, he stepped up against the mighty god himself. Odin mustered the other, "You ask a high price". "Wisdom gives power. MÍMÍR has got what you want. An eye and you may drink." Odin ripped his eyeball out and bent over the rim in the shade of the huge tree. He filled his mouth and drank. The well of MÍMÍR was truly strange. As he devoured the liquid that tasted like water he sensed a new divine feeling. Even without his eye he could see so much further.

--- Free after Norwegian Mythology

2 Odin - The Norse god of wisdom, war, art, culture, and the dead and the supreme deity and creator of the cosmos and human beings.
The CORPORUM products are powered by a technology that was invented by CognIT. It consists of two basic concepts, a concept extraction facility and a resonance algorithm. Concept extraction focuses on the semantics of a text. The approach is rooted in classic information theory dating back to the seminal work of Claude Shannon as well as more formal linguistic analysis. Therefore, MÍMIR looks beyond the mere signal and analyses the concept behind the term. The concept extraction effort combines natural language processing and knowledge intensive methods. The component taking care of this process is the CMCogLib component, which is also the component used for information gathering in the OntoBuild system.

Another important component in many applications of CORPORUM is the resonance algorithm. This algorithm enables comparison between the contents of two texts. It implies that the conceptual structures of the two texts are analysed against each other. The resonance metaphor stems from the idea that the match process triggers violent reactions if the frequency of the emitted signal is close to the natural undamped frequency of the objects themselves. In other words if there is a good match then there is high resonance with respect to the content of the text found.

**Contextual interpretation**

The basic text interpretation capability of MÍMIR is contextual. Framing the context inherent in a document is fundamental if you want to enable prudent indexing and grouping. In such a case you let the content of the document rather than an ad hoc set of keywords determine the indexing. Once you have settled the context it is possible to determine what the document is all about, how it overlaps the content of other documents. Moreover contextual homogeneity assures that quick lexically based searches returns the desired results. The MÍMIR technology also enables mapping the document content. Since the approach focuses on meaning it is possible to visualise the knowledge inherent in a document. Linked with objective parameters such as name of author and creation date it is possible to build a "who knows what" directory and to measure development in knowledge focus for both individuals and groups.

**Information distribution**

Ontologies embedded in text will define its content. Central concepts constituting the core can be listed once the document is analysed. This summary function provides a simple, but important overview of the text found and can serve as a useful "super abstract". In the event that agents work on behalf of the user it is important that the analysis and match operation is made transparent. It is important that the agent conveys its findings and the match results in a way that enables the user to make decisions based on that.

![Figure 4: Histogram showing where the desired content in the document can be found and to what degree it is pertinent. Top of document is to the left. End of document is on](image)
the right hand side of the diagram. If a bar is selected and double-clicked the system will launch the paragraph represented by the bar.

A recent development in the current MÍMIR component that is based upon the information on information distribution. This summarization function is used in CORPORM- OntoExtract for generation of abstracts according to the Dublin Core MetaData set.

3. Prototypes developed during the OTK-project
Systems tested and implemented in cooperation with the project partners.

CORPORM – CCAexport

tool for extracting semantic structures from natural language texts and representing these in XML format. This tool was initialized by the project partners to let CORPORM provide input for visualization tools. For this first extended functionality CORPORM was augmented with an export function for XML (currently there are three formats available). The XML representations could be visualised using CCAviewer software (developed by Administrator BV, NL), or by using freely available Java Applets.3

Figure 5: OTK and the semantic web, visualised semantic structure.

3 such as SUN’s graph viewer, available from http://javaboutique.internet.com/Graph/index.html

report on deliverable 6: software tool for ontology extraction
CORPORUM – Islands
Functionality for enabling context related feedback in search engines. CORPORUM is able to specify words within their context, such that a users of a search engine, scrolling a specific topic, gets domain-specific, related keywords as alternatives to the currently used keywords. In the example, the closest related concepts to the specified concept (“oil-rig”) are represented as an ordered list.

Example:
<oil-rig> : <oil-tanker, pipe-line, gas, ship, helicopter>

The getIsland function is domain specific, i.e. is automatically build from an available set of documents and could be used for analysing such domains.

CORPORUM – OntoBuild
The CORPORUM-OntoBuild software is build up around two different groups of target documents: structured texts (i.e. there is knowledge available about the lay-out of a document), and unstructured texts (no such information available).

