The Workshop Programme

[Please insert timing of sessions, authors and titles of speeches, coffee/lunch breaks using font Times New Roman, 12 pts and interlinear spacing as you prefer]

9:30 Welcome and Introduction

9:45 **Daan Broeder et al.,** *INTERA: A Distributed Metadata Domain of Language Resources*
10:15 **Laurent Romary et al.**, *An on-line data category registry for linguistic resource management*

10:45Break

11:00 **Gil Francopoulo et al.,** *Data Categories in Lexical Markup Framework or how to lighten a Model*
11:30 **Francesca Bertagna et al**.,*The MILE Lexical Classes: Data Categories for Content Interoperability among Lexicons*
12:00 **David Dalby and Lee Gillam,***"Weaving the Linguasphere":LS 639, ISO 639 and ISO 12620*

12:30 Lunch
14:00 **Robert Kelly et al.,** *Annotating Syllable Corpora with Linguistic Data Categories in XML*
14:30 **Thorsten Trippel,** *Metadata for Time Aligned Corpora*
15:00 **Ricardo Ribeiro,** *How to integrate data from different sources*

15:30 **Kiyong Lee et al.,** *Towards an international standard on feature structure representation (2)*
16:00 **Ewan Klein and Stephen Potter,** *An Ontology for NLP Services*

16:30 Discussion and Panel: Evaluation of Metadata and Data Categories

Workshop Organisers

[names and affiliations]

Thierry Declerck, Saarland University and DFKI GmbH
Nancy Ide, Vassar College
Key-Sun Choi, Kaist
Laurent Romary, LORIA

Workshop Programme Committee

[names and affiliations]

Maria Gavrilidou, ILSP
Stelios Piperidis ILSP
Daan Broeder, Max Planck Institute
Peter Wittenburg, Max Planck Institute
Nicoletta Calzolari, ILC
Monica Monachini, ILC
Claudia Soria,ILC
Khalid Choukry, ELRA/ELDA
Mahtab Nikkhou, ELDA
Kiyong Lee, Korea University
Paul Buitelaar, DFKI
Andreas Witt, University of Bielefeld
Scott Farrar, University of Bremen
William Lewis, University of Arizona
Terry Langendoen, University of Arizona
Gary Simons, SIL International
Eric de la Clergerie, INRIA

Table of Contents

INTERA: A Distributed Metadata Domain of Language Resources. ***Daan Broeder, Maria Nava and Thierry Declerck***

An on-line data category registry for linguistic resource management. ***Laurent Romary, Nancy Ide, Peter Wittenburg and Thierry Declerck***

Data Categories in Lexical Markup Framework or how to lighten a Model. ***Gil Francopoulo, Monte George and Mandy Pet***

The MILE Lexical Classes: Data Categories for Content Interoperability among Lexicons. ***Francesca Bertagna, Alessandro Lenci, Monica Monachini and Nicoletta Calzolari***

"Weaving the Linguasphere":LS 639, ISO 639 and ISO 12620. ***David Dalby and Lee Gillam***

Annotating Syllable Corpora with Linguistic Data Categories in XML. ***Robert Kelly, Moritz Neugebauer, MichaelWalsh & StephenWilson***

Metadata for Time Aligned Corpora. ***Thorsten Trippel***

How to integrate data from different sources. ***Ricardo Ribeiro, David M. de Matos, and Nuno J. Mamede***

Towards an international standard on feature structure representation (2): ***Kiyong Lee, Lou Burnard, Laurent Romary, Eric de la Clergerie, Ulrich Schaefer, Thierry Declerck, Syd Bauman, Harry Bunt, Lionel Clément, Tomaz Erjavec, Azim Roussanaly, and Claude Roux***

An Ontology for NLP Services: ***Ewan Klein and Stephen Potter***

Author Index

Syd Bauman
Francesca Bertagna
Daan Broeder
Harry Bunt
Lou Burnard
Nicoletta Calzolari
Lionel Clément
David M. de Matos
Eric de la Clergerie
David Dalby
Thierry Declerck
Tomaz Erjavec
Gil Francopoulo
Monty George
Lee Gillam
Nancy Ide
Robert Kelly
Ewan Klein
Kiyong Lee
Alessandro Lenci
Nuno J. Mamede
Monica Monachini
Maria Nava
Moritz Neugebauer
Mandy Pet
Stephen Potter
Ricardo Ribeiro
Laurent Romary
Azim Roussanaly
Claude Roux
Ulrich Schaefer
Thorsten Trippel
Peter Wittenburg

INTERA: A Distributed Metadata Domain of Language Resources

Daan Broeder, Maria Nava+, Thierry Declerck++

Max-Planck-Institute for Psycholinguistics
broeder@mpi.nl,
+Evaluation and Language Resources Distribution Agency,
++University of Saarland

Abstract

The INTERA and ECHO projects were partly intended to create a critical mass of open and linked metadata descriptions of language resources helping researchers to understand the benefits of an increased visibility of language resources in the Internet and to motivate them to participate. The work was based on using the new IMDI version 3.0.3 which is a result of experiences with the earlier version and new requirements coming from the involved partners. Language resource distribution centers such as ELDA have the opportunity to use and add to this metadata infrastructure and use it to enhance their catalogue and offer more services to their customers such as offering data samples and download of partial corpora .This document sumarizes mainly experiences done in the project INTERA.

Introduction

At LREC 2000 in Athens the first workshop[1] about metadata concepts for making language resources visible in and discoverable via the Internet was organized by some of the authors. At LREC 2002 two groups demonstrated operational frameworks for creating metadata for language resources and to work with them for management and discovery purposes. While OLAC[2] (Open Language Community Archives Community) started form a Dublin Core[3] point of view with the goal to create a set that allows for the description of all types of language resources, software tools, and advice, the IMDI[4] (ISLE Metadata Initiative) activities started with a slightly different approach. The focus was primarily on multimedia/multimodal corpora and a more detailed set was worked out that can be used not only for resource discovery but also for exploitation and managing large corpora. Most importantly, IMDI allows its metadata descriptions to be organized into linked hierarchies supporting browsing and enabling data managers to carry out a variety of management tasks.

The two years since 2002 have been used to improve the metadata sets based on the experience of the communities. They have also been used to create an interoperable domain, i.e., a mapping schema was worked out between the IMDI and OLAC sets and the IMDI domain acts as an OLAC data provider. IMDI records can be searched for from the OLAC domain.

IMDI Metadata Set 3.0.3

Based on the experiences and on a broad discussion process including field linguists, corpus linguists and language engineers, the IMDI set 3.0.3 was designed as part of the INTERA project[5] and is available as an XML-schema. It was adapted to simplify the content description and the artificial distinction between collectors and other participants probably influenced by Dublin Core was removed. Three major extensions were applied: First, it is now possible to describe written resources that are not annotations or descriptions. This was necessary, since most language collections contain written resources in the form of field notes, sketch grammars, phoneme descriptions etc. Second, as a consequence of long discussions with participants of the MILE lexicon initiative[6], it is now possible to describe lexicons with a specialized set of descriptor elements.

Third, it is now possible to define and add project-specific profiles. In the earlier version IMDI supported already the possibility of extensions at various levels in the form of user defined category–value pairs, i.e., the user was able to define a private category and associate values with it.

This feature was used by individuals and also projects to include special descriptors, however, these descriptors were not fully supported by the IMDI tools. In the new version, however, projects or sub-domains such as the Dutch Spoken Corpus respectively the Sign Language community can define a set of important categories and these are supported while editing or searching.

Therefore, IMDI consists of its core definitions that have to be stable to assure users that their work will be exploitable even after many years and of sub-community specific extensions, which nevertheless are result of discussion processes.

A new direction is also given to IMDI in INTERA, which foresees the linking (or merging) of metadata for language data with descriptors in use in catalogues for language technology tools, like the ACL Natural Language Registry, hosted at DFKI[7].

[1] http://www.mpi.nl/ISLE
[2] OLAC: http://language-archives.org
[3] See http://dublincore.org/ for more information on the the Dublin Core Metadata Initiative
[4] IMDI: http://www.mpi.nl/IMDI
[5] Integrated European language Resource Area: http://www.elda.fr/index.html
[6] See http://www.ilc.cnr.it/EAGLES96/isle/complex/clwg_home_page.htm for more details
[7] See http://regsistry.dfki.de

IMDI Catalogue Metadata

The design and development of the IMDI metadata set was directed to offering adequate descriptions at the level of resources. However it was recognized at an early stage that there is a need to describe whole collections of language resources at the level of a finished or published corpus. During a IMDI workshop in 2001 a proposal for the IMDI Metadata Elements for Catalogue descriptions[8] was presented based on information from the Evaluation and Language Resources Distribution Agency (ELDA) or the Linguistic Data Consortium (LDC)

The description of language resources for distribution purposes is essential for data centres. Catalogue management is the core activity of data centres and this function is reflected in their own metadata. Beside the description of the content, i.e. data categories offered within a language resource, it is vital to supply searchable information to potential users trying to locate corpora as units, for instance, for a specific application, from a specific source or distributed under a specific license.

At catalogue management level there are special requirements that are connected with aspects of the dissemination activity that are particular to distribution agencies. A data center like ELDA uses classes of descriptors that account for different features of a corpus as a whole. From this point of view, effective metadata should contain the following information:

- Identification of the language resource;
- Description of the data
- Author(s) and editor(s) of the data
- Objectives for creating the data and intended purpose of the data;
- Data sources and how the data was created;
- Accuracy and reliability of the data
- Distribution and contact information, including prices and licensing policies.

In ELDA's metadata, these categories are used alongside other classes of descriptors that account for the content of any particular resource (speech, written or multimodal corpora, lexica, terminologies, etc.).

The set of the IMDI Metadata Elements for Catalogue descriptions accounts for the need of offering distribution information, though in a less detailed, flatter representation than the one used by data centers like ELDA. Practically all the metadata classes mentioned above are reflected in the IMDI catalogue descriptions, so that it is possible to specify information like the size of whole corpora, the physical medium of the corpus (CD/DVD), prices, etc.

In particular, specific descriptors accounting for the (foreseen) use of the corpus were introduced. Usually, corpora are created with a specific use in mind and, in that case, it is natural to make this information available at the level of the whole corpus as a list of possible "application domains".

The comparison of the IMDI and ELDA metadata sets has also highlighted the need of another specific category of descriptors. The introduction of metadata elements supplying information on feature distribution is currently under study. This class of metadata would be specific for describing varying parameters across a corpus, where their overall distribution is important in order to determine whether a corpus may be suitable for a certain purpose: Among these feature distribution metadata are:

- Age/Gender distribution of participants (age classes, number of age classes, etc);
- Language distribution (number of languages, percentage of languages represented in a corpus, etc.);
- Text-type/Genre distribution.

These distribution parameters are particularly important when there is no means of making selections with the desired characteristics directly from the corpus. The metadata elements accounting for distributional features are currently being formalised and described.

The IMDI Framework

Tools

The IMDI initiative also offers set of tools[9] for the latest metadata set version 3.0.3:

(1) The IMDI-Editor that allows users to create fully IMDI compliant metadata descriptions and that supports all IMDI features such as controlled vocabularies and project specific profiles.

(2) The IMDI-Browser that allows navigating in the distributed domain of linked metadata XML files supporting searching as well as browsing, the setting of bookmarks etc (fig. 1). A tree-builder that allows the user to create new user-specific virtual trees by linking arbitrary metadata descriptions and creating arbitrary nodes.

(3) For large archives with a web-server on-the-fly transformed HTML presentation of the metadata files that allow users to browse in the linked metadata domain with normal web-browsers (fig. 2). Different sites may implement different ways of presenting the IMDI domain.

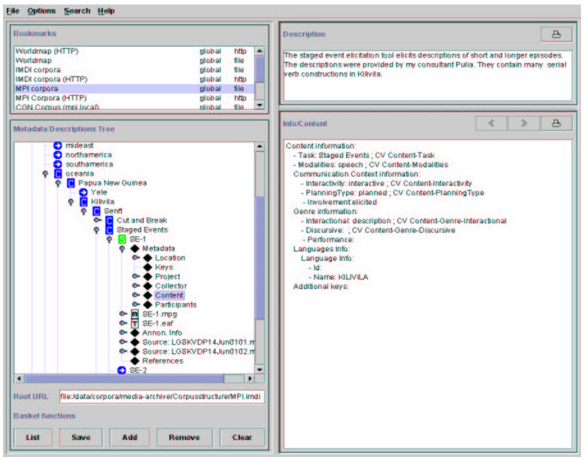
(4) Software for a service that offers access to IMDI records according to the OAI metadata harvesting protocol.

Distributed Metadata

Metadata used for discovery (metadata search) purposes is distributed over several locations. Often to offer an effective Metada discovery service, the metadata needs to

[8] Documentation available at http://www.mpi.nl/IMDI/documents/Proposals/IMDI_Catalogue_2.1.pdf

[9] All tools are Open Source and available at the sites: http://www.mpi.nl/tools or http://www.mpi.nl/IMDI



Worldmap (HTTP)	global	http
Worldmap	global	file
IMDI corpora	global	file
IMDI corpora (HTTP)	global	http
MPI corpora	global	file
MPI Corpora (HTTP)	global	http
CGN Corpus (mpi local)	global	file

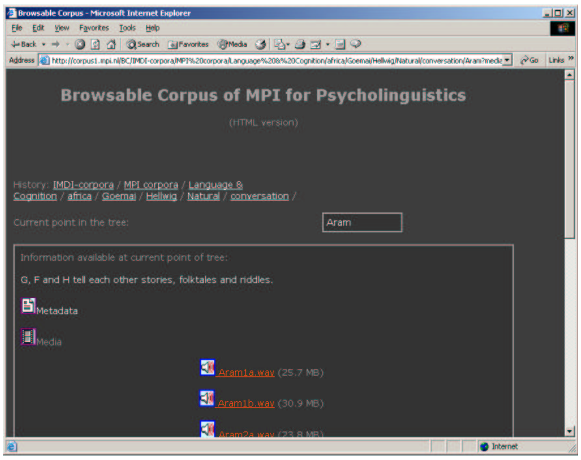
The staged event elicitation tool elicits descriptions of short and longer episodes. The descriptions were provided by my consultant Pulia. They contain many serial verb constructions in Kilivila.

Content information:
- Task: Staged Events ; CV Content-Task
- Modalities: speech ; CV Content-Modalities
Communication Context information:
- Interactivity: interactive ; CV Content-Interactivity
- PlanningType: planned ; CV Content-PlanningType
- Involvement:elicited
Genre information:
- Interactional: description ; CV Content-Genre-Interactional
- Discursive: ; CV Content-Genre-Discursive
- Performance:
Languages Info:
Language Info:
- Id:
- Name: KILIVILA
Additional keys:

Root URL: file:/data/corpora/media-archive/Corpusstructure/MPI.imdi

Figure 1 The interface of the XML-based special browser that offers advances functionality.

be brought together at a single site. The OAI model[10] defines data and service providers related via the metadata harvesting protocol that defines the interaction pattern and the metadata record packaging. The data providers all have to minimally provide Dublin Core records to achieve a minimal level of semantic interoperability. However, the OAI protocol also allows to send records specified by another schema such as IMDI. Based on this, service providers can build services for example a metadata search service that covers a large group of different repositories working internally with different metadata sets.



Address: http://corpus1.mpi.nl/BC/IMDI-corpora/MPI%20corpora/Language%20&%20Cognition/africa/Goemai/Hellwig/Natural/conversation/Aram?media

Browsable Corpus of MPI for Psycholinguistics

(HTML version)

History: IMDI-corpora / MPI corpora / Language & Cognition / africa / Goemai / Hellwig / Natural / conversation /

Current point in the tree: Aram

Information available at current point of tree:

G, F and H tell each other stories, folktales and riddles.

Metadata

Media

Aram1a.wav (25.7 MB)

Aram1b.wav (30.9 MB)

Aram2a.wav (23.8 MB)

Figure 2 the interface for browsing in the metadata domain with the help of normal HTML presentations

The OAI protocol is comparatively simple to implement, the common praxis is still to harvest XML-files. The ECHO project has shown that most of the institutions are not yet prepared to support OAI. The IMDI metadata infrastructure assumed from the beginning that metadata records can be located at various institutions – even on the notebook of remotely working fieldworkers. Therefore, IMDI metadata records can be linked in a simple way – similar as web-sites. The browser only needs a registered URL to integrate the IMDI descriptions into the domain. For searching, the IMDI tools will scan all known metadata links and create indexes that can then be exploited.

However, all IMDI tools expect IMDI type of metadata records, i.e., IMDI is not a concept for establishing interoperability between different metadata sets. Within the ECHO project an integrated metadata domain was built that includes ten different repositories from five different disciplines. It was shown that interoperability at the structural level was mainly achieved by harvesting XML-structured files and at the semantic level by creating special mappings (Wittenburg, 2004a/b). The Dublin Core approach reduces too much of the semantic richness of the provided information. Therefore, it is seen just as another view on the data.

INTERA Metadata Search infrastructure

Usually metadata infrastructures depend on one or several sites harvesting the metadata records, storing them in a DB and offer users a (Web-)interface to search them such as the already mentioned OAI model. INTERA is no different but the IMDI tools also offer access to metadata that is present on the local machine, without any network access, thus empowering users to create their own (small) metadata repositories.

When designing the IMDI infrastructure a requirement was that no external database software should be required so that metadata search can be done with the browser without network connectivity for local corpora. Of course it is possible to search remote IMDI repositories also from the same interface.

Figure 3 (at the end of the paper) shows the metadata search infrastructure as it is foreseen and being realized in INTERA.

(1) There are multiple sites that store metadata of language resources (that may themselves or may not be available on the net also). These metadata records are available in the form of IMDI structured trees [] so they can all be accessed by the top-node for a site and can be found by descending the tree.

(2) Some sites can harvest the IMDI records in this way for example to construct catalogues and/or provide faster metadata search facilities. In the figure ELRA is such a harvesting site.

(3) metadata search can be performed in different ways using either a specific tools such as the IMDI-Browser that is able to access and search IMDI metadata records that a user has created on a local machine (mysite) and also remote metadata repositories such as MPI that offer IMDI records either from local data or metadata that has been harvested from other sites. Sites that harvest metadata can store these in DBMS and offer search facilities through normal WWW browsers.

[10] See OAI http://www.openarchives.org/, and OLAC http://www.language-archives.org/

(4) The definitions and explanations about the terms used in the IMDI metadata set is stored in a central Data Category Repository (DCR), at the moment only access through a normal web-browser is possible. However in future projects we hope to realize to interact with the DCR from special tools so we can for example upload definitions for new metadata descriptors from the IMDI-Editor.

(5) Sites that harvest IMDI metadata records can themselves be harvested by OAI/OLAC service providers[11]. In this way the IMDI records become available to a wider audience.

Critical Mass of Metadata

Within the INTERA project it was the task to convince various data centers and projects to participate in building a distributed IMDI domain. Typically, these data centers have language resources from the area of language engineering. In the mean time the metadata is generated by the following institutions: European Language Resource Agency (Paris), Institut National de la Langue Francais (INALF, Nancy), German Center for Artificial Intelligence (DFKI, Saarbrücken), University of Saarbrücken (Saarbrücken), Bavarian Speech Archive (Munich), Meertens Institute (Amsterdam), University of Florence, Institute for Language and Speech Processing (Athens), Instituto Linguistica Computationale (Pisa), University of Ljubljana, University of Sofia, University of Iasi and the Max-Planck-Institute for Psycholinguistics (Nijmegen).

In the ECHO project[12] it was one of the tasks to motivate researchers and institutions to create metadata descriptions of resources that can be seen as part of our heritage. Here the following institutions can be mentioned: University of Helsinki, Phonogrammarchiv Vienna, University of Groningen, University of St. Petersburg, Kotus (Helsinki), Sweden's national Dialect Archive, European Sign Language Communities (Stockholm, London, Netherlands, Germany), University of Utrecht, University of Uppsala, University of Stavanger, University of Lund, DOBES Programme (Nijmegen).

This new emerging domain including the activities of about 27 partners includes textual corpora, national speech corpora, multimedia/multimodal corpora, parallel corpora, lexicons and various types of written resources.

Yet we don't have a final estimate about the number of individual resources that will be described and available at the end of 2004 when the two mentioned projects will be finished. At the Max-Planck-Institute there are currently about 30.000 sessions described by metadata. Large corpora such as SMARTKOM, the Dutch Spoken Corpus, the LABLITA corpus and the ATILF corpus will be part of the new domain, so we can expect that there will be many more resource units described and therefore searchable.

It is hoped that this emerging domain is large enough to demonstrate the usefulness of metadata for discovery purposes and that it will inspire others to participate. The ENABLER[13] overview has clearly indicated that there is a lack of visibility of language resources in the Internet and that their accessibility is even worse. Therefore, the creation of metadata must be a high priority program to foster re-usage. In a declaration agreed upon at the ENABLER meeting in Paris in 2003 it was stated that the funding agencies should make the generation and integration of proper and openly available metadata descriptions according to one of the two currently existing standards (OLAC or IMDI) obligatory.

Metadata Creation Process

In the first phase of INTERA and ECHO various European data centers and research institutions were approached whether they are interested to participate in creating an integrated metadata domain. The initiative had good response, i.e., most reacted in a positive way. However, the knowledge about the principles and goals of metadata creation and the expectations were very different. Some expected a larger amount of funding support and did not see that metadata is not meant to clean up the state of their repositories.

Most of the data centers that finally participated were aware of the relevance and concept of metadata. Therefore, there was no need for intensive training programs. However, since these centers with large corpora were already using header type of information or some internal database, it was not evident for them that IMDI not only requires metadata records. To create a browsable domain as well it is necessary to create a linked hierarchy of metadata descriptions and meaningful nodes that represent abstract concepts such as "language", "genre" and "age". It would be possible in IMDI to just deliver metadata records, simply create one node representing the institution and link all descriptions to this one node. But that would lead to long and unstructured lists that are not useful for browsing. To help creating such meaningful hierarchies programs would be necessary to create abstractions from the metadata descriptions semi-automatically.

The experiences with projects and institutions in the ECHO project were different. Here training courses and introductions were necessary to inform the researchers about all aspects of standardized metadata. In general these groups had to start from scratch, since they did not work with formal metadata beforehand. Metadata creation then means a considerable amount of work, since interviews are required and analysis work is needed to fill in the values for the metadata elements.

In special cases such as the Sign Language community a discussion process was initiated that led to additional categories that were absolutely necessary. Only with categories such as "Father.deafness." metadata would be easily exploitable by the members of that specific

[12] European Cultural Heritage Online: http://www.mpi.nl/echo

[13] ENABLER: http://www.enabler-network.org/

community. Therefore, the concept of project or community specific profiles was introduced.

Problems

The efforts needed to create metadata descriptions varied considerably as well as the available skills to write scripts to semi-automatically create basic information that can be enhanced manually. Although the IMDI infrastructure offers an editor with useful options to increase the efficiency such as storing and re-using blocks of information, manual metadata creation is very time consuming and often not feasible.

The experience showed that it is much more easier to use spreadsheet tools such as EXCEL for researchers to create and manipulate a large set of records. The same is true for experienced people that prefer to use scripts to create the metadata records. However, these techniques in general create metadata of bad quality. The following types of problems were encountered:

- There is no guarantee that scripts produce well-formed XML files.
- The character encoding often is not UNICODE.
- Most problematic is that the tools used do not provide support for the controlled vocabularies leading to typo errors, spelling variants and many others.

It is the service provider who has to invest time to check the correctness of the produced metadata records and to improve the metadata records in collaboration with the data providers. The OAI[7] model for metadata harvesting only requires a validation at the moment of registration and simply points to the errors. This may in general not be sufficient, without additional help many of the data providers would stop.

Improving the content of the metadata descriptions is very important for successful searching. Two phenomena can be observed: (1) Since metadata creation is a hard job, even in evident cases elements are not filled in. (2) As already indicated all kinds of variations can be found, since the creators partly do not make use of controlled vocabularies.

First, in a very large collection it is a problem to identify such errors or missing values. Second, how to correct them without starting time consuming interactions with the various data providers. To detect errors and variants it makes sense to first run a validation against the controlled vocabularies. Until now, however, the errors have to be corrected manually. Methods that use a formal closeness (one character difference) or other type of heuristics were not yet applied. Variants that occur due to language differences (for example Afrique, Afrika, Africa) could be corrected if one would have suitable online dictionaries or terminology databases.

Third, filling in empty elements is even more difficult, since there can be many reasons why elements were not used. Until now these cases were identified by accident, i.e., someone inspecting metadata records, finding that for example the country is filled in but not the continent. A script using geographic thesaurus information could very easily add information in such a simple case. If the “genre” field, however, is not filled in there is no simple chance to identify this except by producing long lists. Still it would not be evident how such fields have to be filled in, since only the researchers can do this.

Another aspect that was found during the metadata creation work is that many institutions are looking for institutions that can store there collections. They don’t have the human resources to organize them and maintain them in a proper state so that they can be used by others. So we need ingest tools that easily allows researchers to hand over their data to another institution in an easy way. At the MPI such a system is currently in work. Ingestion will be tightly combined with metadata creation.

Future

Much effort is taken to create and maintain metadata descriptions and it is expected that projects such as INTERA and ECHO will help to increase the awareness that metadata is very important. Therefore, we have to assure that the investments will be maintained over a long period.

All IMDI categories were registered within the emerging ISO TC37/SC4 data category repository. In doing so semantic definitions are carried out in a widely agreed and machine readable way. It is expected that also OLAC and TEI categories will be entered in the same way. This would give all definitions a higher degree of stability. It would also allow us to make the semantic mapping between the categories explicit. It would also open the possibilities that researchers create their own mappings between categories and even develop own metadata sets by re-using the existing and well-defined categories.

It is expected that creating metadata will also become more attractive when new applications will become available. The INTERA project has as one other goal to link the domain of language resources with that of tools that operate on such resources. The MIME type concept is not new, however, the requirements go far beyond this. Bundles of resources have to be processed by tools combining several of them in one step. Characteristics of resources such their annotation schemes are relevant to detect the most useful tool. Within the INTERA project an interaction between the IMDI domain and the ACL tool registry[14] is being developed that is based on the open Language Resource Exchange Protocol (LREP) tht is curretnly bein gdefined with the INTERA project.

Conclusions

In this paper we presented the metadata creation work in the INTERA and ECHO projects and the experiences that were made. The creation of high quality metadata descriptions in general costs more effort than was originally expected. Given that many researchers still see metadata creation as an overhead, makes infrastructure projects of this sort a difficult, but nevertheless important enterprise.

[14] ACL Software Registry: http://registry.dfki.de

A sufficiently large metadata domain is expected to become available this year. To convince other institutions and individuals to contribute to this domain more utilities have to be developed to easily create large sets of metadata descriptions, to derive corpus-structured semi-automatically and to enrich the content.

These structures can be easily distributed over different physical locations since the connections between the tree nodes are based on HTTP links and the tree nodes and metadata records are disseminated by normal web servers.

The purpose of these tree structures is to allow users to browse the available metadata and make sub selections for later metadata search.

Obviously some special sites or portals are needed to give users entry points to the metadata domain.

References

Broeder, D.G., Brugman, H., Russel, A., and Wittenburg, P., (2000), A Browsable Corpus: accessing linguistic resources the easy way. In Proceedings LREC 2000 Workshop, Athens.

Wittenburg, P. (2001) *Lexical Structures*. DOBES internal document. MPI Nijmegen.

Wittenburg, P. (2003). WP2-TR16-2003 Version 3 Note on ECHO's Digital Open Resource Area. http://www.mpi.nl/echo/tech-report-list.html

Wittenburg, P. (2004). WP2-TR17-2004 Version 1 Note on an ECHO Ontology. http://www.mpi.nl/echo/tech-report-list.html

http://www.getty.edu/research/conducting_research/standards/intrometadata/1_introduction/index.html

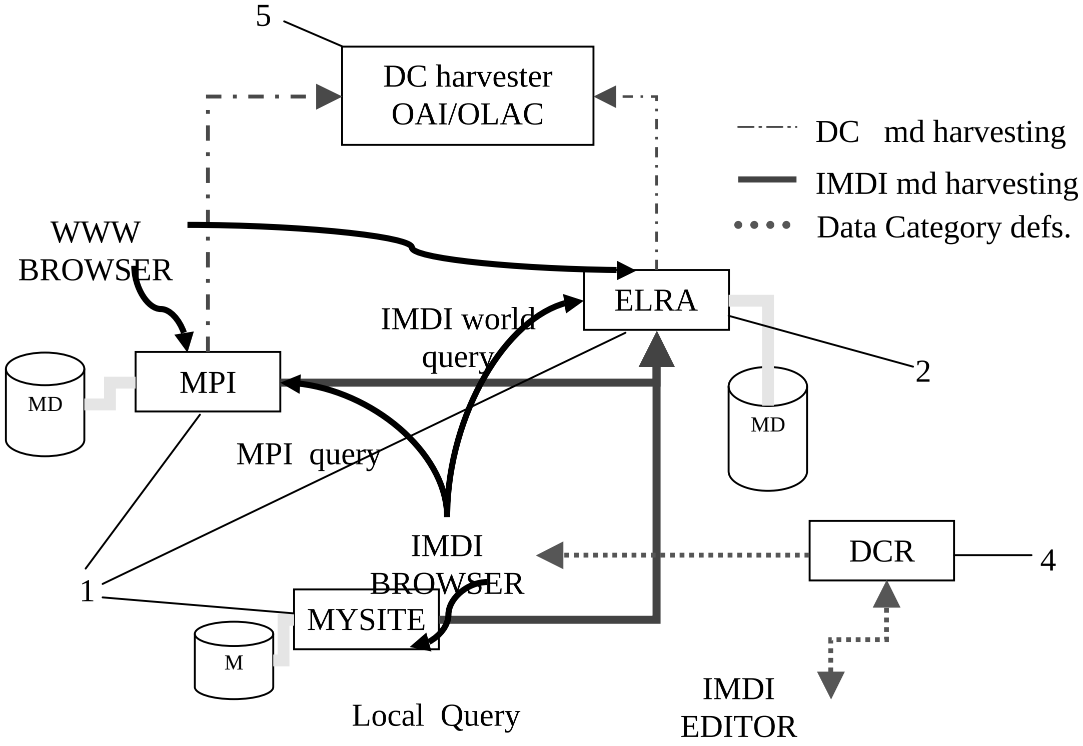


Figure 3: Intera Search Infrastructure

An on-line data category registry for linguistic resource management

Laurent Romary+, Nancy Ide^, Peter Wittenburg# and Thierry Declerck*

+Laboratoire Loria, ^Vassar Colleg, #MPI and *Saarland University and DFKI GmbH
Laurent Romary@loria.fr

Abstract

This document, consisting in an infomral merging of papers by Nancy Ide et al. and actual work done in the INTERA project, describes the principles and technical framework for deploying an international data category registry (DCR) in the domain of language resources. We demonstrate the potential usages of data categories for modelling linguistic representations and annotation schemes and describe mechanisms for localizing the descriptive content of data categories. To exemplify the use of these mechanisms, we demonstrate their application to the IMDI metadata set in the context of the Intera project. We also show the first implementation of an on-line environment that implements the DCR and is currently available to ISO sub-communities to register their own practices before being widely deployed under the auspices of ISO committee TC 37.

Introduction

The description of the metadata attached to any kind of language resource or tool cannot be based on a fully fixed set of fields. The variety of possible types of resources and tools makes it necessary to think of metadata as the combination of a reference set of descriptors (or data categories) that ensures that any two metadata sets using the same data category will be interoperable, while preserving flexibility in combining categories according to the archiving or distribution requirements of the corresponding language resources. For instance, the information describing a lexical database at ILC,Pisa will likely not be the same as the information used to describe recordings of speech data by field linguists at MPI, Nijmegen or a part of speech tagger in the ACL Natural Software Tools repository[1] at DFKI, Saarbrücken. This holds true beyond the metadata level; information included in linguistic annotation of any kind is likely to vary considerably from site to site and, especially, application to application, depending on the underlying language, context of use or theory. Therefore, any international standardizing effort must offer methods and concepts for designing linguistic formats or annotation schemes that strikes the optimal compromise between interoperability across similar applications and flexibility in a specific application.

ISO committee TC 37/SC 4 has initiated an effort to establish a data category registry (DCR) for language resources. The effort implements a combined strategy that relies on general principles for the design of linguistic format are stated in the Linguistic Annotation Framework project (Ide & Romary, 2003; Ide, et al., 2003) together with a general infrastructure for registering and disseminating data categories at an international level. This strategy is intended to allow the implementer of a language archive to compile his/her own data categories by either choosing those available in the DCR or defining his own using standard mechanisms and formats, as well as to compare site-specific categories with those available in the Registry.

