

# COLDIC, a Lexicographic Platform for LMF Compliant Lexica

Núria Bel, Sergio Espeja, Montserrat Marimon, Marta Villegas

Institut Universitari de Lingüística Aplicada

Universitat Pompeu Fabra

Pl. de la Mercè, 10-12, Barcelona, Spain

E-mail: {nuria.bel,sergio.espeja,montserrat.marimon,marta.villegas}@upf.edu

## Abstract

Despite of the importance of lexical resources for a number of NLP applications (Machine Translation, Information Extraction, Event Detection and Tracking, Question Answering, among others), there has been a traditional lack of generic tools for the creation, maintenance and management of computational lexica. The most direct obstacle for the development of such generic tools, that is, independent of any particular application format, was the lack of standards for the description and encoding of lexical resources. The availability of the Lexical Markup Framework (LMF) has changed this scenario and has made it possible the development of generic lexical platforms. COLDIC is a generic platform for working with computational lexica. The system has been specially designed to let the user concentrate on lexicographical tasks, but still being autonomous in the management of the tools. The creation and maintenance of the database, which is the core of the tool, demand no specific training in databases. A LMF compliant schema implemented in a Document Type Definition (DTD) describing the lexical resources is taken by the system to automatically configure the platform. Besides, the most standard web services for interoperability are also generated automatically. Other components of the platform include build-in functions supporting the most common tasks of the lexicographic work.

## 1. Introduction and Motivation

Computational lexica are repositories of information about words in particular languages that are normally issued for specific applications and tools. Despite of the importance of lexical resources for a number of NLP applications (Machine Translation, Information Extraction, Event Detection and Tracking, Question Answering, among others), there has been a traditional lack of generic tools for the creation, maintenance and management of computational lexica.

Every lexicon developer had to create his/her own management tools as every lexical model had characteristics that required customization, and for any particular project or application to develop a sophisticated lexicographical tool is usually out of its scope (and budget). And the lack of generic tools has had a negative impact in the creation and management of large scale lexica for Natural Language Processing (NLP).

Along with different large lexical developments, there were some attempts to produce lexicographical workstations, for instance in the following projects: *Preparatory Action for linguistic Resources Organisation for Language Engineering* (PAROLE; Zampoli, 1997); *Semantic Information for Multifunctional Plurilingual Lexicons* (SIMPLE; Lenci et al., 2000) and *International Standards for Language Engineering* (ISLE; Atkins et al., 2002). But, as mentioned, they were too tied to their specific design of lexical entries, and making impossible to use them for data having other format. Other initiatives, as the *Open Lexicon Interchange Format* (OLIF), have develop toolsets to support tasks related to lexical description, but, again, too restricted to their internal format.

With the proposal for an standard for the representation and encoding of lexica, the Lexical Markup Framework (LMF) (Francopoulo et al. 2006), the scenario must change as is proposing a standard that allows the description and encoding of data coming from different tradition. This recent ISO proposal has opened

new possibilities to the design of generic tools and justifies investments in the development and use of a generic tool that can be used for lexica from different origins, and with different characteristics.

COLDIC is a generic platform for working with computational lexica. The system has been specially designed to let the user concentrate on lexicographical tasks, but still being autonomous in the management of the tools. The creation and maintenance of the database, which is the core of the tool, demand no specific training in databases. A LMF compliant schema implemented in a Document Type Definition (DTD) describing the lexical resources is taken by the system to automatically configure the platform. The LMF schema has been the basis for defining the internal structure of the COLDIC database because the need of guaranteeing interoperability for data integration, exchange and service also to remote tools via web services<sup>1</sup>. Besides, the most standard web services for interoperability are also generated automatically.

A specific aspect taken into account in the development of COLDIC has been to maximize the support to the lexicographer. Despite of the amount of work involved in manually crafting a lexicon, and the importance of the task, there has been a complete lack of support to them in tools proposed in the past. The main objective of COLDIC is to offer lexicographers a tool to focus on lexical development tasks. The creation of the database and its maintenance demand no specific training in databases as the LMF schema implemented in a XML Document Type Definition (DTD) is taken by the system to configure the platform. Further special build-in functions supports the user in the most common tasks of

---

<sup>1</sup>A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards.

the lexicographic work.

In the rest of the paper we present the main characteristics of the tool. We start by presenting the LMF, which is the backbone of the COLDIC platform. Section 3 is devoted to the description of the tool and its components. The Generation Modules are presented in section 3.1; Administration modules in section 3.2, Core interface modules in section 3.3, which include the Advanced Query Tool, the Macro-like Query Constructor, and the Advanced Form Editor Tool. Section 3.5 presents the Web Services modules. Section 4 is a list of some improvements that are planned for the future versions of the tool, whose technical features and availability conditions are addressed at Section 5.

