How to integrate data from different sources

Ricardo Ribeiro[†], **David M. de Matos**[*], **Nuno J. Mamede**[*]

L^{2}F – Spoken Language Systems Laboratory – INESC ID Lisboa
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
{ricardo.ribeiro,david.matos,nuno.mamede}@l2f.inesc-id.pt
[†]ISCTE – Instituto Superior de Ciências do Trabalho e da Empresa
[*]IST – Instituto Superior Técnico

Abstract

We present a dynamic multilingual repository for multi-source, multilevel linguistic data descriptions. The repository is able to integrate and merge multiple/concurrent descriptions of linguistic entities and allows existing relationships to be extended and new ones created. In addition, the repository is capable of also storing metadata, allowing for richer descriptions. We present results from work on large data collections and preview developments resulting from ongoing work.

1. Introduction

In the area of natural language processing we are often presented with the problem of having data sets ready to be used, and yet being unable to use them. This happens due to several facts: the coded information does not satisfy some of the actual needs or the resource format is not appropriate. These situations may lead to incompatibilities between the data used by two applications that perform the same kind of task, preventing the cross reusability of those data sets or even their combination to form a richer set of data.

Usually, in the development process of any kind of resource, some decisions are made that may affect the usability the resource. For instance, if a lexicon is built to be used by language interpretation applications, generally it is not suitable to be used directly for language generation. A generation lexicon is usually indexed by semantic concepts whereas an interpretation lexicon is indexed by words (Ribeiro et al., 2004; Jing and McKeown, 1998).

This paper explores a possible solution to the problem described. The solution consists of the development of a repository capable of integrating data sets that come from different sources. The data to be integrated may cover different levels of description, belong to different paradigms or have different actual formats of representation. Several types of incompatibility may appear, when merging this kind of data, making the coexistence of the involved resources difficult. One of the requisites of this solution is that this repository should be more than a mere storage device, and it should act as a bridge between all imported data sets, providing a canonical representation and respective translation agents for the involved information.

The problem of integrating data from diverse sources, in order to reuse it taking advantage of the best features of each data set, is not new. In fact, since *the mid 1980s, many researchers, language engineers and technology planners became aware of the idea of reusability and of its crucial role in facilitating the development of practical human language technology products that respond to the needs of users* (EAGLES, 1999).

The triggering event of these concerns was the *Automating the Lexicon: Research and Practice in a Multilingual Environment* (Walker et al., 1995) workshop that took place in 1986. Then, several projects were launched that addressed these issues. The EUROTRA-7 Study (EUROTRA-7, 1991) was concerned with accessing the feasibility of designing large scale reusable lexical and terminological resources. The main contributions of this study were an initial specification of a model for a reusable lexicon and several recommendations regarding the importance of standardization. Another important project was Multilex (Paprotté and Schumacher, 1993). This project aimed at providing specifications of standards for multilingual lexicons. The result was a preliminary design for a reusable multilingual lexicon, that continued the work previously started during EUROTRA-7. The GENELEX project had as main objective the development of a generic, application-independent model of lexicon. This model is commonly described as *theory welcoming* since it tries to accommodate data from competing theories. The GENELEX (Antoni-Lay et al., 1994) model was adopted (and adapted) in projects like PAROLE/SIMPLE (PAROLE, 1998; SIMPLE, 2000) which aimed at the development of the core of a set of natural language resources for the European Community languages. Alongside these projects, the EAGLES initiative aimed at accelerating the provision of standards for large-scale language resources; means of manipulating such knowledge; and, means of assessing and evaluating resources, tools and products (EAGLES, 1999).

This document is organized as follows: §2. presents the problems that may appear when trying to reuse data sets coming from different sources, and the requirements for a possible solution; §3. describes the proposed solution: a dynamic repository that tries to accommodate the differences of the data sets and their evolution (in content and structure); §4. describes an implementation of the proposed solution; Data access and maintenance issues are discussed in the following sections. The document concludes with a brief progress report presenting up to date results and some remarks about the advantages of this approach.