This paper demonstrates how the ISO infrastructure can be implemented in a variety of applications, and in particular, its implementation in the Intera project[2] for standardizing metadata descriptors. The paper addresses the following issues:

- The possibilities for use of the framework that have been approved in ISO TC 37/SC 3 (ISO CD 12620-1) in order to provide localization mechanisms to cover a wide variety of user needs;
- The adaptation of the IMDImetadata set[3] to this framework, identifying the information in the IMDI specification can be identified as data categories (in the sense of ISO 12620-1) and how it can be instantiated in the standardized format;
- The localization of names given to the IMDI metadata set;
- The on-line environment for browsing the defined categories, through which all results from the Intera project are made available.

Using data categories to describe language resources

By definition, a *data category* is an elementary descriptor that can be used to specify and implement a linguistic annotation scheme in the broadest sense, which includes:

- descriptive information attached to a language resource or tool (metadata) as well as information used to describe linguistic features at any level (morpho-syntactic, syntactic, discourse, etc;

[1] See http://registry.dfki.de
[2] See http://www.elda.fr/rubrique22.html
[3] See http://www.mpi.nl/IMDI/

- information concerning the provenance of the annotation, e.g., whether it was produced manually (e.g., via hand annotation or transcription of spoken data) or automatically (e.g., as the output of a POS tagger),
- indication of whether the descriptor is a placeholder for some value (e.g. /grammatical gender/, /content modality/) or a possible value for a placeholder (e.g. /masculine/ or /speech/).

In the context of the description of linguistic format, the role of data categories is two-fold.

- First, they provide a uniquely identified reference for implementers, who can utilize the data categories through, for example, the use of Formal Public Identifiers, in order to ensure immediate interoperability.
- Second, the data categories in the registry serve as documentation for an annotation scheme, by providing all the necessary information (definition, examples, etc.) to make the semantics of a given data category as precise as possible.

A major issue to be raised in this context is that of the specificity of a given data category with regards the linguistic context. One basic assumption that has been agreed upon within ISO committee TC 37 is that the implementation of an international data category registry (DCR) should find the right balance between generalization (a data category such as /grammatical number/ is applicable to a large a number of languages), and precision in the applicability of a data category to a single language (e.g. only the two values /singular/ and /plural/ are applicable to the German language, whereas other languages may allow for /dual/, /trial/ or /paucal/ (a few)). The next section describes how this issue is related to that of localization, and makes explicit the interrelation between these two requirements.

Another issue concerns the need to provide a mechanism through which annotators can select from the data category registry. For this purpose we propose an on-line browsing tool that allows the user to create proprietary data category selection (DCS) from the categories available in the DCR, in particular those implemented from the IMDI specification.

Representing data categories

Foreword on object language and working language

When dealing with linguistic data, whether it is expressed in a database or semi-structured document, it is often necessary to identify the *working language* applicable to the data—that is, the language the data itself utilizes. This information can be used to make the right layout choices for presentation (e.g. hyphenation practices), or to select the appropriate spellchecker in an editing environment.. In the case of XML documents, the World Wide Web Consortium (W3C) has introduced the `xml:lang` attribute to identify the working language of a document or document fragment, which enables exploitation of the hierarchical XML information structure and the associated rules of inheritance over embedded XML elements to control scope[4]. ISO 16642, which defines a standard Terminology Markup Language (TML)[5], has adopted this attribute as one of the fundamental mechanisms that should be used in any TML compliant to the standard. For example, the following uses ISO 639-1 country designations to identify the working language of a definition:

```
<feat type="definition" xml:lang="fr">Une valeur entre 0 et 1 utilisée...</feat>
```

In addition, the data may itself include information *about* languages, either because it describes a language, exemplifies some properties of a language, or provides further information about a language sample. This *object language* may be different from the working language used to convey the information, as in the case of grammar books or lexical descriptions.

As defined by the TML, terminological data collections include a “language section level” in the meta-modelthat is specifically intended to organize the information contained within terminological entries into blocks dedicated to specific object languages. It should thus be systematically associated with a language marker, which in turn should be distinguished from any other working language indication. ISO 16642 considers that the object language indication is itself a data category[6] which is mandatory at language section level. For example, the following shows a language section whose object language is English (it describes English terms), but whose working language is French:

```
<struct type="LS" xml:lang="fr">
     <feat type="language identifier">en</feat>
     <feat type="definition">Une valeur entre 0 et 1 utilisée...</feat>
             <struct type="TS">
             <feat type="term" xml:lang="en">alpha smoothing factor</feat>
             <feat type="term type">fullForm</feat>
     </struct>
</struct>
```

The preceding example demonstrates two phenomena. First, any working language can be superseded at a deeper position in the XML representation by a new marking. This is the case when an English term is provided, where the working language should obviously be English. Second, the working language only applies to linguistic data, and not to other information such as numbers or dates, and, in particular, it does not apply to code identifiers such as ‘fullForm’ in the example above. ‘fullForm’ is the identification of a value described in ISO 12620, which should not be treated as linguistic information (for instance, one should not apply a spell checker to such a field).

[4] That is, the scope of the xml:lang attribute includes all the attributes and descendants of the element where it appears.

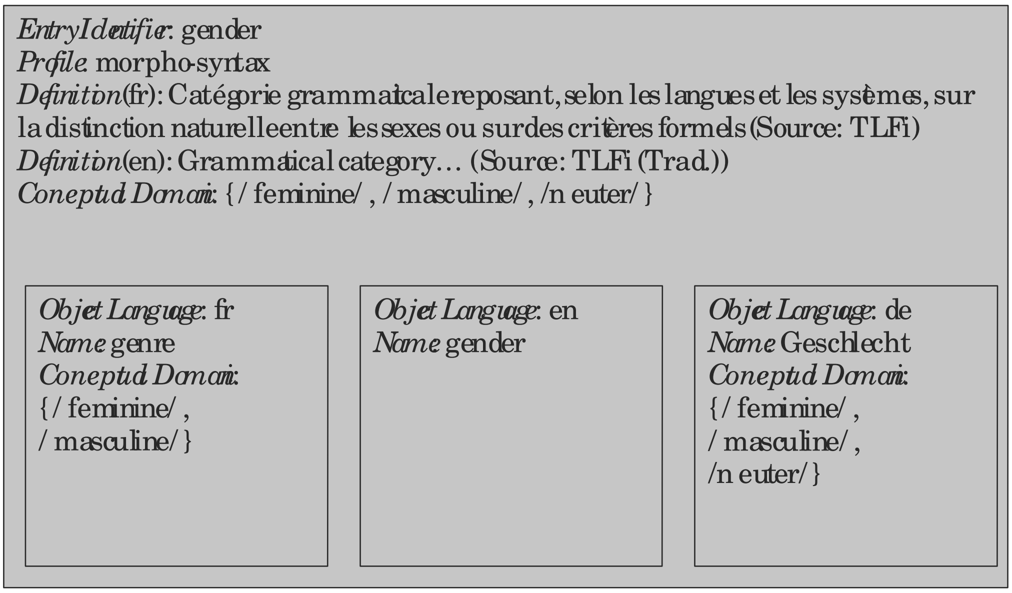
[5] See http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=32347&ICS1=1

[6] /Language Identifier/ in ISO 12620.

The descriptive structure of a data category

Figure 1 shows the overall organization of the descriptive portion[7] associated with a data category. As can be seen, a data category is represented as a concept that is described at two different levels of granularity:

- at the higher level, the data category is seen as a unitary concept covering all its possible usages across languages. It is uniquely related to an identifier and contains descriptive information (definition, explanation, notes, and examples) that is valid for all those usages. Two important fields in the description are *profile*, which indicates the domain of activity in the field of language resources to which this category can be applied (in the case of the IMDI set, the profile is set to 'metadata'); and *conceptual domain,* which lists the possible values that his field can take, independent of its applicability to a certain language.
- the lower level contains language-specific sections ('language section' in ISO CD 12620-1), which provides information concerning the implementation of the data category for a given language. At this level, a more precise definition can be provided for the data category together with specific examples, and, where applicabl,e a subset of the main conceptual domain. Interestingly, it is also possible to give the 'name' of the data category for this language, which is the main entry point for localization that has been chosen for the IMDI metadata set in INTERA.



EntryIdentifier: gender
Profile: morpho-syntax
Definition(fr): Catégorie grammaticale reposant, selon les langues et les systèmes, sur la distinction naturelle entre les sexes ou sur des critères formels (Source: TLFi)
Definition(en): Grammatical category… (Source: TLFi (Trad.))
Conceptual Domain: {/feminine/, /masculine/, /neuter/}

Object Language: fr
Name: genre
Conceptual Domain:
{/feminine/,
/masculine/}

Object Language: en
Name: gender

Object Language: de
Name: Geschlecht
Conceptual Domain:
{/feminine/,
/masculine/,
/neuter/}

Figure 1: An example of the descriptive component of a data category.

Localizing data categories

The preceding structure offers a general framework for localizing data categories:

- from an object language point of view, we can specify the meaning of a data category as it applies to given languages by specializing definitions, explanations and notes, examples for the language[8], or by subsetting the conceptual domain from the generic description to the specific;
- from a working language point of view, each available field at either the generic or language-specific level (i.e. definition, explanation and note) can be translated and re-expressed[9] in another language. For instance, it is possible to provide a Japanese note for the usage of /gender/ in the German language.

One specific way to localize data category definitions is *naming* with a name (or term) that best designates a concept in a given language. As such, naming is language specific, but it merges the two notions of object language and working language: It is related to the object language because it describes the name that can be used in this language, and it is related to the working language because the name itself is provided in the same language.

In the case of data categories for language resource metadata, there will likely be little variation of semantics between the generic level and the specific level. In addition, in the process of unifying the IMDI dataset with the OLAC consortium's metadata proposal, there is no certainty as to the stability of the definitions, explanations, or notes associated with a given data category, since its semantics may evolve when the two schemes are merged.

Therefore, localization of the names in the IMDI metadata set has been performed so far with two objectives in mind:

- facilitating the access for a wide variety of users of the metadata set;
- experimenting ISO CD 12620-1 (D3.1) through its applicability to the IMDI metadata set.

[7] The INTERA project also describes the various issues related to the administration of data categories within the DCR.

[8] In which case there is a conjunction of the object and working language axes.

[9] Under the condition that the semantic scope that is expressed strictly matches the initial or reference description.

Deploying an on-line environment for accessing localized version of the IMDI metadata set

Applying 12620-1 to the IMDI metadata set

The main difficulty in applying ISO CD 12620-1 to the IMDI metadata set, a central goal of the INTERA project, has been to map the fields available in the initial documentation onto the general framework for describing data categories. To date, the following choices have been made:

- the IMDI Identifier has been kept as the identifier for the data category;
- the initial name in English (that is, in the English language section) was determined by taking the IMDI identifier, dropping the spare stop sign, and flattening it in small letters ;
- the IMDI definition has been retained as the general definition for the data category ;
- the IMDI encoding field has been manually modified to either produce a note when expressed in plain text, or to build up the conceptual domain, when a list of descriptors was made explicit ;
- the comments field was transformed into an explanation, except when an explicit example was provided—in this case it was moved to the 'example' descriptor of ISO CD 12620-1.

To exemplify the transformation process, we include below the initial IMDI description for Content.Modalities, together with its representation according to the ISO CD 12620-1 framework.

IMDI entry :
Content . Modalities
Element: Content . Modalities
Identifier: Content . Modalities
Definition: Gives a list of modalities used in the session.
Encoding: Open vocabulary 'Content . Modalities' (4.4).
Comments: The element is not used to give an exhaustive list of all the modalities, but should be used to list the modalities that are typical for the task or of interest for the researcher.
Example: in route direction one would typically look at speech and gestures and not at eye-gaze.

ISO CD 12620-1 representation (expressed in GMT – Generic Mapping Tool):

```
<struct type="DC">
        <struct type="AI">
                <struct type="AR">
                        <feat type="identifier">Content . Modalities</feat>
                </struct>
        </struct>
        <struct type="Desc">
                <feat type="profile">Metadata</feat>
                <feat type="broader concept generic">Content</feat>
                <feat type="conceptual domain">Unknown</feat>
                <feat type="conceptual domain">Unspecified</feat>
                <feat type="conceptual domain">Speech</feat>
                <feat type="conceptual domain">Writing</feat>
                <feat type="conceptual domain">Gestures</feat>
                <feat type="conceptual domain">Pointing gestures</feat>
                <feat type="conceptual domain">Signs</feat>
                <feat type="conceptual domain">Eye gaze</feat>
                <feat type="conceptual domain">Facial expressions</feat>
                <feat type="conceptual domain">Emotionnal states</feat>
                <feat type="conceptual domain">Haptics</feat>
                <brack>
                        <feat type="definition" xml:lang="en">Gives a list of modalities used in the session.</feat>
                        <feat type="source">IMDI Part1 Metadata Elements for Session Descriptions Draft Proposal
                Version 3.02 March 2003</feat>
                </brack>
                <feat type="explanation">The element is not used to give an exhaustive list of all the modalities, but
        should be used to list the modalities that are typical for the task or of interest for the researcher.</feat>
        <feat type="example">In route direction one would typically look at speech and gestures and not at
    eye-gaze.</feat>
    </struct>
    <struct type="LS">
        <feat type="language">english</feat>
        <struct type="NS">
            <feat type="name">Content . Modalities</feat>
        </struct>
    </struct>
</struct>
```

Localizing names and definitions

To simplify the task of the various partners involved in providing localized names for the IMDI metadata set, a simplified table has been produced by applying a specific XSLT stylesheet to the ISO CD 12620-1 compliant representation. This table recalls the name in English of the IMDI category as of version 3.03 of the IMDI documentation, together with what has been identified as the definition (see previous section). The table has been filled by the various partners of INTERA to provide names in the following languages: German, Dutch, Swedish, Italian, Spanish, French and Greek.

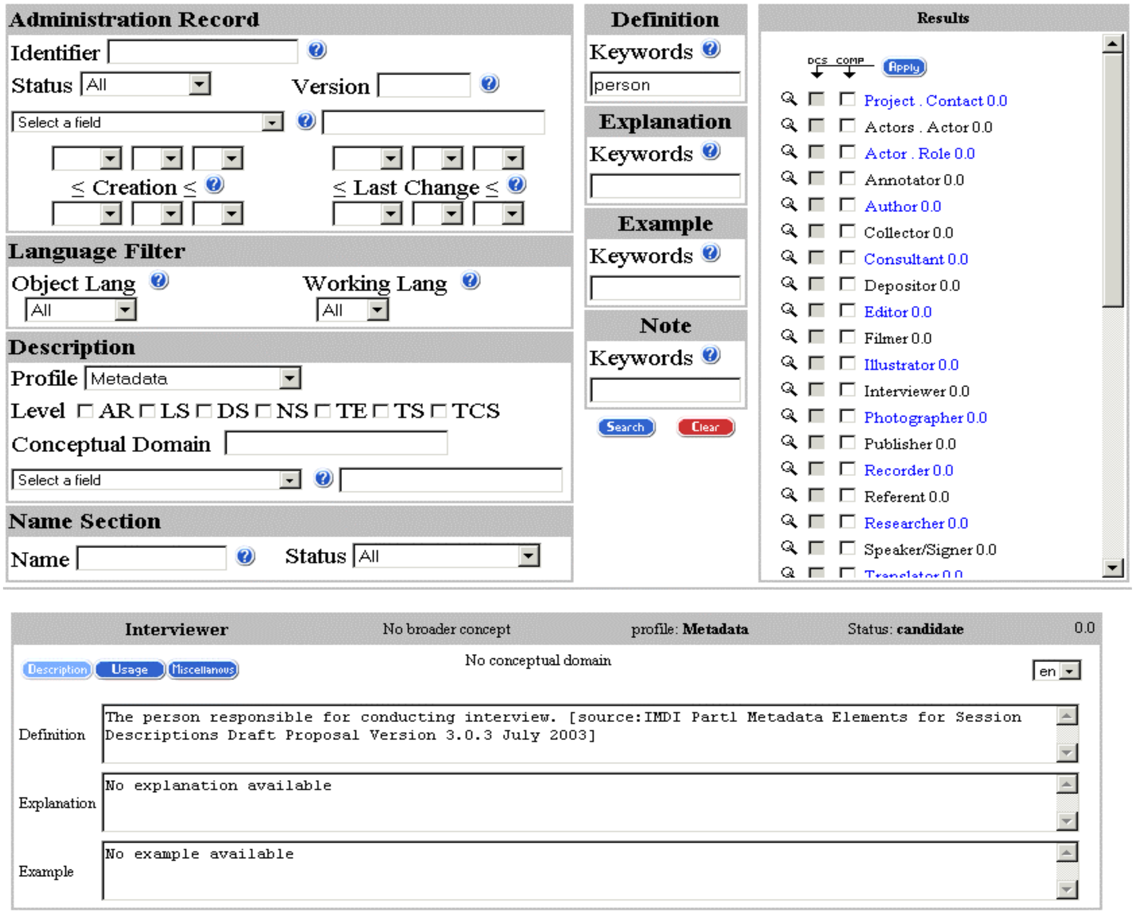
Making the IMDI metadata set available on-line

Putting the IMDI meta-data set in a standardized format is worthwhile only if it is made publicly available for use in the design of specific metadata scheme. We have therefore incorporated the result of the transformation of the IMDI documentation to an ISO CD 12620-1 compliant format into an experimental on-line tool for browsing through the data categories in ISO TC 37. This work, subsidized by INRIA in its corporate action Syntax, has been made freely available to the consortium. under http://syntax.loria.fr and accessible to Intera partners during the period of the project using the following coordinate. This environment is conceived to last far beyond the time of the project and we kindly ask regular users to ask for their own login and password as soon as possible, so that the coordinates provided above are only there for the sake of a quick initial browsing test.

In what follows we describe the main functionalities of the Syntax server when querying the IMDI metadata set, after being standardized in accordance of ISO 12620-1. Figure 2 shows the main query window of the Syntax server. The top left of the window includes a series of query fields; the top right displays the result set associated with a given query; and at the bottom there is a full display of any data category selected using the magnifier symbol in the display list.

The screenshot in figure 3 shows the search fields of the query window for a request for all data categories belonging to the metadata domain (see the Profile field), and for which the definition contains the word ‘person’. Such a query would typically correspond to a situation where a user is looking for possible data categories corresponding to the description of any kind of participant, e.g., in the course of collecting field data.

The screenshot in figure 4 shows the list of data categories that have fulfilled the initial query. In this figure one can see that it is possible to ask for the full description of the data category (magnifier icon), to add the data category to one’s own selection (DCS (Data Category Selection)) or to compare two existing data categories (COMP). This last functionality can be particularly useful when one wants to merge two existing proposal, as contemplated in the case of IMDI and OLAC.



Definition Keywords: person

Results:
- □ □ Project . Contact 0.0
- □ □ Actors . Actor 0.0
- □ □ Actor . Role 0.0
- □ □ Annotator 0.0
- □ □ Author 0.0
- □ □ Collector 0.0
- □ □ Consultant 0.0
- □ □ Depositor 0.0
- □ □ Editor 0.0
- □ □ Filmer 0.0
- □ □ Illustrator 0.0
- □ □ Interviewer 0.0
- □ □ Photographer 0.0
- □ □ Publisher 0.0
- □ □ Recorder 0.0
- □ □ Referent 0.0
- □ □ Researcher 0.0
- □ □ Speaker/Signer 0.0
- □ □ Translator 0.0

Interviewer | No broader concept | profile: **Metadata** | Status: **candidate** | 0.0

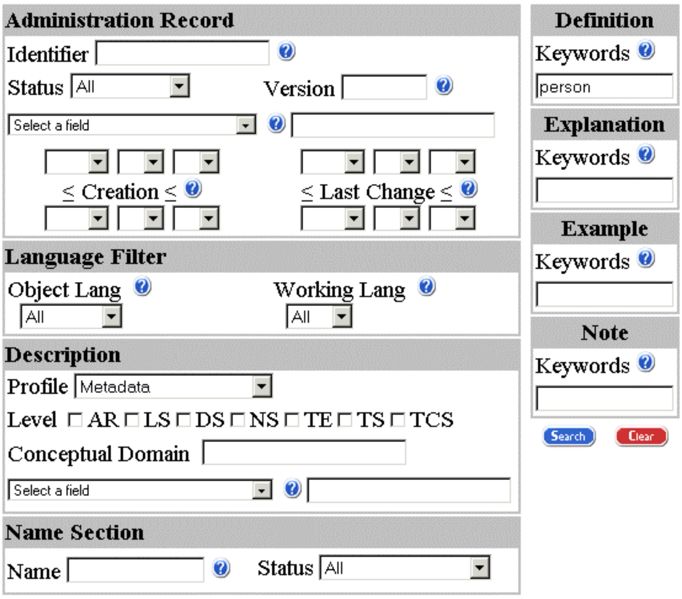
No conceptual domain

Definition: The person responsible for conducting interview. [source:IMDI Part1 Metadata Elements for Session Descriptions Draft Proposal Version 3.0.3 July 2003]

Explanation: No explanation available

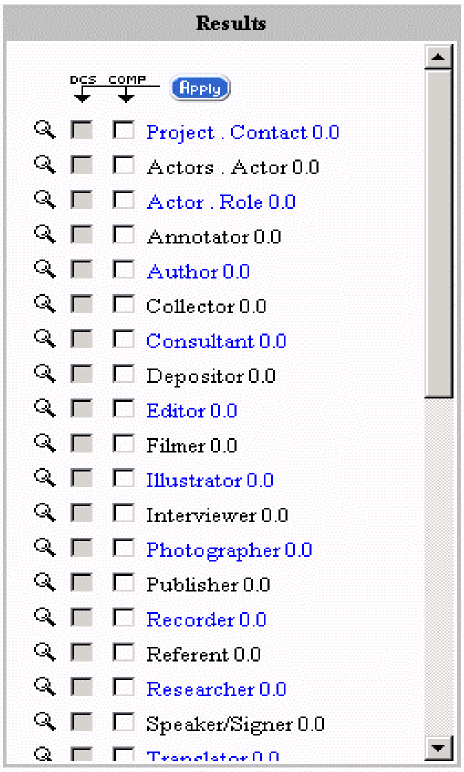
Example: No example available

Figure 2: Overview of the Syntax query interface for a request on the IMDI metadata set



Definition Keywords: person

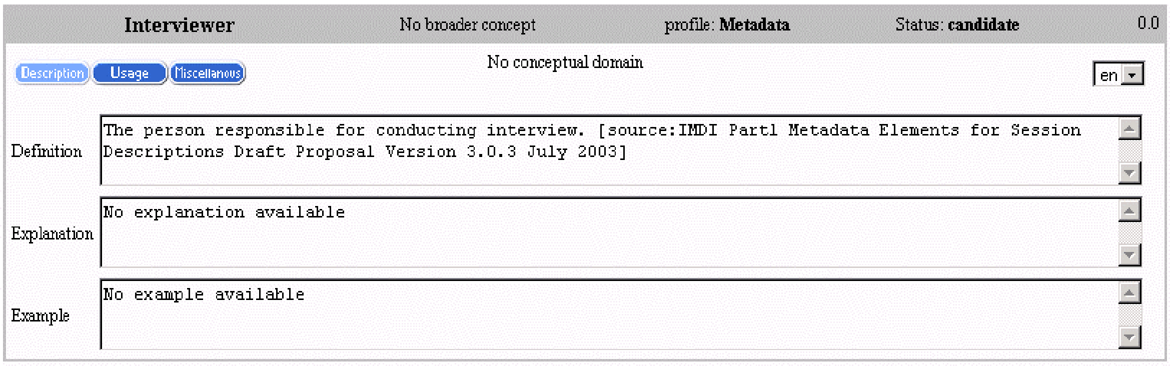
Figure 3: the fields available for querying data categories



□ □ Project . Contact 0.0
□ □ Actors . Actor 0.0
□ □ Actor . Role 0.0
□ □ Annotator 0.0
□ □ Author 0.0
□ □ Collector 0.0
□ □ Consultant 0.0
□ □ Depositor 0.0
□ □ Editor 0.0
□ □ Filmer 0.0
□ □ Illustrator 0.0
□ □ Interviewer 0.0
□ □ Photographer 0.0
□ □ Publisher 0.0
□ □ Recorder 0.0
□ □ Referent 0.0
□ □ Researcher 0.0
□ □ Speaker/Signer 0.0
□ □ Translator 0.0

Figure 4: the result set for all entries in the IMDI data categories for which the definition contains the string ‘person’

Finally, figure 5 shows the full representation of a data category (here Interviewer) with the possibility to have access to the conceptual domain (when applicable), or the namings in available languages.



Interviewer | No broader concept | profile: **Metadata** | Status: **candidate** | 0.0

No conceptual domain

Definition: The person responsible for conducting interview. [source:IMDI Part1 Metadata Elements for Session Descriptions Draft Proposal Version 3.0.3 July 2003]

Explanation: No explanation available

Example: No example available

Figure 5: the visualization of the /interviewer/ data category

Perspectives

The work done in Intera in relation to the standardization and localization of the IMDI metadata set is obviously only a step in the direction of providing an international reference set of metadata categories for language resources and tools. The next step is to use the framework presented in this report not only as a basis for comparing and possibly merging the descriptors offered by the IMDI and OLAC initiatives, but also to relate these to other metadata schemes such as the Dublin Core[10] and the TEI (Text Encoding Initiative) header. To accomplish this broader goal, it will be necessary to extend the simple browsing tool described here to enable submission of data categories on-line, and review of the submissions by a committee of worldwide experts (as described in ISO CD 12620-1). This extension is planned for implementation within WP3, in parallel with the final phase of the project.

References

Ide, N., Romary, L. (2003). Outline of the International Standard Linguistic Annotation Framework. Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model Right, Sapporo, 1-5.

Ide, N., Romary, L., de la Clergerie, E. (2003). International Standard for a Linguistic Annotation Framework. Proceedings of HLT-NAACL'03 Workshop on The Software Engineering and Architecture of Language Technology, Edmunton.

Martin, L.E. (1990). Knowledge Extraction. In Proceedings of the Twelfth Annual Conference of the Cognitive Science Society (pp. 252--262). Hillsdale, NJ: Lawrence Erlbaum Associates.

[10] The OLAC metadata set is actually described as a refinement of the Dublin Core format, and the IMDI set points to the Dublin Core when applicable. There are on-going discussions under the auspices of ISO/TC 37/SC 4 to work together with the TEI to go towards a wide coverage set of metadata descriptors.

DATA CATEGORIES IN LEXICAL MARKUP FRAMEWORK OR HOW TO LIGHTEN A MODEL

Gil FRANCOPOULO AFNOR-INRIA gil.francopoulo@wanadoo.fr
Monte GEORGE ANSI dracalpha@earthlink.net
Mandy PET ANSI-ORACLE mandy.pet@oracle.com

1 Introduction

Previously, ISO TC37 efforts have focused on standards and associated models for dealing with language resources such as terminologies but up until now focus has not been on the various other aspects of language processing. The Lexical Markup Framework (LMF), a proposed standard numbered ISO-24613, addresses lexical resources at a higher level that allows for interoperability with terminological, human oriented lexical resources and machine-based NLP lexica. LMF relies heavily on the ISO-12620 data categories (DC), managed under the auspices of the ISO-12620 revision by Laurent Romary (AFNOR-INRIA). They serve as lego building blocks used to facilitate this operability.

We will see how the DC ease the definition and use of various norms and particularly lexical models.

2 Current situation

Traditionally, concerning linguistics constants, the two following strategies are applied:

Strategy #1:
The lexical model defines the list of all the possible values for a certain type of information. For instance, /gender/ could be /masculine/, /feminine/ or /neutral/.
More precisely, there are two sub-strategies:

- define that /gender/ is /masculine/, /feminine/ or /neutral/ without any more details.
- define that /gender/ is /masculine/ or /feminine/ for French and /masculine/, /feminine/ or /neutral/ for German.

Strategy #2:
The values are not listed at all. The model just states that there is the notion of gender.

An example of the first strategy is applied in the GENELEX [Antony-Lay] and EAGLES models where the DTD contains all the possible values. The drawback of such an approach is that the DTD is necessary huge and could be incomplete, specially for languages unknown to the model authors.

The advantage of the second strategy is that the model is simple and nothing is forgotten. But its drawback is that such a model is useless and we will see that in the next paragraph.

3 Capacities

For a lexical model, we can distinguish two criteria:

- The power of representation: what kind of data the model is able to represent ? what language the model could be applied to ?

- The power of operation: is it possible to compare two words ? how to present a pick list to a user of an interactive workstation ? is it possible to merge two LMF conforming lexica ?

The two criteria are somehow contradictory: the more generic the approach, the more diverse lexica are needed to merge.

Coming back to the second strategy that is to avoid defining the possible values for gender, the power of representation is high but the power of operation is very low. Nothing guarantees that a lexicon defines gender as /m/ and /f/, or /mas/ and /fem/ or worth /neuter/ for French. **In such a situation, comparing words or merging various lexica are difficult operations and the norm becomes useless**.

4 Merging

Let’s detail a bit what is merging.

Merging can take various forms such as the following use cases:

Use Case #1
Situation: Multilingual lexicon in N languages
Goal: Add 1 new language to this lexicon

Use Case #2
Situation: Monolingual lexicon in language L
Goal: Add words in language L

Use Case #3
Situation: Multilingual lexicon in N languages
Goal: Add missing translations

Let’s add that merging is a frequent operation and is an heavy burden for the lexicon manager.

5 Solution

The solution is not easy. We must represent existing data and due to the extension of multilingual databases and various formats used, merging seems to be the most demanding operation.

There is another point to be mentioned. This problem is not specific to lexicon management. The gender definition is shared by other processes like text annotation and features structures.

That means that:

- It is not very wise to duplicate the effort in various norms.
- Text annotation, features structure coding and lexical representation are not independent processes. In case of parsing for instance, the information extracted from the lexicon will be transferred to annotation or feature structures, there is the danger to produce different (and so incompatible) values.

The solution is to define data categories in a separate norm. These values will then be shared by the lexicon, annotation and features structures norms. And of course other future norms could take place in this architecture.

6 Details

The data categories are not only constants like /masculine/ preferred to /m/ or /mas/ but are defined according to the language processed.

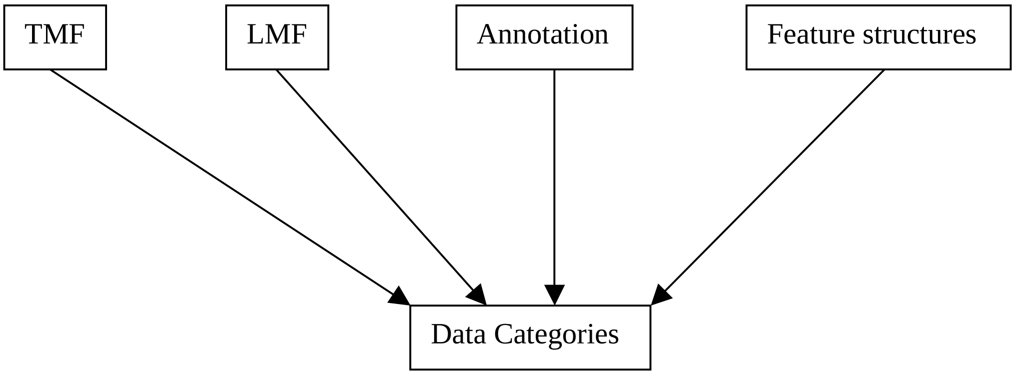
More precisely each feature will be defined as a tree. The top node is /gender/ for instance. One level below, we have /french/ and the possible values are /masculine/ and /feminine/. At the same level as /french/, we have /german/ and the possible values are /masculine/, /feminine/ and /neuter/.
For an unknown language, the possible values are the union of all values extracted from all languages.

As it could be noticed, the number of values is quite important. A management tool is needed in order to ease data category search and selection. Such a tool is provided by INRIA under the auspices of the Syntax project.

7 A family of norms

The process used is similar to the one of TMF (aka Terminological Markup Framework) that is the ISO norm for thesaurus [Romary].

Data categories are located at the lower level of the TC37 family of norms as sketched in the following diagram.



And the four norms are based on data categories, so each norm is light, non redundant and can interoperate with the others.

8 Conclusion

Like the other norms of the family, the base line for LMF is to:

- Concentrate on structuring the elements and linking elements together.
- Relegate language idiosyncrasies in an external and shared norm: ISO-12620.

As we have seen, LMF is part of a more global ISO move in order to define a set of coherent norms based on data categories.

Bibliography:

Antoni-Lay M-H., Francopoulo G. and Zaysser L. 1994
A generic model for reusable lexicons: The GENELEX project.
Literary and Linguistic Computing 9(1): 47-54.
Romary L. 2001
Towards an Abstract Representation of Terminological Data Collections – the TMF model.
TAMA. Antwerp.