CORPORUM – OntoExtract
first, web-based version of a CORPORUM version able to extract ontologies and represent them in XML/RDF/OIL. The latest version of CORPORUM-OntoExtract is able to communicate with and negotiate final format of the to-be-submitted ontology extracted from a specific text.

CORPORUM – OntoWrapper
for tackling the structured documents found, OTK-project deliverable 7 will deliver a tool called CORPORUM-OntoWrapper. Using that tool, one can extract knowledge from pages where the structure is known.
4. CORPORUM-OntoExtract

Based on the functionality described above, CORPORUM-OntoExtract exports RDF in a format that consists of a variety of sections. Generally speaking there is a section on defining all RDF/XML namespaces, followed by a section defining the relation that are used in defining the ontology. The ontology consists of classes, subclasses, instances and a representation based on relations between classes based on semantic information available to CORPORUM-OntoExtract. In the RDF output the following sections can be distinguished:

- **Namespace definitions**
  (RDF/XML): here the namespaces are defined, and the version control string from the CognIT component is provided.

**EXAMPLE**:
```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Lightweight Ontology, generated by 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/">
</rdf:RDF>
```

- **Dublin Core based metadata**
  Based on available information (mainly webserver) and the CMCogLib generated content model Dublin Core based Metadata is created (cf. Dublin Core specification: [http://purl.org/dc](http://purl.org/dc)). The following elements are exported:

**EXAMPLE**:
```xml
<!-- Begin Dublin Core Based Ontology Metadata -->
<rdf:Description about="">
  <dc:Title>This agent searches for information and semantics of the OTK or OntoKnowledge project</dc:Title>
  <dc:Creator>CMCogLib: 1.0.4.28</dc:Creator>
  <dc:description>
    The On-To-Knowledge project applies ontologies to electronically available information to improve the quality of knowledge management in large and distributed organisations. The major outcomes of On-To-Knowledge (or OTK) are a methodology, tools, and best-practises for knowledge management. OTK partners comprise commercial enterprises like Administrator, British Telecom, Swiss Life, Enersearch and CognIT, as well as academic partners as the University of Amsterdam and the University of Karlsruhe.
  </dc:description>
  <dc:publisher>local workstation</dc:publisher>
  <dc:date>2001-06-07</dc:date>
  <dc:type>text</dc:type>
  <dc:format>text/plain</dc:format>
</rdf:Description>
```

report on deliverable 6: software tool for ontology extraction

- 12 -
- **Property definitions**

In this section properties that are used for describing relations between concepts are defined. Properties that are used by CORPORUM-OntoExtract are (apart from properties defined after interaction with the RDF server):

- universal property `hasSomeProperty`, used for definition of properties belonging to an instance where no further information on property type is available in the knowledge base.
- semantic cluster information `<RelatedTo>` properties. These properties are used to describe relationships between concepts and concepts, concepts and instances and relations of instances with each other.

**EXAMPLE:**

```xml
<!-- 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 -->
```

- **Ontology (taxonomical)**

Taking a single text or document as input, CORPORUM-OntoExtract retrieves a document specific light-weight ontology from it. Ontologies extracted by CORPORUM-OntoExtract are basically taxonomies that represent classes, subclasses and instances.
- **Classes**, described in the text which is analysed. Classes can be either defined as a subclass of the universal rdf:resource if no more information of about the class can be found.

- **Subclasses**, classes can also be defined as subclass of other classes if evidence is found that a class is indeed a subclass of another class. Currently a subclass relationship found by Corporum is based on simple information about terms, that is, CORPORUM-OntoExtract is not able to retrieve subclass-of relationships that are described implicitly.

- **Facts/instances**

  The current implementation of CORPORUM-OntoExtract is defined in such a way that class definitions do not contain properties. As properties of classes are found, they will be defined as properties of an instance of that particular class.

  **Example:**

  `<rdfs:Class rdf:ID="news_service">
  <rdfs:subClassOf rdf:resource="#service"/>
  </rdfs:Class>

  `<news_service rdf:ID="news_service_001">
  <hasSomeProperty>financial</hasSomeProperty>
  </news_service>`

  In the example above, there is a special “news_service” found in the text, which is then defined as a subclass of a class “service”. Furthermore CORPORUM-OntoExtract finds an instance of the subclass “news_service”, which is called “news_service_001” in the example. This instance has a property in the current domain, namely that it is a financial “news_service”. Other instances of concepts could be found, but here context within the text is rather important, and CORPORUM-OntoExtract is not always able to make the right representations.

