# Using DEB Services for Knowledge Representation within the KYOTO Project

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

## Abstract

Within the EuroWordNet projects the national wordnets were interlinked with a interlingual index, ILI. In the subsequent Balkanet project, mostly pragmatic decisions stood as the reason for choosing the English as the pivot language instead of ILI. The Global WordNet Grid, as an approach, and the KYOTO project, as an instantiation of this approach are shifting the idea of the pivot from lexical meanings to real semantics - the languages will be interlinked through the shared ontology.

In this paper, we describe the design and implementation of the KYOTO database, which is based on the Dictionary Editor and Browser (DEB) platform. The main ideas and assets of the platform are presented and the necessary additions and adaptation for the needs of the KYOTO project are depicted.

## 1 Introduction

The wordnet semantic networks, regarding the Princeton WordNet (Fellbaum, 1998) as well as its national derivatives in more than fifty languages,[1] have been already used in many projects of intelligent text processing. The main benefits of wordnets are the hypero-hyponymic hierarchy and its translatability, i.e. the fact that most of the national wordnets are linked to the English one as a pivot. Further on, we will show the insides of the database part of a project that moves this pivot to the semantic part, i.e. tries to "replace" the English pivot with a shared ontology.

In the following text, we describe the KYOTO project (Vossen, 2008), which aims at a favourable application of the WordNet like ontologies in the multilingual form (denoted as the *Global WordNet Grid*) and a shared common ontology corresponding to the level of the *Suggested Upper Merged Ontology* (SUMO) as the central knowledge backbone. The ontology here serves as a meaning description tool for all the terms and facts that are extracted, compared and stored within the KYOTO system.

## 2 The KYOTO Project – WordNets, Ontologies and Text

WordNet semantic networks allow to express basic language relations[2] in a multigraph structure directly processable by computer systems.[3] However, description of more complicated structured knowledge, e.g. relations with more than one participants, cannot be encoded in a WordNet-standard way that could be further analysed and used by computers.

In the KYOTO system, this (potential) drawback of WordNet is solved by the idea of extending the WordNet into a *Global WordNet Grid* of multiple languages with a shared ontology in the center. Interlinking of national wordnets is not a new idea, it was introduced e.g. in the EuroWordNet (Vossen, 1998) and Balkanet (Christodoulakis, 2004) projects. In these projects the "pivot," i.e. the *interlingual index*, was represented directly by the English WordNet. This solution had several advantages and several disadvantages. From the point of view of the knowledge analysis, the biggest disadvantage was that the lexical knowledge structure was "hidden" in the English lexicon without the possibility to really extract it for the purpose of further computer processing. The shared ontology provides a way of adding structural semantics to the interlingual links.

The KYOTO project will incorporate and expand the Global WordNet Grid and will be the first system that exploits the benefits of storing the definitions of terms and facts in a computer processable logical system using the Grid's shared ontology.

[1] see `http://www.globalwordnet.org/` for information about particular national wordnets

[2] hyperonymy/hyponymy, synonymy/antonymy, holonymy/meronymy, etc.

[3] deriving sets of similar objects, classes of more general objects or objects with opposite meaning