The MILE Lexical Classes: Data Categories for Content Interoperability among Lexicons

Francesca Bertagna[1], Alessandro Lenci[2], Monica Monachini[1], Nicoletta Calzolari[1]

[1]Istituto di Linguistica Computazionale (ILC) – Consiglio Nazionale delle Ricerche
Via Moruzzi 1, 56100 Pisa, Italy
(francesca.bertagna, monica.monachini, nicoletta.calzolari)@ilc.cnr.it
[2]Dipartimento di Linguistica, Università degli Studi di Pisa,
Via S. Maria 36, 56100 Pisa, Italy
alessandro.lenci@ilc.cnr.it

Abstract

Addressing the issue of content interoperability among lexical resources, the paper aims at testing the expressive potentialities of MILE (Multilingual ISLE Lexical Entry) as a standard environment for Computational Lexicons. An experiment of mapping differently conceived lexicons, FrameNet and NOMLEX, to the MILE general schema of shared and common lexical objects. In order for not being only isolated exercises and promoting such kind of initiatives, a proposal for a set of Data Categories, which represent in RDF the common/shared lexical objects of the MILE semantic layer is formulated. This set, developed on the lines of the already existing RDF schema for the syntactic layer, is intended to be submitted to the ISO TC37/SC4 for evaluation and approval.

1 Introduction

This paper represents an attempt to further elaborate the mapping experiment presented at the LREC Conference in Bertagna *et al.* 2004, where differently conceived lexicons are mapped to MILE. We tried to push forward the potential of MILE as a common standard framework for lexical encoding, by proposing a set of Data Categories, which instantiate the MILE semantic lexical objects in RDF. The advantages and potential to develop a Data Category Registry (DCR) are well known (Ide *et al.*, 2003). The classes and properties defined here correspond to the E-R diagrams described in Calzolari *et al.*, 2003 for the semantic layer. They are developed on the lines of the already existing ISLE RDF schema for the syntactic layer, with the intent to increase the repository of shared lexical objects with those objects necessary for the representation of semantic information. The RDF schema for semantics is presented here in Appendix, whereas, as far as the syntactic schema is concerned, the reader is referred to Calzolari *et al.*, 2003. It should be noted that the RDF statements provided here comply with the goals of ISO TC37/SC4 for Language Resource Management.

1.1 RDF and Interoperability

"Interoperability" is meant as exchange and integration of information between systems (Vckovski, 1999). While HTML and XML allow the access and interchange of data at the formal and structural level, a metadata representation language like RDF/S (further extended with ontology formalization capabilities, e.g. in OWL or DAML+OIL) is expected to enable a new and unprecedented progress towards content interoperability among resources. Such is the main vision of the Semantic Web: a wealth of new possibilities stemming from representing documents and data semantics with metadata defined within ontologies, which will be easy for a machine to interpret and make use of in an intelligent way (Lassila, 1998). Computational lexicons are repositories of syntactic and semantic information. Recently, there have been various efforts to translate existing lexical resources in RDF/S or in DAML+OIL, in the attempt to make their content available in the Semantic Web for various future applications (see Narayanan et al., 2002; Melnik&Decker at www.semanticweb.org/library). However, there is a concrete risk for these experiments to become mere conversion exercises, unless they are backed by an additional framework providing a common/shared compatible representation of lexical objects. Actually, in order to reach a truly content interoperability, intelligent agents must be provided with the possibility to manipulate the objects available in different lexical repositories understanding their deep semantics. This would entail, for instance, that applications should be enabled to understand whether two lexical objects are of the same type so that the same operations can be applied to them. In the paper we will tackle the issue of content interoperability among lexical resources by presenting an experiment of mapping differently conceived lexicons (in the particular case FrameNet and NOMLEX) to a general schema of shared and common lexical objects. The schema adopted in this experiment is MILE (Multilingual ISLE Lexical Entry), a meta-entry for the encoding of multilingual lexical information (Calzolari *et al.*, 2003) developed within ISLE[1] (International Standards for Language Engineering). The aim of the experiment is to evidence problems and collect hints that may emerge while mapping lexicons against an abstract model, while testing the expressive potentialities of the MILE as a standard for computational lexicons.

2 MILE

The MILE Lexical Model (MLM) is described with Entity-Relationship (E-R) diagrams defining the entities of the lexical model and the way they can be combined to

[1] ISLE was an initiative under the FP5 within the EU-US International Research Co-operation, with the aim to develop and promote widely agreed on Human Language Technology standards and best practice recommendations for infrastructural LRs.

design an actual lexical entry. MLM defines a first repertory of "MILE Lexical Classes" (MLCs), which formalize the main building blocks of lexical entries. The MLCs are defined on the basis of an extensive survey of major existing practices in lexicon development. MLCs form a "top ontology of lexical objects", as an abstraction over different lexical models and architectures. The MLM defines each class by specifying its attributes and the relations among them. Classes represent basic lexical notions. Instances of MLCs are the "MILE Data Categories" (MDCs), each of them identified by a URI. MDCs can be either user-defined or reside in a shared repository. Part of the class structures in the MLM has been formalized as a RDF Schema, and data categories have been created using RDF and OWL (Ide *et al.*, 2003).

3 The Mapping Experiment

Two main methodological scenarios concerning the mapping may be envisaged.

(1) The first implies to resort to a high level mapping of the elements in a lexicon onto the MILE lexical objects. This is similar to the proposal in (Peters et al 1998), i.e. a common object model, sitting on top of the resource-specific models, which allows a uniform access procedure for all the resources. In this approach, the expert of the specific lexicon takes a number of decisions concerning the mapping between the linguistic information in the lexicon and the set of available lexical objects in the abstract model. One of the main advantages of such a solution is that resources would retain their native structure, without being submitted to format conversion.

(2) In the second approach, the possibility provided by MILE of creating instances of the lexical classes can be exploited to create lexical entries directly in MILE, which thereby acts as a true interchange format.

The most appropriate mapping strategy clearly depends on the possible applicative scenarios in a distributed and open environment requiring lexical resources content interoperability. The first approach is actually most promising for a "smart" access to lexical repositories. In this sense, mapping the resource data model onto a common schema provide with an explicit formal characterization of object semantics would easy the off-line processing of extracting the required information.

On the contrary, the second approach would be more suitable for the purpose of managing, integrating and merging lexical information residing in different repositories. Creating lexical entries in an MILE-like schema would be a way to make available the semantics of each lexical entry in a fully explicit way, allowing intelligent computational agents to exploit it in inferential systems and knowledge-intensive applications. In what follows, we will present some preliminary results of the experiment we have undertaken to map FrameNet and NomLex onto MILE. We preferred to perform the mapping at lexical object level (following strategy (1), since it is expected that, once the mapping conditions are formally and totally explicitly defined, the conversion at the entry level would follow naturally.

3.1 FrameNet to MILE

Our first experiment concerns the possibility to map the FrameNet (FN) architecture to MILE.

In this paper, we have preferred not to involve in the mapping experiment two other important lexicon models: the WordNet "family" and the PAROLE/SIMPLE lexicons. From the beginning, one of the requirements for the standard was to perfectly represent WordNet notions of *synset* and semantic relations. In this sense, mapping WordNet to MILE is more straightforward and the interested reader can have an exemplification of it in (Lenci, 2003). At the same time, being the MILE architecture grounded on the GENELEX model, it perfectly adheres to SIMPLE. Representing FrameNet with the expressive modalities of MILE is a more difficult task.

FrameNet (Baker *et al.*, 1998) is an important reality in the lexicon scenario and its linguistic design offers original features the standard has to deal with. The notion of Frame as such doesn't belong to the classes provided by MILE. Moreover, Narayan et al. (2002) offer us a ready set of DAML+OIL classes representing the FrameNet notions to work on. We will try to map the Frame, the Frame Element (FE) and the Lexical Unit (LU) on the correspondent MILE classes. The following picture shows how a certain degree of correspondence is possible.

The Frame can be represented by the MLC Predicate, the FrameElement by the Argument and the Lexical Unit by the SemU.

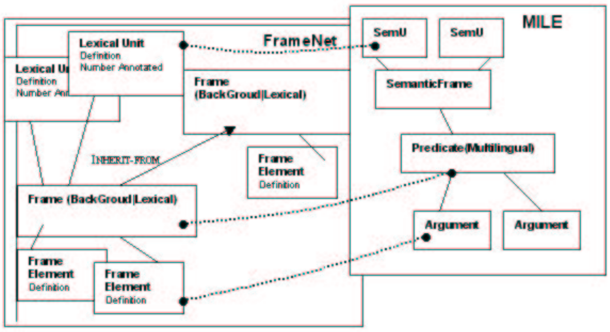


Fig. 1: Mapping FrameNet Lexical Objects to MILE

The Frame is an extended and complex structure of knowledge evoking what we may call "actantial scenarios" and playing the central role in the design of the resource. The Frame Elements are the “actants”, the entities playing a part in the *scenario* evoked by the Frame. So, the Frame “Getting” represents a situation where “a *Recipient starts off without the Theme in their possession, and then comes to possess it* etc..”. In this situation, the Frame Elements are the Recipient, the Theme and others. In MILE the notion that better expresses this same information is the Predicate. It can be *lexical* or *primitive* and it is linked to Arguments by means the *hasArgument* relation. If we want to use the Predicate to represent the Frame we have to choose its Primitive (non-lexical) modality. The MILE notions of Predicate, Argument and SemU are flexible enough to be interpreted in a narrower or in a wider way: the Predicate can be more close in size to the subcategorization frame or more extended and close to the notion of Frame and of *scenario*. In FN, the set of FE types is open in order to better fit the specific needs of the different Frame. Even though providing a recommended set of possible values for the Thematic Roles of the Arguments (derived from

SIMPLE), MILE allows the user to independently choose the most appropriate values; in this way, the MLM allows the representation of the open set of Frame Element names. In mapping FrameNet onto MILE, some mismatches of formal nature emerge. For example, while in FrameNet the Lexical Unit is directly linked to the Frame, in MILE the Predicate is inserted in a more general class, the SemanticFrame, which is in between the SemU and the Predicate. It specifies the predicative argument structure of the lexical entry and de facto contains the Predicate (with its arguments) and the type of link between the SemU and the Predicate, expressed by means of the attribute *TypeOfLink*. As a matter of facts, different words belonging to different POSs may share the same predicate in the predicative representation[2]. The problem is that while in MILE the specification of the *TypeOfLink* is not optional, in FrameNet the nature of the link between the Lexical Unit and the Frame is underspecified (so we find in the same group the Lexical Units *to acquire*, *to gain*, *acquisition_act* etc., all sharing a membership to the same Frame GETTING). A possible solution is to add a new value (Underspecified) for the *TypeOfLink* attribute in MILE. A more serious problem consists in the lack of any inheritance or embedding mechanisms for the MILE Predicate. In FrameNet two types of relations among frames are possible: first of all, the various frames can be organized in a hierarchical way, exploiting a sort of IS-A relation among the frames: "*if frame B inherits from frame A, then B elaborates A, and is a subtype of A.*" (Narayanan et al., 2002). Moreover, a kind of sub-type relation can be established among a complex frame and several simpler frames (the so-called subframes). These important features of FrameNet cannot be represented using MILE: under this point of view, we can state that a complete "translation" from FrameNet to the standard cannot be successfully achieved. The *modularity* of the MILE, however, may be an answer to this problem: it would allow the addition, for instance, of a new object PredicateRelation to the LexicalModel. Even without the availability of a specific class SubPredicate, MILE would be able to represent the semantics of a predicate considered a part/sub-type of a more complex and articulated Frame. By envisaging specific relations among predicates, it would also be possible to express the temporal ordering among the frames (another information we can find in FN). In the next future, we would like to verify if also the FN strong correlation between lexical entry and corpus evidences (by means of annotation) is representable using MILE devices. We will discuss later of this aspect but surely the *flexibility* of the model (i.e. its being open to adaptations and improvements without changing the existent) is an important feature a standard should have in order to represent new linguistic notions and different lexicon "vision".

3.2 NOMLEX to MILE

The second experiment proposes a mapping between the MLCs and NOMLEX (Reeves *et al*. 1999), a syntactic lexicon for English nominalizations. NOMLEX has been designed similarly to COMLEX, a syntactic subcategorization lexicon for English verbs. Basically, the strong reason underlying the choice of such a lexicon for the mapping, is that NOMLEX has an architecture very far form the MILE E-R model: lexicon entries take the form of parenthesized, nested feature-value structures, allowing to express lexical information in a very synthetic and compact way. NOMLEX, basically, describes syntactic frames of nominalization and also relates the noun complements to the verb arguments. All this information, once mapped against the MILE basic notions, proves to be covered by their corresponding MILE Lexical Classes (MLCs). The immediate main divergence consists, hence, in the adopted expressive means. Whereas in the previous experiment, two lexicons both based on an E-R model but not with perfectly overlapping notions have been confronted, viceversa, here, the mapping has to deal with the same linguistic notions, expressed with two conceptually opposite lexicon structures. Another important diverging point characterizes the two lexicons: the definition of the clear cut between the levels of linguistic representation. In a NOMLEX lexical entry, not only purely syntactic properties are provided, but some semantic pieces of information enter into the description. In a same feature value, no clear boundaries between the syntactic and semantic parts are defined: as a consequence, the level of *interface* between syntax and semantics as well is partly hidden in the syntactic description of the lexicon. Conversely, in MILE the representation of lexical information is highly modular, flexible and layered, with notions distinctly distributed over different levels of linguistic representation. These differences make the experiment particularly challenging, thus giving the opportunity to better test the MILE model, in terms of adequacy, expressiveness and potentialities. By way of an example of the mechanism and efforts that two differently conceived lexical organizations involve, notwithstanding the mappability of the linguistic notions, one object class, shared by all NOMLEX entries, is mapped onto the MLCs. This is the class expressed by the feature :nom-type in which the type of nominalization is declared, i.e. if it expresses the event/state of the verb, if includes incorporations of a verb argument. Mutually exclusive values can be specified, depending on the different expressions and possible incorporations of the argument. Expressing that in the MILE model means to *decompress* the information and spread it over different MLCs, belonging to different lexical layers. According to the MILE architecture, indeed, the type of relation between a nominalization and its verb base is more properly of a semantic nature. It involves many MLCs, and, moreover, implies the level of interface between syntax and semantics. Next to an MLC:SynU, a corresponding MLC:SemU is needed, with the object CorrespSynUSemU to state a link between the two. From the SemU, the MLC:SemanticFrame branch out, dominating the MLC:Predicate and its connected MLC:Argument(s)[3]. Two attributes of the class SemanticFrame, the 'typeOfLink' and 'includedArg', respectively, are in charge of specifying the relation between the SemU and the Predicate and the incorporation

[2] For instance, the verb *destroy* and the nouns *destruction* and *destroyer* may share the same predicate DESTROY respectively with a MASTER, VERBNOM, and AGENTNOM type of link

[3] It should be noted that a verb and its nominalizations are supposed to share the same semantic frame.

of the argument. In Fig. 2, the values ‘AGENTNOM’ and ‘0’ instantiate the agent nominalization[4]. The object CorrespSynUSemU, at this level of conceptual mapping, remains empty: if the mapping is pushed at the level of lexical entries it will be instantiated to specify the way the Syntactic and Semantic Frames correspond each other and, particularly, how semantic Arguments are projected on to the syntactic Slots.

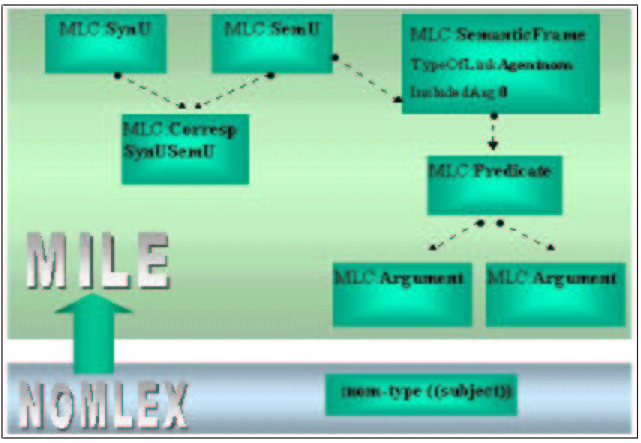


Fig. 2: A NOMLEX class mapped onto MILE MLCs.

The mapping between such a models is highly costly, since information expressed in a very compact and dense way should be explicitly decompressed and distributed over pertaining levels. This operation is due to the high level of granularity in MILE, which, however, has been thought up exactly to allow the compatibility with differently packaged linguistic objects.

4 Experiment Results and Open Issues

In this paper, we presented an experiment aiming at testing the expressive potentialities of the MILE as a standard for computational lexicons. The fundamental idea is that, by providing an efficient standard for the representation of notions at different level of linguistic description, we can obtain the key element for content interoperability among lexical resources. FrameNet and NOMLEX, two important, representative yet differently conceived lexicons, were chosen for the mapping experiment. The results of both experiments are promising, yet some reflections need to be made.

In the FrameNet to MILE experiment, we see that, even with some limits and approximations, all the FN basic notions can be, in some ways, represented using the MILE Lexical Classes. The possibility to work on a lexicon whose design follows a relational model allows an easier recognition of the lexical objects playing central roles at architectural level. MILE adheres to a relational model of the lexicon, where the *semantics* of each object is made explicit by the many relations the object has with the other objects available in the data structure. FrameNet is a lexicon of this type: the meaning of the Frame is not given by a description, a label or a code, but rather by the relations the Frame has with the Lexical Units, the Frame Elements etc.. When trying to map the FN structures on MILE, we have to verify if:

i) among the MLCs there is a valid correspondent for each FN lexical object,

ii) the internal coherence of FN is preserved when passing to MILE (i.e. if the reciprocal relations between the Frame, the Frame Element and the Lexical Unit are mirrored by the relations between the Predicate, the Argument and the Semantic Unit),

iii) there is no loss of information (and we saw that the danger of losing the important inheritance and embedding mechanisms among the Frames can be averted adding new specific modules to the MLCs).

The underlying models of NOMLEX and MILE are instead deeply different and the mapping is much more difficult. While the MILE pushes at the extreme the E-R model, NOMLEX adopts a type feature structure formalism to represent syntactic phenomena. The difference between the two is extremely evident when we observe how what in MILE belongs to distinct layers of representation (usually the semantic and syntactic layers) is represented in NOMLEX simply by juxtaposed labels within the same description code. Performing the mapping of a non-E-R lexicon onto MILE presents more difficulties and it is much more costly in terms of human intervention in the definition of the mapping conditions. It seems, however, an unavoidable price that we have to pay if we want to open the semantics and make the data structure more explicit, comparable with other lexical architectures and repositories. All in all, it can be a very useful enterprise when wanting to share and make interoperable the lexicon content in a distributed environment.

The two experiments are promising in showing how the highly expressive MILE can be used to represent both FN and NOMLEX. The modular, granular and flexible framework of the MILE model seems well suited for acting as a true interface between differently conceived lexical architectures, since it provides well recognizable, atomic, primitive notions that can be combined, nested and inherited to obtain more complex ones.

The described experiments are a first small-scale attempt to establish mapping conditions from some existing lexicons and the MILE. If we want MILE to become a really used standard, we should work intensively in the next future to provide mapping conditions between the most important lexicon models and architectures and MILE. It is obvious that this can be achieved only with the participation and help of the lexicon community, in order to benefit by the competence of each lexicon developer.

Furthermore, in order to foster the adoption of MILE as a standard framework for computational lexicons and strengthening its potential, we tried to increase the already available Data Category Registry (DCR) for the syntactic layer, by providing a draft RDF schema for the lexical objects of the semantic layer. The schema is included below in Appendix A and contains the RDF instantiations for the classes and properties corresponding to the E-R diagrams presented in Calzolari *et al.* 2003 for the MILE semantic layer. An RDF Data Category Registry represents one of the most important key issues for

[4] The mechanism applies to all the values: changing the value in NOMLEX means to change the value in one of the MILE MLCs.

starting developing multi-lingual lexicons and reusing existing ones. The proposed set of RDF Data Categories is situated in the framework of ISO TC37/SC4 and is intended as a draft to be submitted for evaluation and approval within the Lexicon Markup Framework (LMF) Working Group.

Appendix A

```
<!-- RDF Schema for ISLE lexical classes
for semantics-->

<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:owl
="http://www.w3.org/2002/07/owl#
         xmlns:mlc
="http://www.cs.vassar.edu/~ide/rdf/isle-
schema-v.6#">
         xmlns:mlcs
="http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics#">

         <!-- ISLE/MILE lexical classes for
semantics -->

<rdfs:Class
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics#SemanticFrame">
<rdfs:label>SemanticFrame</rdfs:label>
<rdfs:comment> specifies the predicative
argument structure of a lexical
entry</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #Predicate">
<rdfs:label>Predicate</rdfs:label>
<rdfs:comment> defines the predicates
entering into the Semantic
Frame</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #Argument">
<rdfs:label>Argument</rdfs:label>
<rdfs:comment> arguments entering into the
specification of a predicate
</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #SemFeature">
<rdfs:label>SemFeature</rdfs:label>
<rdfs:comment> specifies a semantic
feature-value pair and is used to describe
SemU, Synset or to specify selectional
preferences on the semantic
arguments</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #ThematicRole">
<rdfs:label>ThematicRole</rdfs:label>
<rdfs:comment>defines the thematic (or
semantic roles) that can be used to specify
the arguments within a semantic
frames</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #Synset">
<rdfs:label>Synset</rdfs:label>
<rdfs:comment>A set of synonyms that can be
related to other synsets.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #SemFeatureName">
<rdfs:label>SemFeatureName</rdfs:label>
<rdfs:comments> Specifies the semantic
features entering into the semantic
feature-value pairs. Features are defined
by their range of values.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #SemValue">
<rdfs:label>SemValue</rdfs:label>
<rdfs:comments> Defines the possible values
taken by features </rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #SelectionalPreferences">
<rdfs:label>Selectional
Preferences</rdfs:label>
<rdfs:comments> Selectional preferences is
a cluster of information that semantically
constrain the possible realizations of the
semantic frame arguments. Selectional
Proferences may include: semantic features,
synsets, collocations, particular semantic
units, a combination of all these types of
lexical information. Moreover, it is
possible to express “logically” complex
selectional preferences using logical
operators.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #LogicalOp">
<rdfs:label>Logical Operators</rdfs:label>
<rdfs:comments> This entity can be used to
express logical combinations of lexical
objects: selectional preferences,
etc.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics #Collocation">
<rdfs:label>Collocation</rdfs:label>
```

```
<rdfs:comments> This class can be used to
specify the collocations of the lexical
entry</rdfs:comment>
</rdfs:Class>

       <!-- Properties between MILE classes
for semantics -->
       <!--Properties from SemU to other
classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-
semantics#SemanticRelation">
<rdfs:label>SemanticRelation</rdfs:label>
<rdfs:domain
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemU"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics#SynsetRelation">
<rdfs:label>SynsetRelation</rdfs:label>
<rdfs:domain
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#Synset"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Synset"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics#belongsToSynset">
<rdfs:label>belongsToSynset</rdfs:label>
<rdfs:domain
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Synset"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-
semantics#hasSemanticFrame">
<rdfs:label>hasSemanticFrame</rdfs:label>
<rdfs:domain
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SemanticFrame"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #hasSemFeature">
<rdfs:label>hasSemFeature</rdfs:label>
<rdfs:domain
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #hasCollocation">
<rdfs:label>hasCollocation</rdfs:label>
<rdfs:domain
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Collocation"/>
</rdf:Property>

       <!-- Properties from synsets to
other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #consistsOfSemU">
<rdfs:label>consistsOfSemU</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Synset"/>
<rdfs:range
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #hasSemFeature">
<rdfs:label>hasSemFeature</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Synset"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics##SemFeature"/>
</rdf:Property>

       <!-- Properties from SemFeature to
other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#hasSemFeatureName">
<rdfs:label>hasSemFeatureName</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SemFeatureName"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#hasSemFeatureValue">
<rdfs:label>hasSemFeatureValue</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
```

```
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SemFeatureValue"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #isaSemFeature">
<rdfs:label>isaSemFeature</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
</rdf:Property>

       <!-- Properties from Semantic Frame
to other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #hasPredicate">
<rdfs:label>hasSemFeatureName</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SemanticFrame"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Predicate"/>
</rdf:Property>

       <!-- Properties from Predicate to
other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#isDescribedByFeature">
<rdfs:label>isDescribedByFeature</rdfs:labe
l>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Predicate"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #hasArgument">
<rdfs:label>hasArgument</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Predicate"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Argument"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics#isDescribedBy">
<rdfs:label>isDescribedBy</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics #Predicate"/>
```

```
<rdfs:range rdf:resource="
http://www.ilc.cnr.it/~bertagna/rdf/isle-
schema-semantics#SemFeature"/>
</rdf:Property>

       <!-- Properties from Argument to
other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#hasThematicRole">
<rdfs:label>hasThematicRole</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Argument"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#ThematicRole"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#hasSelectionalPreferences">
<rdfs:label>hasSelectionalPreferences</rdfs
:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Argument"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
</rdf:Property>

       <!-- Properties from Thematic Role
to other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#isaThematicRole">
<rdfs:label>isaThematicRole</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#ThematicRole"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#ThematicRole"/>
</rdf:Property>


       <!-- Properties from Selectional
Preferences to other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #selectsSemU">
<rdfs:label>selectsSemU</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#SemU"/>
```

```
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#selectsSemFeature">
<rdfs:label>selectsSemFeature</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#SemFeature"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #selectsSynset">
<rdfs:label>selectsSynset</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Synset"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#selectsCollocation">
<rdfs:label>selectsCollocation</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Collocation"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics
#selectsCollocation">
<rdfs:label>selectsCollocation</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Collocation"/>
</rdf:Property>

      <!-- Properties from Logical Op to
other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #firstArgument">
<rdfs:label>firstArgument</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
</rdf:Property>

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #secondArgument">
<rdfs:label>secondArgument</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
<rdfs:range
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-
semantics#SelectionalPreferences"/>
</rdf:Property>

      <!-- Properties from Collocation to
other classes -->

<rdf:Property
rdf:about="http://www.ilc.cnr.it/~bertagna/
rdf/isle-schema-semantics #hasCollocate">
<rdfs:label>hasCollocate</rdfs:label>
<rdfs:domain
rdf:resource="http://www.ilc.cnr.it/~bertag
na/rdf/isle-schema-semantics#Collocation"/>
<rdfs:range
rdf:resource="http://www.cs.vassar.edu/~ide
/rdf/isle-schema-v.6#MU"/>
</rdf:Property>

</rdf:RDF>
```

References

Baker C.F., Fillmore C.J., Lowe J.B. (1998). The Berkeley FrameNet Project. In Coling-ACL 1998: Proceedings of the Conference (pp. 86-90).

Bertagna, F., Lenci, A., Monachini, M., Calzolari, N. (2004). Content Interoperability of Lexical Resources: Open Issues and “MILE” Perspectives. Proceeding of LREC2004. Lisbon, Portugal.

Calzolari, N., Bertagna, F., Lenci, A., Monachini, M. (2003). Standards and best Practice for Multilingual Computational Lexicons and MILE (Multilingual ISLE Lexical Entry). Pisa.

Ide, N., Lenci, A., Calzolari N.: RDF Instantiation of MILE/ISLE Lexical Entries (2003). Proceedings of the ACL 2003 Workshop on Linguistic Annotation: Getting the Model Right, 11 July 2003 Sapporo, Japan.

Lenci A. (2003). Lexicon Design in the Age of the Semantic Web. Eurolan-2003 Summer School Tutorial, Bucharest, Romania.

Narayanan S., Fillmore C. J., Baker C.F, Petruck R.L. (2002). FrameNet Meets the Semantic Web: a DAML+OIL Frame Representation. In AAAI Workshop Proceedings.

Peters W., Cunningham H., McCauley C., Bontcheva K, Wilks Y. (1998). An Uniform Language Rosource Access and Distribution. In LREC-1998 Proceedings.

Reeves R., Macleod C., Meyers A. (1999). Manual of NOMLEX: The Regularized Version. Computer Science Department. New York University.

Vckovski A. (1999). Interoperability and Spatial Information Theory. Kluwer.

“WEAVING THE LINGUASPHERE”: LS 639, ISO 639 and ISO 12620

David Dalby
Linguasphere Observatory /
Observatoire Linguistique,
Hebron, Wales SA34 0XT
dalby@linguasphere.com

Lee Gillam
Department of Computing,
University of Surrey,
Guildford GU2 7XH
l.gillam@surrey.ac.uk

Abstract

In this paper we propose and describe a framework for the future integration of spoken, written, signed and audio-visual materials in and for all known languages. This paper builds on descriptions of language coding presented elsewhere (Dalby et al, 2004), and their potential for use within metadata registries, specifically the proposed data category registry of ISO TC 37. The aim of this paper is to develop the debate about the inclusion of language identification systems within this registry and to set the scene for an eventual harmonization of the world's languages and language resources as a continuous electronic system.

1. The Concept of the Linguasphere

For the first time, it is possible to foresee the integration and interoperability of all forms of language, so that every human linguistic system, both contemporary and historic, may be accessed as part of a wider, but hitherto neglected, reality. This reality, which humankind may now perceive clearly for the first time, is the “linguasphere”, the worldwide linguistic environment and the shared cultural heritage and communicational resource of one planetary species.

The scientific handling of the ever-increasing volumes of electronic data, generated by the spoken, written and signed languages of the world, requires an efficient, flexible, coherent and all-embracing system of identification, classification and organisation. Unfortunately, because language is such an integral part of our individual and collective existence, it is often the subject of confused thinking. When we refer to the "English language", for example, it is frequently unclear whether we are referring to the standardised written (and formal spoken) language, including the minor differences between American, British and other conventions, or whether we are referring collectively to *all* forms of the English language, including every spoken "dialectal" variety in the world and all recorded written forms, past and present.[1]

In earlier centuries, the sheer complexity of the global environment, and the impossibility of accessing, let alone harnessing, the combined resources of all the world's languages, helped maintain the "Tower of Babel" image. Languages were the most obvious manifestation of the disunity of human communities and nations and "races". Viewed as separate objects, and arranged into family trees, they have been regarded as the cultural possessions of historical communities.

There was, and still is, debate about which of the languages of this planet may have been selected by the creator of the universe to convey religious ideas to chosen communities. Languages have been treated like competing zoological or botanical species and, after the near extinction of the languages of two continents by the most expansive language of all, a desperate race has begun to save the most endangered of those still remaining. The recording of the linguistic heritage of dying communities has brought new life and much needed funding to the academic discipline of linguistics.

Today, however, our complex but always continuous linguistic environment has moved into a new, electronic, stage of its development, even more sensational and much more rapid in its effects than the earlier inventions of writing and printing. The concept of the "linguasphere" may be defined today as

[1] It is for this reason that we propose that the now ubiquitous alpha2 language tags of the ISO 639-1 standard be formally restricted to the identification of *literary and standardised written languages* (including their spoken, read and signed representations): see Dalby, Gillam et al. (2004). The unambiguous tagging of well documented standard languages can be covered by the few hundred combinations available (26x26), whereas the specification of <u>all</u> forms of spoken, written and signed language, as discussed below, requires at least tens of thousands of identifiers.

"the languages of humankind, viewed collectively as components of a worldwide vehicle for thought, communication and documentation and for the collective maintenance of communal and individual identities". We now have the technical and communicational resources to view the workings of the linguasphere within three complementary dimensions:

- the *"organic linguasphere"* is the underlying continuum of all spoken, written and signed conventions through space and time – lexical, phonological, grammatical and semantic – which in differing combinations and geographical patterns constitute the structure of all human languages, against the evolving background of communal identities and cultures;
- the *"mental linguasphere"* may be abstracted as the total linguistic knowledge and competence of all communicating members of humankind at any one time;
- the *"recorded linguasphere"* is the cumulative physical and electronic collection of all recorded materials in all languages through time, that collective "library" which is fast becoming our most progressively organized, collectively accessible and dynamically exploited human resource, involving the increasing partnership of machines.

The linguasphere has always existed, in the sense of the continuum of all human languages, linked together haphazardly in the minds of bilingual and multilingual speakers. Today, however, we have the first opportunity of organising the linguasphere as an integrated global entity, as a meta-base for the classification and communication of knowledge about ourselves, in all languages simultaneously. Having identified and classified everything around us, albeit in ways that remain the subject of debate, we will be able to identify and classify ourselves on a global scale, using languages as both our collective tool and our shared medium.

In this paper, we present the principles and composition of a structural meta-base, the "Linguasphere System", which has been created at the outset of the 21st century to meet the needs and scale of the future co-ordinated development of languages worldwide. We then suggest how this system might be used in combination with metadata registries, as a means to organise unlimited streams of multilingual and multimedia data and materials around a standardized corpus of "language" identifiers. The initial result of this work will be akin to a multilingual electronic thesaurus of all the world's languages.

