Migrating Cornetto Lexicon to New XML Database Engine

Aleš Horák and **Adam Rambousek**
NLP Center
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
{hales,xrambous}@fi.muni.cz
http://deb.fi.muni.cz

Abstract

The original Cornetto project started to develop a new complex-structured lexicon for the Dutch language. The lexicon building process works with information from two current electronic dictionaries – the Referentie Bestand Nederlands (RBN), which contains FrameNet-like structures, and the Dutch wordnet (DWN) with the usual wordnet structures. The resulting Cornetto lexicon is stored in a system called Cornetto database, which is built over the Dictionary Editor and Browser platform.

In this paper, we describe a transition of the Cornetto database system to a new database backend based on large set of tests that were run on four selected (out of twenty) available XML database systems. We present the technical details of the Cornetto editing process and the results before and after the database transition.

1 Introduction

The Cornetto database system (Horák et al., 2009) is based on the DEB (Dictionary Editor and Browser) development platform (Horák et al., 2006a). The general purpose of DEB is to offer common client-server functionality for different types of lexicographic resources, including dictionaries, wordnet semantic networks, classical ontologies or lexical databases.

The Cornetto lexico-semantic database[1] combines Wordnet with FrameNet-like information (Fillmore et al., 2004) for the Dutch language. During the lexicon building process the Dutch Wordnet (Vossen, 1998) and the Referentie Bestand Nederlands (Maks et al., 1999) are the most consulted external language resources. The Dutch Wordnet (DWN) is similar to the Princeton Wordnet for English, and the Referentie Bestand Nederlands (RBN) includes frame-like information as in FrameNet plus additional information on the combinatoric behaviour of words in a particular meaning. The combination of the two lexical resources results in a rich linguistic database that improves natural language processing (NLP) technologies, such as word sense-disambiguation, and language-generation systems. In addition to merging the Wordnet and FrameNet-like information, the database is also mapped to a formal ontology to provide a more solid semantic backbone.

Both DWN and RBN are semantically based lexical resources. RBN uses a traditional structure of form-meaning pairs, so-called Lexical Units (Cruse, 1986). Lexical Units contain all the necessary linguistic knowledge that is needed to properly use the word in a language. Lexical Units in Cornetto are organized into Synsets (synonymical sets). For Cornetto, the Synsets follow the concept of near synonymy from EuroWordNet (Vossen, 1998).

The DEB platform is currently employed in more than 15 national and international projects, e.g. the KYOTO EU project (Vossen, 2008) or a five-year project of the New Encyclopaedia of the Czech Language. Two projects with nearly thousand active users all over the world are DEBDict and DEBVisDic (Horák et al., 2006b). DEBDict as a general dictionary browser offers access to many dictionaries and lexical resources in several languages, and DEBVisDic, wordnet editor and browser has already been used to build more than fifteen wordnets in different languages from all over the world. The freely available DEB server is currently installed in ten institutions from three continents, where it is used mostly as an XML-based data storage, presentation and manipulation system.

