Towards an XML Representation of Proper Names and Their Relationships

Béatrice Bouchou, Mickael Tran*, and Denis Maurel

Université François-Rabelais de Tours - Laboratoire d'Informatique
DI de l'EPU de Tours, 64 avenue Jean Portalis
37200 Tours, France
{beatrice.bouchou,denis.maurel}@univ-tours.fr,
mickael.tran@etu.univ-tours.fr

Abstract. The presented work is a part of the Prolex project, whose aim is the design and implementation of a multi-lingual dictionary of proper names and their relationships. It focuses on the design of a standard XML representation for this kind of information. We first present the main lines of the conceptual model for proper names (a classical Entities / Relationships model), then we report on our experiment in designing an XML schema from this conceptual model. We describe the current resulting schema and discuss its main features.

1 Introduction

Since 1996, the Prolex project concerns proper names processing, particularly toponyms and inhabitant names [13], and stresses the need to link proper names together, e.g. in Foreign Affairs [14]. We have recently extended our project to every kind of proper names in a multilingual context [15]. We are creating a multilingual database of proper names, the Prolexbase, with linguistic information for natural language processing.

In the Prolex project, the need for an XML representation of proper names and their relationships has appeared first for interface purposes: a standard XML schema could enhance other ways for importing and exporting data, leading to more flexible exchanges or integration of data.

Indeed, according to classical database design, we have built a conceptual model, which has been translated into a logical model in order to efficiently store, maintain and use the dictionary of proper names. This has been done for the relational model: the french table counts more than 323000 entries and 55000 links of relation (these data have been translated into English, Italian, German, Spanich, etc.). As relationships between proper names are stored in the database, we can check whether some proper names are related, we can query for translations, etc. These are typical needs for our target applications: semantic tagging of texts, classification, translation, etc.

Now, there are several motivations for translating the conceptual model *also* into an XML schema:

- In the last few years, XML has become a logical data model, integrated into database applications: it appears however that the process of translating a conceptual model into an XML schema is an open challenge in itself.

* Supported by the RNTL-Technolangue project financed by the French Ministry of Industry.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 44–55, 2005.

- We wish the linguistic resource we are building to be widely used and nowadays XML is the standard way to integrate and/or exchange data: thus, XML can be a convenient interface layer for our relational database.
- Our schema represents a specialized vocabulary for proper names and should be used to describe terminal nodes in tagging models.

Our main contributions in this paper are to present a concrete experiment of XML schema design on the one hand, using an abstract notation to specify both the structure (schema) and the integrity constraints, and on the other hand to report on the current status of the XML schema for proper names (and their relationships), designed mainly on the basis of case studies of French and Serbian, for the moment.

The paper is organised as follows: in section 2 we present the conceptual model of proper names and their relationships. In section 3 we define the notation that we use for our XML schema, we describe the schema (and integrity constraints) and we discuss some of its features. In section 4 we conclude and present future work.

2 The PROLEX Conceptual Model

We have built a conceptual model, shown in Figure 1, derived from our ontology of proper names [10] which results from studies on their typology and on their inflectional and derivational mechanisms in different languages. In Figure 1, ontological concepts and their links are represented by entities (rectangles) and relationships (ovals). This model is structured in four layers which can be grouped in two parts: a multilingual part (*conceptual* and *metaconceptual* layers) and a monolingual part (*instances* and *linguistic* layers). Notice that each layer contains one main entity, which represents words in the layer of *instances*, lemmas in the *linguistic* layer, pivots in the *conceptual* layer and types in the *metaconceptual* layer.

2.1 Multilingual Part

The general architecture has been designed to be flexible enough in order to be applied to different languages without changing the interlingual structure, represented by the *conceptual* layer. The major concept for multilingual aspects is the *pivot*, a conceptual proper name used to connect proper names that represent the same concept in different languages (via the relationship *concept*). Relationships between proper names that are common to every languages are defined on pivots.