The Linguasphere System, known as LS 639[2], already provides over 25,000 unique four-letter language identifiers (referred to here as alpha4 langtags). These langtags are applied equally but unambiguously to:

(i) individual languages,

(ii) individual groupings of two or more languages,

(iii) individual varieties or "dialects" within any language, written, spoken or signed,

(iv) individual components or communities within those varieties,

(v) historical periods within each diachronically recorded language, and

(vi) individual scripts [3] and varieties of script or spelling system.

The coverage of each langtag is correlated and defined in terms of its adjacent identifiers at identical, wider and/or narrower layers of linguistic classification. Each alpha4 langtag has its place within a global hierarchy, but can also stand alone as a unique and unambiguous identifier.

This system is based on the key principles outlined below, and the structure of its identifiers (alpha4 langtags) allows for a potential expansion to cover up to 450,000 identified units of linguistic reference.

[2] the number 639 reflects the intended function of the LS 639 alpha4 langtags as a complement to, or extension of, the existing ISO 639 alpha2 and alpha3 langtags.

[3] including identical forms, wherever appropriate, to the pre-existing alpha4 langtags of ISO 15924 (2004) Code for…Scripts.

2. Principles of the Linguasphere System

The development of a large-scale system of language identifiers, with capacity for future expansion and the means to cope with continuous changes in languages, requires a clear theoretical basis. The development of the Linguasphere System is based on the following key principles:

- A distinction is maintained between the preferably discouraged use of the term "language" to describe a written (and formal spoken) *standard language*, and its basic, more comprehensive application to a *language* in all its written, spoken and/or signed forms.
- *Languages* are not independent objects, like apples, but are parts of a fluid continuum of human communication, the ever-changing surface of the linguasphere.
- *Languages* have dual roles as means of communication and of communal identity, so that the organisation of data on languages needs to embrace data on the communities who use them.
- *Languages* and the *varieties* and *components* of a language cannot always be classified with certainty, either over periods of time or across geographical areas, so any necessarily flexible form of classificatory coding (including hierarchical sub-tags) needs to be kept separate from the accurate and static tagging of linguistic and communal identities.

The development of a system of specification based on these principles has required not only the production of a list of unique identifiers (alpha4 langtags), but also the means by which to answer questions about the geographical, cultural and linguistic contexts of these identifiers, both historical and current. These will include global questions such as “what is the current spread of language communities within the world's megacities[4]?” or "what is the current distribution of bilingualism and .translingualism[5] among language communities in contact?".

3. Structure of the Linguasphere System

LS 639, the system of Linguasphere alpha4 langtags, provides for correlation with and unambiguous conversion to/from the alpha2 tags of ISO 639-1, the alpha3 tags of ISO 639-2 (parts 2/B and 2/T), the SIL/Ethnologue (proposed ISO 639 part 3), and other related systems (RFC 3066 etc). Since LS 639, potentially the basis for ISO 639 part 6, is more granular than other identifiers and classifiers, this correlation will provide a precise form of definition and mapping for other systems of language coding and classification (including the existing ISO 639 parts 1 and 2, and the expected parts 3 and 5).

The Linguasphere System consists of three parts:

- a fixed numeric framework of 10 sectors and 100 zones of global linguistic reference, known as the **Linguasphere Key**
- an adjustable alphabetic scale of relationship within each of those 100 zones, known as the **Linguasphere Scale**
- 25,000+ unique LS 639 alpha4 langtags, or fixed "language labels" known as **Linguasphere Identifiers**, each having an assigned place on the Linguasphere Scale of the relevant zone.

These parts are briefly presented under the following three sections.

3.1 Linguasphere Key

The framework of global reference is composed of ten referential sectors, with these sectors containing a total of one hundred referential zones. This allows any language in the world, or any defined group of languages, or any variety or component or community of any language, to be simply and

[4] As exemplified by research on the languages of London, commenced in the 1990's within the "Logosphere" language mapping programme at the London School of Oriental and African Studies and the Observatoire Linguistique (now Linguasphere Observatory): see Baker and Eversley (2000).

[5] The term "translingual" describes a speaker able to navigate competently between two or more closely related languages, or a community in which the majority of speakers are able to do so (e.g. from Catalan to Spanish). The distinction between translingualism and other forms of bilingualism is useful, since it involves differences in the processes of language learning and of translating, as well as in the way languages may influence each other. See Dalby (2000a), pp.70, 108.

unambiguously located within the linguasphere by means of a pair of digits. This numeric framework is referred to as the Linguasphere Key [6], and the two digits represent information about the relevant zone.

The first digit of this key is used to refer to one of the ten referential sectors that establish a major division of the linguasphere between:

- languages classified outside five major 'families' or affinities, and
- all those languages which have been classified within them.

Languages in the first of these two categories are initially classified, according to purely geographical criteria, within five *geosectors* corresponding to the continent where they are spoken.

Languages in the second category (including, as it happens, all major languages with an "intercontinental" distribution) are classified within five linguistic *phylosectors*, corresponding to the continental or intercontinental affinity to which each of them belongs.

The ten sectors are ordered, both numerically and alphabetically, so that:

- the five *geosectors* are each indicated by an even digit: 0=AFRICA; 2=AUSTRALASIA; 4=EURASIA; 6=NORTH-AMERICA; 8=SOUTH-AMERICA
- the five *phylosectors* are each indicated by an odd digit: 1=AFRO-ASIAN (containing languages of the Afro-Asiatic or Hamito-Semitic affinity); 3=AUSTRONESIAN (containing languages of the Austronesian affinity); 5=INDO-EUROPEAN (containing languages of the Indo-European affinity); 7=SINO-INDIAN (containing languages of the Sino-Tibetan affinity); 9=TRANSAFRICAN (containing languages of the Atlantic-Congo affinity)

The second digit of the Linguasphere Key is used to subdivide these ten *sectors* (five *geosectors* + five *phylosectors*) into one hundred *zones*, representing the referential sub-division of each sector into a further ten parts.

Within the five *phylosectors*, the component zones (or *phylozones*) are based on the known linguistic subdivisions of each of the *affinities* (or 'families') concerned, selected subdivisions being either combined or further divided to arrive at a total of ten referential parts. 5=Indo-European, for example, divides readily into ten phylozones, corresponding to so-called "branches" of the Indo-European wider affinity or "family", whereas in the case of 1=Afro-Asian, a total of ten phylozones is arrived at by allocating three zones (rather than one) to the more complex Chadic "branch" of the Afro-Asiatic intercontinental affinity, representing the three actual linguistic groupings within that branch, i.e. 17=Charic, 18=Mandaric, 19=Bauchic.

Within the five *geosectors*, twenty-five of the fifty component zones[7] are themselves *phylozones*, corresponding to wider or narrower affinities, as in the case of 00=Mandic in 0=Africa, for example, or 41=Uralic in 4=Eurasia. The remaining twenty-five zones are *geozones*, corresponding to convenient geographical groupings of languages that may sometimes share a geo-typological relationship, as in the case of 43=Caucasus or 44=Siberia, or may simply be isolated languages or groupings of languages spoken in the same geographic area, as in the case of 87=Amazon.

The sectors and zones form a consistent table of reference covering the totality of modern languages in the world, to which any past or future system of historical classification may be specifically cross-referenced. A stable framework – or linguistic "workbench" - is thus provided, on which pieces of the historical and contemporary jigsaw of linguistic relationships can be assembled and re-assembled as necessary. The underlying framework of reference will no longer need to be changed each time a new 'family-tree' of remoter or closer affinities is proposed or established. The scale of relationships within this framework (see below) will allow for future changes of classification.

3.2 Linguasphere Scale

The proven or assumed relationships among the languages of each zone are recorded by means of an alphanumeric code, composed of the two digits of the Linguasphere Key, followed by an alpha code[8].

[6] See Dalby (2000a), pp.58-62.

[7] The fact that exactly 25 of these 50 zones may be treated as *phylozones* is a statistical coincidence.

[8] The term "alpha code" refers to the function of the Linguasphere Scale in the standardised encoding of relationships among languages, in contrast to the purely identifying function of the LS 639 "alpha4 langtags".

This alpha code, known as the Linguasphere Scale, is variable in length, serving to encode the intermediate and close relationships among languages in the same zone (including groups and varieties of languages), based on current scholarship and documentation.

The working of this alpha code is not described in detail within this paper, but may be summarized and exemplified as follows. The Linguasphere Scale is composed of 2 sequences of up to 3 letters each, distinguished by case. The first sequence (in upper-case) represents "outer layers", a graduated coding of relationships, ranging from a substantial minority to a substantial majority of the lexical materials present in the languages of each zone. The second sequence (in lower-case) represents "inner layers", a geographical and/or linguistic ordering of the closely related varieties of a specific "language" or tight cluster of "languages". Unlike the Linguasphere Key, applied for stable referential purposes to each zone and to each language assigned to that zone, the alpha code of the Linguasphere Scale can be reset at any point within a zone, whenever it is necessary to incorporate new or revised information, or re-classification, into updated versions of the Linguasphere System. This cascade updating of the hierarchical alpha code has no effect on the alpha4 identifiers of the defined languages involved, or of any other unchanged components. An example of the use of this scale is given in the following section.

3.3 Linguasphere Identifiers (LS 639 alpha4 langtags)

The Linguasphere Identifiers, known collectively as LS 639, form an expanding series of over 25,000 unique "four-letter language labels" or alpha4 langtags, each of which has a specific and if necessary adjustable place against the Linguasphere Scale of the relevant zone. The system, already indexed to over 70,000 language names and variant names, has the potential for expansion to over 450,000 identifiers[9].

The LS 639 alpha4 langtags have been selected and designed to cover every known language, written, spoken, and signed, either modern and/or recorded from the past, as well as a growing catalogue of the component dialects and communities, historical periods and writing systems within individual languages. The application of these identifiers extends not only inwards, however, but also outwards, to include the names of groups of languages up to and including major affinities or 'families'. Their purpose is to provide unique and unambiguous labels for every *unit of linguistic reference*, from isolated or extinct language communities to the most widely distributed families of modern languages.

The alpha4 langtags have been added to the Linguasphere System since a selective outline of the system was first published in 1993[10] and since its complete global register appeared in 1999-2000[11]. These LS 639 identifiers are designed to serve as unambiguous machine-readable access tags to all relevant data on and in any unit of linguistic reference at whatever level. The linkages among them, represented and controlled by the Linguasphere Scale, will make it possible for machines and human users to navigate the Linguasphere System - and hence the linguasphere itself - in all directions, outwards to wider categories, inwards to narrower varieties, and sideways to adjacent and other related units of linguistic reference. This universal application of alpha4 langtags as static identifiers means that the reassignment of any unit of linguistic reference to a wider or narrower layer of classification does not affect its established identifier or langtag. Each langtag gives access to information on and in each relevant language (or variety or group), and its components, and enables the information to be viewed in the context of the relevant languages' wider relationships.

The classification of linguistic relationships provides an obvious framework for organising data on natural languages. Yet how can such a framework, often based on complex hypotheses[12], be protected from the inevitable upheavals caused by any reassessment of linguistic relationships[13]? One remembers the way in which books on African languages, for example, needed to be reclassified in the

[9] In practice the number will be less than 450,000, since readily pronounceable sequences are avoided as much as possible in the composition of alpha4 langtags, for obvious reasons.

[10] Dalby (1993)

[11] Dalby (2000a), including preview editions published in 1997 and 1998 in accordance with the objevtives of the UNESCO Linguapax project.

[12] See for example the complex language family index reproduced in Grimes (2000b).

[13] Historical relationships among languages are sometimes described as "genetic". This is misleading in that languages are not independent objects when in close contact within the minds of bilingual speakers, who are key players in the evolution of the linguasphere.

mid-20th century to cater for major changes in their classification.[14] This problem is overcome in LS 639 by treating the comprehensive *identification of inter-relationships* among languages as a fundamental category of metadata attached to, but not determining, the alpha4 identifiers of individual languages or varieties of language. A continually updatable *roadmap of the linguasphere* may consequently serve as a logical supplement to – but not necessarily a part of - the proposed expanded structure of ISO 639.

The Linguasphere System may be briefly illustrated by the following example, tracking the hierarchy of relationships from the most widely distributed of all language families (Indo-European) through to the local form of southern Welsh spoken around the Preseli Hills, where the Linguasphere Observatory is currently situated. The Linguasphere Key is represented by one or two initial digit(s), the outer layer(s) of the Linguasphere Scale by the subsequent upper-case letter(s), and the inner layer(s) by the final lower-case letter(s). The Linguasphere alpha4 identifiers are cited between forward slashes.

Example of the Linguasphere Hierarchy

	***scale**= reference name /**alpha4 langtag**/* :	*cf. parallel example in each case* :
sector	**5**= Indo-European /**ineu**/	cf. 4= Eurasia /euas/
zone	**50**= Celtic /**celt**/	cf. 51= Romanic /rmnc/
outer layers	**50B**= Brythonic /**brtn**/	cf. 50A= Gaelic /gael/
	[50BA= Cymraeg (Welsh) /cymr/] [15]	cf. 50BB= (Breton+ Cornish) /brkr/
inner layers	**50BAa**= Cymraeg (Welsh) /**cymr**/	cf. 50BBb= (Breton) /brzg/
	50BAad= Cymraeg y De (South Welsh) /**cyde**/	cf. 50BAab= (North Welsh) /cyst/
	50BAdda= Iaith y Preseli (Preseli Welsh) /**prsl**/	cf. 50BAdba= (S. Central Welsh) /cycd/

The totality of Indo-European languages is thus identified by the same form of alpha4 langtag, in this case **/ineu/,** as the local form of the Welsh language in west Wales, identified by **/prsl/**. Between these two extremes, alpha4 langtags are likewise used to identify the Celtic languages within Indo-European, **/celt/**; the "Brythonic" or Britannic languages within Celtic, **/brtn/**; the "Welsh" or Cymraeg language itself, **/cymr/**; and the inner layer of "Southern Welsh", /**cyde**/.

Note the duality of language names in Welsh (autonyms) and in English (exonyms, in brackets).

The application of LS 639 alpha4 langtags to all levels of linguistic identification has the following advantages:

1. With hundreds of thousands of potential combinations, LS 639 is able to represent the actual scale of complexity of spoken languages around the world.[16]
2. The full range of 25,000+ alpha4 langtags is already established and will be available from August 2004, for software development (as XML lang tags) and other purposes.[17]
3. The mnemonic form of most alpha4 langtags favours human readability alongside an essential machine readability. Although machines have no need for mnemonic identifiers, communities of speakers are likely to prefer the “meaningful” tagging of their languages based on their own autonyms.

[14] When major groupings such as “Sudanic” were replaced by new groupings such as “Niger-Congo”.

[15] An extra outer layer is necessary at this point (although 50BA is identical in content to 50BAa), because 50B=Brythonic subdivides first into 50BA & 50BB, i.e. Welsh *versus* Breton+ Cornish, before subdividing into the three related languages (50BAa, 50BBa & 50BBb, i.e. Welsh, Cornish & Breton).

[16] In contrast to alpha3 tags, which are limited to just over 17,500 combinations, adequate for the designation of entire languages but insufficient for the more comprehensive task of distinguishing linguistic varieties and components.

[17] If LS 639 is accepted as the basis of a NWIP (New Work Item Proposal) by ISO/TC37/SC2, meeting in Paris in August 2004, then a period of public review of the 25,000+ identifiers will need to be agreed and organised before they are confirmed as part of ISO 639-6 or other international standard.

4. High granularity gives LS 639 a refined power of definition, allowing "languages" to be identified in terms of their components rather than the reverse.[18]

5. The correlation of LS 639 tags with all other forms of language identifiers will support all legacy databases with fixed 2- or 3-character fields for language identifiers.

6. LS 639 supports the parallel use of ISO 639-2, with its proposed extensions (639-3 and 639-5), since each alpha3 tag will be precisely definable in terms of its alpha4 equivalents, covering its components and wider linguistic context.[19] See section 4 below.

7. The use of alpha4 langtags at all levels will facilitate, whenever required, the future redefinition of any "language" as a "variety" of a wider language, or as a "collection" of two or more languages, without changing its LS 639 tag. Such changes of layer of classification (i.e. level) need not affect the application of the relevant identifiers.

8. Each fixed alpha4 langtag is located by reference to its coded and potentially adjustable place on the Linguasphere Scale [20]. Information on the classification of each referent is contained in the relationship scale rather than the alpha4 langtag itself.

4. LS 639 and ISO 12620

The proposed expansion and refinement of ISO 639 coincides with proposed and ongoing work regarding the development of metadata registries for language resources (by sub-committee TC37/SC4). Within these registries, language identifiers will be needed both for use in the language resources they are used to describe or standardise, and also to act as *keys* to metadata in the definition of the metadata (e.g. XML coding) itself.

Based on work carried out initially in ISO 12620:1999, which described so-called “Data Categories” found in terminological collections, ISO 12620 is being revised in conformity with ISO 11179-3 to describe the management of data categories, with subsequent parts providing descriptions of validated data categories, for example part 2 for terminological data categories. The parallel development of these (sets of) standards will provide a link between the creation and management of language identifiers and their management and use within software systems via metadata registries, enabling and ensuring interoperability between language resources that may use differing systems of language identifiers (at the very least). Use of Data Categories for specific types of language resources has been described for terminologies in ISO 16642 (Terminological Markup Framework): see Gillam et al 2002 for a discussion of these standards.

The revision of 12620 has produced a data model that can be used to describe identifiers. This model for description, based on ISO 11179-3, requires at minimum a canonical reference name to be attached to the identifier, and to its description. The reference name can be considered as a “conceptual” identifier that can have various forms in different languages: this model parallels the terminological metamodel of ISO 16642, although we are effectively outside the language, perhaps at a metalanguage level. Since ISO 12620 deals with language resources, a prime consideration of language resources is the language, so the various existing and proposed parts of ISO 639 provide various “pick list” values that could be used, in combination with a field such as “language identifier” from ISO 12620:1999, for the description of a language resource.

Setting aside the administrative aspects of both LS 639 (and by extension the current and proposed parts of ISO 639) and ISO 12620, the question of what information is contained in such a registry remains open. The alpha2, alpha3 and alpha4 identifiers with their canonical names, and their language names are essential. Here it is important also to consider the mapping between these sets of identifiers: ISO 639-1 provides the single **cy** identifier; ISO 639-2 provides a bibliographical **wel** and a terminological **cym**. The SIL Ethnologue provides the tag /**WLS**/ to cover all Welsh, and lists Northern Welsh, Southern Welsh and Patagonian Welsh without further coding. The structure for the wider

[18] In contrast to alpha3 tags, which depend on the *a priori* definition of individual "languages" and "language names".

[19] In this context, the Linguasphere Observatory welcomes close consultation and collaboration with ISO TC37 and SIL/*Ethnologue.*

[20] Dalby (2000a), pp.58-74

classification of the SIL identifier is similar to that of the LS 639 /**cymr**/ identifier, except that an historical “Insular Celtic” level is added by SIL.

SIL: Indo-European > Celtic > Insular > Brythonic > Welsh /**WLS**/ (United Kingdom)

LS 639: Indo-European > Celtic > Brythonic > Welsh or Cymraeg /**cymr**/….

To support these multiple systems, some form of mapping between the identifiers is required, but there is also potentially the need for supporting multiple hierarchies amongst these identifiers, since points of convergence between the systems allow the potential for additional divergence, and access to further identifiers such as those presented earlier. Consideration of a standardized *reference name* for each language and for its language variants is also important (for the 70,000+ names in LS 639 and the 41,000 plus in Ethnologue).

Hence, if we consider /**cymr**/ as the identifier, this can be named in English as /**Welsh**/ or in Welsh as /**Cymraeg**/ (these names are independent of their presentation in a specific resource that uses them). This has equivalents **cy**, **wel**, **cym**, **WLS**. Describing the description is interesting, since we need to say that /**cymr**/ = Cymraeg (in the language itself) or Welsh (in English). This could be expressed /**cymr**/ = /**cymr**/ Cymraeg or /**engl**/ Welsh, or as /**cymr**/ = Cymraeg or /**engl**/ Welsh (where the lack of a tag implies the use of the "autonym", i.e. the speakers' own name for the language). The languages of the names of languages therefore need to be specified using the same system of identifiers which are identified by the names! This is further complicated by the fact that the name of a language may vary among the varieties of the language, so that the Welsh autonym could in fact be subdivided to provide tags that represent these varieties, for example /**cyst**/ Cymraeg, /**cyde**/ Cymrâg… (etc.).

Finally, to fully include LS 639 identifiers, and to an extent also SIL Ethnologue identifiers, in a 12620 specified Data Category Registry, the identifiers used to describe the language identifiers can also be adopted or included. To cater for the structure of these systems, categories such as /continent/, /Africa/, /Australasia/, /Eurasia/, /North America/, /South-America/ are required. The system of language identifiers is then itself a language resource, produced by selecting and organising such identifiers.

5. Conclusion and Discussion

The role of the Linguasphere Observatory in the next stage of the development of ISO 639 was recognised in a generous resolution at the ISO meeting held in Oslo in August 2003:

> "ISO/TC37/SC2[21] appreciates the valuable practical and theoretical input from the Linguasphere Observatory (Wales) and the British Standards Institution in conjunction with the work with language coding carried out in ISO/TC37/SC2/WG1 and … requests David Dalby of the Linguasphere Observatory to develop further the proposal *ISO 639-6 Codes for the representation of names of languages – Extension coding for language variation* for use in conjunction with other parts of ISO 639 … and to submit a New Work Item Proposal with a corresponding Working Draft by 2004-05-31 for discussion at the next meeting [of ISO/TC37/SC2/WG1, to be held in Paris in August 2004]."

LS 639, its potential adoption as ISO 639-6, and the use of the LS 639 alpha4 langtags within metadata registries will facilitate the following:

- a "road-map" for the adoption of its more extensive set of alpha4 identifiers, including optional migration from alpha2 and alpha3 identifiers,
- the geographical mapping of alpha4 tagged items. Some of this work has already been undertaken among members of the Linguasphere network, including cartography in UK (centred on Africa), in France (centred on the Himalayas) and in Russia (centred on the Caucasus)
- referential transparency. For example, when we refer to the "English language" it is often unclear whether or not we are referring to the standardised written (and spoken) language. Are we ignoring the minor differences between American, British and other conventions in the standard written language? Alternatively, are we referring collectively to *all* forms of the

[21] Sub-Committee SC2 of TC37 is responsible for language coding.

English language, including every spoken "dialectal" variety in the world and all recorded written forms, past and present?

- no duplication of identifiers of ISO 15924 (used to designate the scripts of the world, or – by extension – the communities who use each script). LS 639 and ISO 15924 identifiers may be combined as required, providing that a standardized means for doing so is adopted also.

The dissemination and use of such a system will be important in the fields of business, government, education, social research and the media. Assisting international consortia by the introduction and use of the LS 639 system will be a valuable scientific contribution from Europe.

The authors invite discussion regarding development of a road-map for the implementation of a 12620 compatible set of language identifiers which covers aspects identified in previous sections, and which will include a description for the so-called "concatenation" of identifiers from separate systems, for example to clarify the use of en-GB-Latn (denoting the British variety of contemporary Standard English, as written in the Latin script) versus eng-Latn or engl-latn (covering all forms of English written in that script at any time).

Beyond this important technical task lies the prospect of laying the carefully planned foundations and architecture for the next stage in the progressive harmonisation of a multilingual world. This will involve the progressive integration of the dictionaries, thesauri, and electronic translation programs of the languages of the world into a single, multilingual database. Specification of languages will be the central parameter in the construction of this global resource for documentation, translation and interpretation, and in the parallel identification, assessment and development of individual language communities. An efficient system of identifiers for the languages and language communities of the world, in a framework which allows for both change and growth, will be an essential foundation for the global documentation of humankind.

The increasing mobility and dissemination of language communities around the urbanised planet has been paralleled by the electronic transformation of the spoken word into the principal medium of worldwide communication and instant documentation. The observation, understanding and creative exploitation of both phenomena will require the transparent, accurate and unambiguous identification of every spoken, written and sign language, including each component variety, community and recorded corpus, from the most globalised to the most localised.

This need for a coherent system of universal linguistic identification is accelerating as electronic communications and speech applications become available to communities of all sizes around the globe in their own languages. The role for such a system is rapidly expanding as demands increase for multilingual translation and interpretation, including subtitling and dubbing. As the multilingual character of megacities - and of the world itself - develops and changes, there is urgent need for a global system of linguistic and ethnic identification and documentation.

Any institutions or individuals who wish to participate in the further development of the Linguasphere System (including LS 639), and in the updating and expansion of the Linguasphere Register, are asked to contact editors@linguasphere.com without delay.

Acknowledgements

In the development of the Linguasphere System, David Dalby acknowledges the cumulative support of the Agence de la Francophonie (formerly ACCT), UNESCO, School of Oriental and African Studies (University of London), Leverhulme Trust Fund, British Academy, British Standards Institution, Opportunity Wales and ICT Marketing Ltd. He is grateful to Debbie Garside and to Chris Cox (BSI) for helpful advice during the preparation of this paper.

Lee Gillam acknowledges partial support for this work by the EU co-funded projects SALT (IST-1999-10951) and GIDA (IST-2000-31123), the EPRSC funded SOCIS project (GR/M89041) and the ESRC FINGRID project.

References

Constable, P. (2004-01), “Issues to resolve in ISO 639”, ISO/TC 37/SC 2/WG 1 N 115

Baker, P. and Eversley, J. (ed.), *Multilingual Capital: the languages of London's schoolchildren and their relevance to economic, social and educational policies*, Battlebridge: London. 92pp. ISBN 1 903292 00 X

Dalby, D. (1966), "Levels of relationship in the comparative study of African languages", *African Language Studies* VII (SOAS), 1966, pp.171-179

Dalby, D. (1977), *Language Map of Africa and the adjacent islands*. International African Institute: London.

Dalby, D. (1993), *Les langues de France et des pays et régions limitrophes au 20ème siècle* (Essai de classification… précédée d'une introduction théorique et pratique…). 45pp. L'Observatoire Linguistique: Cressenville. ISBN 2 9502097 5 0

Dalby, D. (2000a) *Linguasphere Register of the World's Languages and Speech Communities*: Volume 1 (Introduction and Index). 300 pp. Linguasphere Press: Hebron (Wales). ISBN 0 9532919 1 X

Dalby, D. (2000b) *Linguasphere Register of the World's Languages and Speech Communities*: Volume 2 (The Register). 743 pp. Linguasphere Press: Hebron (Wales). ISBN 0 9532919 2 8

Dalby, D., Gillam, L., Cox, C., Garside, D. (2004) “Standards for Language Codes : developing ISO 639-6”. *Proceedings of 4th International Conference on Language Resources and Evaluation* (forthcoming).

Fitzgibbon, A. and Reiter, E. (2003) “‘Memories for life’: Managing information over a human lifetime”. http://www.nesc.ac.uk/esi/events/Grand_Challenges/proposals/Memories.pdf (4 March 2004)

Gillam, L., Ahmad, K., Dalby, D. and Cox, C. (2002) “Knowledge Exchange and Terminology Interchange: The role of standards”. In Proceedings of Translating and the Computer 24. ISBN 0 85142 476 7

Grimes, B.F. (Ed.) (2000a) *Ethnologue*: Volume 1 (Languages of the World). 14th Edition. 866 pp. SIL International: Dallas (Texas). ISBN 1 55671 103 4

Grimes, B.F. (Ed.) (2000b) *Ethnologue*: Volume 2 (Maps and Indexes). 14th Edition. 735 pp. SIL International: Dallas (Texas). ISBN 1 55671 104 2

ISO 639-1 (1988) “Codes for the representation of the names of languages – Part 1 (Alpha2 code)”

ISO 639-2 (1998) “Codes for the representation of the names of languages – Part 2 (Alpha3 code)”

ISO 15924 (2004) "Codes for the representation of the names of scripts"

ISO 3166-1 (1997) “Country codes”

ISO 11179-3 (1994) “Information technology – Specification and standardization of data elements. Part 3 (Basic attributes of data elements)”.

ISO 12620 (1999) “Computer Applications in Terminology – Data categories”

ISO 16642 (2003) “Computer Applications in Terminology – Terminological markup framework (TMF)”

Annotating Syllable Corpora with Linguistic Data Categories in XML

Robert Kelly, Moritz Neugebauer, Michael Walsh & Stephen Wilson

Department of Computer Science
University College Dublin
Belfield, Dublin 4
Ireland
{robert.kelly, moritz.neugebauer, michael.j.walsh, stephen.m.wilson}@ucd.ie

Abstract

The usefulness of high quality annotated corpora as a development aid in computational linguistic applications is now well understood. Therefore it is necessary to have systematic, easily understandable and effective means for annotating corpora at many levels of linguistic description using. This paper presents a three step methodology for annotating speech corpora using linguistic data categories in XML and provides a concrete example of how such an annotated corpus can be exploited and further enhanced by a syllable recognition system.

1. Introduction

The need for high quality annotated corpora to assist in the development of speech applications is now well-understood. Furthermore, much research has been conducted into the development of tools which support for the acquisition of these corpora. This paper presents one particular methodology which assists the linguist in the development and maintenance of annotations. The annotation methodology combines both user driven and purely data driven techniques. Each stage of the process incrementally enriches a syllable labelled corpus using well-defined and universal data categories. The resulting resource adheres to a standard structure employing linguistic data categories familiar to speech researchers.

Annotated corpora for languages are vital linguistic resources from both a language documentation and an applications perspective. Recently much emphasis has been placed on developing multilingual resources for syllable recognition. In order to support ubiquitous multilingual resource development, a standard language and corpus independent annotation methodology must be identified. In addition, this methodology must employ a standard registry of linguistic data categories which are also independent of the language in question and corpus being annotated. If such a methodology can be developed then the resulting annotations will necessarily adhere to a standard format. This has far reaching implications for the manner and the extent to which the annotations can be used. For example, multilingual speech applications utilising the annotations can be implemented in a generic fashion such that the annotations can be used as plug and play resources.

The annotation procedure outlined in this paper assumes the existence of a syllable labelled data set. Such a syllable data set may not always be available, especially since corpora tend to be labelled at the segment and word level but not at the syllable level. However the procedure has been recently adapted such that annotations can be derived from segment labelled data (see section 2.). The technique aims at structuring the existing syllable annotations in a standard representation. The representation used here is the *Multilingual Time Map* (MTM)(Carson-Berndsen, 2002). An MTM is an XML document that structures the corpus of syllables using a standard tag set. Thus, an MTM can be seen as a standardised registry of data categories where each XML tag corresponds to a single data category. It is also important to note that the underlying structure of an MTM encodes a finite-state machine. More specifically an MTM is an extension of a phonotactic automaton, a finite state representation of the allowable segment combinations at the syllable level. Thus, an MTM is in fact a multi-tape finite-state transducer where the state transition structure describes at least the syllables in the original training corpus. The state-transition structure may also have undergone generalisation to further account for well-formed syllables not observed in the corpus but which are considered well-formed for the language in question. Generalisation in relation to finite-state structures and further details regarding MTMs are discussed further in section 2.. Once the annotations have been structured in an MTM the syllables can be recovered by identifying all acceptance paths described by the underlying finite state structure of the MTM. Each acceptance path describes a single syllable which can be extracted by concatenating the segment annotations which must be present on every transition of the path. In addition to the required segment annotations each transition of the MTM describes a number of further annotations which include at least the following levels of linguistic description; segment, frequency of segment with respect to the particular corpus in each phonotactic context, probability of segment in each phonotactic context, features associated with segment and also implications between phonological features. Also, if timing information is available, the MTM will provide annotations detailing the average duration and standard deviation of duration of each segment occurring in each phonotactic context. In addition, the core MTM supports task specific data categories, examples of which are discussed in section 5..

The core annotation procedure itself consists of three stages. Firstly a phonotactic automaton learner takes the set of syllable annotations and constructs the initial finite-state structure of the MTM for those syllables. Thus, after this first stage the MTM describes annotations at the syllable level in terms of the induced state transition structure. Following the induction of the initial MTM structure, the second stage of the procedure serves to augment the MTM with an annotation describing the articulatory features associated with each segment annotation. Users can define phonological features which are associated with the segments labelling the transitions. This segment-feature bundle correspondence is stored in a separate XML structure called a feature profile. The third and final stage of the core annotation procedure is to examine the feature annotations of the second stage and using a feature hierarchy. This identifies phonological feature implications which are then integrated into the annotation using two categories. The first specifies those features which are introduced by the associated segment and the second those that are shared with segments appearing elsewhere in the MTM.

The three step procedure detailed above describes a language independent approach to structuring corpora (which may differ widely in structure) in a homogeneous annotation scheme by utilising standardised data categories. Thus, the annotation procedure can be applied to corpus of syllables from any language, thus supporting the development and extension of a multilingual phonotactic resource catalogue. Furthermore, since the first and third stages are completely data driven the resources can be acquired rapidly and at low cost. The acquired MTMs can then be stored in a central resource repository. Also, since each MTM has an underlying finite-state structure, the acquired MTMs can be efficiently processed (van Noord, 1997). This ensures that these phonotactic resources can be easily linked to speech applications which may require the inspection of the annotations and/or the use of the finite-state structure described by the MTMs.