[1] see Figure 1

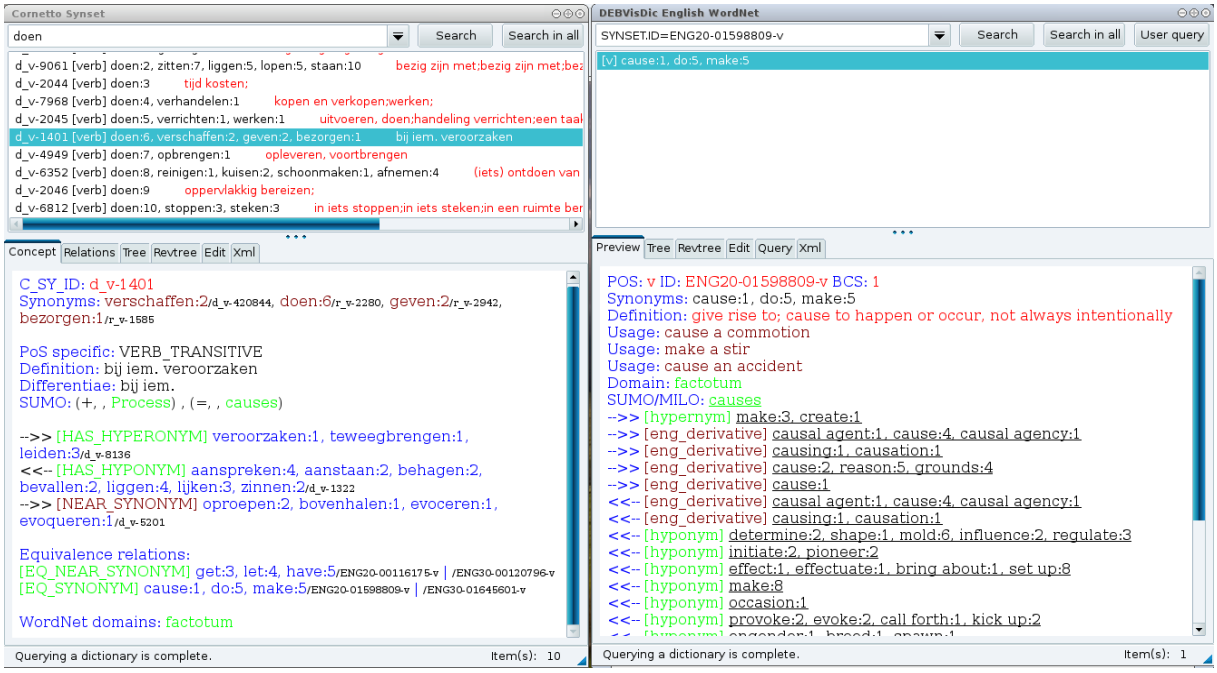

Figure 1: An example of the Cornetto editing interface.

In the following text, we first describe the needs of a DEB storage backend and the selection process between possible XML databases used as a storage backend in the DEB platform, and then we present the details of changing the backend within the Cornetto database system.

2 The DEB Database Backend

With the current deployment of the DEB platform, several complex tasks have appeared with growing needs for the employed database storage. To resolve such problems and to offer reserves in the speed of the DEB database backend (i.e. the XML storage engine used for saving the data processed by the system), available native XML database systems were analyzed and compared, with the resulting recommendation of the best performance for knowledge and ontology systems.

The DEB (Dictionary Editor and Browser[2]) is an open-source software platform for the development of applications for viewing, creating, editing and authoring of electronic and printed dictionaries. The platform is based on the client-server architecture. Most of the functionality is provided by the server side, and the client side offers (computationally simple) graphical interfaces to users. The client applications communicate with the server using the standard web HTTP protocol.

The server part is built from small, reusable parts, called servlets, which allow a modular composition of all services. Each servlet provides different functionality such as database access, dictionary search, morphological analysis or a connection to corpora.

The overall design of the DEB platform focusses on modularity. The data stored in a DEB server can use any kind of structural database and combine the results in answers to user queries without the need to use specific query languages for each data source. The main data storage is currently provided by the Oracle Berkeley DB XML (Chaudhri et al., 2003). However, it is possible to switch to another database backend easily, without any changes to the client parts of the applications.

Database systems working with XML data (both native XML databases and XML enabled relational databases) are already widespread and used in many areas. Their performance was benchmarked by many projects using several benchmarks. However, conclusions of previous publications (Böhme and Rahm, 2008; Nambiar et al., 2002; Lu et al., 2005) do not provide one definitive answer as for the choice of the best XML database. XML-enabled and native XML

[2] `http://deb.fi.muni.cz/`, see e.g. (Horák et al., 2006a)