  **Example:**

  `<commercial_enterprise rdf:ID="#CognIT"/>`

- **Cross-taxonomic relations**

  The last category that is exported by CORPORUM-OntoExtract is the so-called cross-taxonomic relations. 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

---

4 Naturally there are a variety of other ways in which to represent such knowledge. The above decision is purely a design decision that was taken when developing CORPORUM-OntoExtract. Different export formats can be defined when necessity comes up.

---

report on deliverable 6: software tool for ontology extraction - 14 -
using the `<isRelated>` structure.

Example:

```xml
<rdf:Description rdf:about="sophisticated_automatic_support">
   <weaklyRelatedTo rdf:resource="#information"/>
</rdf:Description>
```
5. Technical architecture

CORPORUM-OntoExtract is part of the CORPORUM-OntoBuild software and has been build as a set of four main components and a language specific support component. Later (upon completion of OTK-project deliverable 7) the CORPORUM-OntoWrapper software will be integrated in the CORPORUM-OntoBuild component. The software is implemented using a client/server architecture. Its main components are:

- **CMOntoBuild**: taken care of overall control of the system and co-ordinating all information flows,
- **CMWebHandler**: responisble for collecting all (text-) documents from a specific site,
- **CMCogLib**: analysis texts, extracts information, exports a variety of formats
- **CMLexEn**: language dependent support module for CMCoglib
- **CMWebInteract**: communication component that takes care of all interaction of CORPORUM-OntoExtract with the RDF database. Responsible for querying the RDFDB, as well as submitting final analysis results.
- **DOMhandler**: integrated in CMWebInteract, the OpenXML DOM handler takes care of the interpretation of the results which are returned from the RDF server

![Client/Server based System Architecture of CORPORUM-OntoExtract](image-url)

**Figure 6: Client/Server based System Architecture of CORPORUM-OntoExtract**

Interaction with CMOntoBuild is taken care of by means of simple webserver calls. After invoking CMOntoBuild (see section [3] for an overview of the parameters needed by
CMOntoBuild), the system will automatically retrieve documents and build a “knowledge model” using the retrieved text and already available knowledge from the RDF server.

CORPORUM-OnToExtract then performs the following tasks:

- CMOntoBuild is invoked by the user
- CMWebHandler is invoked by CMOntoBuild,
- CMWebHandler retrieves the domain that is specified from the intra/internet and returns it to CMOntoBuild,
- CMOntoBuild passes texts to the CMCoglib that analyses, interprets and extracts information from these texts, and returns a basic RDF representation to CMOntoBuild,
- CMOntoBuild now analyses the generated RDF and queries the RDF Ontology repository to try to find knowledge that can augment the previously generated RDF,
- When all querying that could be performed is done, and the RDF is augmented, the final RDF ontology for a specific document is sent to the RDF server together with a reference to the original text.

CORPORUM-OnToExtract closes the session and waits for the next call. The information that it put in the RDF DB can now be either queried or loaded into OntoEdit for further analysis and refinement.
6. Programmers Guide

CORPORUM-OntoExtract is meant to require as little user-interaction as possible. For that reason, the system architecture is laid out in such a way that after invoking the initial process no further user-intervention is required. The specified domain is analysed, texts are retrieved, analysed, compared with knowledge in the database, and finally submitted to the knowledge base. CORPORUM-OntoExtract assumes the usage of the RDF server (Sesame) that is developed within the project.

Invoking CORPORUM-OntoExtract

Currently, CORPORUM-OntoExtract is available from the following location:

http://195.70.181.43/ontoextract

The parameters which are needed by CORPORUM-OntoExtract and should be provided by the user of the system (or the software communicating with the webserver):

- aDbQueryLocation
- aDbaddLocation
- aDbName
- aPassword
- aTargetURL
- aRecursionLevel

aDbQueryLocation
The location (URL) of the query engine, including the query string (it uses the GET protocol).
Example:
aDbQueryLocation :=
'http://rdf.cs.vu.nl/a12r/sesame/query?format=xml&query=

aDbaddLocation
The location (URL) to which the final results of the OntoBuild process are to be submitted.
Example:
aDbAddLocation :=
'http://rdf.cs.vu.nl/admin'

aDbName
The name of the database that is to be used. Both querying and submission will be done

\[^5\] Sesame is currently accessible at ‘http://rdf.cs.vu.nl’ and the specifications of CORPORUM-OntoExtract are based on the specifications specified for Sesame in May 2001.

\[^6\] To date is June 2001, in case of inaccessibility of the system, please contact info@cognit.no

\[^7\] operational as from 1.7.2001
to the database specified here.