2. The problem

In general, the problems that afflict data sets and their reusability refer to miscellaneous incompatibilities:

(i) at the description level, i.e., how existing objects are described (the problem manifests itself frequently as tag incompatibility); (ii) at the level of what is described: some descriptions may describe objects missing from other descriptions; (iii) basic incompatibilities: format/representation: XML (W3C, 2001a) vs. tabular data; and (iv) expressiveness: e.g. “United States of America” as a single entity vs. composition of separate entities.

Figure 1 presents the description of the word *algo* (Portuguese for *something*) in several lexicons. The examples were taken from PAROLE, SMorph (Aït-Mokhtar, 1998), LUSOlex/BRASILex (Wittmann et al., 2000) and EPLexIC (de Oliveira, n.d.) lexicons. It is possible to observe cases for all the incompatibilities described above.

Description (in the first sense), representation and expressiveness incompatibilities can be observed, in this example, for the word *algo*. Concerning description incompatibilities, PAROLE and LUSOlex lexicons present two different categorizations (adverb and indefinite pronoun) for that word, while SMorph and EPLexIC have only one (in SMorph, *algo* is described as indefinite pronoun and in EPLexIC, as adverb); in what regards to representation, PAROLE uses XML (obtained from the original SGML), while the others use a textual (tabular based) format; and, concerning expressiveness, PAROLE and LUSOlex present a higher (similar) description granularity. In what concerns described objects, PAROLE and LUSOlex use several objects to describe the word *algo*, while SMorph and EPLexIC define only an object corresponding to the line where *algo* is described. The PAROLE lexicon also includes syntactic as well as semantic information (the latter from the SIMPLE part), omitted in this figure.

To address the incompatibility issues presented above, we identified a set of requirements: (i) preserving existing information (this is in an “at least” sense); (ii) allowing data reuse among different applications; (iii) allowing data to be imported/exported across existing formats (existing applications keep working, but they now use potentially better/richer data); and (iv) easy maintenance and documentation of changes.

These requirements are ideal in the sense that they may be addressed in various ways. A given solution for one of them may be optimal, but not suit all of them: some solutions may be better than others and some solutions may give rise to new problems. Our proposal seeks to find a balance, minimizing the negative aspects while meeting the requirements.

3. Proposal

Although models like the one proposed by GENELEX are generic, application-independent and, in this case, even theory welcoming, they are also static and do not describe means of evolving, in order to acommodate, for example, different kinds of information than the ones initially foreseen.

We propose a canonical model for storing/manipulating data, and a dynamic maintenance model for keeping the data model synchronized with new data developments. Even though a canonical model has its own set of problems, it presents distinct advantages: it is easier to maintain and document a single format than multiple different ones; the effort dedicated to maintenance tasks is concentrated, possibly further improving them; it allows for deeper understanding of data, which in turn facilitates reuse (the reverse would require a user to understand multiple models). Figure 2 shows how data moves around within the proposed

PAROLE

```
<mus id="r592" naming="algo"
     gramcat="adverb" autonomy="yes"
     synulist="usyn23987 usyn23988">
  <gmu range="0" reference="yes"
       inp="mfgr1">
    <spelling>algo</spelling>
  </gmu>
</mus>
<mus id="pi1" naming="algo"
     gramcat="pronoun"
     gramsubcat="indefinite"
     autonomy="yes"
     synulist="usyn23320">
  <gmu range="0" reference="yes"
       inp="mfgempty">
    <spelling>algo</spelling>
  </gmu>
</mus>
<ginp id="mfgr1" example="abaixo">
  <combmfcif combmf="combtm0">
    <cif stemind="0">
      <removal/>
      <addedbefore/><addedafter/>
    </cif>
  </combmfcif>
</ginp>
<ginp id="mfgempty" comment="empty Mfg">
  <combmfcif combmf="combtmempty">
    <cif stemind="0">
      <removal/>
      <addedbefore/><addedafter/>
    </cif>
  </combmfcif>
</ginp>
<combmf id="combtmempty"/>
<combmf id="combtm0" degree="positive"/>
```