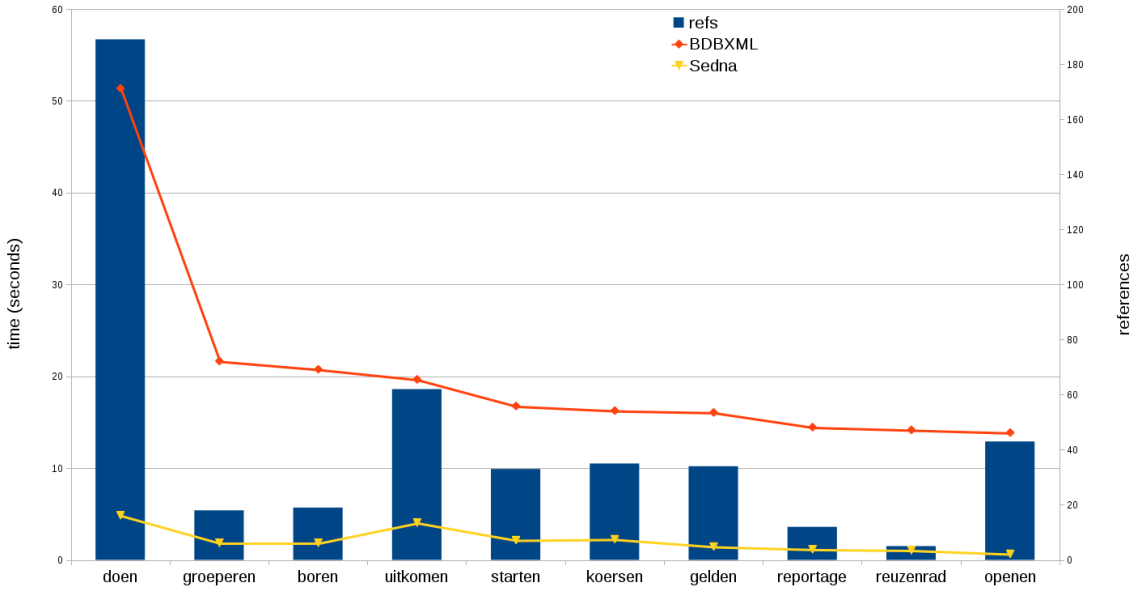

Figure 2: Graph comparison of the most time-consuming Cornetto synset queries in Oracle Berkeley DB XML and Sedna

databases. Generally, the results suggest that different XML benchmarks can show different weak and strong points of each database systems. When comparing the two classes of XML databases, i.e. relational databases with XML support and native XML databases, we can see that XML enabled relational databases process data manipulation queries more efficiently, and native XML databases are faster in navigational queries which rely on the document structure.

We have thus performed extensive testing with the selection of 4 from more than 20 native XML or XML-enabled databases. The selection was driven by the requirements of effective XML processing, an open source licence, active development and support of XML-related standards. The tested databases were:

- *eXist*. The eXist database (Meier and others, 2003) is developed in Java and licensed under LGPL, active since 2000 and currently developed by the group of independent developers. The database supports XQuery, XSLT and XUpdate standards for data manipulation, and DTD, XML Schema, RelaxNG and Schematron for validation.

  Users are able to specify structural indexes (element and attribute structure in documents), range indexes (*contains*, *starts-with* and similar functions), and full-text indexes (Apache Lucene (Foundation, 2006) is used for full-text indexing).

- *MonetDB*. The MonetDB database (Boncz et al., 2006) is developed by CWI Amsterdam and several Linux distributions and MS Windows are officially supported. The database is licensed under a customised Mozilla Public License.

  The main goal of MonetDB is to design a database for processing very large (in GBs) XML documents. The default database settings are optimized for document reading, offering indexing for quick query execution, although the indexes have to be rebuilt after every document update. Another option is an optimization for document updating, with simpler index structure and slower performance for search queries.

  The database supports XQuery and partly XQuery Update (W3C, 2009). It is also possible to use MonetDB internal query language. Indexing is automatized, without the possibility to alter settings in any way. The PF/Tijah (Hiemstra et al., 2006) text search system is utilized for full-text searching.

Table 1: Complex synset searches in Oracle Berkeley DB XML and Sedna (in seconds)

The slowest queries

in Oracle Berkeley DB XML

	# of refs	DBXML	Sedna
doen	189	51.4	4.9
groeperen	18	21.7	1.9
boren	19	20.8	1.9
uitkomen	62	19.7	4.1
starten	33	16.8	2.2
koersen	35	16.3	2.3
gelden	34	16.1	1.5
reportage	12	14.5	1.2
reuzenrad	5	14.2	1.1
openen	43	13.9	0.7

in Sedna

	# of refs	Sedna
proper	31	8.3
omlaaglopen	12	7.9
gaan	216	7.0
houden	143	7.0
zin	145	6.8
stuk	168	6.8
oppassen	44	6.7
hol	58	6.6
hand	43	6.5
slag	117	6.3