## 2. Lexical Markup Framework

The Lexical Markup Framework was a joint ISO TC37/SC34 and LIRICS (a EU funded project in the e-content program, <http://lirics.loria.fr>) initiative to build a standardized abstract framework for the construction of computational lexicons. The aim of LMF was to create a metamodel inclusive of the specificities of main lexicographical styles. This metamodel provides the user with a representation of lexical objects, the structure of the information underlying its description and its use.

Used as a standard for the creation and use of electronic lexical resources, LMF is the basis for a real exchange of data between and among these resources, and for the use of these resources in remote, distributed applications based on web services. The ultimate goal of LMF is to facilitate true content interoperability across all aspects of electronic lexical resources.

LMF is crucially structured into the LMF core package, together with two main extensions: a Machine Readable Dictionary one and a NLP lexical resources one. While the core package describes the basic hierarchy of information of a lexical entry, the extensions (that use the LMF core components) address specific requirements for particular functionalities. In addition, LMF has only been devised to handle with the structure of the information. Linguistic features that describe the lexical items and which are also used by other fields (such as corpus annotation, language technologies, etc.) are defined in the ISO 12620 Data Category Registry, further guaranteeing standardization and interoperability.

## 3. COLDIC

COLDIC is an integrated lexicographic platform for the creation and management of electronic lexica. The platform consists of a database, a graphical interface for lexicographers and a web services interface. Its core functionality is a database whose creation and maintenance demand no specific training in databases. The system takes the LMF schema implemented in a XML Document Type Definition (DTD) to configure the database and other functionalities of the platform. Another of its main features is that it provides an interface not only for human users, but also for consultation and information delivery as asked by remote systems via web services. This interface is automatically generated and needs no previous knowledge of web services by the user.

The main features of the platform are:

- Reading and parsing of a LMF compliant lexical model DTD and generation of a relational database that can be managed with a core web based interface.
- Offering of a query builder tool that supports the user in the creation of content based queries, with advanced features as macro like queries with parametrized arguments.
- Automatic generation of a graphical view of the lexical model that is used as a support in the query builder tool.
- Support in the creation of encoding forms to assist lexicographers in the introduction of new data, search and validation of encoding features via Data Category Registration remote look up.
- Automatic creation of a number of standard and lexically oriented web services by exploiting the interoperability capabilities of the LMF standard and implementing the LMF for lexica API.

Because user requirements might vary according to different needs, COLDIC comes equipped with a set of basic functionalities:

- Query and browse facilities,
- import, export and migration of data,
- easy encoding of new data by means of user customized forms, and
- test and validation of both the data and the model,

COLDIC consists of the following modules:

- The application generation module, which handles the creation of a relational database in terms of a LMF compliant DTD.
- The administration module, which handles profiles, imports and exports data.
- The core interface module, a graphical interface with facilities the building and executing of queries and forms.
- The web services module, which offers a LMF compliant API.

### 3.1 Generation modules

In order to generate automatically a lexicographic management platform, every user's lexicon model must be declared in a DTD complying with the LMF schema. For the DTD to serve as complete COLDIC initialization information, it must obey certain simple restrictions that will allow the system to infer correctly all the required entities, attributes and the relations holding among them. These restrictions refer mainly to the name of the attributes. When defining an element within an attribute list, which refers to another element in the DTD, it must have the name of these attribute but in plural. As you can see in Figure 1, attributes under SyntacticBehaviour are plural names (`subcategorizationFrames`, `subcategorizationFramesSets`) that refer to the element `SubcategorizationFrame`.

```

<!ATTLIST SyntacticBehaviour
id ID #IMPLIED
  senses IDREFS #IMPLIED
  subcategorizationFrames IDREFS #IMPLIED
  subcategorizationFrameSets IDREFS #IMPLIED>
<!ELEMENT SubcategorizationFrame (feat*, LexemeProperty?,
SyntacticArgument*)>
<!ATTLIST SubcategorizationFrame
id ID #IMPLIED
inherit IDREFS #IMPLIED>
<!ELEMENT LexemeProperty (feat*)>

```

Figure 1: Partial view of the LMF DTD to show the naming convention in COLDIC DTD

In case of changes in the DTD, the user can reparse it and the system will modify the database and tools accordingly. The data will remain unmodified unless the new DTD demands for the deletion of a particular class.

### 3.2 Administration Modules

Administration modules are mainly for dealing with users' accounts and the import and export of data.