SMorph

```
algo            /pr_i/s/GEN:*/pri.
```

LUSOlex

```
Adv191 <algo> ADVÉRBIO - FlAdv2 <algo>
Pi1 <algo> PRONOME INDEFINIDO - <algo>
FlAdv2  <abaixo>
        __P____ 0       <><>
$
```

EPLEXIC

```
algo/R=p/"al˜gu/algo
```

Figure 1: Lexicon comparison of *algo* descriptions.

solution.

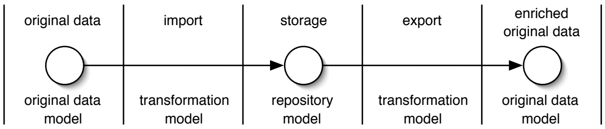

Figure 2: Data circulation.

It is also important to state that the proposed model allows evolution of both data and data structure.

Any sufficiently expressive high-level modeling language should be suitable for describing our models: one such is UML (Booch et al., 1999; OMG, n.d.); another would be XML Schema Definitions (XSD) (W3C, 2001b). Also to consider is their being amenable to automated processing, as well as their usefulness as model documentation languages (both UML and XSD fulfill these criteria: XSD, directly; UML, partly, via XMI (OMG, 2002)). We chose UML for its relative ease of use and rapid learning curve.

Since UML can be converted into XMI (i.e., XML), it allows a wide range of processing options. This, in turn, allows for the repository's data model to be used as the starting point for a set of processes that not only create the actual database, but also facilitate access to its data items (this may be done, e.g., through the use of code automatically generated from the UML model, as carried out by our prototype (de Matos and Mamede, 2003)).

In addition to the above, UML diagrams provide a useful source of documentation for the current state of the repository model. In fact, meta-information present in the UML diagrams may even be included in the database, thus enriching the data sets already there with a meta level.

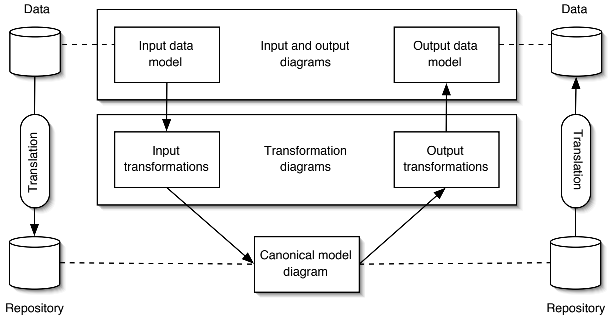

Figure 3: Models.

3.1. Canonical model

The canonical model consists of a set of class diagrams that specify the entities involved in the description of language components. Such components are morphological entities, inflection paradigms, predicates and their arguments, and so on.

The canonical model is based on existing large coverage models, i.e., we seek a broad coverage linguistic description that crosses information from various levels, including but not limited to morphology, syntax, and semantics. Examples of existing models, as mentioned before, are the ones resulting from the PAROLE project and its follow-up, the SIMPLE project.

In figure 3, we show the relations between the data input and output models, the data transformation models and the repository model, described in the following subsections.

3.2. Data input and output models

Data input/output models are used to describe external formats, i.e., formats of data to include in or to obtain from the repository. These models may already exist in some form (e.g. an SGML DTD) or they may be implicit (e.g. SMorph, ispell (Gorin et al., 1971–2003) use tabled data).

We isolate these models to clearly separate the repository's canonical model from the outside world. Nevertheless, we maintain open the possibility of interaction with other ways of storing/representing data. The following aspects must be taken into account.

3.2.1. Information aggregation

The repository is not limited in its capacity for storing objects by differences in the various descriptive levels of data to be imported, nor because of information concerning a particular domain. In fact, the repository is able to support multiple levels and domains, as well as the relationships between their objects, thus becoming an important asset for the tasks of information aggregation and organization.