The most frequent searches

in Oracle Berkeley DB XML

	# of searches	# of refs	DB XML	Sedna
god	594	24	2.7	1.8
artikel	550	53	4.7	1.2
jacht	337	57	0.9	0.5
schijf	265	36	1.3	2.2
officier	260	75	0.9	2.0
arm	252	61	1.6	3.1
college	214	65	1.1	2.1
academie	206	30	0.8	1.5
rijkdom	197	38	0.9	2.0
president	194	9	0.5	0.5

in Sedna

	# of searches	# of refs	Sedna
artikel	128	53	1.2
aardig	127	62	1.4
intreden	120	17	0.9
gewoonte	113	26	1.1
komen	77	146	0.6
krijgen	73	31	1.1
vallen	64	115	1.4
slaan	62	73	0.5
inbrengen	60	21	0.9
heer	58	138	4

- *Sedna*. The Sedna database system (Fomichev et al., 2006) is developed by the Russian Academy of Sciences, and released under Apache Licence. Official packages for Windows, Linux, MacOS, FreeBSD and Solaris are available.

  The database supports XQuery and custom variant of XQuery Update for data manipulation, and XML Schema for validation. Indexes have to be set manually and a special function must be used in the query to access the index. Full-text indexing is provided by external commercial tool dtSearch. Sedna offers several extensions, such as the capability of an SQL connection from XQuery, or the trigger support.

- *Oracle Berkeley DB XML*. Oracle Berkeley DB XML (Chaudhri et al., 2003) was created as an extension of Berkeley DB. The database is now developed by Oracle and released for Windows and Linux. Users can choose between open source and commercial licences.

  The underlying structure is still based on Berkeley DB and each document container is stored in a single file. The database supports XQuery and part of XQuery Update. The document validation according to a supplied XML Schema is checked only during document storage, later changes can render the document invalid. Users have to specify indexes manually, full-text indexing is also supported, although it is not possible to use regular expressions in queries.

Because of the special focus on dictionary writing systems, we ran different test suites designated to both "raw speed" of the database and to specific requirements of knowledge and ontology systems. According to the results of the tests (see (Bukatovič et al., 2010) for the details of the tests re-

Table 2: Lexical unit search in Oracle Berkeley DB XML and Sedna (in seconds)

The slowest queries

in Oracle Berkeley DB XML

	DBXML	Sedna
uitkomen	25.3	0.5
vervallen	22.9	0.4
steken	21.2	1.0
opstaan	20.9	0.8
trekken	20.7	0.9
opzetten	20.7	0.5
sterven	20.6	0.8
treden	20.5	0.5
plaatsen	20.5	0.4
springen	20.5	0.3

in Sedna

	Sedna
sterk	9.1
doteren	6.6
prioriteit	4.5
gelijk	4.3
aanvaarding	4.2
zwaar	4.2
onbeschaafd	3.9
vurig	3.7
open	3.3
onmogelijk	3.3

The most frequent searches

in Oracle Berkeley DB XML

	# of searches	DBXML	Sedna
god	560	1.8	0.6
artikel	533	3.2	0.2
eindy	183	1.0	1.0
gewoonte	152	5.4	0.3
vis	143	0.4	0.6
gang	127	1.4	1.1
richtlijn	123	0.2	0.7
beeld	114	1.4	0.3
huis	102	1.3	0.7
goed	101	0.9	1.4

in Sedna

	# of searches	Sedna
schilderen	1173	0.1
draaien	520	0.8
slaan	453	0.8
gebruiken	413	0.1
keren	349	0.5
branden	343	0.5
verliezen	317	0.2
blazen	306	0.2
hechten	294	0.3
steunen	279	0.2

sults), none of the available native XML databases can supersede the others for all kinds of operations needed for knowledge and ontology storage and manipulation. Berkeley DB XML cannot efficiently solve the queries involving multiple nodes and full-text queries. The eXist database contains the Lucene module for text search and supports many XML standards, so it can be recommended for deployment where these features are more important than the database performance. On the other hand the MonetDB database can be, according to its specific architecture, conveniently used for when working with very large amounts of XML data. For middle-size data collections, the Sedna database can provide the same performance as MonetDB, while offering richer set of features. The potential drawbacks of Sedna are the need to use special queries for the defined data indexes and the use of commercial tool for optimized full-text queries. However, the full-text queries without this optimization are already comparably fast.