2. Data Driven Induction of Initial MTM

The first stage of the annotation procedure is to induce the initial finite-state structure of the MTM. This first stage is completely data driven requiring no user intervention however it requires that a corpus of syllable labelled utterances be available from which the initial structure can be induced. If such a syllable labelled corpus is not available, which may well be the case since corpora are typically labelled at the phoneme and word level but not at the syllable level, then a semi-automatic procedure has been developed allowing syllable annotations to be derived from phoneme annotations with a minimum of user supervision. This semi-automatic approach to deriving syllable annotations is discussed after a description of the primary topic of this section, namely the data driven induction of MTM finite-state structures from syllable labelled data. Firstly, however a discussion of phonotactic automata is required since the structure of these automata underlie that of MTMs.

A phonotactic automaton is a finite-state representation encoding the allowable sound combinations that are valid for a language at the syllable level. Since a phonotactic automaton is a finite-state structure, it consists of a number of states with some state designated as the initial or start state[1]; a subset of the states designated as accepting or final states; and finally a finite set of state transitions over a given alphabet. In the case of phonotactic automata the alphabet is the inventory of segment labels, thus labels on transitions represent single sound segments and the allowable sound combinations are modelled by the state-transition structure. As an example, figure 1 illustrates a subsection of a phonotactic automaton for English showing only a subset of the possible sound combinations observed in well-formed syllables. Note that this automaton is nondeterministic with a unique start state (labelled 0) and transitions labelled with SAMPA[2] phoneme symbols. Also final states are denoted by double circles in figure 1. Phonotactic automata have proven useful in speech applications, in particular these finite-state models of phonotactic constraints are used as the primary knowledge component in a computational phonological model of syllable recognition, the Time Map model (Carson-Berndsen, 1998), discussed further in section 5.. A phonotactic automaton allows the Time Map recognition engine to decide on the well-formedness of putative syllables. Given such a syllable, a phonotactic automaton for a language allows the recogniser to determine if the syllable is well-formed for the language by attempting to trace an acceptance path through the state-transition structure using the individual segments of the syllable as input symbols. Returning to figure 1, it is easy to see that according to this finite-state structure the combinations $/s\ p\ l\ aI\ n/$ and $/T\ r\ aI/$ would be considered well-formed while the combinations $/s\ p\ l\ aI\ p/$ and $/T\ aI/$ would be considered ill-formed.

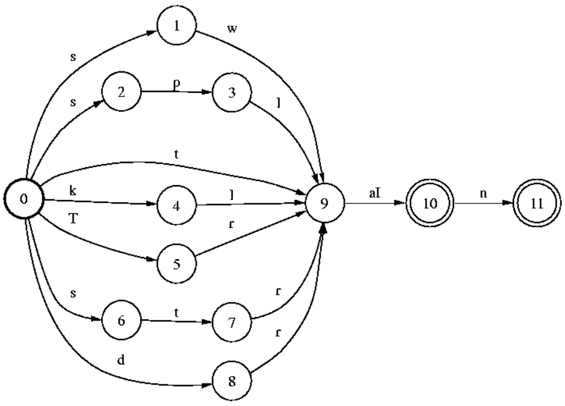


Figure 1: Phonotactic Automaton for English (subsection).

An MTM for a language represents an extension of the basic finite-state structure of the phonotactic automaton for the language to a multitape finite-state machine with each tape describing an additional annotation with respect to the original segment label. The induction procedure outlined in this section will further annotate the original segment label by introducing two additional transition tapes describing the frequency and relative frequency (probability) of occurrence of the segment in each phonotactic context with respect to a supplied training corpus. Also, if timing information is available, two further tapes will be added enhancing the annotation to include the average duration and

[1] Note that we assume here a unique start state for all finite-state structures. It can be easily shown that given a machine with multiple start states, an equivalent machine can be constructed having a single start state.

[2] `http://www.phon.ucdl.ac.uk/home/sampa/`

standard deviation of duration of each segment occurring in each phonotactic context.

The additional annotations described above can all be derived through the application of a regular grammatical inference algorithm applied to the initial corpus of syllable labelled data. We assume here that syllable labelled data consists of a phonemically labelled utterance (with or without timing/durational information) with syllable boundaries marked. This initial corpus and represents the training corpus for the inference algorithm and corresponds to the base annotation which is to be structured and further annotated using the MTM schema . As discussed in (Kelly, 2004, Section 3) it is meaningful to apply a regular inference algorithm here since the number of syllables in any given language represents a finite language (in the formal sense) and also since syllable phonotactics have been shown to be representable as finite-state machines (i.e. as phonotactic automata). Since we wish to further annotate the syllable labelled corpus with probabilities, a stochastic inference algorithm is required. Note that if probabilities are not required then we can apply the stochastic inference procedure and ignore the inferred probabilities. For the task at hand the choice of stochastic inference procedure is in fact arbitrary, however for the annotation methodology outlined here we utilise an implementation of the ALERGIA regular inference algorithm (Carrasco and Oncina, 1999).

ALERGIA uses the syllables of the training sample to first build a deterministic *Prefix Tree Automaton* (PTA) accepting exactly the supplied syllable data set. Following this, each pair of states in the PTA is compared using state frequencies derived from counts of syllables that terminate at states and transition frequencies derived from the common prefixes of syllables that occur in the sample. If a pair of states is found to (statistically) generate the same tail language based on the above frequencies then they are deemed (statistically) identical and merged. Through this state merging process a minimal stochastic deterministic automaton is inferred. Space prohibits a full discussion of the ALERGIA algorithm however further details concerning the inference algorithm applied to the task of learning phonotactic automata can be found in (Kelly, 2004) and (Carson-Berndsen and Kelly, 2004).

Since the automata inferred by ALERGIA are stochastic in nature, the frequencies and probabilities of states and more importantly transitions can both be output as transition tapes. The transition frequencies correspond to the frequency with which segments on the associated segmental tapes occur in particular phonotactic contexts. Similarly, the probability of transitions corresponds to the relative frequency with which the segments on the associated segmental tapes occur in particular phonotactic contexts. Further, the ALERGIA algorithm can be easily extended to take account of timing information that may be available in the corpus. Thus, if durational information relating to the segments of syllables is supplied as part of the initial annotations, e.g. start and end times of segments as seen in the TIMIT corpus (Garofolo et al., 1993), then an average duration and standard deviation of duration for each segment in each phonotactic context can easily be extracted and integrated into the inferred MTM as two additional tapes. This then provides two additional levels of annotation in the MTM.

As mentioned previously the initial MTM inferred from the corpus of syllable labelled data is stored as an XML document, the structure of which is rigorously governed by an XML schema. The schema specifies the set of allowable annotation tags that can appear in an MTM. MTMs specified according to this schema can be easily reused for continuing the annotation procedure described here and also for other applications that can take advantage of the chosen interface format. A portion of an inferred MTM showing the marked up structure of the unique start state and a single transition detailing the different levels of annotation discussed above is shown in figure 2. The single transition is from the state labelled 0 to the state labelled 2 with five tapes of information; a segment label (phoneme $/s/$), a frequency of occurrence, average duration and standard deviation of $/s/$ in the phonotactic context of the transition from state 0 to state 2 (in seconds) and also a weight tape denoting an inverse log probability for the transition.

```
<MTM language="ENG">
  <startStates>
    <state>0</state>
  </startStates>
  <finalStates>
    ...
  </finalStates>
  <transition>
    <sourceState>0</sourceState>
    <destinationState>2</destinationState>
    <phonemeTape>s</phonemeTape>
    <frequencyTape>2</frequencyTape>
    <durationTape>0.1045</durationTape>
    <deviationTape>0.0059</deviationTape>
    <weightTape>2.302</weightTape>
  </transition>
  ...
</MTM>
```

Figure 2: Portion of the XML representation of an MTM.

The application of a regular inference algorithm at this stage of the annotation procedure to automatically induce syllable phonotactics requires that a corpus of syllable labelled utterances be available, which may not always be the case. To counter this potential shortcoming, a semi-automatic incremental approach has been developed to derive syllable annotated data from phonemically labelled utterances. The derivation is carried out using an annotation assistant which successively displays phoneme labelled utterances from a given corpus to a user together with suggested syllable boundaries. The suggested boundaries are derived from a partial phonotactics that the system has built from previously annotated utterances. Consequently, the boundaries may or may not be correct and are subject to user verification after which the syllable annotation is integrated into the partial phonotactics. As successive utterances are syllabified the partial phonotactics becomes more complete and following a number of user supervised annotations the system can run in a fully automatic mode, syllabifying the remaining utterances and building a more complete phonotactics as it annotates. The system uses the chosen regular inference algorithm (again, ALERGIA is used in this particular case) to build the partial phonotactics after each syllable annotation and consequently a syllable phonotactics based on the corpus is produced in addition to the syllable annotations. The phonotactics can be output as an MTM in XML and delivered directly to the second phase of the annotation procedure as discussed in the following section. Further details on the annotation assistant can be found in (Kelly, 2004) and (Carson-Berndsen and Kelly, 2004).

3. Feature Set Definition

This section describes a *feature definition module* which facilitates user driven association of a multilingual feature set with a set of phonological symbols. This information is stored in an XML tree structure, a *feature profile*, and is used to annotate a particular MTM with segment specific symbol-to-phonological feature associations. An important consideration in the design of the module was to remove all necessity for technical knowledge of the operational and denotational workings of the technologies employed on the part of the user. At the same time, it was requisite that associations between symbols and features be defined within a coherent and useful structure that allowed easy access to the data by a range of applications and processes. The module provides an intuitive environment allowing users to define mappings between symbols and phonological features using only graphical representations of the data. The module encodes these feature definitions internally within an XML based feature profile. The structure of a feature profile is shown below (figure 4). Using XML as the data exchange format guarantees data portability across platforms and applications, while the module's interfaces ensure that the user need only deal with the data graphically for the purposes of feature definition, editing and display.

3.1. Feature Profile Creation

Feature profiles have an underlying XML representation that comprises any number of user defined feature associations, each of which individually consist of a symbol and a feature bundle. In addition, each association is annotated with a `<languages>` tag, denoting those languages for which that particular symbol-feature association is valid. In this way the feature set may be described as multilingual, as it is intended to provide a full inventory of phonological features for a complete symbol set across a number of languages. While the feature set is shared by all languages, the symbol set is language dependent. For the definition of the multilingual feature set, a dynamic approach to interface creation is adopted. Using the data from the induced MTM of section 2., the module automatically generates a feature input interface by extracting every unique symbol occurrence from the automaton's network. Since specifying the full state-transition structure of the phonotactic automaton underlying an MTM is an incremental process, subsequent passes through the growing network generate input interfaces only for those symbols which do not yet appear in the feature inventory. In this way we seek to reduce input replication and redundancy. The symbol-feature associations can be encoded in one of three ways: as mappings between symbols and unary, binary or multilevel feature structures. Unary features may be considered to be properties that on their own can be assigned to segments; binary features are attribute/value pairs which have two mutually exclusive values; multilevel feature structures consist of a number of tiers of information, each of which has an associated set of phonological features as parameters, from which one is chosen. Mappings between symbols and unary feature entities require that the entire set of possible features be first input to the module. From this data a Document Type Definition (DTD) is automatically created. This DTD is used to provide top-down constraints on the validation of future feature profiles which make use of the same feature set. From the DTD, a graphical feature input interface is generated. Symbol-to-feature mappings are created by using the module's graphical interfaces. Users first select a symbol and then click on those features which they wish to associate with it. Binary and multilevel features are defined in a similar fashion, with the additional step for multilevel structures of first inputting the tiers required, followed by the values each tier can take. Once the associations have been defined, the module adds them to the internal feature profile structure.

```
<!ELEMENT featureProfile (featureAssociations)*>
<!ELEMENT featureAssociations (symbol,languages,features*)>
<!ELEMENT symbol(#PCDATA)>
<!ATTLIST symbol notation ( IPA | SAMPA) #IMPLIED>
<!ELEMENT languages (language*)>
<!ELEMENT (#PCDATA)>
<!ELEMENT features (phonation?,manner?)>
<!ELEMENT phonation (#PCDATA)>
<!ELEMENT manner (#PCDATA)>
<!ELEMENT place (#PCDATA)>
```

Figure 3: Example Document Type Definition.

3.2. Interfaces

The feature definition module provides a number of user interfaces that allow the users to display, modify or add to the data stored within the feature profile. In accordance with the objective outlined above, all interfaces enable users to manipulate the data using only its graphical representations. Working with the graphically displayed feature profile tree, users can perform a number of editing functions: adding or removing features; changing symbols; deleting or modifying tier information etc. Any changes which affect the underlying document's structure as defined by its DTD - the inclusion of an additional tier, for example - are automatically updated within the DTD, subject to user confirmation. The default symbol set used throughout has an underlying representation as IPA-Unicode. However, a *notation transducer* allows users map a set of defined feature associations from this IPA representation to a number of alternative phonetic alphabets (e.g. SAMPA, WordBet, ARPAbet etc.). Interfaces are also provided for performing some data manipulation, e.g. extracting a language specific profile from the feature profile's multilingual superstructure.

Having defined a rich set of multilingual features, we seek to extract as much useful information from it for corpus annotation. The following section describes how we generalise over the data within feature profiles, seeking to optimise the information and highlight feature dependencies.

4. Data-driven Induction of Phonological Implications

Phonological implications are commonly specified in terms of hand-crafted rules which capture the set-theoretical relation of subsumption between two sets of phonemic units (Gazdar et al., 1985). This relation is often assumed for instance between the feature *nasal* and the feature *voiced*, expressing the observation that every nasal segment is also voiced but not vice versa. While we on the one hand believe in the usefulness of such rules even for the annotation of speech corpora, we argue on the other hand that it is favourable to induce these rules automatically for two reasons: firstly, the user-defined $\langle symbol, feature\rangle$-pairs which have been defined in the annotation during the step described in the previous section, may eventually be changed by the user at a later stage; this would require an undesirable manual re-evaluation of all rules. Secondly, there is no limit on the size of the phonological feature set which means that the set of implication rules may be hard to establish by hand, if this is possible at all. Thus, we present an automated method to achieve this informative set of implicational rules which is then used to further augment the present format of multilingual time maps.

The above motivations for phonological feature profiles lead to an expressive knowledge base which provides a fine-grained level of description for the modelling of individual phonological segments. However, we acknowledge the fact that such a rich set of features – despite its descriptive value – might not be easily accessible for manual optimisation, such as identification of implicational relations between individual features as well as possible combinations of features. Segment entries of the kind illustrated in figure 4, in this case segment [l], represent the input data for our automated method to extract information about feature distributions in our database.

```
<featureProfile>
   <featureAssociations>
      <symbol notation="SAMPA">l
      </symbol>
      <languages><lang>ENG</lang>
                 <lang>GER</lang>
      </languages>
      <features> consonantal,
                 lateral,
                 nonvocalic,
                 ...,coronal
      </features>
   </featureAssociations>
   <featureAssociations>...
   </featureAssociations>...
</featureProf>
```

Figure 4: Entry for [l] after the first module.

To obtain this valuable information, while equally eliminating the need for manual effort, we propose a computational method based on automated deduction that delivers correspondences between individual features as well as correspondences between all sets of sounds created by these combinations. Once the phonological feature associations have been defined via the previous module, we traverse the XML-trees with the aim of performing as much deterministic inference as possible. We apply our algorithm to automatically generate feature hierarchies similar to inheritance hierarchies in unification-based grammar formalisms, where features are ordered with respect to the size of their extents, i.e. the segment set they denote. This procedure is illustrated in figure 5. With regard to implication rules, the feature [rd] implies the feature [semihi] for the example given below, since the set of round vowels is subsumed by the set of semi-high vowels.

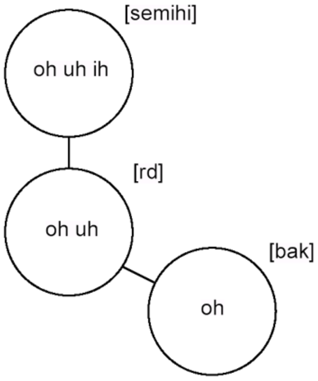


Figure 5: Induction of subsumption hierarchies.

We choose to “multiply out” every single combination of features to achieve its extent in terms of phonological segments. Finally this information is used to enrich the current phonological feature profiles with two elements distinguishing between bi-directional and unidirectional implications. To carry out efficient updates on our lexical knowledge base we use XSL which is a stylesheet language for transforming XML documents. In our case we intend to unify the set of all feature profiles with generalisations over this particular set yielding a more expressive feature profile. The following example, figure 6, displays the feature association for the segment [l] after it has been enriched with all logical implications gained from multiple tree traversal in our lexical knowledge base (note: the “features” subtree has been omitted):

We can see from this single entry that we are now able to say that the segment [l] introduces the feature [lateral] to our feature trees which in turn means that we can infer the presence of a segment [l] and all its additional features simply given the featural information [lateral]. Furthermore, we can observe that all features apart from [lateral] do not imply presence of the segment in question since they also occur in feature associations of other segments. All this information is based on automatically generated feature hierarchies as described in earlier work (Neugebauer and Wilson, 2004). The core of this work consists of an algorithmic method to deduce hierarchies which encode inheritance re-

```
<featureProfile>
   <featureAssociations>
      <symbol notation="SAMPA">l
      </symbol>
      <languages><lang>ENG</lang>
                 <lang>GER</lang>
      </languages>
      <features> ... </features>
          <introducing>lateral
          </introducing>
          <sharing>consonantal,
                   nonvocalic,
                   ...,coronal
          </sharing>
   </featureAssociations>
   <featureAssociations>...
   </featureAssociations>...
</featureProf>
```

Figure 6: Entry for [l] after the second module.

lationships among sets of segments and features for a single language or even for different languages. For the purposes of this paper, the following three steps summarise the procedure which augments the *introducing* and *sharing* nodes in the feature trees.

1. for each feature defined in the feature profile, traverse the individual feature associations to determine the extent (the segments for which the feature is defined) of each single feature

2. create all the sets of segments which are denoted by single features and if the set contains only exactly one element, add an *introducing* node to the feature tree of that particular element

3. compute the complement of the feature subset in question and store the result as a value of the created *sharing* node of the feature tree

Our method does not only serve to establish implications which hold for the domain of single segment entries but since we generalise over the all entries we also achieve similar information for whole sets of sounds. Consider for instance the following implicational generalisations which are provided in the table below: if we know the features in the leftmost column, we can infer the features to their right. The set of sounds in the final column is the set of sounds which share the union of unique and shared features; in the last row all elements which carry the feature [round] are displayed which maps onto the set of round voiced vowels which is expressed in the following implication rule: [round] → [voiced, vocalic].

The information within the resulting feature tree is used to further annotate the MTM of section 3.. The fully specified feature associations for each symbol within a particular feature profile are extracted and used to construct an additional input tape for the MTM. The mapping component of the optimisation procedure discussed in this section traverses the MTM and inserts a tape containing the phonological feature information for each occurrence of the associated symbol within the network. Similarly, once the feature profiles have themselves been augmented with information regarding optimised feature sets, further tapes indicating feature redundancy or uniquity can be extracted and dispersed throughout the MTM.

high	voiced, vocalic, round	{iy, uw}
front	voiced, vocalic, round	{iy}
back	voiced, vocalic, round	{ao, uw}
semilow	voiced, vocalic, round	{ao}
round	voiced, vocalic	{iy, ao, uw}

Figure 7: Examples for induced implication rules

The next section provides a practical example of how the results of the methodology presented can be explicitly utilised by a linguistic application, more specifically by a syllable recognition system.

5. Enrichment for Syllable Recognition

The previous sections of this paper have discussed an incremental approach to enrich phonological corpus annotation at different levels. This section discusses how the MTM and its contents can be used by a computational phonological model of syllable recognition, and how the testing phase can be employed to further enrich the annotation of the corpus.

The Time Map model (Carson-Berndsen, 1998) is a computational phonological model, which is directly applicable to speech recognition, that employs a phonotactic automaton and axioms of event logic to interpret multilinear feature representations of speech utterances. The model distinguishes two temporal domains. The first domain is absolute signal time where features are considered as events with temporal endpoints. The features are extracted from the speech signal of an utterance using Hidden Markov Model techniques and have extraction probabilities associated with them. The utterance is then represented as a multilinear structure of phonological events. Figure 8 illustrates an example of a multilinear representation (in absolute time) with events on different linguistic tiers (such as phonation, manner, place etc.). Note that the model is not procedurally bound to any one particular feature set. The second temporal domain is relative time and considers only the temporal relations of overlap and precedence as salient. Input to the model is in absolute time. However the parsing process takes place in relative time using only the overlap and precedence relations, and is guided by the phonotactic automaton which imposes top-down constraints on the feature overlap relations that can occur in a particular language. The *Time Map* model has been implemented in both a generic framework (Carson-Berndsen and Walsh, 2000) and a multi-agent environment (Walsh et al., 2003).

In brief, the MTM (XML) representation of the phonotactics of a given corpus is used by the parsing algorithm as an anticipatory guide. For example, from the initial node of the phonotactic automaton, which defines well-formed syllables of the corpus, a number of candidate segments are anticipated. The MTM representation specifies these segments with respect to their constituent temporally overlap-

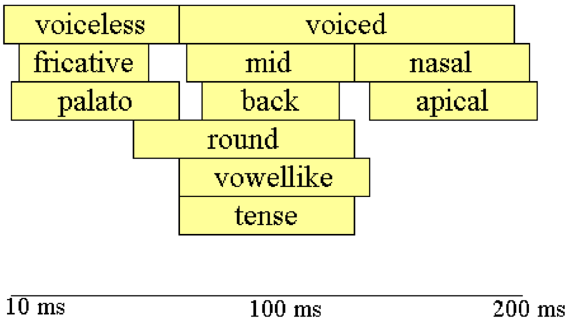


Figure 8: Multilinear representation of [So:n].

ping phonological features. It is these overlapping features in the multilinear feature representation of the speech utterance which the parser seeks to identify for each of the anticipated candidate segments.

The parser gradually windows though the utterance, guided by the phonotactics, and each time the constituent features of an anticipated segment are found the segment is considered recognised and the parser traverses the transition in the phonotactics. Each time a final node in the phonotactics is reached a syllable hypothesis is recorded. This represents a concrete and novel application of how a richly annotated analysis of a corpus can be exploited by a syllable recognition system.

It is worth noting however that aspects of the recognition life-cycle can be harnessed to further add to the annotation. While the XML-based phonotactics presents segments with their constituent overlapping phonological features there is no indication of the relative importance of these feature overlaps. In the context of Time Map syllable recognition, where each segment is recognised through the satisfaction of a number of feature overlap constraints, it would be desirable to know the relative importance of these constraints to the recognition of the segment as a whole. In particular, given that the phonological features present in the multilinear representation of the utterance may be extracted with low probability from the signal, or that underspecification occurs in the representation due to background noise etc., it would be beneficial to know which overlap constraints are most important. With respect to the phonotactics it would be desirable to enhance the MTM representation with a ranking of feature overlap constraints for each phoneme segment. In this way the MTM representation explicitly captures the relationship between features. Furthermore, given the declarative nature of the representation it can be used not only for speech recognition but also in other speech domains, for example speech synthesis.

This augmentation of the phonotactics is achieved by adding the following tags, illustrated in figure 9, as children of the `<transition>` element of figure 2.

The `<overlapConstraint>` element and its children define a temporal overlap constraint between two features and assign a rank value, a probability, for that constraint. Obviously a number of such elements are required in order to capture all the feature overlaps which constitute a phoneme segment. The `<threshold>` element denotes the total sum of all the ranks. Each time a constraint is recognised in the input its rank is added to a running total. If this total reaches the threshold then the segment is deemed to be recognised. The rank and threshold values presented above are arbitrary, provided merely for the purposes of illustration. The next step is to acquire real values for ranks and thresholds.

```
<overlapConstraint>
    <ranking>0.7</ranking>
    <feature-info1>
        <feature-name>labial</feature-name>
    </feature-info1>
    <feature-info2>
        <feature-name>voiced</feature-name>
    </feature-info2>
</overlapConstraint>
<threshold>1</threshold>
```

Figure 9: Adding overlap Constraints.

The acquisition process is divided into three stages. The first stage involves HMM training, for each feature in the feature set, using 70% of the corpus. The second stage takes this same 70% for testing, i.e. running the parser over this subset of the corpus. It should be noted that this is performed with retraining in mind, not for the purposes of producing overall system recognition results (which typically involves testing with the remaining 30%). The result of parsing the corpus subset is a corresponding output file for each utterance in the subset, containing a string of phonemes. The following example is used to describe the third stage. Figure 10 presents a window between 150ms and 180ms of a speech utterance where three phonological features *x*, *y* and *z* (presented with their extraction probabilities) all overlap in time.

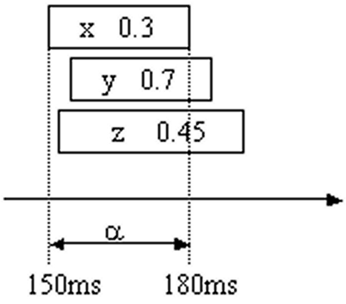


Figure 10: A sample window.

A phonemic segment α is recognised only when three overlap constraints are satisfied, namely $x \;^{o}\; y$, $x \;^{o}\; z$, and $y \;^{o}\; z$, as in figure 10.[3] The third stage involves taking each output file and comparing it against the corresponding phonemically labelled reference file in the corpus. An edit distance measure is used to calculate where the recognition output and the reference match. Each time a phoneme seg-

[3] The ^{o} symbol denotes overlap.

ment is correctly recognised the window of the multilinear representation of the utterance in which it was recognised is re-examined. Each overlap relation in the window which contributed to the recognition of the segment is considered, the extraction probabilities of both contributing features being multiplied in order to arrive at a rank value. Taking figure 10 as an example, α is recognised in the input and is also present in the same position in the reference file. This indicates a successful recognition. At this point the window is re-examined in order to rank the overlaps relative to each other by multiplying the probabilities in each relation. For example the contribution of $x \ ^o \ y$ is 0.21, for $x \ ^o \ z$ it is 0.135, and for $y \ ^o \ z$ it is 0.315. These figures (once normalised to add to 1) could form the rank value for each of the overlap constraints for α if α only occurred once. However it is likely that a given segment would occur numerous times and hence this process is repeated for each occurrence and a running total kept for the rank of each constraint. An average rank value is then derived by dividing by the number of occurrences.

Following the description above it is possible to further enrich the MTM representation of the phonotactics and provide data driven statistical values for the relative importance of feature overlap constraints. Knowing which constraints are most significant is beneficial as it allows certain constraints to be relaxed, in the case of underspecification, effectively meaning that a segment could be considered satisfactorily recognised even though not all of its constraints had been satisfied. This is equivalent to lowering the threshold.

6. Conclusions & Future Work

This paper has presented a systematic language independent methodology for richly annotating speech corpora at multiple levels of linguistic granularity using common data categories familiar to speech researchers. The three stage procedure combines data-driven and user-driven approaches to annotation. The resulting annotations are stored in an XML marked-up document known as a Multilingual Time Map embodying a data category registry. This portable structure incorporates data categories useful and common in many areas of speech research and by way of example we have shown how such a Multilingual Time Map can be employed and indeed further enriched by a syllable recognition system.

7. Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 02/IN1/I100.
The opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Science Foundation Ireland.

8. References

Carrasco, Rafael C. and Jose Oncina, 1999. Learning deterministic regular grammars from stochastic samples in polynomial time. *ITA*, 33(1):1–19.

Carson-Berndsen, Julie, 1998. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*. Dordrecht, Holland: Kluwer Academic Publishers.

Carson-Berndsen, Julie, 2002. Multilingual time maps: Portable phonotactic models for speech technology applications. In *Proceedings of the LREC 2002 Workshop on Portability Issues in Human Language Technology*.

Carson-Berndsen, Julie and Robert Kelly, 2004. Acquiring reusable multilingual phonotactic resources. To Appear in Proceedings of the 4th International Conference on Language Resources and Evaluation.

Carson-Berndsen, Julie and Michael Walsh, 2000. Interpreting multilinear representations in speech. In *Proceedings of the Eighth International Conference on Speech Science and Technology*. Canberra.

Garofolo, John S., Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren, 1993. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. Technical report, National Institute of Standards and Technology.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag, 1985. *Generalized phrase structure grammar*. Oxford: Blackwell.

Kelly, Robert, 2004. A language independent approach to acquiring phonotactic resources for speech recognition. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*. CLUK04.

Neugebauer, Moritz and Stephen Wilson, 2004. Phonological treebanks - issues in generation and application. To Appear in Proceedings of the 4th International Conference on Language Resources and Evaluation.

van Noord, Gertjan, 1997. FSA Utilities: A toolbox to manipulate finite-state automata. In Darrell Raymond, Derick Wood, and Sheng Yu (eds.), *Automata Implementation*, Lecture Notes in Computer Science 1260. Springer Verlag, pages 87–108.

Walsh, Michael, Robert Kelly, Gregory M.P. O' Hare, Julie Carson-Berndsen, and Tarek Abu-Amer, 2003. A multi-agent computational linguistic approach to speech recognition. In *Proceedings of the Eighteenth International Joint Conference on Artifical Intelligence*. Acapulco, Mexico: IJCAI-03.

Metadata for Time Aligned Corpora

Thorsten Trippel

Fakultät für Linguistik und Literaturwissenschaft
Universtität Bielefeld, Germany
ttrippel@spectrum.uni-bielefeld.de

Abstract

For a detailed description of time aligned corpora, for example spoken language corpora and multimodal corpora, specific metadata categories are necessary, extending the scope of traditional metadata categories. We argue that it is necessary to allow metadata on all levels of annotation, i.e. on a general level for catalogues, on the session level for each recording, on the annotation level for multi tier score annotation, even on the level of individual annotation segments. We use existing standards where they allow this distinction and introduce metadata categories for the layer level.

1. Motivation

Metadata descriptions for spoken language corpora and multimodal corpora are different from textual resources due to fundamental differences in structure of multimodal corpora. Nevertheless, the detailed description is of highest importance for the classification of resources enabling reusability and portability, which is crucial due to the costs of the creation of corpora that are built on this sort of data.

Spoken language corpora are extremely expensive, much more than collections of (written) texts. According to various investigations, annotation time ranges from the factor 10, i.e. 1 minute signal requires 10 minutes for the annotation, to the factor 100 for the annotation of multimodal data (see for example Gibbon et al., 1997a), which results from the time needed for segmentation and transcription depending on the linguistic level.

Spoken language corpora and multimodal corpora require larger storage and transmitting capacity, as such a corpus consists not only of annotations, based on audio or video recordings, but the signal is also part of the corpus, as it is the signal that enables the use of certain corpora, for example, for applications in psycholinguistics, speech engineering and phonetics, but also to a certain degree for sociolinguistics and language varieties studies. To avoid additional costs for obtaining corpora and for file transfer in addition to locating relevant information, very detailed descriptions of the content are required.

2. Spoken Language Corpora and Multimodal Corpora

2.1. Characteristics of Spoken Language Corpora and Multimodal Corpora

The representation of spoken language corpora and multimodal corpora is different from textual ones. For example, Leech, 1993, argues that it is impossible to distinguish between the representation and the interpretation for spoken language corpora, as the textual representation of speech implies the interpretation by an annotator. As annotation of speech is part of multimodal corpora, the same is true for this more general class of corpora. In fact, in the following the terms *spoken language* and *multimodal* in the context of corpora will be used interchangeably, as the latter is a more general form of the former and the difference in practice is currently related to the available tools, signal coding and annotation schemes rather than the descriptions thereof.

Gibbon et al., 1997b, differentiate between spoken language corpora and non-spoken language corpora by eight characteristics, which are taken from a technical perspective but address also content and ethical differences. A technical problem is the volatility of speech data, which is one main characteristic of speech. The signal disappears as soon as it is released. This stresses the need for persistent storage of audio signals, covering problems which range from recording quality (environmental conditions, quality of equipment, etc.) to storage (including data formats, compression algorithms — if any— and bandwidth, storage space, among others). A major difference, related to the volatility, is the processing of the language that is oriented towards the actual performance time. This has consequences for error handling — while, in non-spoken language, a writer might make sure not to show his or her corrections, false starts, etc., in spoken language repairs and hesitations occur. The same is true for the recognition of words and structures, which, in a written format is given by categories identified by letters, spaces and typographical symbols. For spoken language utterances the segmentation needs additional processing , i.e. unit separation and identification.

2.2. Data Formats for Spoken Language Corpora and Multimodal Corpora

Spoken language corpora and multimodal corpora consist of two parts:

1. the signal, which is usually an audio or video signal, which needs to be stored in a processable format. The format algorithms and specifications are not part of this description, but they need to be documented in the metadata.