3.2.2. Multiple levels

We consider multiple information levels referring to the ones described in the literature (morphology, syntax, and so on). But we are not limited to these “traditional” descriptions: it may be of interest to include support for other levels, e.g. one halfway between morphology and syntax. The design of the repository must provide support both to existing descriptions and to descriptions resulting from either cross-references of existing data or from including new data in the repository. Evolution to improve support must, however, ensure that current uses remain possible.

3.2.3. Multiple sources

In addition to the aspect presented in §3.2.2., we must also consider the existence of multiple information sources in the context of a given domain: data may originate from different projects and/or applications. The main concern here is maintaining the expressiveness of the original data, as well as the integrity of their meaning and the consistency of the data already in the repository. The danger stems from using different formats and descriptions for stored and imported data. As an example, morphology models defined by the PAROLE project are much more expressive than those defined by, say, a morphological analyzer such as JSpell (de Almeida and Pinto, 1994). The repository must be able to import/export both data sets according to their original models.

The coexistence of multiple sources is a non-trivial problem, especially if the canonical model assumes links between description levels: importing data from sources without those links may require additional assumptions. An example: PAROLE/SIMPLE morphological entities may be associated with syntactic units and these with semantics units; in contrast, syntactic data from project Edite (Marques da Silva, 1997), while also associated with semantic information (different from that of SIMPLE), is not directly associated with the morphological level.

Regarding integrity, consider a morphological entity: it may be defined in different ways by different models. However, when stored in the repository, it must be represented as a single object with the semantics of each original source model. This implies that the canonical model must be sufficiently flexible and expressive to ensure that the original semantics of imported objects is not destroyed.

3.2.4. Relationships and non-linguistic data

Beyond data items, which may come from various independent sources and possibly unrelated domains, the repository must contemplate the possible existence of relationships between the objects it stores. We have seen examples of those relationships (e.g. between morphological and semantics objects, or those existing between syntax and semantics objects). Other relationships may be created and stored, to account for any aspect deemed of interest: e.g. relationships with non-linguistic data, such as ontologies.

In general, relationships are not restricted in what concerns the number of related objects, i.e., the repository supports any multiplicity.

3.3. Data transformation models

These models allow resources from the repository to be adapted to diverse applications. Some of these applications may precede the repository and require proprietary formats. This compatibility issue is just one example of the more general problem of exporting data described according to the canonical model to formats described according to outside models. The export capability is of great importance, since the repository must ensure its usefulness for existing applications.

Two sets of models have, thus, been defined: the first contains models of the transformations needed for converting from data described by external models and the canonical model. The second set contains models of the transformations needed for converting from data described by the canonical model and external models.

4. Implementation

We now present implementations for each of the previous concepts.

4.1. The canonical model

Implementing the canonical model consists of defining the model proper and deploying it using some kind of data storage solution. Requirements as defined in §3.1. must be satisfied.

Work on the modeling task started with the study of existing large coverage models defined by the PAROLE/SIMPLE projects. Their models, published as SGML DTDs, were enriched according to the requirements for supporting both the new concepts and existing concepts that underwent some refinements. The resulting data model differs from the original, but is still very close and has, so far, proven to be sufficient for covering other models.