Example:
ADbName := ‘cognit-test-db’

aPassword
The password belonging to the database name specified in “ADbName”

aTargetURL
The root URL of the (set of) pages that is to be analysed.

Example:
ADbName := ‘http://www.ontoknowledge.org’

ARecursionLevel
The recursion level that is to be used for domain extraction. The recursion level is restricted to “in-domain” pages, i.e. no pages outside of the domain specified in aTargetURL are taken into consideration, even if such a page is found within the specified recursion depth.

Example:
ARecursionLevel := 3
7. Discussion on use of CORPORUM technology in OntoExtract

CORPORUM shows a rather diverse functionality in the currently available version. CORPORUM is not always capable of capturing such intentional knowledge. The main cause for this often is the lack of extensive world knowledge, or CORPORUM aims at capturing contextual knowledge instead of performing deep semantic analysis according to (predefined) world models.

Currently there is much more knowledge and information available from analysis processes than the information that is communicated to the user. Obvious improvements of the system as it is now are visualisations of Central (or core) Concept Areas, allowing for browsing through document sets by using visual maps and extending its linguistic capabilities to deeper understanding of smaller discourses.

Within the OnToKnowledge project, the aim is to help create and later utilise the so-called “semantic web”. CORPORUM-OntoExtract plays to role of an annotation engine that analyses single web pages and augments them with ontologic information. Several problems show up when using information extraction tools in web-based scenarios:

**Content in natural language vs. content in structure**

Whereas CORPORUM takes his pride in being able to capture intention and content from documents without taking into consideration the lay-out and structure of the texts contained in it. However, use-cases performed within the OTK-project show that taking into consideration eventually available knowledge about document types might actually be necessary in cases where documents represent many small knowledge items (cf. contracts, licences, etc.). Whereas CORPORUM can get into difficulties analysing such texts for the sole purpose of information extraction, it will usually be able to deal with these documents when used for comparison of such texts to other texts (i.e. when the neuro-fuzzy comparison engine can enter the scene).

Deliverable 7 (OTK-project deliverable on extracting information from structured texts) will deliver a component which is able to extract knowledge from structured documents, purely based on a document’s lay-out. This component is currently under development and first versions will become available as CORPORUM-OntoWrapper in fall 2001.

**Diversity of web pages (unknown intention)**

The most difficult issue for any text and information retrieval engine is the diversity of documents on the web. There is no control possible whatsoever on type and content of documents returned. Since CORPORUM-OntoExtract expects standard ASCII text, the system needs a component that is robust enough to be able to retrieve all documents from the web, and extract text from documents that are often ill-formed, intermingle HTML code with scripts, use very diverse ways to reach the wanted lay-out and sometimes are written in completely different character sets. Since the intention of a document is not known beforehand, it is often difficult to analyse a text according to the intention of the writer (as explained above, the system has no current knowledge on document types like contracts, news group items, information channels etc.). Based on the experience gained in the case-studies, it became clear that a combination of CORPORUM-OntoExtract with CORPORUM-OntoWrapper might solve a few of these issues (provided that the OntoWrapper architecture enables recognition of document types). Truly closed environments like often encountered in intranets of (larger) companies are also easier to extract information from using both retrieval tools, since these domains are typically dealing within a less complex domain (as the “real-world”).
Representational issues (A-box vs. T-box reasoning)

Besides of technical challenges like the ones mentioned above, there are more philosophical issues to be resolved as well. CORPORTUM-OntoExtract has an internal knowledge representation that is rich in nature. Using this knowledge to export RDF(S) annotations is not always straight-forward. At current the system is restricted to the export of relatively simple class definitions, subclass relations, definition of properties belonging to instances of certain classes and (in many practical settings extremely useful) cross-taxonomic relationships. Most of the CORPORTUM-OntoExtract generated knowledge is TBox assertional knowledge (i.e. frame-based representations of facts and terminological assertions like class-subclass relationships). There is a whole field to be further explored with regard to ABox reasoning that is currently not dealt with yet. From a practical point of view it is not always pragmatic to implement such complex reasoning, not because of technical impossibilities, but simply because of the end-users capabilities to make an effective use of such systems.