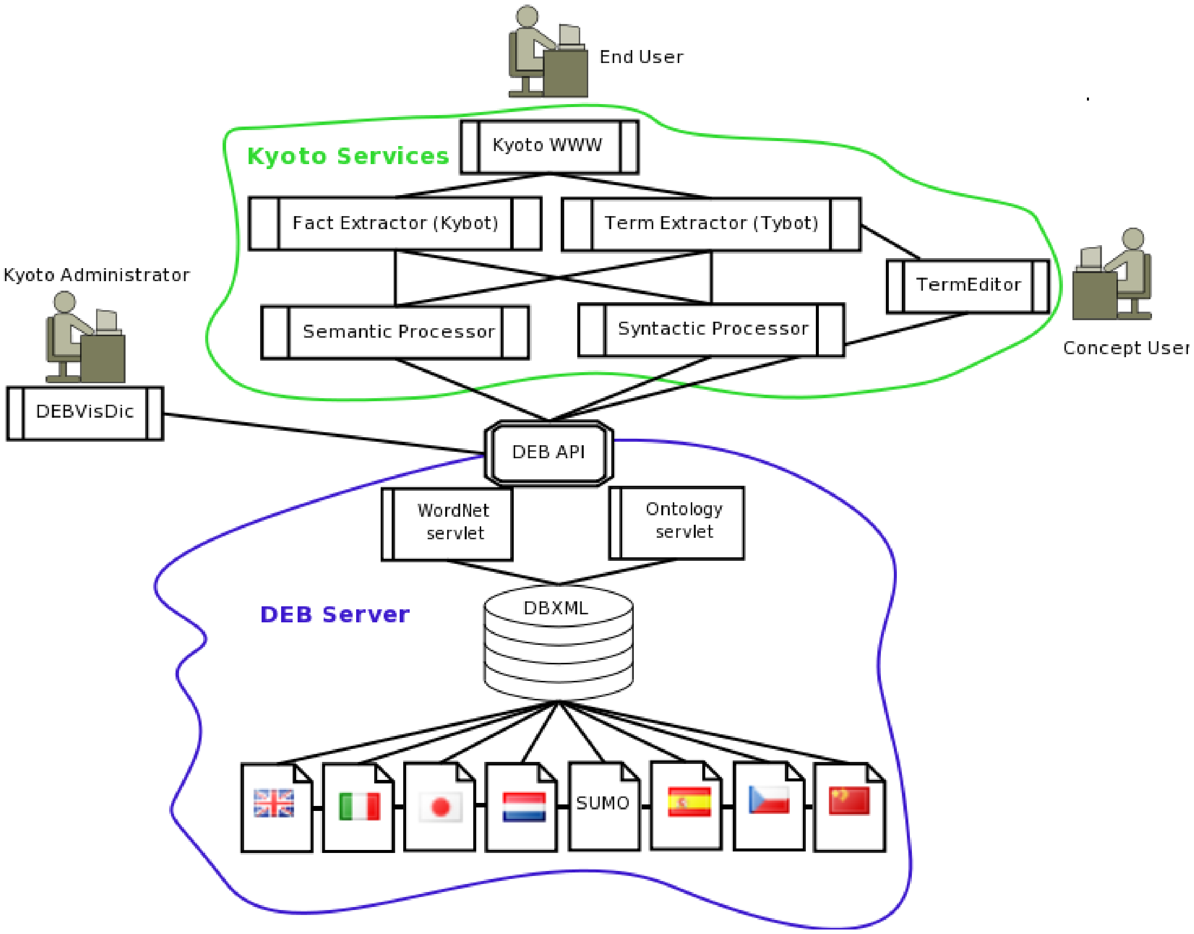


Figure 1: The schema of the KYOTO database within the KYOTO system.

## 3 The KYOTO Databases

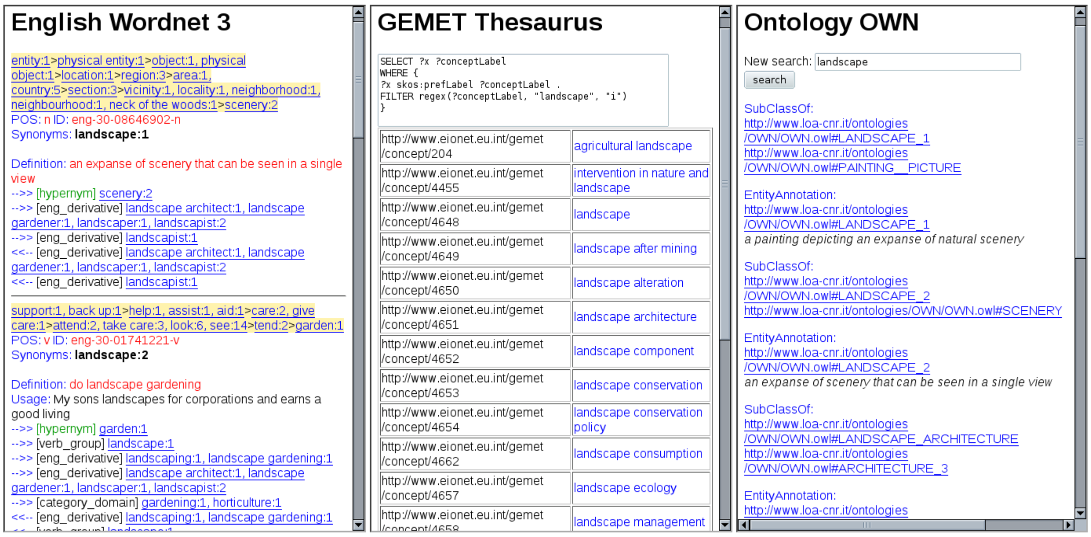
The KYOTO database is built over the DEBVisDic application with the DEB server either set up at one central locality or it can be set up by several KYOTO partners. The DEB platform provides important backgrounds for the KYOTO project universal features (see Figure 1).