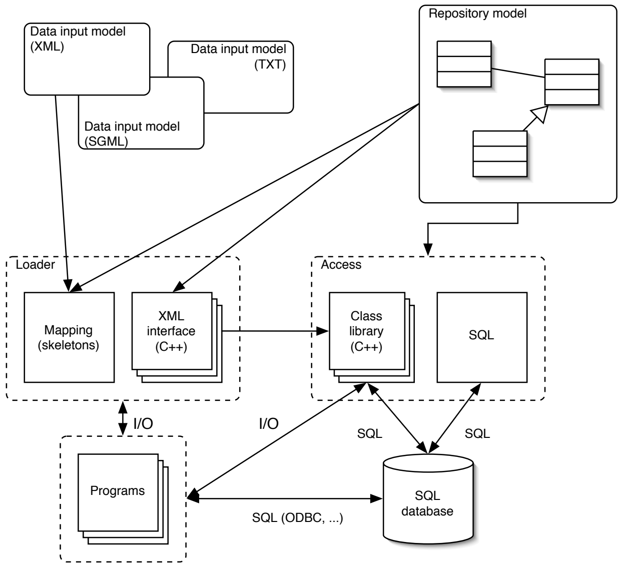

Figure 4: Models and code generation as implemented by the current prototype (de Matos and Mamede, 2003).

We chose a relational database (RDB) to implement the repository. RDBs confer flexibility to the global design of the systems that use them. The flexibility is directly linked to fine data granularity provided by database tables and by the operations provided to work with them, e.g., dynamic changes are possible, making it possible to perform changes to data structures while in use. RDBs are also flexible in the possible views they allow to be defined over data: they allow finer selection according to the client's interests.

Any candidate RDB engine must possess some way of verifying and enforcing data integrity constraints (e.g. references to foreign keys). The exact nature of these mechanisms is not important in itself, but must be taken into account when processing data.

Our choice for storage and data management was MySQL (MySQL, n.d.). Tables and integrity maintenance constraints were generated using XSLT scripts taking as input the original UML repository models (de Matos and Mamede, 2003). Note that only the canonical model diagrams are used in this task, i.e., the data input/output and data transformation models are not used.

4.2. Data input and output models

As mentioned above, these models are used to describe data to be imported/exported to/from the repository, i.e., to be converted to/from the canonical data model.

These models may be described using UML (same advantages as for the canonical model), but other data description languages, such as XML Schema Definitions (XSD), may be acceptable as long as their expressiveness is deemed sufficient for automatic processing and documentation purposes. If the original description does not exist, it is possible that one or more UML models may cover the data to be processed. Selecting the appropriate external model will depend on the current canonical model and on how well the external model allows the external data to be mapped onto the canonical representation.

These models do not require further implementation or support (they are assumed to be supported by some outside application/system). In what concerns our work, they are to be used as input for the code derived from the data transformation models (see §3.3.).

4.3. Data transformation models

Our work with these models is limited to selected cases. Namely, we defined input transformation models for the Portuguese data resulting from the PAROLE/SIMPLE projects. Although preliminary, at the time of this writing, the work allows us to envision the best way of implementing other filters for loading arbitrary data. Data from EPLexIC and LUSOlex/BRASILex underwent a different set of transformations, namely, they were converted to the external representation of the canonical prior to loading.

Output transformation models have not been explicitly implemented: currently, we obtain data from the database, either through the programming interface, associated with the canonical model, or directly, via SQL commands.

5. Data Access

Accessing the database implies no special requirement. It is the nature of the transfered information that introduces the requirements that should be fulfilled.

5.1. Access to the canonical repository

For convenience and flexibility, a network interface should be provided. This feature, present in almost all modern RDBs, should not prove difficult to implement. It may be either a proprietary or an open protocol implementing some kind of distributed SQL transport mechanism. Examples are ODBC (Microsoft Corporation, n.d.) and JDBC (Sun Microsystems, Inc., n.d.). We privileged openness, since it facilitates portability and maintenance (on this topic, see for instance (Norris, 2004)).

Since our main source of interaction would come from a set of C++ applications we started by defining a programming interface for this language. A connectivity library (DTL/ODBC (Gradman and Joy, n.d.)) was used to link the lower level RDB access with the higher level program interface (a set of automatically generated C++ classes representing database concepts). As mentioned before, the generation of these classes was done using XSLT, taking as input the original canonical model UML diagrams. Since this was the method used for building the database itself, we are able to guarantee close mappings between the different levels, thus minimizing concept mismatches.