In order to accommodate to different users' needs and requirements, an administration module allows for the definition of different profiles. Profiles are intended to define access to different sections of the database, and different views of its contents.

For uploading contents into the database, XML data compliant with the DTD can be imported straightforward into the database. Another administration module exports contents into different formats, one of these being XML.

### 3.3 Core Interface Modules

For easy the description we will refer separately to the core interface modules: the Advanced Query Tool, the Advanced Form Editor Tool and the Content Management Tool.

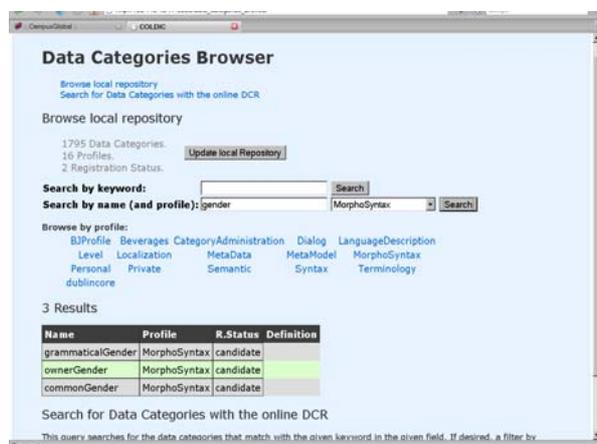


Figure 2. Searching and browsing the ISO Data Category repository.

Because LMF only provides the structure of the lexical entry, all the core modules give an interface to ISO

DataCategories (ISO 12620) for describing lexical elements. The platform access the *syntax* registry (<http://syntax.inist.fr>) using the API developed by Kempes-Snijders et al. (2007). The interface is shown in Figure 2.

### 3.3.1 Advanced Query tool

Advanced Query Tool handles the queries to the database. COLDIC facilitates the creation of queries with the support of graphical views of the underlying lexical model. In order to allow for different degrees of complexity, the user has to handle a query builder form that fixes attributes and conditions on the values. The query builder further assists the user by selecting the entities and conditions that apply to the previously selected item.

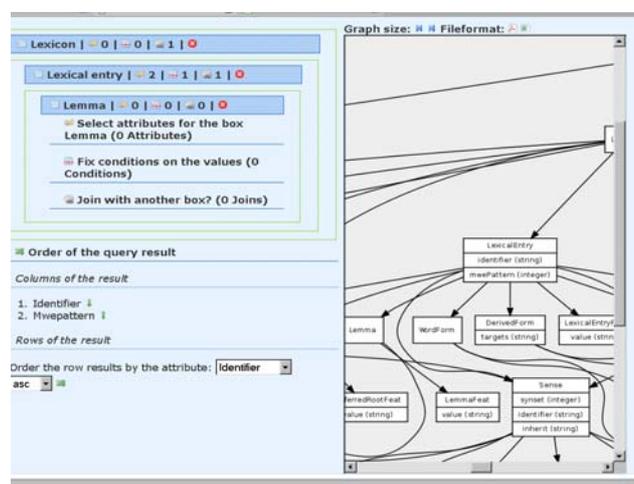


Figure 3: View of the query builder and the graphical view of a part of the complete LMF model

The results of a query are shown, with the information defined by the user, as an HTML web page, and can be saved to a file in this format as well as in XML, Excel formats, CSV and tabulated text.

### 3.3.2 Macro-like Query constructor

Queries built by the user can be stored for later use. Besides, for queries that express singular search conditions, for instance searching entries of a particular inflection paradigm, the condition can be set as a parameter or attribute whose value can vary according to different values for the condition. Thus, queries can be built as macro-like parametrizable elements where the user only has to change the value of the condition. This feature converts COLDIC in a powerful tool that helps creating queries that can later be shared allowing easy modification of its parameters.

### 3.3.3 Advanced Form Editor Tool

In addition to the possibility of uploading data via scripts, COLDIC offers the possibility to create forms for introducing new data. In order to create LMF compliant

data easily, the tool offers a form creator that hides the internal complexities of the formal model, by allowing the definition of defaults, inferences and inherited values. These forms can be shared and issued by more expert lexicographers to help non-experts to encode new data easily.

| attribute name (in table) | name (shown in form)                      | field type                                 | hidden?                  | compulsory               |
|---------------------------|---|--|--------------------------|--------------------------|
| identifier                | <input type="text" value="identifier"/>   | <input type="text" value="string"/>        | <input type="checkbox"/> | <input type="checkbox"/> |
| mwePattern                | <input type="text" value="mwePattern"/>   | <input type="text" value="integer"/>       | <input type="checkbox"/> | <input type="checkbox"/> |
| partOfSpeech              | <input type="text" value="partOfSpeech"/> | <input type="text" value="data_category"/> | <input type="checkbox"/> | <input type="checkbox"/> |

update model attributes

Relate model  
 DerivedForm

Figure 4. Partial view of a form being edited by the user. Names of attributes can be changed and hidden. The value can be set compulsory, and some default value can also be introduced.