### 3.1 The DEB Architecture

The Dictionary Editor and Browser (DEB) platform (Horák et al., 2006; Horák and Rambousek, 2007; Horák et al., 2008) has been developed as a general framework for fast development of wide range of dictionary writing applications. The DEB platform provides several very important foundations that are common to most of the intended dictionary systems.

These foundational features include:

- strict client-server architecture with communication based on standard HTTP(s) protocol including authentication.
- the communication between the server and the client is based on predefined Application Programming Interface (API) and works with data in the XML form. The actual storage system (denoted as “storage backend”) is hidden for the user. Thus it is possible to replace the backend and add new backend as the request arises. For instance, in the following text we describe adopting the OpenLink Virtuoso database as a new DEB backend for its SPARQL data query language abilities.
- the standard DEB clients (DEBDict, DEBVisDic, DEBTerm, PRALED, ...) for the data presentation and manipulation use the Mozilla Extensions (Oeschger and others, 2002), which allow a separation of the graphical interface from the application logic.
- data checking and presentation are provided by means of XML standards such as XML Schema or XSLT.



English Wordnet 3

entity:1>physical entity:1>object:1, physical object:1>location:1>region:3>area:1, country:5>section:3>vicinity:1, locality:1, neighborhood:1, neighbourhood:1, neck of the woods:1>scenery:2
POS: n ID: eng-30-08646902-n
Synonyms: **landscape:1**

Definition: an expanse of scenery that can be seen in a single view
-->> [hypernym] scenery:2
-->> [eng_derivative] landscape architect:1, landscape gardener:1, landscaper:1, landscapist:2
-->> [eng_derivative] landscapist:1
<<-- [eng_derivative] landscape architect:1, landscape gardener:1, landscaper:1, landscapist:2
<<-- [eng_derivative] landscapist:1

support:1, back up:1>help:1, assist:1, aid:1>care:2, give care:1>attend:2, take care:3, look:6, see:14>tend:2>garden:1
POS: v ID: eng-30-01741221-v
Synonyms: **landscape:2**

Definition: do landscape gardening
Usage: My sons landscapes for corporations and earns a good living
-->> [hypernym] garden:1
-->> [verb_group] landscape:1
-->> [eng_derivative] landscaping:1, landscape gardening:1
-->> [eng_derivative] landscape architect:1, landscape gardener:1, landscaper:1, landscapist:2
-->> [category_domain] gardening:1, horticulture:1
<<-- [eng_derivative] landscaping:1, landscape gardening:1
<<-- [verb_group] landscape:1

GEMET Thesaurus

```
SELECT ?x ?conceptLabel
WHERE {
?x skos:prefLabel ?conceptLabel .
FILTER regex(?conceptLabel, "landscape", "i")
}
```

| | |
|---|---|
| http://www.eionet.eu.int/gemet/concept/204 | agricultural landscape |
| http://www.eionet.eu.int/gemet/concept/4455 | intervention in nature and landscape |
| http://www.eionet.eu.int/gemet/concept/4648 | landscape |
| http://www.eionet.eu.int/gemet/concept/4649 | landscape after mining |
| http://www.eionet.eu.int/gemet/concept/4650 | landscape alteration |
| http://www.eionet.eu.int/gemet/concept/4651 | landscape architecture |
| http://www.eionet.eu.int/gemet/concept/4652 | landscape component |
| http://www.eionet.eu.int/gemet/concept/4653 | landscape conservation |
| http://www.eionet.eu.int/gemet/concept/4654 | landscape conservation policy |
| http://www.eionet.eu.int/gemet/concept/4662 | landscape consumption |
| http://www.eionet.eu.int/gemet/concept/4657 | landscape ecology |
| http://www.eionet.eu.int/gemet/concept/4658 | landscape management |

Ontology OWN

New search: landscape
search

SubClassOf:
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#LANDSCAPE_1
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#PAINTING__PICTURE