Regardless of these methods, access to the repository is open to other methods. This is one of the advantages of using a RDB engine as a separate data management agent. In particular, use of other languages is possible, as long as they support the concepts in the repository, e.g., via the object abstraction. We introduce this requirement to prevent the high costs associated with explicit management of non-native concepts in the target language. Another requirement is that a low-level RDB interaction library (either native/proprietary or open/standard) exists that supports the chosen language. Once again, this is to avoid pursuing expensive solutions.

5.2. Programming interface

More than being just a source of passive data, the repository supports “online” uses. To support online clients, the repository must support some kind of communication mechanism with its users, regardless of them being humans or machines. Thus, in addition to being able to import/export data using existing formats, the repository also provides a clearly defined programming interface.

6. Maintenance

There are two main aspects regarding maintenance. The first is repository content management: this aspect accounts for future expansion both of data content and expressiveness of data models, i.e., their descriptive power. In fact, current work already points to a few troublesome aspects (e.g. paradigm shifts). So far, though, we have been able to find elegant solutions for all of them and still maintain the original data semantics (of course, in some cases, semantics has been augmented to account for the new uses).

The second maintenance aspect concerns management of data models: this item covers aspects relating to miscellaneous remodeling operations and possible redefinition of the canonical model. This implies transition operations between successive canonical models, which in themselves are no more than instantiations of data import/export operations, albeit possibly more complex than the ones used by applications such as a morphological analyzer.

In spite of having started work on maintenance aspects, content and model maintenance remain to be fully addressed. Data model maintenance has already been partially addressed by the use of UML diagrams and subsequent code generation operations that allow full access to the corresponding repository data.

7. Final remarks and future directions

Results so far, obtained with large data sets, allow us to conclude that our approach addresses the requirements stated above. Moreover, importing the lexicons presented in table 1, enriched the repository and the sets themselves at three different levels: (i) enrichment obtained from the data integration, which provides an easier selection of the needed data for a specific purpose and, given the user of the data has only one format and one set of data to consider, easier reutilization and improvement of the data itself; (ii) the interaction with the data is promoted by means of the data access possibilities mentioned above. This interaction promotes more interaction and consequent data enrichment (by allowing simple/easy extension/expansion of data relationships); (iii) implicit enrichment which is a consequence of importing new data into the existing structure. For example, when importing the phonetic information of the word forms of EPLexIC to the existing phonetic paradigms structure, all related word forms of the imported one were enriched with corresponding phonetic information.

We are also able to conclude that our work points to a more general solution to the problem of data reuse and

Lexicon	Size (entries)	Imported entries
PAROLE	20k	fully loaded
LUSOlex	65k	fully loaded
BRASILex	68k	fully loaded
EPLexIC	*80k*	partially loaded
SMorph	35k	not loaded

Table 1: Data sets under study. Since the repository is multilingual we are using the ISO 639 and ISO 3166 standards to encode respectively the language and region. Thus, PAROLE, LUSOlex, EPLexIC, and SMorph are all marked as `pt_PT` and BRASILex as `pt_BR`. Although we have data samples for other languages they have yet to be considered. Italicized numbers in the table refer to word forms.

integration. In addition, it opens the door to seamless integration with other data description levels, such as language-oriented ontologies.

8. References

Antoni-Lay, Marie-Hélène, Gil Francopoulo, and Laurence Zaysser, 1994. A Generic Model for Reusable Lexicons: the Genelex Project. *Literary and Linguistic Computing*, 9(1):47–54. Oxford University Press.

Aït-Mokhtar, S., 1998. *L'analyse présyntaxique en une seule étape*. Thèse de doctorat, Université Blaise Pascal, GRIL, Clermont-Ferrand.