2. the annotation, which is aligned with the signal using references to the signal time, so called *timestamps*. The annotation itself could be interpreted as metadata for the signal, but for the present the metadata discussed are the descriptions for the annotations.

The last point here already refers to another problem, which is the distinction between data and metadata, which can become rather obscure and fuzzy. For example for a wordlevel annotation, a wordclass characterisation can be interpreted as meta information on the word, though on a different tier wordclasses could be annotated separately as data.

To avoid this, a strong restriction for metadata categories is required, where the content categories are reduced as much as possible, editorial information can be applied automatically by applications, based on the user name and a system date, besides application inheritant technical descriptions. Other information can be coded to distinguish different annotation layers from each other and to identify them.

Annotations are available in two classes resulting from a different background.

Document-centred annotation formats, which resemble a textual structure with paragraphs and possibly with headings. The reference timestamps are included using so called *milestones* (Barnard et al., 1995), which are pointers to a specific point on the linear scale. In XML this is typically done with empty elements holding attributes that can occur at any position in the document. Intervals are defined implicitly as the region between two milestones or pointers to the timeline.

Document centred annotations typically do not use more than one annotation of a signal, everything is given on one tier only.

A document centred application for signal data annotation is *Transcriber* (Barras, 1998-2003). The TEI (Sperberg-McQueen and Burnard, 1994, Section 11) provides another structure for the annotation of speech, though it does provide a reference to time only by pointers to an external timeline (see Sperberg-McQueen and Burnard, 1994, Section 14), instead of a direct timestamp.

Annotation is bound to the word or phrase level, other linguistic levels of annotations are not intended.

Data centred annotation formats, in which the annotation is clearly structured into the units according to the level of annotation. These formats explicitly or implicitly define intervals on the timeline, i.e. they either represent start and end points of the interval, or one or the other is inferred from the preceding or following segment, respectively. Text-like structures, such as paragraphs, headings, and sections are not represented; subject classification by keywords is part of the metadata which can be included in some formats.

Data centred annotation formats can easily include more than one annotation level by adding new tiers referring to the same timeline.

Data centred applications include software and data formats from the area of phonetics such as *Praat* (Boersma and Weenink, –2004), *wavesurfer* (Sjölander and Beskow, 2004, Sjölander and Beskow, 2000), ESPS waves$+$ (Entropic, 1998), and tools for multimodal corpora, such as the *TASX-annotator* (Milde and Gut, 2002), *ELAN* (ELAN, –2004), *ANVIL* (Kipp, 2000-2003) and *EXMARaLDA* (Schmidt, –2004), which itself uses the data format of the *Annotation Graph Toolkit* (Bird et al., –2004).

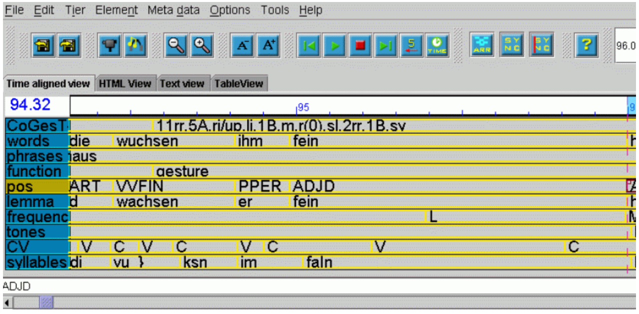


Figure 1: Multi tier annotation using the TASX-annotator. Annotation levels: Gesture, words, phrases, gestural function, part of speech, lemma, prosody, and syllables.

The data-driven formats imply a graph structure which is easily modelled by the Annotation Graph Model (Bird and Liberman, 2001). These formats for the annotation are mostly equivalent language bindings of this model, having certain differences in the metadata and in the interpretation and typing of individual tiers.

Data driven annotation formats are the most frequently used formats in the annotation of spoken and multimodal data and provide the *de facto* standard in multi level annotation. Even document-centred annotations can easily be transformed into these formats. The TASX format, for example, has been used as an intermediate data format for the interchange between XWaves, Praat, Transcriber, TASX-Annotator, etc.

Figure 1 illustrates a multi tier annotation using the TASX-annotator.

3. Encoding Metadata

Metadata as the description of resource properties does not in itself describe a structure. Nevertheless, existing metadata descriptions include and propose structures for these metadata. These structures vary from simple attribute-value structures via more structured triples to deeply structured trees.

3.1. AV-Structures

Most existing metadata standards, such as Dublin Core and OLAC, provide an attribute-value structure. Data categories are named and assigned a specific value. This type of structure corresponds to simple Attribute-Value (AV) Structures that have been in use in knowledge engineering for a long time. It provides a flat structure that can easily be stored in relational databases.

The relation between the attribute and value is predefined and can be described as a *ISA*-relation, for example, the *author* of a resource is the *person* denoted by the name given in the value.

3.2. Triples

The predefined relation between attribute and value is a shortcoming of AV-structures, if other relations are required. In knowledge engineering the concept of triples was introduced to avoid this shortcoming. In these triples, just as in AV-structures, a value (an object) is assigned to a property (a subject), but in contrast to AV-structures, the relation (a predicate) is explicitly given, as well. Consequently, every AV-structure can be represented as a triple, by explicitly stating the predicate. One example for this is the *Knowledge Interchange Standard* (KIF) (see KIF, 1998).

Another example is the *Resource Description Framework* (RDF, (Beckett, 2004)) standard issued by the World Wide Web Consortium (W3C).

3.3. Tree Structures

A different kind of structure is given by tree structures. These describe a hierarchical relation between metadata categories. The relation between metadata categories can be of a different nature, as well. Usually it is a taxonomic relation (*ISA*-relation) between subordinate and superordinate categories. One reason for this kind of hierarchy is the inheritance of information from superordinate categories. This is either done by subclassification, i.e. the data categories are specified — given a meaning — by their position in the hierarchy, or the granularity of the metadata is determined by the position in the hierarchy, i.e. a subdividable value is either given on a superordinate level or subdivided on a lower level.

For language resources, the IMDI (IMDI, 2001) standard as well as the TEI metadata header (Sperberg-McQueen and Burnard, 1994) provides a hierarchy of metadata. Both use the hierarchy for the classification of categories and not for determining granularity.

Tree structured metadata categories contain the most information but are relatively difficult to cope with in terms of relational databases with their flat structure. However, the relation of categories to a hierarchy can easily be expressed using a pointer to an existing ontology, such as in Farrar and Langendoen, 2003.

4. Metadata Levels

Metadata sets for the description of corpora are available in different forms, allowing the cataloguing of these corpora and accessing them, providing a general description of these metadata. Widely used are the following:

DC (see Dublin Core, 2003) developed for the cataloguing of resources in libraries the DC metadata is suitable for bibliographical sources such as texts, articles and books; however for corpora and multimodal data a lot of metadata categories are missing.

OLAC (see Simons and Bird, 2002): developed for the cataloguing of linguistic resources in data repositories. OLAC metadata provides additional linguistic data categories and some technical information of electronic material for linguistic resources of one language, one annotation, one annotator and one medium.

TEI metadata header (see Sperberg-McQueen and Burnard, 1994): developed for encoding texts, the TEI metadata allows the encoding of metadata categories relevant for textual sources which follow the TEI document grammar.

IMDI (see IMDI, 2001 and IMDI, 2003): from the language engineering perspective approaches resources from the catalogue[1] and session level[2] differently, allowing an inclusion of multimodal data in data repositories and describing annotations to a certain degree.

The main shortcomings of all metadata sets remain the following:

underspecification of data categories: Underspecification in this context is a problem related to the granularity and semantics of metadata categories, which is defined in terms of human perception rather than on a formal basis. This results in a basic absence of a classification. An example is a data category *language*, which does not by itself provide information whether the language of description, language of the content, or native language of a speaker is under consideration. Introducing new subcategories does not necessarily solve the problem because a less standardized vocabulary or a larger number of words in the controlled vocabulary result in too large adaptions for tools. However, by defining content models for all data categories — besides already existing closed vocabularies — and defining the semantics of the data categories formally, this problem could be solved.

multilingual sources: Almost all metadata standards are targeted at monolingual resources, though for simple implementations a multiple use of data categories is allowed. For example in Dublin Core, 2003, it is only possible to describe that a resource contains information in a number of languages, not which part of the resource contains information in a particular language. This can be solved by differentiating between different languages in different annotations layers and describing these annotations separately.

multi-participant sources: The same as for the multilingual resources always applies if some characteristic is used with different values, also with persons. Though a subclassification exists for different persons connected to a resource, such as publisher, author, editor, etc., the standards allow listing persons and items only. It seems to be assumed that the resource can be described as a whole, though the parts may be clearly distinguishable, such as the speaking person at a given position of the signal or authorship of a specific section.

Distinguishing the differences by inserting layers or tiers for every level or speaker enables direct access to this information.

However, this does not solve the problem of metadata description of this information provided on different layers, such as standoff annotation (McKelvie and Thompson,

[1] The catalogue level is a general description of the corpus as such.

[2] The session level can be described as one individual recording within a corpus.

1997), primary data identical annotation (Witt, 2002), or even multi-tier score as common in signal annotation (e.g., annotations on different linguistic levels represented on tiers with `Praat`, `TASX-Annotator` or `wavesurfer`) where different levels can include

- different kinds of annotation units, following different annotation standards and linguistic theories,
- annotations by different annotators,
- a variety of different annotation dates and periods,
- a variety of annotation tools, resulting in different restrictions to the annotations,
- different languages, where for multilingual signals (e.g. interpreted speech) each language is annotated on a separate tier,
- ...

To approach this problem a further abstraction is required, namely the introduction of metadata levels or metadata categories for different uses.

4.1. Metadata Categories for Different Uses

The problems of metadata categories for different annotation units, linguisic theories, etc., can easily be solved by distinguishing different types of metadata. These are illustrated by Figure 2.

- Metadata on the catalogue level for a *Resource Description Library*: This includes the bits of information used in large data repositories for locating a specific resource providing basic information for the retrieval of further metadata, such as the file format and location of the resource and infrastructure requirements for retrieval (such as software to access the data). This can be seen in comparison to an abstract for an article, or a sales brochure for a product. The information given is highly conventionalized and relatively independent of the resource under consideration.
- Metadata on the individual annotation levels for a detailed *Linguistic Description*: This information is used for applications and for detailed research questions. Metadata for linguistic description are the specification of the annotation of a corpus or can be interpreted as a sort of *user manual*. These descriptions include:
  - Metadata on the session level: On the session level information is needed with regard to the structure — the data format — and the content of the individual primary data.
  - Metadata on the layer level: this includes information about the specific annotation such as the annotator, annotation formalisms including data format and encoding, technology used in annotation, etc.
  - metadata about the actual annotation event, which might include deviations from the layer metadata or technical information for retrieval software.

These different metadata levels are interrelated by sharing data categories and information. However, the linguistic description needs to be far more detailed.

4.2. Suggested Metadata Encoding

As the representation of metadata in tree structure has advantages in guiding the user, the metadata encoding should refer to an ontology of metadata categories. However, to allow more efficient storage and processing, all categories (*leaves* in tree terminology) should have unique names. In IMDI, for example, *type* is used context dependently. To provide context independent naming of categories *type*, should be qualified as *type of recording, type of medium*, *type of resource*, etc. If this is provided the metadata can be processed in AV-form or, if a predicate is given, in RDF or another knowledge format.

5. Metadata for Time Aligned Corpora

Spoken language corpora and multimodal corpora are both time aligned and can be described on various levels of granularity: firstly, generally as a whole, called *catalogue level*, secondly a description for every part of the signal, called *session level*, thirdly, a detailed documentation of the annotation for every level of annotation or every annotation tier, called *layer level*, and finally for every annotated segment, called *event level*. A similar structure can be found in the MPEG-7 standard, where regions, segments, objects, etc. can be described (MPEG-7, 2003).

The MPEG-7 standard was created to enable easier access and querying multimedia resources such as videofilms and audio recordings, based on a time aligned annotation. Salembier and Smith, 2001 describe the scheme for multimedia data, based on categories similar to Dublin Core, 2003, describing recording information (called creation information), storage media information, and information related to intellectual property rights. However, linguistic data categories are not intended and the use is intended for large media archives and not for linguistic corpora. Hunter and Armstrong, 1999 describe different schemas for video annotation, based on an early version of RDFS, RDF, DTD, etc., allowing arbitrary metadata categories, mentioning the problems free metadata categories cause in the context of non-standardized archives.

The metadata categories listed were motivated by the creation of multimodal corpora for the creation of multimodal lexica[3]. A detailed description of the metadata for this specific corpus can be found in Trippel and Baumann, 2003. The metadata classification is used for the automatic induction of lexica from corpora as described by Trippel et al., 2003 to allow to create a lexicon microstructure.

[3] The corpora were funded by the German Research Council grant to the project *Theory and Design of Multimodal Lexica*, Research Group *Text Technological Information Modelling.*

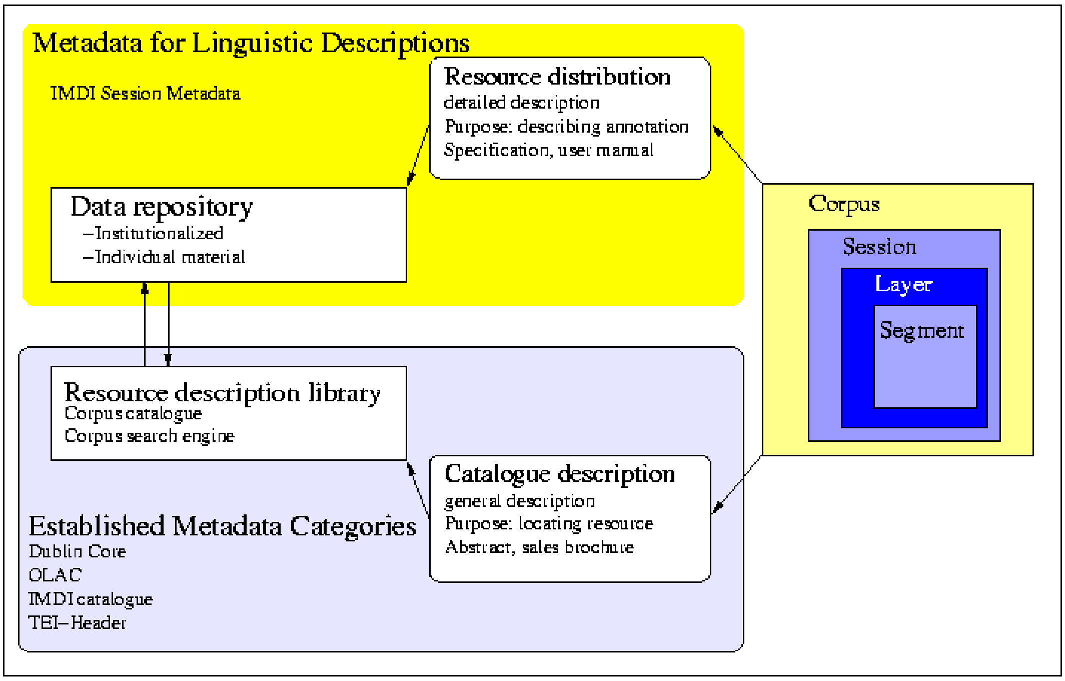


Figure 2: Metadata categories for corpora by intended use

5.1. Catalogue Level Metadata for Time Aligned Corpora

For the catalogue level the descriptions of Simons and Bird, 2002, or IMDI, 2001, can easily be adapted for specific project requirements. The latter is more detailed and provides a mapping to the former as well as to Dublin Core, 2003, and can therefore be used in contexts where these are used as a standard.

In IMDI, 2001, the data categories are structured hierarchically, and the lowest level elements still have unique names, resulting in the option of storing and processing them in table formats such as relational databases.

5.2. Session Level Metadata for Time Aligned Corpora

The most detailed description for session metadata for multimodal corpora is given with the IMDI proposal (IMDI, 2003), providing a hierarchy of metadata for information such as *Session*, with subcategories *Name*, *Title*, *Date*, *Location*, etc.

Due to the reduplication of category names on a low level, this system cannot be used directly in a relational database. A solution is to qualify the leaves of the tree in order to make sure they are unique. This includes the combination of the category name with the name of the superordinate category.

As the session is distinguished from the specific annotation level, some metadata categories are not required if they are recorded on the other layers. The metadata categories on this level can indeed be inferred from information on lower levels; for example, the list of annotators for one session can be inferred from the annotators of the individual levels. However, this inference is part of the *MetaLex* approach as described by Trippel et al., 2004, and is often implied as the description levels are not distinguished in many contexts. In the corpora mentioned above, all categories that are not directly related to annotation on a specific annotation level have been recorded here, using the IMDI categories.

5.3. Layer Level Metadata for Time Aligned Corpora

The data categories on the layer level are defined according to appropriate categories from session and catalogue level. As these are not available in other systems they are described in detail. The categories are given in a hierarchy, implying an ontology, though the naming allows processing in table form, as well.

Information class: classes of information with the following subclasses:

phonemic: annotation on phonemic level, for example describing individual constituents.

syllabic: annotation on syllable level, for example:

- orthographic: orthographic syllable annotation
- phonemic: phonemic syllable annotation
- phonetic: phonetic syllable annotation

word: annotation based on word level segmentation, such as:

- orthographic: orthographic word annotation in standard orthography
- phonemic: phonemic word annotation
- phonetic: phonetic word annotation
- syntactic: syntactic word annotation
- lemma: lemmatization
- morphemic: morphemic segmentation

prosodic: prosodic annotation based on:

- tones
- breaks

phrase: annotation on larger units, again on different levels:

- orthographic,
- tones,
- syllables,

gloss: interlinear gloss with a specification of the gloss language

hand/arm gesture: arm gesture annotation

- left: left hand and arm
- right: right hand and arm
- pair: annotation of the movement of the limb pair
- complex: annotation of complex gestures
- function: functional gesture annotation
- spatial relation: spatial relation between the limb pair

data warehousing: information concerning annotating personnel, annotation and version with subcategories:

annotator: description of each annotator in terms of:

annotator name

annotator native language

annotator other language

annotator qualification

annotator comment: comments on the annotator

annotator role: function of the annotator, which is relevant especially if more than one annotator is involved

annotator affiliation

annotation date

annotation revision

annotation software

annotation media: media used for annotation, for example audio or video for speech annotation

annotation status: status of the annotation, e.g. *finished*, *work in progress*, *to be revised*

layer title

misc: prose text with other relevant information

5.4. Event Level Metadata for Time Aligned Corpora

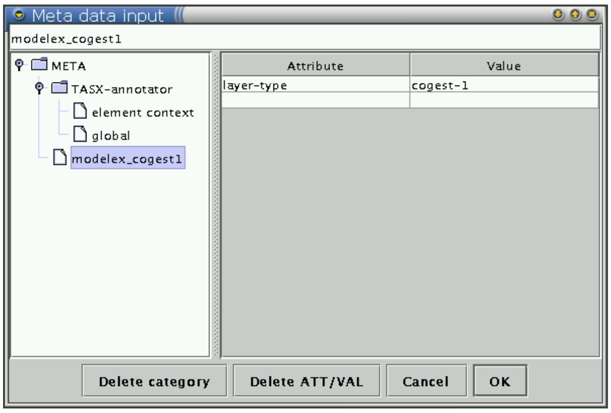
Every information deviating from the layer level needs to be recorded with each segment. For example an annotation can be done by one person, who will be the *annotator*, but one segment is corrected by somebody else, who needs to be specified at this segment.

This feature is currently used to store technical information for a segment, such as font selection by the TASX-annotator.

6. Technical Realisation of the Level Based Metadata Concept

The level based metadata concept has been implemented and used in the TASX format, which is used by the TASX-Annotator (Milde and Gut, 2002) and the PAX audio concordance system (Trippel and Gibbon, 2002). The TASX grammar allows the flexible insertion of metadata on all levels of annotation. This was motivated by the idea of interchanging data created with different tools without data loss, and allowing to store the metadata with the original data.

Figure 3 shows the TASX-annotators metadata editor creating a metadata entry on one annotation level.



Attribute	Value
layer-type	cogest-1

Figure 3: Metadata editor of the TASX-annotator

7. Summary and Outlook

The necessity for metadata on all levels of annotations has been explained. For some of these levels there are recommendations and proposals for a metadata inventory. These proposals have to be further adapted to allow processing in hierarchical form by linking to ontologies and in flat table structures and lists.

For the other levels for time aligned corpora, especially on levels below session level, there is need for furhter discussion. This paper was intended to contribute to the discussion that may lead to a standard for the documentation of this class of corpora.

Further work has to be done to allow the inference of metadata within annotation tools, in order to increase consistency and usability.

Acknowledgement

The work presented in this paper was funded mainly by the German Research Council grant to the project *Theory and Design of Multimodal Lexica*, Research Group *Text Technological Information Modelling.* The author is grateful to the researchers affiliated with this group.

8. References

Barnard, David T., Lou Burnard, Jean-Pierre Gaspart, Lynne A. Price, C.M. Sperberg-McQueen, and Giovanni Battista Varile, 1995. Hierarchical encoding of text: Technical problems and sgml solutions. In Nancy

Ide and J. Véronis (eds.), *The Text Encoding Initiative: Background and Context*. Kluwer Academic Publishers, pages 211–231.

Barras, Claude, 1998-2003. Transcriber – a tool for segmenting, labeling and transcribing speech. Software, download at http://www.etca.fr/CTA/gip/Projets/Transcriber/, checked January 2004.

Beckett, 2004. Resource description framework (RDF) model and syntax specification. Http://www.w3.org/TR/rdf-syntax-grammar/.

Bird, Steven and Mark Liberman, 2001. A formal framework for linguistic annotation. *Speech Communication*, (33 (1,2)):23–60.

Bird, Steven, Xiaoyi Ma, Haejoong Lee, Kazuaki Maeda, Beth Randall, Salim Zayat, John Breen, Craig Martell, Chris Osborn, and Jonathan Dick, –2004. AGTK: Annotation graph toolkit. Software, download at http://agtk.sourceforge.net/.

Boersma, Paul and David Weenink, –2004. Praat. http://www.praat.org, checked January 2004.

Dublin Core, 2003. DCMI metadata terms. http://dublincore.org/documents/2003/03/04/dcmi-terms/.

ELAN, –2004. EUDICO linguistic annotator. Software, download at http://www.mpi.nl/tools/elan.html.

Entropic, 1998. Xwaves. Software. No longer maintained.

Farrar, Scott and D. Terence Langendoen, 2003. Markup and the gold ontology. In *Workshop on Digitizing and Annotating Text and Field Recordings*. LSA Insitute, Michigan State University. Published at http://saussure.linguistlist.org/cfdocs/emeld/workshop/2003/langoen-paper.pdf.

Gibbon, Dafydd, Roger Moore, and Richard Winski (eds.), 1997a. *Handbook of Standards and Resources for Spoken Language Systems*, chapter SL corpus collection. Berlin and New York: Mouton de Gruyter.

Gibbon, Dafydd, Roger Moore, and Richard Winski (eds.), 1997b. *Handbook of Standards and Resources for Spoken Language Systems*, chapter SL Corpus Design. Berlin: Mouton de Gruyter.

Hunter, Jane and Liz Armstrong, 1999. A comparison of schemas for video metdata representation. In *Proceedings of the Eighth International World Wide Web Conference*. Toronto. Http://www8.org/w8-papers/3c-hypermedia-video/comparison/comparison.html.

IMDI, ISLE Metadata Initiative, 2001. Metadata elements for catalogue descriptions draft proposal version 2.1. http://www.mpi.nl/world/ISLE/documents/draft/IMDI_Catalogue_2.1.pdf.

IMDI, ISLE Metadata Initiative, 2003. Metadata elements for session descriptions, draft proposal version 3.0.3. http://www.mpi.nl/world/ISLE/documents/Proposals/ISLE_MetaData_3.0.3.pdf.

KIF, 1998. Knowledge interchange format. http://logic.stanford.edu/kif/.

Kipp, Michael, 2000-2003. Anvil — annotation of video and spoken language. Software, download at http://www.dfki.de/ kipp/anvil/, checked January 2004.

Leech, Geoffrey Neil, 1993. Corpus annotation schemes. *Literary and Linguistic Computing*, 8(4):275–281.

McKelvie, David and Henry S. Thompson, 1997. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML Europe'97*. Barcelona.

Milde, Jan-Torsten and Ulrike Gut, 2002. The tasx-environment: an xml-based toolset for time aligned speech corpora. In *Proceedings of LREC 2002*. Las Palmas.

MPEG-7, 2003. Mpeg-7 overview. Technical report, ISO/IEC.

Salembier, Philippe and John R. Smith, 2001. Mpeg-7 multimedia description schemes. *IEEE Transactions on Circuits and Systmes for Video Technology*, 11(6):748–759.

Schmidt, Thomas, – 2004. EXMARaLDA. Software, download at http://www.rrz.uni-hamburg.de/exmaralda/, checked January 2004.

Simons, Gary and Steven Bird, 2002. OLAC metadata. http://www.language-archives.org/OLAC/metadata.html.

Sjölander, Kåre and Jonas Beskow, 2000. Wavesurfer - an open source speech tool. In *Proceedings of ICSLP 2000*. Beijing, China.

Sjölander, Kåre and Jonas Beskow, 2004. Wavesufer. Software.

Sperberg-McQueen, C. M. and Lou Burnard (eds.), 1994. *TEI P3 Guidelines for electronic text encoding and interchange*.

Trippel, Thorsten and Tanja Baumann, 2003. Metadaten für Multimodale Korpora. Technical Document 3, Bielefeld University, Bielefeld. ModeLex Technical Document, Research Group Text Technological Modelling of Information.

Trippel, Thorsten and Dafydd Gibbon, 2002. Annotation driven concordancing: the PAX toolkit. In *Proceedings of LREC 2002*. Las Palmas.

Trippel, Thorsten, Felix Sasaki, and Dafydd Gibbon, 2004. Consistent storage of metadata in inference lexica: the metalex approach. In *Proceedings of LREC 2004*. Lisbon: ELRA.

Trippel, Thorsten, Felix Sasaki, Benjamin Hell, and Dafydd Gibbon, 2003. Acquiring lexical information from multilevel temporal annotations. In *Proceedings of Eurospeech 2003*. Geneva.

Witt, Andreas, 2002. Meaning and interpretation of concurrent markup. In *Proceedings of the ALLC / ACH 2002 - Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities*. Tübingen. Http://www.uni-tuebingen.de/cgi-bin/abs/abs?propid=40.

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 of 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). The work done by EAGLES was then continued in the scope of the ISLE project, and, in this context, specifically by the ISLE Computational Lexicon Working Group (CLWG). This group is *committed to the consensual definition of a standardized infrastructure to develop multilingual resources for HLT applications [...]*. Currently, the ISLE CLWG is focused on aspects of computational lexical semantics and multilingual lexicons (Atkins et al., 2002).

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, LUSOlex, and EPLexIC present two different categorizations (adverb and indefinite pronoun) for that word, while SMorph has only one (in SMorph, *algo* is described only as indefinite pronoun); in what concerns 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.

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
algo/Pi=nn/"al˜gu/algo
```

Figure 1: Lexicon comparison of *algo* descriptions. Phonetic description according to SAMPA (SAMPA, n.d.).

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. Thus, the proposed model allows evolution of both data and data structure.

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 solution.

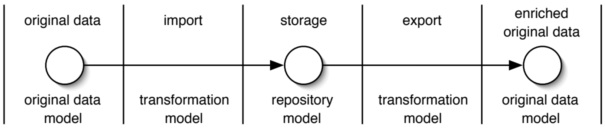


Figure 2: Data circulation.

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 they can be represented in XMI (i.e., XML), UML diagrams allow for 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.[1]

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.

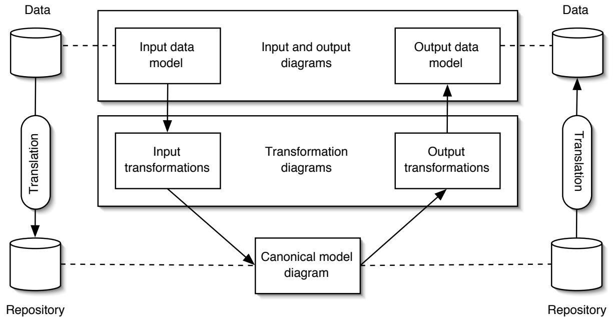


Figure 3: Models.

In figure 3, we show the relationships 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 corresponding 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,

[1] As an example of the usefulness of metadata in the database, our prototype uses this meta-information for ensuring the integrity of data relationships and for synthesizing information concerning some aspects of data enumerations (de Matos and Mamede, 2003).

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 semantic 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 semantic objects, or those existing between syntactic and semantic 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: that is, 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 predate 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 external models. The export capability is of great importance, since the repository must guarantee 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.

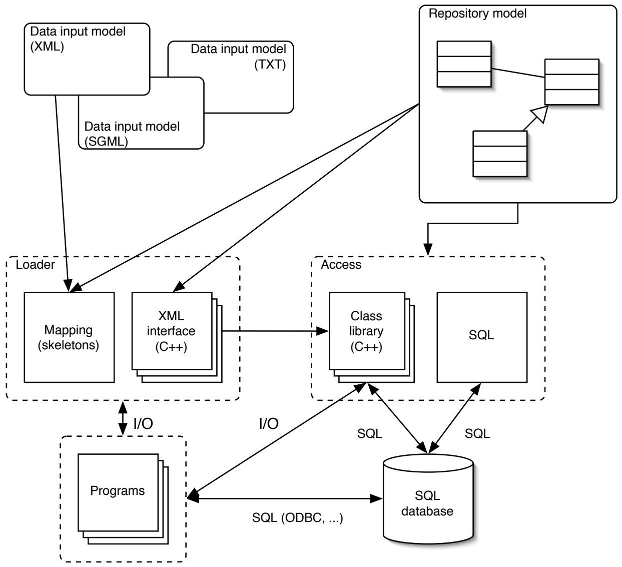


Figure 4: Models and code generation as implemented by the current prototype (de Matos and Mamede, 2003).

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 providing coverage for other models.

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 the 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 they are 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 so far limited to selected cases. Namely, we defined input transformation models for the Portuguese language 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 model prior to loading. More study is needed for comparing these two approaches.

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

Figure 5 presents the output obtained when extracting the description of the word *algo* using the PAROLE output model. It is possible to observe how the description of the entry *algo* has been enriched by the information imported from EPLexIC: a phonetic morphological unit (pmu) has been added to each morphological unit and the corresponding phonetic infection paradigms are also part of the output.

During the import process of EPLexIC, each entry of this lexicon produced a new morphological unit in the repository, unless the information of that entry could be appended (as a phonetic description element) to an existing one. Any diverging data was subjet to an individual analysis.

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 satisfied.

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 implement-