### 3.4 Web services modules

Following the LMF specifications, the system offers all the web services needed that conform the LMF API for lexica. Thus, any other program, local and remote, can obtain information from the catalogue, any component structure, search for a particular feature value, and can be programmed to update, add or delete entries or information related to them.

In particular, a specific web service has been devised for parsing XML written data files against the LMF schema, and in case of success automatically updating the data according to the schema.

The application comes also equipped with a web service that consults the Syntax Data Category Registry, where the ISO compliant features to describe lexical items are to be stored for consultation and use of lexicographers.

### 4. Future work

Future plans for improving the lexicographic management platform COLDIC include:

- The administration module should contain a users management administration tool that, for instance, creates logs and historic records of users activities.
- Merging and blending of different resources.
- Additional security measures to prevent from unauthorized access.
- Lexicographic tools such as type definition, class extraction, evaluation, validation and statistical facilities.

### 5. Technical features and availability

COLDIC has been developed in Ruby ([www.ruby-lang.org](http://www.ruby-lang.org)) and uses the Ruby on Rails framework ([www.rubyonrails.org](http://www.rubyonrails.org)). Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. Ruby on Rails is an open source web framework that favors convention over

configuration in order to get less and more understandable code.

The core of COLDIC, the automatic building of the platform out of a DTD, is based in what we have called MetaRails plugins. MetaRails plugins are open source Ruby on Rails plugins that handle the following modules of the platform (Figure 5): Automatic database generation, web services generation, the *querier* and the forms editor. The open source project MetaRails, created for COLDIC development, can be found at [meta-rails.rubyforge.org](http://meta-rails.rubyforge.org) and is distributed under the GPL License. COLDIC is released as a open source project that can be found at [COLDIC.sourceforge.net](http://COLDIC.sourceforge.net) distributed under the GPL license.

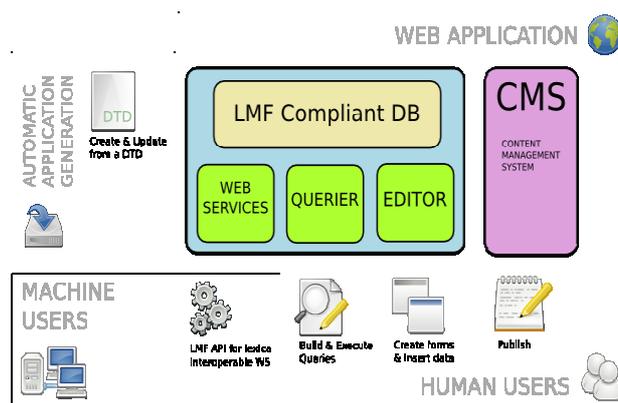


Figure 5. Architecture of the COLDIC platform

## 6. Acknowledgements

This research was supported by the Spanish Ministerio de Educación y Ciencia: project AAILE, HUM 204- 5111-02- 01 and Ramón y Cajal and Juan de la Cierva Programs.

## 7. References

- Francopoulo G., George M., Calzolari N., Monachini M., Bel N., Pet M., Soria C. 2006 Lexical Markup Framework (LMF), *Proceedings of LREC 2006*, Genoa.
- Kemps-Snijders et al. (2007). Data Category Registry API, *Lyrics Delivery 5.1*. <http://lyrics.loria.fr>.
- Kemps-Snijders and J. Nioche. (2007). API for Lexica, *Lyrics Delivery 5.1*. <http://lyrics.loria.fr>.
- Lenci A., Bel N., Busa F., Calzolari N., Gola E., Monachini M., Ogonowski A., Peters I., Peters W., Ruimy N., Villegas M., Zampolli A. 2000. SIMPLE: A General Framework for the Development of Multilingual Lexicons, *International Journal of Lexicography 13(4)*. Oxford University.
- Open Lexicon Interchange Format (OLIF). The OLIF2 Consortium. 2002. <http://www.olif.net>.
- Villegas, M.; Bel, N. (2002). "From DTDs to relational dBs. An automatic generation of a lexicographical station out off ISLE guidelines" in *Proceedings of LREC2002*
- Zampolli, A. (1997). "The PAROLE project in the general context of the European actions for Language Resources". In *Proceedings of the Second European Seminar: Language Applications for a Multilingual Europe*. IDS/VDU, Manheim/Kaunas.