Booch, Grady, James Rumbaugh, and Ivar Jacobson, 1999. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc. ISBN 0-201-57168-4.

de Almeida, José João Dias and Ulisses Pinto, 1994. Jspell – um módulo para a análise léxica genérica de linguagem natural. In *Encontro da Associação Portuguesa de Linguística*. Évora.

de Matos, David Martins and Nuno J. Mamede, 2003. Linguistic Resources Database – Database Driver Code Generator (DTL/ODBC). Technical Report RT/007/2003-CDIL, L^{2}F – Spoken Language Systems Laboratory, INESC-ID Lisboa, Lisboa.

de Oliveira, Luís Caldas, n.d. EPLexIC – European Portuguese Pronunciation Lexicon of INESC-CLUL. Documentation.

EAGLES, 1999. Final Report: Editor's Introduction. EAGLES Document EAG-EB-FR2, Expert Advisory Group on Language Engineering Standards.

EUROTRA-7, 1991. Feasibility and project definition study of the reusability of lexical and terminological resources in computerised applications.

Gorin, R. E., Pace Willisson, Walt Buehring, and Geoff Kuenning, 1971–2003. International ispell. `http://www.gnu.org/software/ispell/ispell.html`.

Gradman, Michael and Corwin Joy, n.d. *Database Template Library*. See: `http://dtemplatelib.sf.net/`.

Jing, H. and K. McKeown, 1998. Combining multiple, large-scale resources in a reusable lexicon for natural language generation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*.

Marques da Silva, Maria Luísa, 1997. *Edite, um sistema de acesso a base de dados em linguagem natural. Análise Morfológica, Sintáctica e Semântica*. Tese de mestrado, Instituto Superior Técnico, Lisboa.

Microsoft Corporation, n.d. ODBC – Microsoft Open Database Connectivity. Specifications and implementations may be found, among other places, at: `http://msdn.microsoft.com/library/en-us/odbc/htm/odbcstartpage1.asp`, `www.iodbc.org`, or `www.unixodbc.org`.

MySQL, n.d. *MySQL Database Server*. MySQL A.B. See: `http://www.mysql.com/products/mysql/`.

Norris, Jeffrey S., 2004. Mission-Critical Development with Open Source Software: Lessons Learned. *IEEE Software*, 21(1):42–49.

OMG, 2002. *XML Metadata Interchange (XMI) Specification, v1.2*. Object Management Group (OMG). See: `www.omg.org/technology/documents/formal/xmi.htm`.

OMG, n.d. *Unified Modelling Language*. Object Management Group (OMG). See: `www.uml.org`.

Paprotté, Wolf and Frank Schumacher, 1993. MULTILEX – Final Report WP9: MLEXd. Technical Report MWP8-MS Final Version, Westfälische Wilhelms-Universität Münster.

PAROLE, 1998. *Preparatory Action for Linguistic Resources Organisation for Language Engineering – PAROLE*. `http://www.hltcentral.org/projects/detail.php?acronym=PAROLE`.

Ribeiro, Ricardo, Nuno Mamede, and Isabel Trancoso, 2004. *Morphossyntactic Tagging as a Case Study of Linguistic Resources Reuse*. Colibri. To appear.

SIMPLE, 2000. *Semantic Information for Multifunctional Plurilingual Lexica – SIMPLE*. `http://www.hltcentral.org/projects/detail.php?acronym=SIMPLE`.

Sun Microsystems, Inc., n.d. JDBC Data Access API. See: `http://java.sun.com/products/jdbc/`.

W3C, 2001a. *Extensible Markup Language*. World Wide Web Consortium (W3C). See: `www.w3c.org/XML`.

W3C, 2001b. *XML Schema*. World Wide Web Consortium (W3C). See: `www.w3c.org/XML/Schema` and `www.oasis-open.org/cover/schemas.html`.

Walker, D., A. Zampolli, and N. Calzolari (eds.), 1995. *Automating the lexicon: Research and practice in a multilingual environment*. Oxford: Oxord University Press.

Wittmann, Luzia, Ricardo Ribeiro, Tânia Pêgo, and Fernando Batista, 2000. Some language resources and tools for computational processing of portuguese at inesc. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*. Athens, Greece: ELRA.

Acknowledgments

This paper was supported by the DIGA Project (POSI/PLP/41319/2001).