EntityAnnotation:
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#LANDSCAPE_1
*a painting depicting an expanse of natural scenery*

SubClassOf:
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#LANDSCAPE_2
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#SCENERY

EntityAnnotation:
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#LANDSCAPE_2
*an expanse of scenery that can be seen in a single view*

SubClassOf:
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#LANDSCAPE_ARCHITECTURE
http://www.loa-cnr.it/ontologies/OWN/OWN.owl#ARCHITECTURE_3

EntityAnnotation:
http://www.loa-cnr.it/ontologies

Figure 2: Linking wordnets, thesauri and ontologies within KYOTO database.

Query: `https://`*`server_name`*`/wneng30?action=runQuery&outtype=editor`
`&displayonly=POS&query=eng-30-02084071-n`
Result: `{"SYNSET":{"ID":{"$":"eng-30-02084071-n"},"POS":{"$":"n"}}}`

Query: `https://`*`server_name`*`/wneng30?action=save&id=eng-30-02084071-n`
`&data=New definition&saveonly=DEF`
Result: *Synset definition is changed.*

Table 1: API call for obtaining the Part-of-Speech and changing the synset definition instead of full synset information, useful in lightweight clients.

## 4 New DEB Features within the KYOTO Project

### 4.1 Synset API Calls Granularity

Standard DEBVisDic API supports loading and saving synsets including the complete synset data. However, modern AJAX-like[4] lightweight applications frequently need to change just specific parts of the whole synset structure. Supporting this requirement, the client applications that access the wordnet data by means of the DEB application programming interface (API) do not need to get or save all the data at once and can parse less data faster. For these reasons, the API was extended with arguments to read or write specific parts of the synset. Example of such “micro” read and write calls are presented in Table 1.

### 4.2 Translate Synsets

One of the main advantages of wordnets is their multilinguality, i.e. the design of interlingual index used as a pivot between several national wordnets. Currently, the most commonly used pivot is the English wordnet due to its size, completeness and good maintenance. The link to the pivot is encoded either by assigning the English wordnet synset ID directly to the national synset, or by means of *external relations* (ELR) as specific “pointers” to other synsets outside the actual dictionary.

A common operation in multilingual projects is thus “translating” a word (in all synsets) to another language by means of the selected pivot. Since the mapping from national wordnets to English wordnet is not always unambiguous, each synset can point to one or more synsets in common wordnet. The DEB API call thus provides all possible synsets in the target language.

For instance, when we take the word *bosque* (forest) in Spanish, we can find this word in two synsets *bosque:1* and *bosque:2* that are linked to *forest:1, wood:2, woods:1* and *forest:2, woodland:1, timberland:1, timber:4* in the English

[4] *Asynchronous JavaScript and XML*, see (Rosenfeld and Morville, 1998)

```
Query:  https://server_name/wnspa?action=translate&query=bosque
        &target=wnjpn
Result: { ”translated”: [{
          ”elr”: [ ”eng-30-09284015-n” ],
          ”value”: ”jpn-09-09284015-n”,
          ”label”: ”[n] 森林地:, 森:, 樹林:, 森林地帯:, ラフォーレ:, 林:, 森林:, 林地:”
        },{
          ”elr”: [ ”eng-30-08438533-n” ],
          ”value”: ”jpn-09-08438533-n”,
          ”label”: ”[n] 森:, 樹林:, 林:, 森林:, 林地:”
        }]}
```

Table 2: Translating API call – translate *bosque* (forest) from Spanish to Japanese

```
Query:  https://server_name/wnen30_d?action=nextSense
        &literal=Dusky+gopher+frog&callback=jsonp1253866807430
Result without the ‘callback’ parameter:
        {"literal":"Dusky gopher frog","recommended_sense":3}
JSONP compatible response:
        jsonp1253866807430({"literal":"Dusky gopher frog",
                            "recommended_sense":3});
```

Table 3: Variant of API calls supporting the JSONP protocol with callbacks.

wordnet. Through the obtained English IDs, we can enlist direct equivalents e.g. in the Japanese wordnet – see the example in Table 2.