```
<mus id="r592" naming="algo"
     gramcat="adverb" autonomy="yes"
     synulist="usyn23987 usyn23988">
  <gmu range="0" reference="yes"
       inp="mfgr1">
    <spelling>algo</spelling></gmu>
  <pmu range="0" reference="yes"
       inp="pt_PT.FlAdv2p">
    <spelling>"al˜gu</spelling></pmu>
</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>
  <pmu range="0" reference="yes"
       inp="mfpempty">
    <spelling>"al˜gu</spelling></pmu>
</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>
<pinp id="pt_PT.FlAdv2p" example='"oZ@'>
  <combmfcif combmf="combtm0">
    <cif stemind="0">
      <removal/>
      <addedbefore/><addedafter/>
    </cif>
  </combmfcif>
</pinp>
<pinp id="mfpempty">
  <combmfcif combmf="combtmempty">
    <cif stemind="0">
      <removal/>
      <addedbefore/><addedafter/>
    </cif>
  </combmfcif>
</pinp>
<combmf id="combtmempty"/>
<combmf id="combtm0" degree="positive"/>
```

Figure 5: Repository output description for concept *algo* using the PAROLE data model.

ing 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. 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, in turn, 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.

As an example of the previous, when importing the phonetic information of the word forms of EPLexIC to the phonetic paradigms structure of the canonical model, all related word forms of the imported one were enriched with corresponding phonetic information.

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.

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

Atkins, Sue, Nuria Bel, Francesca Bertagna, Pierrette Bouillon, Nicoleta Calzolari, Christiane Fellbaum, Ralph Grishman, Alessandro Lenci, Catherine MacLeod, Martha Palmer, Gregor Thurmair, Marta Villegas, and Antonio Zampolli, 2002. From Resources to Applications. Designing the Multilingual ISLE Lexical Entry.

In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*. Las Palmas de Gran Canária, Spain.

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. Technical report, IMS, University of Stuttgart, Stuttgart.

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/`, `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.

SAMPA, n.d. `http://www.phon.ucl.ac.uk/home/sampa/home.htm`.

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: Oxford 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).

Towards an international standard on feature structure representation (2)

Kiyong Lee[1], Lou Burnard[2], Laurent Romary[3], Eric de la Clergerie[4]
Ulrich Schaefer[5], Thierry Declerck[6], Syd Bauman[7]
Harry Bunt[8], Lionel Clément[4]
Tomaz Erjavec[9], Azim Roussanaly[3], Claude Roux[10]

[1]Korea University Linguistics, Seoul, Korea
klee@korea.ac.kr
[2]Oxford University Computing Services, UK
lou.burnard@oucs.ox.ac.uk
[3]LORIA, France
{Laurent.Romary—Azim.Roussanaly}@loria.fr
[4]INRIA, France
{lionel.clement—Eric.De-La-Clergerie}@inria.fr
[5]DFKI, Germany
ulrich.schaefer@dfki.de
[6]Saarland University & DFKI, Germany
declerck@dfki.de
[7]Brown University, USA
syd_bauman@brown.edu
[8]Tilburg Univeristy, The Netherlands
Harry.Bunt@uvt.nl
[9]Jozef Stefan Institute, Slovenia
tomaz.erjavec@ijs.si
[10]XEROX Research Center Europe, France
claude.roux@xrce.xerox.com

Abstract

This paper describes the preliminary results of a joint initiative of the TEI (Text Encoding Initiative) Consortium and the ISO Committee TC 37SC 4 (Language Resource management) to provide a standard for the representation and interchange of feature structures. The paper published in the proceedings of this workshop is in fact an extension of a paper published in the LREC 2004 proceedings, and about 50% are identical with it.

1. Introduction

This paper describes some preliminary results from a joint initiative of the TEI (Text Encoding Initiative) Consortium and the ISO Committee TC 37/SC 4 (Language Resource management), the goal of which is to define a standard for the representation and interchange of feature structures. The joint working group was established in December 2002, and its proposals are now progressing to Draft International Standard status.

1.1. TEI

Initially launched in 1987, the Text Encoding Initiative (TEI) is an international and interdisciplinary effort the goal of which is to help libraries, publishers, and individual scholars represent all kinds of literary and linguistic texts for online research and teaching, using an encoding scheme that is maximally expressive and minimally obsolescent. The TEI has also played a major role in the development of European language engineering standards since the days of EAGLES. Its recommendations, the “TEI Guidelines”, underpin such key standards as the Corpus Encoding Standard, and address many other areas of language resource documentation and description, as well as lexicographic and terminological databases. Since 2000, maintenance and development of the TEI has been managed by an international membership Consortium, which announced publication of a complete XML version of the TEI Guidelines, known as P4 in 2002, and is now overseeing production of a major new revision, known as P5.[1]

1.2. TC 37/SC 4

The research areas of ISO/TC 37/SC 4 include computational linguistics, computerized lexicography, and language engineering. Language resources consist of contents represented by linguistic data in various formats (e.g., speech data, written text corpora, general language lexical corpora). Text corpora, lexica, ontologies and terminologies are typical instances of language resources to be used for language and knowledge engineering. In both monolingual and multilingual environments, language resources play a crucial role in preparing, processing and managing the information and knowledge needed by computers as well as humans. With a view to mobile computing and mobile content etc., the availability of language resources, having to be considered as multilingual, multimedia and

[1] See also http://www.tei-c.org/

multimodal from the outset will be one of the key success factors.[2]

1.3. Current topics of the joint group

The joint TEI and ISO activity has focussed on the following topics:

- articulation of a detailed technical proposal for an XML format able to represent a feature structure analysis with a precise description of the underlying formal mechanism to ensure the coherence and soundness of the standard in line with major theoretical works in this domain;
- provision of specific mechanisms to deal with reentrant structures, clearly distinguished from a generic pointing mechanism;
- provision of a coherent description of the notion of type, which will enable further development of the standard to include a complementary set of proposal relating to declaration of a Feature System.
- integration of this proposal into the on-going revision of the TEI Guidelines (TEI P5) due for publication in 2004;

2. Goal of the paper

The paper first introduces the basic concepts of the features structure formalism. Section 4 briefly describes the proposal currently being developed as an ISO Standard and its relation to other ongoing work relating to the deployment within ISO TC 37 of a general data category registry for linguistic description. The current proposals will include this and other external sources for use, as a reference, in the declaration of particular feature sets. Finally some conclusions are drawn.

3. Feature structures

Feature structures (FSs) form an essential part of many language processing systems, whether their focus is on the description, enrichment, storage, or management of linguistic data. The FS formalism itself has a formal background in graph theory, and supports powerful unification mechanisms for combining elementary structures, which have facilitated its use in many real-world applications. There are many possible ways of representing FSs, but the basic notions have an intrinsic legibility which make them very useful for representing linguistic information in interchange situations, both between people and between processing systems. To take full advantage of this capability, a standard way of representing such structures in electronic format should be made available so that a) specialists from diverse application fields can share detailed expertise from diverse domains and b) implementers can share basic libraries dedicated to the manipulation of FSs, thus reducing the overall cost of application development.

FSs are uniform objects that can be used to represent a wide range of objects, ranging from very simple structures consisting of simple lists of feature-value pairs, to highly complex typed and nested structures with reentrancy, as found for instance in HPSG (Pollard and Sag, 1994b), LFG (Bresnan, 1982), etc. More recently, FSs have also been used as the internal representation for shallow and robust NLP systems based on finite state technologies, or for merging information sources coming from distinct modalities in multi-modal systems.

4. The proposal

This proposal combines a basic set of tags for representing features and feature structures covering in a uniform way the full range of complexity attested by current implementations, together with additional mechanisms to describe libraries of values, feature value pairs and feature structures. As an example, consider the following simple morpho-syntactic annotation for the word ‘vertes’ in French:

```
<fs>
   <f name=’token’>
     <string>vertes</string>
   </f>
   <f name=’lemma’>
     <string>vert</string>
   </f>
   <f name=’pos’>
     <symbol value=’adj’/>
   </f>
   <f name=’gender’>
     <symbol value=’fem’/>
   </f>
   <f name=’number’>
     <symbol value=’plural’/>
   </f>
</fs>
```

In this XML representation, the element `<fs>` is used to encode a feature structure, and the `<f>` element is used for each of five feature-value pairs making up this structure. Each feature-value pair has a name, given by the name attribute, and contains a primitive or atomic value, marked (in this case) by either a `<string>` or a `<symbol>` element, depending on its datatype. Other possible child elements for the <f> element include <binary> for binary- or boolean-values such as PLUS or MINUS, and <numeric> for various kinds of numeric values and ranges. Complex values can also be represented: collections or multivalues such as lists, sets or multisets (bags) are tagged using a <coll> element; feature structures may also be used as feature-values, thus providing a recursive ability. The components of particular feature structures may be represented directly or referred to by using pointers to previously stored “libraries” of features or feature values. We believe that this XML representation has equivalent expressive power to the classical AVM (Attribute-Value-Matrix) notation, but is more readily processed.

In developing the XML representation, the work group was able to simplify considerably the original TEI proposals as described in (Langendoen and Simons, 1995b), by fo-

[2] See also http://www.tc37/sc4.org/

cussing on applications of the formalism in linguistic analysis alone. The availability of new XML-based tools, in particular the relax-NG schema language now used to express the TEI markup scheme, also proved beneficial for developing a powerful and expressive formalism, adequate to the needs of those using feature structure analysis.

Applications for this formalism have demonstrated the need for more complex mechanisms, which are needed to handle elaborated linguistic information structures. Following on from reference works by Shieber (PATR-II) (Shieber, 1986) or (Carpenter, 1992), there has been a whole range of implementations of FSs in computational linguistics applications. Examples include LOGIN/LIFE (Ait-Kaci and Nasr, 1986), ALE (Carpenter and Penn, 1996), Profit (Erbach, 1995), DyALog (de la Clergerie, 2002), ALEP (Simpkins and Groenendijk, 1994), WAM-like Abstract Machine for TFS (Wintner and Francez, 1995), etc. From another point of view, one can consider the variety of linguistic levels concerned with such representations, e.g. phonology, morpho-syntax, grammars (unification grammars: LFG, HPSG, XTAG), linguistic knowledge base or practical grammar implementation guide (LKB, (Copestake, 2002)), underspecified semantics (MRS, (Copestake et al., 1999)), or integration of NLP components (Schaefer, 2003).

In our work, we have identified and discussed a certain numbers of concepts and topics introduced in the works cited above and we are proposing an XML-based way of representing the corresponding feature structures. As examples, given for this short paper, we show the actual XML implementation of *structure-sharing* (also called *reentrancy*) and the XML treatment of *types*, two topics mentioned in 1.3.:

4.1. Structure Sharing

As shown in most of the works cited above, *structure sharing* (or *reentrancy*) requires the use of labelling for representation in graphic notation such as AVM. For example, to show that a given feature-value pair (or feature structure) occurs at multiple points in an analysis, it is customary to label the first such occurrence, and then to represent subsequent ones by means of the label.

In discussing how to represent this in an XML-based notation, we first proposed making use of a global attribute `label` or `n`, as in the following simple example:

```
<fs>
 <f name="specifier">
  <fs>
    <f name="agr" n="@1">
    <fs>
       <f name="number">
          <symbol value="singular"/>
       </f>
    </fs>
    </f>
    <f name="pos">
         <sym value="determiner"/>
    </f>
  </fs>
 </f>
 <f name="head">
   <fs>
    <f name="agr" n="@1"/>
    <f name="pos">
       <sym value="noun"/>
    </f>
   </fs>
 </f>
</fs>
```

The feature named “agr” is here labelled “@1”. Its first occurrence contains a feature-value pair (“singular number”); its second references this same feature-value pair.

An alternative way of representing this phenomenon is to use the XML ID/IDREf mechanism, as follows:

```
<fs>
 <f name="specifier">
  <fs>
    <f name="agr" id="N1">
    <fs>
       <f name="number">
          <symbol value="singular"/>
       </f>
    </fs>
    </f>
    <f name="pos">
         <sym value="determiner"/>
    </f>
  </fs>
 </f>
 <f name="head">
   <fs>
    <f name="agr" fVal="N1"/>
    <f name="pos">
       <sym value="noun"/>
    </f>
   </fs>
 </f>
</fs>
```

The working group has identified a need to distinguish the case where co-reference implies copying (or transclusion) of shared structures or values, from the case where co-reference simply implies multiple references to the same object, but has not yet reached a resolution as to which of the possible approaches best meets this need.

4.2. Typed Feature Structure

The *typed feature structure* has become a key tool in the linguistic description and implementation of many recent grammar formalisms,

4.2.1. Types

Elements of any domain can be sorted into classes called *types* in a structured way, based on commonalities of their properties. Such linguistic concepts as *phrase*, *word*, *pos* (parts of speech), *noun*, and *verb* may be represented as features in non-typed feature structures. But in typed feature structure particular feature-value pairs may be treated as types.

By *typing*, each feature structure is assigned a particular type. A feature specification with a particular value is then constrained by this typing. A feature structure of the type *noun*, for instance, would not allow a feature like TENSE in it or a specification of its feature CASE with a value of the type *feminine*.[3]

4.2.2. Definition

The extension of non-typed feature structure to typed feature structure is very simple in a set-theoretic framework. The main difference between them is the assignment of types to feature structures. A formal definition of typed feature structure can thus be given as follows:[4]:

Given a finite set of **Features** and a finite set of **Types**, a typed feature structure is a tuple $\mathcal{TFS} = <\mathbf{Nodes}, r, \theta, \delta>$ such that

i. **Nodes** is a finite set of nodes.

ii. *r* is a unique member of **Nodes** called *the root*.

iii. θ is a total function that maps **Nodes** to **Types**.

iv. δ is a partial function from **Features**×**Nodes** into **Nodes**.

First, each of the **Nodes** must be rooted at or connected back to the root *r*. Secondly, there must one and only one root for each feature structure. Thirdly, each of the **Nodes**, including the root *r* node and terminal nodes, must be assigned a type by the typing function θ. Finally, each of the **Features** labelling each of **Nodes** is assigned a unique value by the feature value function δ.[5]

This type type of information can be encoded in an XML notation, as an example (simplified, due to the length of the paper) shows below:

```
<fs type="word">
  <f name="orth">
    <string>love</string>
    </f>
  <f name="syntax">
    <fs type="verb">
      <f name="valence">
        <symbol value="transitive"/>
      </f>
    </fs>
  </f>
</fs>
```

Note here that the line <f name="pos"><sym value="verb"/></f> in the embedded feature structure <fs> has been replaced by typing that <fs> as in <fs type="verb">.

The use of *type* may also increase the expressive power of a graph notation. On the typed graph notation, for instance, multi-values can be represented as terminating nodes branching out of the node labelled with the type *set*, *multiset* or *list*. This node in turn is a terminating node of the arc labelled with a multit-valued feature, say SLASH. Each arc branching out of the multi-valued node, say *set*, is then labelled with a feature appropriate to the type.

4.3. The Equivalence of the XML Representation and the AVM Annotation

The proposed XML representation having equivalent expressive power as the classical AVM notation for feature structures, from a semantic point of view the XML expressions can be interpreted as graphs in the classical way (Carpenter, 1992). In this approach, feature structures are viewed as a *graphs*, i.e., as a certain class of set-theoretical constructs. Carpenter defines a typed feature structure as, given a set Feat of features and a set Type of (hierarchically ordered) types, a quadruple

$$(1)\ < N, n_0, \theta, \mathcal{F} >$$

where N is a finite set whose elements are called *nodes*; where $n_0 \in N$, where θ is a total function from N to Type (typing) and where $\mathcal{F}$ is a partial function from $N \times \mathsf{Feat}$ to N (defining arcs, labelled with feature names, that connect the nodes). The node n_0 is the root of the graph; every node in N is required to be reachable from the root node. Pollard and Sag (1987) use this view when they introduce feature structures as semantic entities in the interpretation of representations of linguistic information. They refer to graphs as "modelling structures", i.e., as structures that play a role in models, and they introduce AVMs as structures in a "description language" that is to be interpreted in terms of feature structures-as-graphs: *"Throughout this volume we will describe feature structures using attribute-value (AVM) diagrams"*. (Pollard & Sag, 1987, 19–20).

This view corresponds to the following metamodel that distinguishes nonterminal and terminal nodes and types:

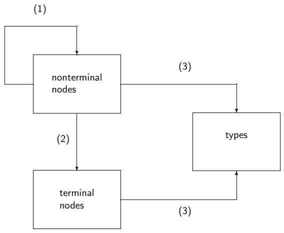


Diagram 1: Metamodel with graphs as model elements

Relations of type (1) in this metamodel correspond to features like HEAD-DAUGHTER in HPSG, those of type (2) to atomic-valued features like GENDER, and those of type (3) to the typing function θ.

An alternative view is that of graphs as *representations*, as a notational alternative to AVMs rather than as the objects interpreting AVMs. For example, Lee (2004) introduces feature structures as ways of capturing information, and mentions graphs as a *notation* for feature structures. Aware of these alternative possible views, Pollard & Sag (1987) note that *"A common source of confusion is that feature structures themselves can be used as descriptions of other feature structures."* One way to avoid confusion is to consider the metamodels corresponding to alternative views.

[3] Note that atomic feature values are considered *types*, too.

[4] Slightly modified from (Carpenter, 1992).

[5] The unique-value restriction on features does not exclude multi-values or alternative values because even in these cases each feature ultimately takes a single value which may be considered complex in structure.

In the graphs-as-representations view, the graph (2) and the AVM (3) are seen as equivalent representations that can both be interpreted as representing the complex predicate (4).

(2)

```
               AGR                NUM
noun     ———————>   agr ———————-> sing
                    |
                    +———————————> fem
                          GENDER
```

(3) $$\begin{bmatrix} \textit{noun} & \\ \text{AGR} & \begin{bmatrix}\text{NUM sing}\\ \text{GENDER fem}\end{bmatrix}\end{bmatrix}$$

(4) $\lambda x : noun(x) \wedge num(x) = sing \wedge gender(x) = fem$

(simplifying slightly). This interpretation reflects a similar view on information as that of first-order logic, with two kinds of individuals: the kind of things that x stands for (words and phrases) and the kind of atomic attribute values like ‘fem’ and ‘sing’. These values are associated with word-like individuals through two-place predicates that are in fact functions; moreover, types such as ‘noun’ correspond to unary predicates. This corresponds to the metamodel visualized in Diagram 2.

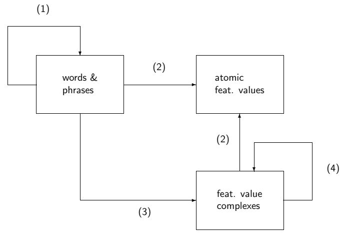


Diagram 2: First-order metamodel for feature structures

Relations of type (1) in this diagram (1) correspond again to features like HEAD-DAUGHTER; (2) to atomic-valued features like GENDER; (3) to features like SYNSEM, and (4) to features like AGR(EEMENT).

5. The role of feature structure markup transformation for the integration of NLP components

One of the main motivations for XML feature structure markup is the interchange of linguistic data. This can be done *offline*, e.g., for the exchange of lexica, grammatical resources, or annotated documents.

A further application is *online* integration of NLP components, where several, specialised modules contribute to improved (e.g., disambiguated or more precise) linguistic analyses. Examples for such hybrid architectures are Whiteboard (Frank et al., 2003; Schäfer, 2003) and DeepThought (Callmeier et al., 2004).

In both cases, online or offline integration, different representations of linguistic data can be involved, where feature structures can either form the source or the target representation or even both.

In general, conversion or translation of different XML representations is required. In the case of XML, such a translation is called transformation, and the established W3C standard language for XML transformation is XSLT (eXtensible Styleheet Transformation; (Clark, 1999)).

The input of an XSL transformation is always XML, while the output can be of any syntax, including XML as a well-supported target format.

To illustrate the use of XML transformation for of feature structure markup, we give concrete examples.

5.1. Feature structures as target representation

Construction of (typed) feature structures from other XML representations that are e.g. produced by a shallow NLP system. Specific elements with attributes are translated to possibly nested feature-value pairs, e.g. for input to a HPSG(Pollard and Sag, 1994a) parser etc. In the following example, `<infl num="singular"/>` is translated to the corresponding feature structure. Of course, also symbolic names e.g. sg to singular etc. can be translated.

```
<xsl:template match="infl">
   <fs type="infl">
       <f name="number">
          <symbol value="@num"/>
       </f>
   </fs>
</xsl:template>
```

Grammar exchange format or meta syntax like in SProUT (Drozdzynski et al., 2004), where a TDL-like grammar syntax (Krieger and Schäfer, 1994) is translated to an internal representation based on feature structure XML. The internal representation is used as input for type checking and compilation.

Data exchange between NLP components, e.g. the so-called SProUTput DTD that is used for exchange of typed feature structures with external NLP components (input and output) in SProUT[6].

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- SProUTput DTD (2003) -->
<!ELEMENT SPROUTPUT ( DISJ )* >
<!ELEMENT DISJ ( MATCHINFO )+ >
<!ATTLIST DISJ id ID >
<!ELEMENT MATCHINFO ( FS ) >
<!ATTLIST MATCHINFO id ID      #IMPLIED
                  rule NMTOKEN #IMPLIED
                cstart NMTOKEN #IMPLIED
                  cend NMTOKEN #IMPLIED
                 start NMTOKEN #IMPLIED
                   end NMTOKEN #IMPLIED >
<!ELEMENT FS ( F )* >
<!ATTLIST FS      type NMTOKEN #REQUIRED
                 coref NMTOKEN #IMPLIED >
<!ELEMENT F  ( FS ) >
<!ATTLIST F       name NMTOKEN #REQUIRED >
```

5.2. Feature structures as source representation

Extraction or projection of information encoded in feature structures such as morphology to other formats or as API-like accessors. e.g. an XPath expression like

[6] Element names are uppercase in the SProUTput DTD

```
  <xsl:template match="fs[@type=’infl’]">
    <infl num="f[@name=’number’]/symbol
                                /@value"/>
  </xsl:template>
```

is the inverse of example 5.1. above.

AVM visualisation tools or editors like the feature structure renderer in SProUT or Thistle (Calder, 2000) both take (different) descriptions of typed feature structures and render a graphical representation of the feature structure.

Extraction of tree structures etc. from a a complex HPSG feature structure, e.g. for further linguistic processing or visualisation in Thistle.

Generation of semantics representation. An example is a transformation of typed feature structures to RMRS XML markup (Copestake, 2003) which e.g. forms the basic representation for the exchange of deep and shallow NLP processing results in the DeepThought architecture (Callmeier et al., 2004), cf. Fig. 1.

5.3. Feature structures as both source and target representation

Translation between different feature structure syntaxes or systems. We give an example of list values that can be encoded differently in typed feature structure markup. The XSLT template below takes a list encoded as nested `FIRST-REST` list typed `*cons*` and translates it to the proposed `<list>` with embedded elements from the `FIRST` attribute values in the input. The template works recursively on `FIRST-REST` lists of any length.

```
<!-- =====================================
Initial template. Enclose list elements from
FIRST-REST list in <list> element
======================================= -->
<xsl:template match=’fs[@type="*cons*"]’>
 <xsl:element name="list">
  <xsl:call-template name="listlist">
   <xsl:with-param name="node" select="."/>
  </xsl:call-template>
 </xsl:element>
</xsl:template>
<!-- =====================================
recursive template: list all list elements
======================================= -->
<xsl:template name="listlist">
 <xsl:param name="node"/>
 <xsl:copy-of select=’$node/f[@name=
                              "FIRST"]/fs’/>
 <xsl:if test=’$node/f[@name="REST"]/fs/
                            @type="*cons*"’>
  <xsl:call-template name="listlist">
   <xsl:with-param name="node"
        select=’$node/f[@name="REST"]/fs’/>
  </xsl:call-template>
 </xsl:if>
</xsl:template>
```

5.4. Reentrancies and Transformation

A general issue that arises for the case where feature structures are source representations is reentrancies. Here, ‘dereferencing‘ is necessary on the basis of lookup in the XML source in order to have access to every node in the DAG (e.g. for feature path access); XML ID/IDREF declarations support faster access as discussed already before. If cyclic reentrancies are disallowed, copying of shared values when generating the features structure representation is an easy and probably quicker way in order to get the full access to shared values. Identity information is preserved through the reentrancy attribute anyway.

6. Related work within the ISO framework

A distinctive feature of the TEI Guidelines is its use of an integrated model of documentation and documentation outputs. The ODD system used to produce its recommendations, both as printed documentation and as formal syntax expressed in XML Schema or DTD languages, has recently been revised and re-expressed. This new modular system for documentation is likely to have wide take up in many different domains. In applying it to the expression of the feature structure analysis language, we have identified a number of potential areas of synergy with the ongoing ISO work on data category registry[7].

7. Conclusions

The work reported has proved to be an excellent opportunity for experimenting with the new descriptive framework being developed for the TEI Guidelines themselves. The feature structure activity has been a useful opportunity to experiment with the creation of relevant tagging systems and tools in a relatively limited but formally complex domain.

In general, the activity reported in the paper shows that there is great scope for further convergence between the TEI consortium and ISO committee TC 37/SC 4, and many benefits to be gained from joint work on issues which require complementary expertise in textual representation methods and in the representation of linguistic concepts.

[7] See for more details: http://jtc1sc36.org/doc/36N0581.pdf

```
<MATCHINFO rule="en_city" cstart="3" cend="7">
  <FS type="sprout_rule">
    <F name="OUT">
      <FS type="ne-location">
        <F name="LOCNAME">
          <FS type="&quot;Paris&quot;"/>
        </F>
        <F name="LOCTYPE">
          <FS type="city"/>
        </F>
      </FS>
    </F>
  </FS>
</MATCHINFO>
```

```
     <rmrs cfrom="3" cto="7">
       <label vid="1"/>
       <ep cfrom="3" cto="7">
         <gpred>ne-location</gpred>
         <label vid="2"/>
         <var sort="x" vid="2"/>
-->    </ep>
       <rarg>
         <label vid="2"/>
         <rargname>CARG</rargname>
         <constant>"Paris"</constant>
       </rarg>
     </rmrs>
```

Figure 1: Transformation of feature structure XML markup (SProUT) to RMRS (DeepThought).

8. References

Ait-Kaci, H and R Nasr, 1986. Login: A logic programming language with built-in inheritance. *J. Log. Program.*, 3(3):185–215.

Bering, Christian, Witold Drozdzyski, Gregor Erbach, Clara Guasch, Petr Homola, Sabine Lehmann, Hong Li, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, Atsuko Shimada, Melanie Siegel, Feiyu Xu, and Dorothee Ziegler-Eisele, 2003. Corpora and evaluation tools for multilingual named entity grammar development. In *Proceedings of Multilingual Corpora Workshop at Corpus Linguistics 2003*. Lancaster.

Bresnan, Joan (ed.), 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

Busemann, Stephan, Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, Hans Uszkoreit, and Feiyu Xu, 2003. Integrating information extraction and automatic hyperlinking. In *Proceedings of ACL-2003, Interactive Posters/Demonstrations*. Sapporo, Japan.

Calder, Joe, 2000. *Thistle: Diagram Display Engines and Editors*. HCRC, U. of Edinburgh.

Callmeier, Ulrich, 2000. PET — A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6 (1):99 – 108.

Callmeier, Ulrich, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel, 2004. The DeepThought core architecture framework. In *Proceedings of LREC-2004*. Lissabon, Portugal.

Carpenter, Bob, 1992. *The Logic of Typed Feature Structures*. Cambridge University Press.

Carpenter, Bob and Gerald Penn, 1996. Efficient parsing of compiled typed attribute value logic grammars. In H. Bunt and M. Tomita (eds.), *Recent Advances in Parsing Technology*. Kluwer, page Recent Advances in Parsing Technology.

Clark, James, 1999. *XSL Transformations (XSLT)*. W3C, http://w3c.org/TR/xslt.

Clark, James and Steve DeRose, 1999. *XML Path Language (XPath)*. W3C, http://w3c.org/TR/xpath.

Copestake, Ann, 2002. *Implementing Typed Feature Structure Grammars*. Stanford, CA: CSLI publications.

Copestake, Ann, 2003. Report on the design of RMRS. Technical Report D1.1b, University of Cambridge, Cambridge, UK.

Copestake, Ann, Dan Flickinger, Ivan Sag, and Carl Pollard, 1999. Minimal recursion semantics: An introduction. Draft.

Crysmann, Berthold, 2003. On the efficient implementation of german verb placement in HPSG. In *Proceedings of RANLP-2003*. Borovets, Bulgaria.

Crysmann, Berthold, Anette Frank, Bernd Kiefer, Stefan Müller, Jakub Piskorski, Ulrich Schäfer, Melanie Siegel, Hans Uszkoreit, Feiyu Xu, Markus Becker, and Hans-Ulrich Krieger, 2002. An Integrated Architecture for Deep and Shallow Processing. In *Proceedings of ACL 2002*. Philadelphia, PA.

de la Clergerie, Éric Villemonte, 2002. Construire des analyseurs avec dyalog. In *Proceedings of TALN '02*.

Drozdzynski, Witold, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu, 2004. Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, 1:17–23. Http://www.kuenstliche-intelligenz.de/archiv/2004_1/sprout-web.pdf.

Erbach, Gregor, 1995. Profit: Prolog with features, inheritance, and templates. In *Proceedings of EACL '95*.

Frank, Anette, Markus Becker, Berthold Crysmann, Bernd Kiefer, and Ulrich Schäfer, 2003. Integrated shallow and deep parsing: TopP meets HPSG. In *Proceedings of ACL-2003*. Sapporo, Japan.

Kasper, Walter, Jörg Steffen, Jakub Piskorski, and Paul Buitelaar, 2004. Integrated language technologies for multilingual information services in the MEMPHIS project. In *Proceedings of LREC-2004*. Lissabon, Portugal.

Krieger, Hans-Ulrich and Ulrich Schäfer, 1994. *TDL*— a type description language for constraint-based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*.

Langendoen, D. Terence and Gary F. Simons, 1995a. A rationale for the TEI recommendations for feature structure markup. In Nancy Ide and Jean Veronis (eds.), *Computers and the Humanities 29(3)*. The Text Encoding Initiative: Background and Context, Dordrecht: Kluwer Acad. Publ. Reprint.

Langendoen, Terence D. and Gary F. Simons, 1995b. A rationale for the tei recommendations for feature-structure markup. *Computers and the Humanities*, 29:191–209.

Lee, Kyiong, 2004. Language resource management – feature structures part1: Feature structure representation. Document iso/tc 37/sc 4 n 033, ISO.

Pollard, Carl and Ivan A. Sag, 1994a. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. Chicago: University of Chicago Press.

Pollard, Carl J. and Ivan A. Sag, 1987. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Pollard, Carl J. and Ivan A. Sag, 1994b. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Sailer, Manfred and Frank Richter, 2001. Eine XML-Kodierung für AVM-Beschreibungen. In Henning Lobin (ed.), *Proceedings der GLDV-Frühjahrstagung 2001*. Gesellschaft für linguistische Datenverarbeitung.

Schaefer, Ulrich, 2003. What: An xslt-based infrastructure for the integration of natural language processing components. In *Proceedings of HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems*.

Schäfer, Ulrich, 2003. WHAT: An XSLT-based Infrastructure for the Integration of Natural Language Processing Components. In *Proc. of the Workshop on the Software Engineering and Architecture of LT Systems (SEALTS), HLT-NAACL03*. Edmonton, Canada.

Shieber, Stuart M., 1986. *An Introduction to Unification-Based Approaches to Grammar*, volume 4 of *CSLI Lecture Notes Series*. Stanford, CA: Center for the Study of Language and Information.

Simpkins, N. and M. Groenendijk, 1994. The alep project. technical report. Technical report, Cray Systems / CEC.

Thompson, Henry S. and David McKelvie, 1997. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML-EU-1997*.

Uszkoreit, Hans, 2002. New Chances for Deep Linguistic Processing. In *Proceedings of COLING 2002*. Taipei, Taiwan.

Wintner, Shuly and Nissim Francez, 1995. Parsing with typed feature structures. In *Proceedings of the Fourth International Workshop on Parsing Technologies*.

An Ontology for NLP Services

Ewan Klein, Stephen Potter

School of Informatics
University of Edinburgh, Edinburgh, Scotland
{ewan,stephenp}@inf.ed.ac.uk

Abstract

1. Introduction

The main focus of this paper is a framework for describing and discovering NLP processing resources. In many ways, the most difficult aspect of this task is the huge space of options. Even disregarding the wide variety of theoretical models for describing natural languages, and even if we restrict attention exclusively to NLP tools, there is sufficient diversity within the NLP community to provoke much disagreement about the best way to describe such tools. In this paper, we try to narrow down the range of choices by focusing on the following issues. First, we emphasize the role of description in supporting **tool interoperability**. Second, we place interoperability within the context of **service composition**. Third, we develop an ontology of NLP services that is informed by the OWL-S semantic framework (OWL-S).

2. Service Use Cases

2.1. Describing NLP Resources

Before embarking on proposals for how language resources should be described, it is important to consider what requirements need be met. Consequently, we describe two use cases which will guide our design objectives. We are currently interested in addressing the needs of two, rather different, communities of users: **NLP researchers**, on the one hand, and **users of domain-specific text mining services**, on the other.

2.2. The NLP Workbench

NLP researchers frequently wish to construct application-specific systems that combine a variety of tools, some of them in-house and some of them third-party. For example, to carry out a named entity recognition task, Janet might want to run a statistical classifier over a corpus that has been tokenized, tagged with parts of speech (POS) and chunked. She might use her own tokenizer and chunker but use someone else's tagger — say the TnT tagger (Brants, 2000). She will almost certainly have to write some glue code in a scripting language to plug these different tools together.

Suppose now that John needs to work with Janet's system a few months later. He wants to try the same experiment, but using a different tagger – say the CandC tagger (Curran and Clark, 2003). A number of issues arise. First, how likely is it that John can simply re-run Janet's system as it was? Can he be sure that he's using the same versions of the software, trained on the same data? Can he be sure that he's getting the same results against the same Gold Standard test data? For such a scenario, it would be useful to have some method for recording and archiving particular configurations of tools, together with a record of the results. Moreover, the configuration needs to include information about the location of the relevant versions of the tools. Rerunning the experiment should ideally be as simple as firing up a configuration manager and reloading the configuration script from a pull-down menu.

Assuming that all went smoothly, how likely is it that John can simply remove the call to TnT in Janet's script and splice in CandC instead? Do they have the same formatting requirements on their input and output? Do the available pre-trained versions of these taggers use the same tagsets? Do they require the same number of command-line arguments? Unfortunately, the answer to these three questions is No.[1] This problem is hard for the human to deal with; consider how much harder it would be to develop a workbench that would automatically check whether two tools could be serially composed. A crucial obstacle to automatic composition is that we lack a general framework for describing NLP tools in terms of their inputs and outputs.