During the testing of both database engines within the DEB platform, we found out that the MonetDB programming interface for the Ruby language used in the DEB platform is not stable enough and not developed actively at the moment. Because of that, MonetDB is not ready yet to be included in the platform. Fortunately, Ruby interface for Sedna is stable and maintained and better suited for DEB platform. That is why Sedna was chosen for the DEB database backend transition for one of the very active projects with tens of concurrent editors, the Cornetto project.

3 Cornetto Backend Transition and Evaluation

The Cornetto data are split to four main databases (lexical units, synsets, ontology terms, and Cornetto identifiers), plus two databases with more detailed linguistic information and the English wordnet database. Usually, user queries combine data from several databases and different informa-

tion are merged to form complex entries.

Database sizes and number of links between them grew over time, currently the main databases contain:

- lexical units – 117 967 entries;
- synsets – 70 507 entries;
- identifiers – 106 305 entries;
- ontology – 3 080 entries.

As the database size and complexity increased, search queries were getting slower even with indexes set up to speed up the most common queries in the current database backend Oracle Berkeley DB XML. Because the user experience was an issue, Cornetto was chosen as the first project to migrate to new database backend.

The database module of DEB platform was exchanged from Oracle Berkeley DB XML to Sedna and no explicit indexes were set up. Even without the indexes, the improvement in search queries was considerable. We have analyzed the database logs for both implementations. The speed was measured in the same conditions (machine, workload, number of users and their interaction with the software). Logs for Berkeley DB XML cover the usage from April 2010 to March 2011. Logs for Sedna cover the usage from March 2011 to September 2011. All presented times are averages of all the searches for the respective entry words and represent just the time needed to run the query in database and to prepare the result list, not the time of the client-server communication over the Internet.

Tables 1 present query times for complex database queries on synsets, with references to lexical units, English Wordnet and ontology. The tables show top 10 slowest queries and 10 most frequent queries in both DB XML and Sedna. For the DB XML words the corresponding times in Sedna are also displayed for comparison. It is clear that the search times are affected by the number of references each synset contains. For example, there are 10 synsets for the word *doen*, with 146 references to other synsets, 28 references to lexical units and 15 references to the English Wordnet. Similar

Tables 2 show statistics for lexical units. The tables present top 10 slowest queries and 10 most frequent queries in Oracle Berkeley DB XML and Sedna.

Table 3: Average time to execute queries in Cornetto database (in seconds)

	Berkeley DB XML	Sedna
Synset	6.1	2.4
Lexical units	2.7	0.9

Finally, Table 3 summarizes average search times (in seconds) for Berkeley DB XML and Sedna databases.

Considering the results of XMark and the custom knowledge and ontology benchmark, the MonetDB and the Sedna databases represent a good choice for the knowledge and ontology systems. MonetDB offers very good performance for very large documents, on the other hand, Sedna provides much more advanced features. Sedna supports index usage only with its own special functions, so the queries need to be changed accordingly.

According to the experiences and evaluation of the Sedna database deployment for Cornetto, performance improvement is significant and enhances user experience. Database performance will be enhanced even more by utilizing specific indexes to speed up the query execution. Based on this pilot transition, the Sedna database will be included in the DEB platform and will gradually replace Oracle Berkeley DB XML as the main database backend.

Based on the preliminary tests, the performance is also greatly affected by the XML parser library included in the DEB platform. Currently, REXML (Russell, 2008) parser is used for parsing each entry during search. However, other parser libraries can improve the speed significantly. Evaluation of available XML parsers will be carried out to find the best option for DEB platform.

4 Conclusions

In the paper, we have presented a successful adaptation of the Cornetto database system to a new XML database backend. The Cornetto system is based on the Dictionary Editor and Browser (DEB) development platform that is designed to provide modular framework for dictionary writing systems. The structure of the Cornetto system and the DEB platform thus allows to change the underlying data storage without the need to make substantial changes to the system as a whole.