### 4.3 Links between Wordnets and Ontologies

All wordnets in the KYOTO database are interlinked using the common central ontology. The solution is not limited to one ontology only, and different domain ontologies can extend the information for some synsets. Apart from the KYOTO Central Ontology, four different thesauri are used:

- GEMET (GEneral Multilingual Environmental Thesaurus)[5]
- SPECIES 2000[6]
- WWF Ecoregions database
- EUNIS[7]

All the ontologies are converted to the standard RDF/SKOS (Miles and Bechhofer, 2009) format and stored in the OpenLink Virtuoso Database backend (Wilensky and Idehen, 2009) (see Figure 2 for an example of interlinking these resources).

The main reason for using the Virtuoso database as a new DEB storage backend is the built-in support for the RDF SPARQL query language (Prud’hommeaux and Seaborne, 2009), which is designed for complex queries over ontological relations encoded in RDF triplets. Also the open-source license of Virtuoso is a necessary prerequisite for inclusion into DEB. Thanks to DEB platform architecture, the Virtuoso service is seamlessly integrated to all DEB interfaces. A user enters a SPARQL query (or a client application prepares one as a result from graphical user formulation) in the DEBVisDic interface and the results are retrieved by the server using Virtuoso AJAX API and presented to the user in the same format as other wordnets.

### 4.4 Importing Full Subtree

During the work on adding new items to the wordnet ontology, users often consults different ontological resources, such as the above mentioned GEMET or EUNIS. Including specific parts of these ontologies into WordNet often works with the same hierarchy as it is defined in the source ontology.

For such cases, the new DEB API provides a

[5] http://isegserv.itd.rl.ac.uk/skos/gemet/
[6] http://www.sp2000.org
[7] http://eunis.eea.europa.eu

Query: `https://server_name/wnen30_d?action=saveTree`

```
[ { ”SYNSET”: {
   ”INTERNAL_ID”: ”1”,
   ”SYNONYM”: { ”LITERAL”: { ”$”:”frog”, ”@sense”:”2” } },
   ”DEF”: { ”$”:”Def frog” },
   ”POS”: { ”$”:”n” },
   ”ELR”: [ { ”$”:”term_frog_id”,”@type”:”equivalent”,”@system”:”KYOTOterminology” },
   { ”$”:”eng-30-01639765-n”,”@type”:”equivalent”,”@system”:”enwn30” } ]
  },{ ”SYNSET”:
   ”INTERNAL_ID”: ”2”,
   ”ILR”: [ { ”$”:”1”,”@type”:”hypernym” } ],
   ”SYNONYM”: { ”LITERAL”: { ”$”:”robber frog”, ”@sense”:”2” } },
   ”DEF”: { ”$”:”Def robber frog” },
   ”POS”: { ”$”:”n” },
   ”ELR”: [ { ”$”:”term_robber_frog_id”,”@type”:”equivalent”,”@system”:”KYOTOterminology” } ]
  },{ ”SYNSET”:
   ”INTERNAL_ID”: ”3”,
   ”ILR”: [ { ”$”:”1”,”@type”:”hypernym” } ],
   ”SYNONYM”: { ”LITERAL”: { ”$”:”poison frog”, ”@sense”:”3” } },
   ”DEF”: { ”$”:”Def poison frog” },
   ”POS”: { ”$”:”n” },
   ”ELR”: [ { ”$”:”term_poison_frog_id”,”@type”:”equivalent”,”@system”:”KYOTOterminology” } ]
  },{ ”SYNSET”:
   ”INTERNAL_ID”: ”4”,
   ”ILR”: [ { ”$”:”3”,”@type”:”hypernym” } ],
   ”SYNONYM”: { ”LITERAL”: { ”$”:”endemic poison frog”, ”@sense”:”4” } },
   ”DEF”: { ”$”:”Def endemic poison frog” },
   ”POS”: { ”$”:”n” },
   ”ELR”: [ { ”$”:”term_endemic_poison_frog_id”,”@type”:”equivalent”,”@system”:”KYOTOterminology” } ]
  },{ ”SYNSET”:
   ”INTERNAL_ID”: ”5”,
   ”ILR”: [ { ”$”:”1”,”@type”:”hypernym” } ],
   ”SYNONYM”: { ”LITERAL”: { ”$”:”golden frog”, ”@sense”:”5_sense_number” } },
   ”DEF”: { ”$”:”Def golden frog” },
   ”POS”: { ”$”:”n” },
   ”ELR”:[ { ”$”:”term_golden_frog”,”@type”:”equivalent”,”@system”:”KYOTOterminology” } ]
  } } ]
```

Result: *All synset from the tree are added with correct link IDs.*

Table 4: API call for merging a full sub-tree from a selected source to a wordnet.

technique for efficient saving of several synsets and their hierarchical structure in one step. With this API call, DEBVisDic can store several synsets at once, while keeping their defined structure. Of course, before saving the synsets, the user does not know the unique synset database IDs regarding the new synsets. To be able to define the synset hierarchy, the user uses temporary identifiers in the request and the DEBVisDic server replaces them with real IDs.