2.3. Text Mining for e-Science

There is increasing interest in deploying text mining tools to assist scientists in various tasks. One example is knowledge discovery in the biomedical domain: a molecular biologist might have a list of 100 genes which were detected in a micro-array experiment and wishes to trawl through the existing published research for papers which throw light on the function of these genes. Another example is the astronomer who detects an X-ray signal in some region of the sky, and wants to investigate the online literature to see if any interesting infra-red signals were detected in the same region. These brief scenarios are special cases of a more general interest in using computing technologies to support scientific research, so-called "e-Science" (Hey and

We are grateful to Steven Bird, Denise Ecklund, Harry Halpin and Jochen Leidner for helpful discussion and comments.

[1] For example, in the case of formatting requirements, TnT uses multiple tabs as a separator between word and tag, while by default, CandC uses the underscore as a separator; it can, however, be configured to use a single tab as separator.

Trefethen, 2002).

In such cases, we expect that the researcher will be using a *workflow tool*; that is, something that "orchestrates e-Science services so that they co-operate to implement the desired behaviour of the system".[2] In this context, we would want the researcher to have access to a variety of text mining services that will carry out particular tasks within a larger application. A service might be essentially a document classification tool which retrieves documents in response to a string of keywords. However, it might be a more elaborate information extraction system which, say, populates a database that is then queried by some further service.

Each text mining tool which is accessible to the scientist end-user must be able to describe what kind of service it offers, so that it can be discovered by the workflow tool. We would expect there to be a more coarse-grained functionality in this use case: the scientist is unlikely to care which tagger is being used. Nevertheless, it will not always be easy to predict in advance where the external boundary of a text mining service will lie, so in principle the challenge of developing explicit and well-understood interfaces for text mining services overlaps with the previous use case. An additional constraint is that the NLP tools must be interoperable with other services provided by the e-Science workflow environment, and must be accompanied by descriptions which are intelligible to non-NLP practitioners.

3. Design Influences and Goals

In order to tease out requirements, let's reflect further on our first use case. Assume that we are given a simple pipeline of processors which carries out some well-defined text processing task. We wish to remove one processor, say a POS tagger, and splice in a new one, while ensuring that we preserve the overall functionality of the pipeline. This means that we need to abstract away from particular taggers to a class of such tools, all of which carry out the same transformation on their input. At this level, we can talk broadly about **interchangeability** of functionally equivalent processors. On the other side of the coin, information about the input and output parameters of two taggers *A* and *B* must be detailed enough for us to tell whether, when *A* is replaced by *B*, *B* will accepts input from the immediately preceding processor and produce output that is acceptable for the immediately following processor. In other words, we require a processor to be accompanied by metadata which enables us to make decisions about **interoperability**.

de Roure and Hendler (2004) argue that interoperability is a key notion for the e-Science research programme, and that technologies from both the Grid (Foster et al., 2001) and the Semantic Web will underpin the programme. The integration of the two technologies has been dubbed the Semantic Grid and both approaches interoperability as being achieved through deployment of *services*. Foster et al. (2002) give a general characterization of services as "network enabled entities that provide some capability through the exchange of messages", and argue that a service-oriented perspective supports virtualization in which resources can be accessed uniformly despite being being implemented in diverse ways on diverse platforms.

We would like, then, an environment that offers an infrastructure for the discovery, orchestration and invocation of services, and one that is flexible and permits a high degree of re-use and automation of workflows. This desire coincides with the aims of much of the effort in the Semantic Web Services community, and so it is to this community that we look for guidance.

The Semantic Web 'vision' is one of enhancing the representation of information on the web with the addition of well-defined and machine-processable semantics (Berners-Lee et al., 2001), thereby encouraging a greater degree of 'intelligent' automation in interactions with this complex and vast environment. One thread of this initiative concerns the provision of web-based services: the web has great potential as a medium for, on the one hand, web service-providers to advertise their services and conduct their business, and on the other, for those with particular service needs to publicise these needs to the environment so as to have them satisfied.

3.1. Description Logics and OWL-S

A number of *de facto* standards exist for locating and invoking web services; these include Unversal Description, Discovery and Integration protocol (UDDI; Bellwood et al., 2002), a protocol for building and using registries of services, the Web Services Description Language (WSDL Christensen et al., 2001), an XML-based language for describing the operations a service offers, and SOAP (Gudgin et al., 2003), an XML-based messaging protocol for communicating with a service. At the time of writing, however, the discovery and use of the relatively few services which exist relies to a large extent on syntactic matching of terms and on human engineering of the content of the invocation calls to them. In order to move towards a semantic service environment, efforts have been made over the last couple of years to develop the OWL-S (previously DAML-S) upper ontology for describing web services. The intention of this initiative is to provide an XML-based ontology which stipulates the basic information that services should expose to the environment in order to facilitate their automatic discovery, invocation, composition and monitoring (OWL-S). This ontology is specified in the OWL Web Ontology Language which provides a language for specifying Description Logic constructs in the syntax of XML and building on top of the RDF data model. Description Logics (e.g. Baader et al., 2003) form a subset of first-order logics which are particularly suited to the description of hierarchical ontologies of concepts, and possess appealing tractability characteristics. Hence, an OWL document describes a machine-

[2] See `http://www.nesc.ac.uk/esi/events/303/`, which offers a fairly recent overview of current work in the workflow area.

processable ontology or fragment of an ontology.

3.2. OWL-S: Profile, Process and Grounding

The OWL-S ontology is divided into three principal areas (cf. Figure 1). The **Service Profile** is used to describe the purpose of the service, and so primarily has a role in the initial discovery of candidate services for a particular task. For the purposes of this paper, we will concentrate on the use and description of profiles, and hence on NLP service discovery. The **Service Model** describes how the service is performed, and is intended for more detailed consideration of the adequacy of the service for the task, to allow the precise composition and coordination of several services and to enable the execution of the service to be monitored. Finally, the **Service Grounding** specifies in concrete terms how the service is actually invoked, the nature of the messages it expects, the address of the machine and port to which these messages should be addressed and so on. We assume that, in general, if a service profile meets the requirements of a client, then any grounding of that service will be an adequate instantiation.

The role of the Profile, then, is to describe the essential capability of the service by characterizing it in functional terms (in addition, non-functional aspects of the service can be specified through additional 'service parameters'). This functional characterization is expressed by detailing the inputs a service expects, the outputs it produces, the preconditions that are placed on the service and the effects that the service has. As well as characterizing services, the Profile has an additional use: to allow potential clients to specify and query for their desired services (which may be partial or more general in nature where details are irrelevant to the client).

Through the use of these *IOPE (Input-Output-Preconditions-Effects)* parameters, a service (or query) may be described in terms of a transformation of its input data into its output data (for example, a POS tagging service can be described as transforming a document into a tagged document). By 'typing' data in this fashion, we gain the ability to define and instantiate 'semantic pipelines' of data through a workflow consisting of a number of distinct services.

However, another mode of use is possible: by extending the core OWL-S ontology within a particular domain with subclasses of the **Profile** class, we also gain the ability to advertise and request services in terms of their categorization; so one might ask for, say, an **NL-Tagger** if one knew that a tagger was required at this point in the workflow. Both the 'transformation' and 'categorization' modes have their uses, and so it is desirable that they be supported in any environment.

This leads to consideration of precisely how particular services in a particular domain are to be described. The OWL-S ontology is (necessarily) domain-independent: to express concepts of particular domains one has to extend the OWL-S ontology through the introduction and use of additional ontological knowledge. However, the use of domain-specific ontologies in this manner places certain obligations on agents in this domain. For service discovery to be possible, both the service providers and potential clients must use the same ontologies: the former to advertise their services, the latter to formulate their requests.[3] Accordingly, there is a need for a standardization effort within domains in order to develop useful and useable ontological descriptions of services. Section 4. describes one such extension of the OWL-S Profile, for describing NLP services, and in such a manner as to permit both the transformation and categorization modes of use described above.

3.3. Reasoning with Profiles: Brokering

Another implication of our approach is that there is at least one 'broker' agent in the domain that acts as a repository for service advertisements and is able to answer service requests.[4] The locations of these brokers would of necessity be known *a priori* to agents in the domain.

Among the fundamental reasoning capabilities of Description Logics are the subsumption of class terms and the classification of individuals into their appropriate categories or classes. Brokers can exploit these abilities to perform service discovery in a number of different ways. For example, on its advertisement, the profile description can be used to classify this service instance into its appropriate location in the domain ontology. Subsequent queries can be interpreted as defining a class description of the desired services; the instances of classes in the service hierarchy which are equivalent to or subsumed by this class are considered to satisfy this query.[5]

It is with this sort of reasoning in mind that we approach the formalization of the NLP domain.

4. A Profile Hierarchy for Linguistic Resources

If we view NL resources as classes arranged in a hierarchy, then a number of taxonomies are possible. It seems rela-

[3] Alternatively, one could envisage the use of different ontologies, along with descriptions of equivalence mappings between their entities, but this introduces additional engineering and processing overheads. The automation of ontology mapping is a difficult problem, for which there are currently no general solutions.

[4] Different types of broker are possible. The simplest (sometimes termed a 'matchmaker' agent) would return matching advertisements to the requesting agent, which is then responsible for selecting and invoking one of these services. More sophisticated brokers might try to dynamically construct composite 'services' consisting of a number of individual services were none of these alone can satisfy the query, or else to apply heuristics to select, negotiate with and invoke services on behalf of the requester. Cf. (Paolucci et al., 2002) for further discussion.

[5] This basic approach can be extended, if more solutions are required, to return instances of classes *which subsume* the query class, or even of those which are merely not necessarily disjoint with the class (although the solutions returned in these cases can no longer be 'guaranteed', in any sense, to satisfy the query).

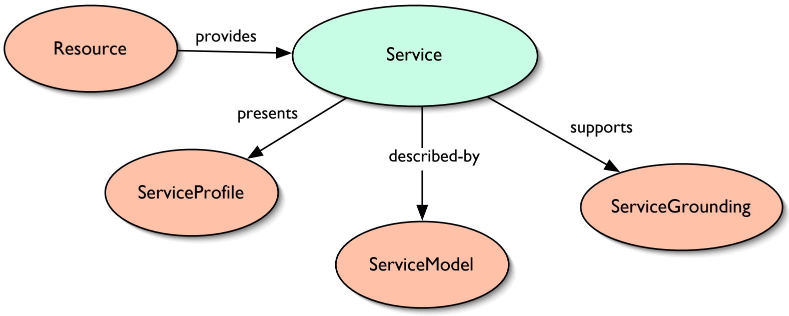


Figure 1: OWL-S Service Ontology

tively uncontroversial to posit a class NL-Resource which is partitioned into two subclasses, NL-StaticResource and NL-ProcessingResource (cf. Cunningham et al., 2000). By ‘static resources’ we mean things like corpora, probability models, lexicons and grammars; by ‘processing resources’ (or processors) we mean tools such as taggers and parsers that use or transform static resources in various ways. As mentioned earlier, the main challenge is to find a motivation for imposing a further taxonomy onto NL-ProcessingResource. Our proposal rests on the following ideas:

1. NLP processors have documents as both input and output.
2. Documents have properties which impose preconditions on processors and which also record the effects of processing.
3. A specification of the properties of documents, as input/output parameters, induces a classification of NLP processors.

We make the assumption that NLP tools are in general **additive**, in the sense that they contribute new annotation to an already annotated document and do not remove or overwrite any prior annotation.[6] As a result, at any point in the processing chain, the annotated document is a record of all that has preceded and thereby provides a basis for making subsequent annotation decisions. This general approach is particularly prominent in XML-based approaches to linguistic markup, but is also prevalent elsewhere.

4.1. Document Properties

Figure 2 illustrates the Document class, together with its main properties. We do not wish to be prescriptive about the allowable class of values for each of these properties. Nevertheless, we will briefly describe our current assumptions.

Document	
hasMIME-Type	MIME-Type
hasDataFormat	anyURI
hasAnnotation	Annotation
hasSubjectLanguage	ISO-693
hasSubjectDomain	Domain

Figure 2: The Document class

hasMIME-Type: The obvious values to consider are `audio` for processors which allow speech input, and `text/plain` and `text/XML` for text processing tools. However, we also wish to allow cases where the value of hasMIME-Type is underspecified with respect to these second two options. Consequently, we treat Text as a subclass of MIME-Type, partitioned into subclasses TextPlain and TextXML.

hasDataFormat: The value of this property is a URI, more specifically, the URI of a resource which describes the data format of the document. By default, the resource will be an XML DTD or Schema, but any well-defined specification of the document’s structure would be acceptable in principle.

hasAnnotation: We treat Annotation as an enumerated class of instances, namely the class {word, sentence, pos-tag, morphology, syntax, semantics, pragmatics}. Although we believe that these annotation types are fairly non-controversial, any broadly-accepted restricted vocabulary of types would be acceptable. The presence of word and sentence reflect the fact that tokenizers will typically segment a text into tokens of one or both these types. Types such as syntax are intended to give a coarse-grained characterization of the dimension along which annotation takes place. However, the specific details of the annotation will depend on the data model and linguistic theory embodied in a given processing step, and we wish to remain agnostic about such details.

hasSubjectLanguage: Following Bird and Simons (2001), we use the term ‘subject language’ to mean “the language which the content of the resource describes or discusses”. Values for this property

[6] In practice, some removal of low-level annotation might take place, and we could also envisage approaches in which ambiguity is reduced by overwriting previous annotation. Nevertheless, for current purposes the assumption of additivity seems a reasonable simplfication,

are presumed to come from ISO 639 (i.e., two- or three-letter codes).[7]

hasSubjectDomain: We are focussing here on tool-related properties, rather than application-related properties; consequently the domain or subject matter of a document is outside the scope of our discussion. However, within a given application, there may well be domain ontologies which would provide useful detail for this property. Moreover, it is obviously of interest to test whether a statistical tool that has been trained on one domain can be ported to another.

At least some of the document properties that we wish to record fall within the scope of Dublin Core metadata, and indeed we might want augment the properties mentioned above with further elements from the Core, such as `publisher` and `rights`. Bird and Simons (2003) have argued in favour of uniformly building metadata for describing language resources as extensions of the Dublin Core. On the face of it, this is an attractive proposal. However, there is at least a short term obstacle to implementing it within our current framework: as an intellectual resource, an OWL-S ontology also needs to be provided with metadata, and the obvious solution is to encode such information using Dublin Core elements. Thus, we would need to carefully distinguish between metadata concerning the ontology itself, and metadata concerning classes of objects (such as Document) within the ontology. We therefore postpone consideration of this issue to the future.

4.2. Processing Resources

In Figure 3, we sketch a portion of the Profile Hierarchy in order to illustrate the classification of processing resources. The class NL-ProcessingResource is shown with two properties, hasInput and hasOutput: both take values from the class Document. Now, we can create subclasses of Document by restricting the latter's properties. For example, consider the class Document $\sqcap \exists$ hasMIME-Type . Text. This is interpreted as the intersection of the set of things in the extension of Document with the set of things whose hasMIME-Type property takes some value from the class Text.

To create a subclass of NL-ProcessingResource, we restrict the class of the inputs, outputs, or both. For example, if the property hasInput is restricted so that its value space is not the whole class Document, but rather just those documents whose MIME type is Text, then we thereby create a new subclass of NL-ProcessingResource; i.e., those processors whose input has to be text rather than audio. We call this the class NL-Analyzer (implicitly in contrast to speech recognizers, whose input would be audio). Note that since the domain of the property hasMIME-Type is in any case restricted to the class Document, we can simplify hasInput . (Document $\sqcap \exists$ hasMIME-Type . Text) to hasInput . ($\exists$ hasMIME-Type . Text), as shown in the property specification for NL-Analyzer in Figure 3.

Every subclass of NL-Analyzer will of course inherit these restrictions, and will in turn impose further restrictions of their own.[8] Thus, we might insist that every tokenizer identifies and annotates word tokens. That is, NL-Tokenizer's output will be a Document with the additional restriction that the set of annotation types marked in the document contains word. Similarly, NL-Tagger will require that its input document has been marked for the annotation type word (i.e., has been tokenized), and will output a document which has additionally been marked for the annotation type pos-tag.

Recall that as a value of hasMIME-Type, Text is underspecified: it can be specialised as either TextPlain or TextXML. Consequently, a tagger which was able to deal equally with both kinds of input could advertise itself as having the more general value for hasMIME-Type, namely Text. This would allow us to compose the tagger with a tokenizer whose output had the property hasMIME-Type . TextXML—that is, composition is allowed if the input of the tagger subsumes the output of the tokenizer. However, the reverse is not true. Suppose the tagger only accepts input with hasMIME-Type . TextXML. Then it cannot straightforwardly be composed with a tokenizer whose output is more general, namely hasMIME-Type . Text.

Although we have concentrated on Document as the input parameter for processors, we need to allow additional inputs. For example, we allow the NL-Tagger class to have the input parameter usesTagset, where possible instances would include the Penn Treebank Tagset, the CLAWS2 Tagset, and so on. Moreover, the subclass of probabalistic taggers would require an additional input parameter, namely the probability model acquired during training.

Within the framework of OWL-S, we would expect a concrete service to be an instance of a class defined in the Profile Hierarchy. Thus, a particular tagger, say `TnT`, would advertise itself by declaring that it was an instance of NL-Tagger, and further specifying values for properties that were mandatory for this class.

4.3. Data Format Requirements

In our earlier discussion, we said that the value of hasDataFormat would be a file URI. An alternative would be to allow processors to specify abstract data types as inputs and outputs (Sycara et al., 2002; Zaremski and Wing, 1997). For example, we might say that a tagger takes as input a sequence of sentences, each composed of a sequence of word tokens, and outputs a sequence of sentences, each composed of a sequence of word-tag pairs. However this doesn't fit in well with the limitations of ontology languages such as Description Logic. For the purposes of matchmaking, a pointer to a format definition file outside the profile hierarchy seems sufficient and more tractable.

[7] Cf. `http://www.loc.gov/standards/iso639-2/`.

[8] Note that Description Logic, and thus OWL-S, only supports strict inheritance—defaults are not accommodated.

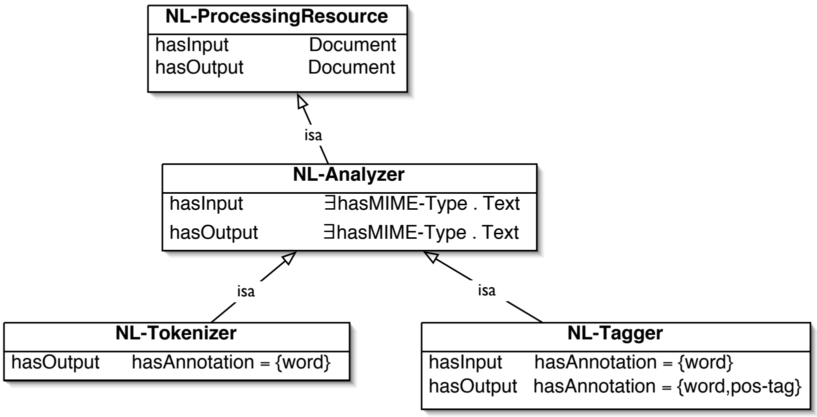

NL-ProcessingResource
hasInput Document
hasOutput Document
isa
NL-Analyzer
hasInput ∃hasMIME-Type . Text
hasOutput ∃hasMIME-Type . Text
isa
isa
NL-Tokenizer
hasOutput hasAnnotation = {word}
NL-Tagger
hasInput hasAnnotation = {word}
hasOutput hasAnnotation = {word,pos-tag}

Figure 3: The ProcessingResource class

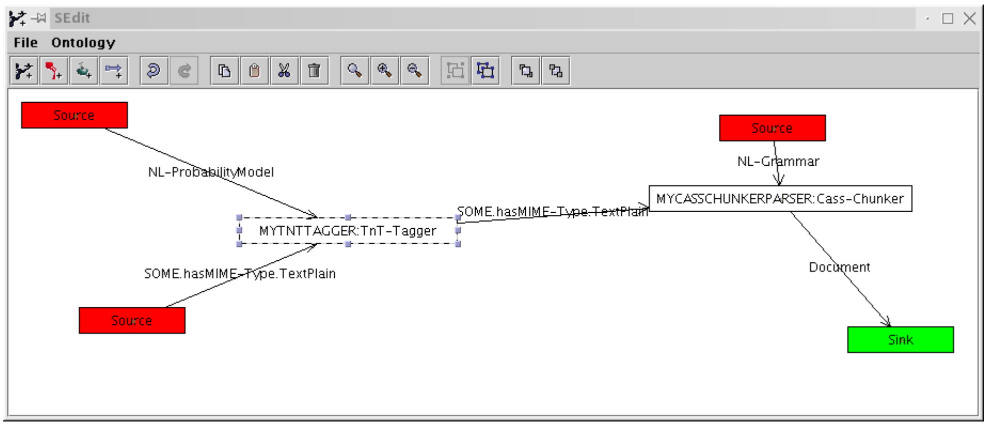


Figure 4: An NLP Web Service Client tool

5. Towards Implementation

To experiment with some of the ideas proposed in this paper, we have developed a prototype environment for the discovery, coordination and (eventually) invocation of NLP Web Services. The NLP Profile Hierarchy described in section 4. has been implemented as an OWL ontology, using the Protégé editor and OWL plugin.[9] Unfortunately, the versions of the OWL-S ontologies available at the time of writing fail to validate in Protégé, and we therefore based our approach on the modified versions made available by Péter Mika at `http://www.cs.vu.nl/~pmika/owl-s/`. The version of the NLP Profile Hierarchy described here can be found at `http://gridnlp.org/ontologies/2004/`.

In order to be able to reason about NLP services, we have used a broker service, built on top of the RACER (Haarslev and Moller, 2001) Description Logic engine. This broker maintains a description, based on the NLP ontology, of the available language processing resources in the environment; when it receives service advertisements, described using OWL-S and this domain ontology, it classifies these and stores them as instances of the appropriate class in the hierarchy. On receiving an OWL-S query, it composes a class description from this and then returns, as potential solutions, (the URLS of) any service instances of classes equivalent to or subsumed by this description.

This broker is itself a web service, accessed through a WSDL end-point. On the client side, we have developed a prototype composition tool for composing sequences of services and querying the broker. The user is able to specify either the type of processing resource that is needed, or the constraints on the data inputs and outputs to some abstract service (or a combination of both) and the tool constructs the appropriate OWL-S, sends this to the broker (via WSDL and SOAP messaging which is hidden from the user) and then presents the alternative services — if any — to the user. Once a user selects one of these, the tool fetches the URL of the service to extract more detailed information about the service, and the user's composition view is updated accordingly.

Figure 4 shows a screen-shot of the tool being used to define a workflow; data, represented by directed edges in this graph (with labels describing the class of the data) flows from 'Sources' to 'Sinks' via one or more services, represented as nodes, labelled with the service name and class. Hence, the screen-shot shows a two-service workflow, producing a DOCUMENT output. To illustrate the use of the tool, and its interaction with the broker, we will now step through the process by which this simple workflow was

[9] See `http://protege.stanford.edu/` for details.

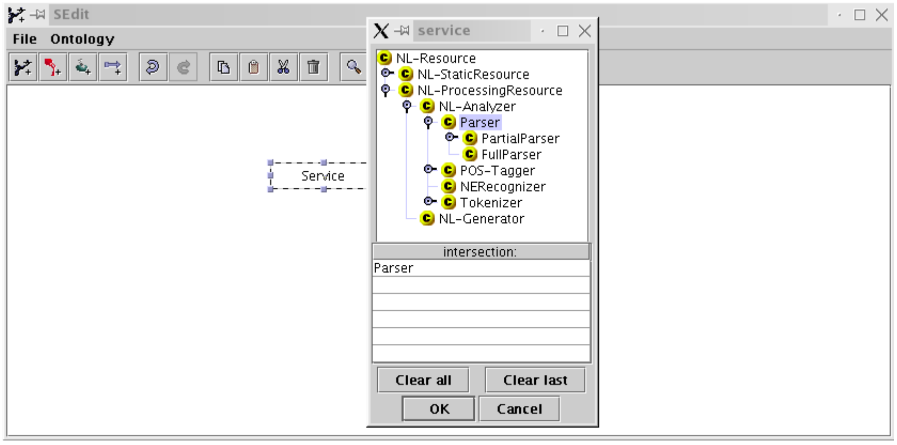


Figure 5: Specifying the class of a desired service

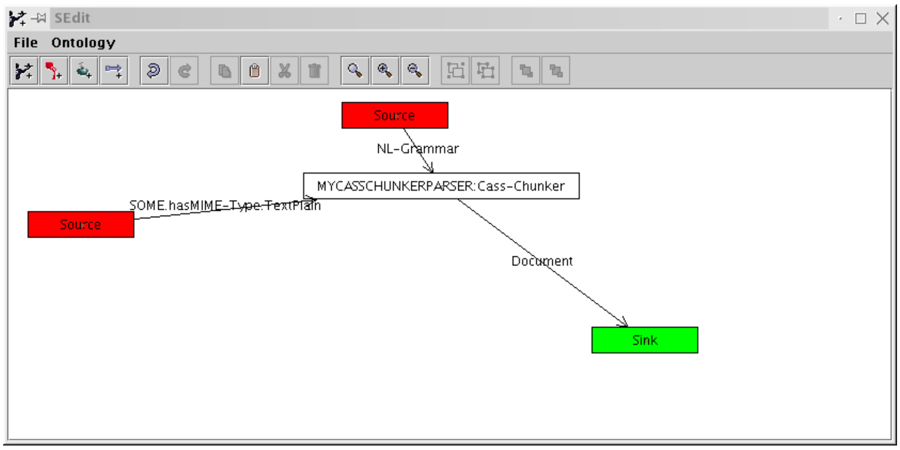


Figure 6: Elaborating the workflow through interaction with the broker

constructed. (To keep this example reasonably clear, the services described are described in rather less detail than might be expected in reality.)

The user — an NLP researcher — here begins with the desire to produce a parsed document. Accordingly, she begins by defining an (anonymous) service of class Parser. The tool has access to the NLP ontology, and a pop-up window allows the class of the desired service to be specified (Figure 5).

Now, the user, via a drop-down menu, places a call to the broker (the address of which is hard-wired into this tool) for details of available services that meet this specification. This has the effect of creating an OWL-S document, the **Profile** of which is an instance of class Parser. Since this is a query, the broker uses this information to find and return the URIs of (the OWL-S descriptions of) all advertised instances of this class and of any of its child classes. In this case, there are two such instances, called **MyLTChunkerParser** and **MyCassChunkerParser**. These are presented to the user as alternatives; she arbitrarily chooses the latter (which is of class Cass-Chunker, an (indirect) subclass of Parser), and its OWL-S description, which specifies the required inputs (namely an NL-Grammar and some (Document) thing which hasMIME-Type of class TextPlain) and the output (a Document), allowing these to be automatically added to the workflow (Figure 6).

Knowing that she needs to first tag the latter Document input, she now replaces its source node with an anonymous service of class POS-Tagger (Figure 7). The broker can now be queried for services of this class which produce an output which hasMIME-Type of class TextPlain. Among the matching services returned by the broker is **MyTNT-Tagger**, which is selected and added to give the workflow shown in Figure 4.

If satisfied with this workflow, the next steps would involve checking and elaborating the workflow further using the OWL-S **Model** of each individual service, and then invoking the workflow using the **Grounding** of each. However, the description of these elements of services will require further conceptualization of the domain, and as a result these steps are not yet implemented. The development of a similar tool to allow human service providers to construct and *advertise* the OWL-S descriptions of their services is also envisaged.

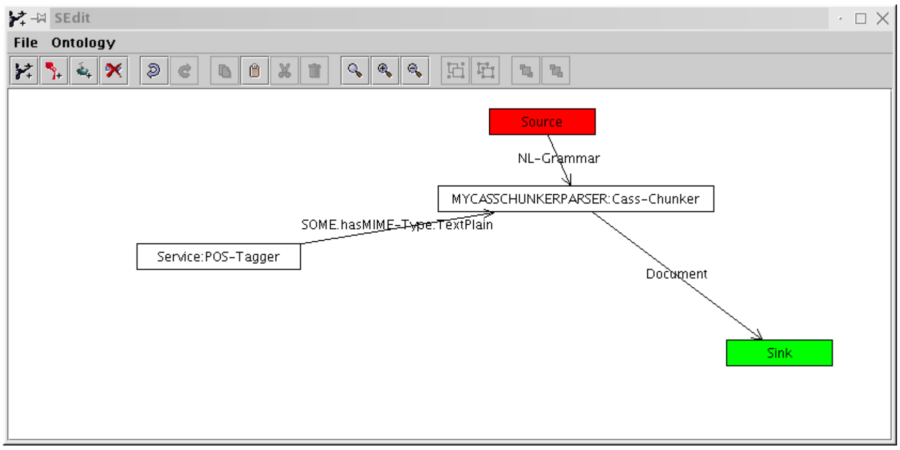


Figure 7: Extending the workflow

6. Conclusion and Future Work

We have argued that a service-oriented view of NLP components offers a number of advantages. Most notably, it allows us to construct an ontology of component descriptions in the well-developed formalism of Description Logic. This in turn supports service discovery and service composition. We have only considered serial composition here, but there is no reason in principle not to allow more complex forms of interaction between components.

One of the most interesting recent frameworks for constructing workflows of NLP components is that proposed by Krieger (2003). His approach deserves more detailed consideration that we have space for here. However, an important difference between our approach and Krieger's is that we do not require components to interact within a specific programming environment such as Java. By wrapping components as services, we can abstract away from issues of platform and implementation, and concentrate instead on the semantics of interoperability. In future work, we will spell out in detail how NLP services described at the OWL-S Profile level can be grounded in concrete resources.

References

Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics as ontology languages for the semantic web. In Dieter Hutter and Werner Stephan, editors, *Festschrift in honor of Jörg Siekmann*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2003.

Tom Bellwood, Luc Clément, and Klaus von Riegen. UDDI technical white paper. `http://uddi.org/pubs/uddi-v3.00-published-20020719.htm`, 2002.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

Steven Bird and Gary Simons. The OLAC metadata set and controlled vocabularies. In *Proceedings of the ACL/EACL Workshop on Sharing Tools and Resources for Research and Education*, Toulouse, 2001. Association for Computational Linguistics.

Steven Bird and Gary Simons. Extending Dublin Core Metadata to support the description and discovery of language resources. *Computing and the Humanities*, 37: 375–388, 2003.

Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000 of the 6th Applied NLP Conference, ANLP-2000*, 2000. URL `\url{http://acl.ldc.upenn.edu/A/A00/A00-1031.pdf}`.

E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language. `http://www.w3.org/TR/2001/NOTE-wsdl-20010315`, 2001.

Hamish Cunningham, Kalina Bontcheva, Valentin Tablan, and Yorick Wilks. Software Infrastructure for Language Resources: a Taxonomy of Previous Work and a Requirements Analysis. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2)*, Athens, 2000.

James Curran and Stephen Clark. Language independent NER using a maximum entropy tagger. In Walter Daelemans and Miles Osborne, editors, *Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Alberta, Canada, 2003. Association for Computational Linguistics. In association with HLT-NAACL 2003.

David de Roure and James A. Hendler. E-science: The Grid and the Semantic Web. *IEEE Intelligent Systems*, 19(1): 65–70, January/February 2004.

Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. Grid services for distributed system integration. *Computer*, 35(6):37–46, 2002.

Ian Foster, Carl Kesselman, and Steven Tuecke. The Anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3), 2001.

M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, and H. F. Nielsen. Simple object access protocol (SOAP). `http://www.w3.org/TR/soap12-part1/`, 2003.

V. Haarslev and R. Moller. RACER system description. In *Proceedings of the First International Joint Conference on Automated Reasoning*, pages 701–706. Springer-Verlag, London UK, 2001.

Tony Hey and Anne E. Trefethen. The UK e-Science core programmed and the Grid. *Future Generation Computer Systems*, 18(8):1017–1031, 2002. ISSN 0167-739X.

Hans-Ulrich Krieger. SDL—A description language for building NLP systems. In Hamish Cunningham and Jon Patrick, editors, *HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pages 83–90, Edmonton, Alberta, Canada, May 2003. Association for Computational Linguistics.

OWL-S. OWL-S: Semantic markup for web services. `http://www.daml.org/services/owl-s/1.0/`, 2003.

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, and Katia Sycara. Semantic matching of web services capabilities. In *Proceedings of the 1st International Semantic Web Conference (ISWC2002)*, pages 333–347, 2002.

Katia Sycara, Seth Widoff, Matthias Klusch, and Jianguo Lu. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5:173–203, 2002.

Amy Moormann Zaremski and Jeannett M. Wing. Specification matching of software components. *ACM Transactions on Software Engineering and Methodology*, 6(4): 333–369, 1997.