We have described the details of the database

selection process and the evaluation of the database transition. The presented results, as well as positive user experience, clearly justify that the new database is very well suited to the kind of operations needed for the development of the Cornetto Lexicon as a new complex lexico-semantic language resource.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and LINDAT-Clarin project LM2010013, by the Czech Science Foundation under the projects P401/10/0792 and 102/09/1842, and by the Ministry of the Interior of CR within the project VF20102014003.

References

T. Böhme and E. Rahm. 2008. Multi-user evaluation of XML data management systems with XMach-1. *Efficiency and Effectiveness of XML Tools and Techniques and Data Integration over the Web*, pages 148–159.

P. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner. 2006. MonetDB/XQuery: a fast XQuery processor powered by a relational engine. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, page 490. ACM.

M. Bukatovič, A. Horák, and A. Rambousek. 2010. Which XML storage for knowledge and ontology systems? In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 432–441. Springer.

Akmal B. Chaudhri, Awais Rashid, and Roberto Zicari, editors. 2003. *XML Data Management: Native XML and XML-Enabled Database Systems*. Addison Wesley Professional.

D.A. Cruse. 1986. *Lexical semantics*. Cambridge, England: University Press.

C.J. Fillmore, C.F. Baker, and H. Sato. 2004. Framenet as a 'net'. In *Proceedings of Language Resources and Evaluation Conference (LREC 04)*, volume vol. 4, 1091-1094, Lisbon. ELRA.

A. Fomichev, M. Grinev, and S. Kuznetsov. 2006. Sedna: A Native XML DBMS. *Lecture Notes in Computer Science*, 3831:272.

Apache Software Foundation. 2006. Apache Lucene. `http://lucene.apache.org`.

D. Hiemstra, H. Rode, R. van Os, and J. Flokstra. 2006. PF/Tijah: text search in an XML database system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR)*, pages 12–17.

A. Horák, K. Pala, A. Rambousek, and M. Povolný. 2006a. DEBVisDic–First Version of New Client-Server Wordnet Browsing and Editing Tool. In *Proceedings of the Third International WordNet Conference (GWC 2006)*, pages 325–328, Jeju Island, South Korea.

Aleš Horák, Karel Pala, Adam Rambousek, and Pavel Rychlý. 2006b. New clients for dictionary writing on the DEB platform. In *DWS 2006: Proceedings of the Fourth International Workshop on Dictionary Writings Systems*, pages 17–23, Italy. Lexical Computing Ltd., U.K.

A. Horák, I. Maks, A. Rambousek, R. Segers, H. van der Vliet, and P. Vossen. 2009. Cornetto Tools and Methodology for Interlinking Lexical Units, Synsets and Ontology. In *Current Issues in Unity and Diversity of Languages*, pages 2695–2713, Seoul, Republic of Korea. The Linguistic Society of Korea.

H. Lu, J.X. Yu, G. Wang, S. Zheng, H. Jiang, G. Yu, and A. Zhou. 2005. What makes the differences: benchmarking XML database implementations. *ACM Transactions on Internet Technology (TOIT)*, 5(1):154–194.

I. Maks, W. Martin, and H. de Meerseman, 1999. *RBN Manual*.

W. Meier et al. 2003. eXist: An open source native XML database. *Lecture Notes in Computer Science*, pages 169–183.

U. Nambiar, Z. Lacroix, S. Bressan, M. Lee, and Y. Li. 2002. Efficient XML data management: an analysis. *E-Commerce and Web Technologies*, pages 261–266.

Sean Russell. 2008. Rexml. `http://www.germane-software.com/software/rexml/`.

P. Vossen, editor. 1998. *EuroWordNet: a multilingual database with lexical semantic networks for European Languages*. Kluwer.

Piek Vossen. 2008. KYOTO Project (ICT-211423), Knowledge Yielding Ontologies for Transition-based Organization. `http://www.kyoto-project.eu/`.

W3C. 2009. XQuery Update Facility 1.0. (`http://www.w3.org/TR/xquery-update-10`).