For example, we want to enrich the WordNet with the following hierarchy of ontological concepts:

- frog
  - robber frog
  - poison frog
    - endemic poison frog
  - golden frog

The user will copy the hierarchy to the WordNet editor and add more synset data, like definition, other synonyms or more relations. When the editing is done, all the data are processed by the DEB server part and stored in the database. An example of the corresponding request and result is displayed in Table 4. In this example, the INTERNAL_ID elements are temporary identifiers that will be replaced with the actual IDs during the save process. This new extension to the DEBVisDic API offers a very effective way of building new WordNets.

### 4.5 JSONP Support

For security reasons, JavaScript client applications may send requests only to server on the same domain as the application. However, when working with several services hosted on different servers, the application needs to overcome this limitation. This kind of API requests is supported by so-called JSONP (JSON with padding) protocol. JSONP is a jQuery extension that passes the obtained server response to a specified JavaScript function. To be able to provide the results of all API calls in the form of the JSONP protocol, all the DEBVisDic API calls accept a new parameter 'callback'. With this parameter, the response is encapsulated in the requested JavaScript function. An example of the JSONP support in the DEB API calls is showed in Table 3.

### 4.6 External WordNet Relations

In a complex system like Global WordNet Grid or KYOTO database, where the different wordnets are connected together, usually through one pivot wordnet, sometimes with several center wordnets.

There are several types of external relations used in inter-wordnets links. The most common relations are:

- EQ_Synonym,
- EQ_Near_Synonym,
- EQ_Has_Hyperonym, and
- EQ_Has_Hyponym.

New API function allows to quickly find all the synsets from several wordnets that are related with the pivot synset. This is very useful for a multi-language projects.

## 5 Conclusions

We have presented the exploitation of the DEB platform as the main part of the database system within the KYOTO project. The DEB architecture shows here the benefits of its versatility and adaptability to news requirements, which allow to add new storage backend of OpenLink Virtuoso database or add the JSONP support to all previous API calls.

Even though the KYOTO project is just in the middle, we believe that the project will be a valuable step forward in defining future standards for semantic network architectures.

## Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009 and by the Czech Science Foundation under the project 102/09/1842.

## References

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Aleš Horák and Adam Rambousek. 2007. Dictionary Management System for the DEB Development Platform. In *Proceedings of the 4<sup>th</sup> International Workshop on Natural Language Processing and Cognitive Science (NLPCS, aka NLUCS)*, pages 129–138, Funchal, Portugal. INSTICC PRESS.

Aleš Horák, Karel Pala, Adam Rambousek, and Pavel Rychlý. 2006. 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.

Aleš Horák, Karel Pala, and Adam Rambousek. 2008. The Global WordNet Grid Software Design. In *Proceedings of the Fourth Global WordNet Conference*, Szegéd, Hungary. University of Szegéd.

D. Christodoulakis. 2004. *Balkanet Final Report*. University of Patras, DBLAB. No. IST-2000-29388.

Alistair Miles and Sean Bechhofer. 2009. SKOS Simple Knowledge Organization System, `http://www.w3.org/2004/02/skos/`.

Ian Oeschger et al. 2002. *Creating Applications with Mozilla*. O'Reilly and Associates, Inc., Sebastopol, California.

Eric Prud'hommeaux and Andy Seaborne. 2009. SPARQL Query Language for RDF, `http://www.w3.org/TR/rdf-sparql-query/`.

Louis Rosenfeld and Peter Morville. 1998. *Information Architecture for the World Wide Web*. O'Reilly and Associates, Inc., Sebastopol, California.

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/`.

Alan Wilensky and Kingsley Idehen. 2009. OpenLink Virtuoso Database, `http://virtuoso.openlinksw.com/`.