Domain specificity of extracted knowledge

Since the ontologies are extracted from specified, restricted domains, the extracted information is expected to by rather domain specific. Since CORPORTUM was originally not intended to make use of previous knowledge whatsoever, the system regards each and every document as a single, independent piece of prose. Of course it is possible to capture discourse in a larger set of documents by simply changing the granularity of an analysis (i.e. take a set of documents together in an analysis, or the opposite: pealing a document so that its smaller units can be analysed seperatable). This domain-specificity has positive and negative sides. Positive, while many of the searches will also be rather domain specific, and knowledge about cross-taxonomic relations might come in very handy in such cases. Negative, while it is sometimes the case that one would like to build up domain independent knowledge bases (cf. CYC). This is still possible, but the knowledge retrieved with this aim in mind will have a much “simpler” character, i.e. is much more restricted to basic class/subclass relationships and the like.

Virtues of the CORPORTUM-OntoExtract approach

The CORPORTUM-OntoExtract approach shows that usage of automated text – extraction represents an effort that is often worthwhile. It also shows that it is not a simple task to perform without any type of pre-existing knowledge about document types and/or some layout information on documents. However, taking into consideration that CORPORTUM-OntoExtract, with its recently implemented ability to reason with knowledge bases, it becomes clear that such approaches have a large potential to enhance usability of formal knowledge representations for end-users and to decrease initial efforts when defining an ontology in new domains.

CORPORTUM-OntoExtracts functionality is directed towards generation of initial ontologies when engineering domains, help refining existing domains and on top of that, is able to grasp non-typical, but domain-specific knowledge items like the discussed cross-taxonomies. The broad-span of possibilities in using information extraction has as drawback that it easily becomes difficult to manage expectations when defining applications for the OTK case-studies. The extension of the target document set in the OTK project (intranet/internet documents), in all their languages, appearances and formats, together with the fact that CORPORTUM-OntoExtract does not yet utilize such lay-out based information, required large amounts of extensions and reengineering of

---

software in order to deal with special scenarios (cf. discussion above and the short history-listing at the introduction).

Given the very valuable experiences and results so far, several lessons can be learned:

- due to the high complexity of documents \((\text{languages*formats*document-types})\), proof-of-concept demonstrations as in the OTK project are in need of clearly defined use-cases at the very beginning of the project, such as to restrict the “play-ground” as much as possible,
- followed by a good (different) set of use-cases at the end of the software development, such that results can be validated,
- the project made us aware that very rich representation languages and complex solutions are not necessarily justified by real-world scenarios, and may thus obstruct from reaching final, (end-) user satisfaction in user-friendliness and usability of the system,
- systems like CORPORUM-\textit{OntoExtract} are able to do much, but not everything. It is important (as mentioned before) to manage expectations before exposing users to automated approaches. This holds, however, for many automated processes that find application out in the “real-world”,
- virtues of systems like CORPORUM-\textit{OntoExtract} are the
  - \textit{dynamicy} of the analysis, i.e. ease of use in dynamic environments,
  - usefulness for \textit{automated ontology proposal} in new domains, such that typical users are really refrained from many of the burdens of acquisition of new knowledge,
  - its ability to offer new ways of \textit{navigating} knowledge bases and documents sets by \textit{visualisation} of contents and by means of semantic-based, graphic structures,
  - \textit{extraction} of \textit{content-based meta-data} from documents (like the most important concepts, semantic structures (graphically represented as well as in other formats), summary extraction from target texts and more of the pieces of knowledge about documents (cf. dublin core based meta data: \url{http://purl.og/dc})),
  - ability to offer domain-specific information as \textit{related-keywords} (used for search scenarios).

From a technology point of view there are several ways ahead that could be pursuit, and the best follow-up for the work done so far would be to do as proposed, and follow up with a new, independent use-case in which the OntoBuild software can be tested and enhanced. Such use-cases should then be well-defined beforehand, with objectives in line with project goals. This would allow for a close follow up and a thorough definition of the further improvements that could be made on the technological side.

CORPORUM-\textit{OntoBuild}, as it exists of CORPORUM-\textit{OntoExtract} and CORPORUM-\textit{OntoWapper} is a very promising tool for ontology engineering and extraction. While meant as a back-end solution (cf \textbf{Figure 3}), it relies on the availability of the powerful tools developed by the other project partners for user-interaction (such as the ontology editor OntoEdit, the semantic structure visualisation tool CCAviewer, and inclusion in GUIs like WebFerret and the like).