This is the case for the *synonymy*, which links pivots with a similar meaning (in a specific context called the *register*: politic, stylistic, diachronic, etc.). For instance, the synonymy in the diachronic register links pivots which represent a concept whose lemma has changed for historical reasons, *e.g. Saint-Petersburg* and *Leningrad* are linked to two different conceptual proper names related by this relationship.

The *predication* links two pivots which are arguments of the same predicate. It has been inspired at first by the lexical function *Cap* of Mel'čuk [12]. But it also includes other relationships like *London* is the capital of *England*, *Jacques Chirac* is the president of *France*, *Aaron* is the brother of *Moses*, etc. Notice that the relation of predication corresponds to a predicate of at least one language (instances of predicates are *president*, *capital*, etc.). The *meronymy*, inspired by WordNet, represents the link between a

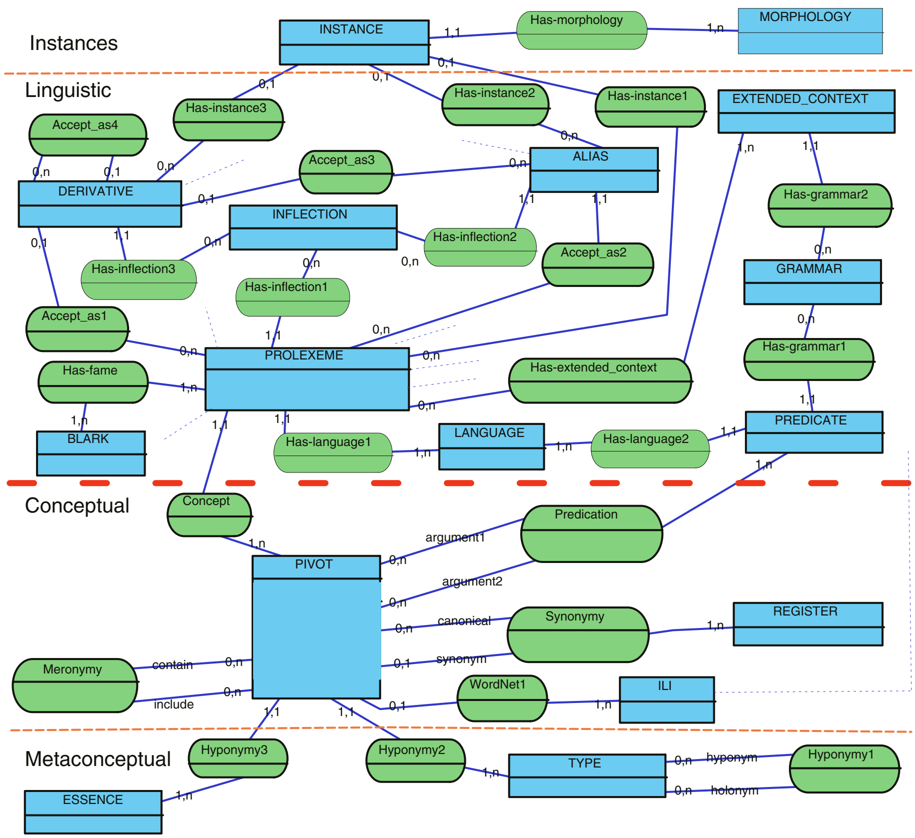

Fig. 1. The conceptual model of the Prolexbase.

whole and its parts. The *WordNet* relationship links a prolexeme and its EuroWordNet ILI (Inter-Lingual-Index) [17].

The metaconceptual layer contains metadata for pivots: *types*, which are hierarchically structured. There are four lexical classes of supertypes, *anthroponyms* (personal and collective names), *toponyms* (place names), *ergonyms* (artefacts and work names) and *pragmonyms* (event names). Simple types are restricted to a set of twenty-six lexical classes, that are determinated by close semantical characteristics. These classes include *organization*, *country*, *celebrity*, etc. The relationship *hyponymy1* is for type hierarchy, *hyponymy2* relates one pivot to its most specific *type*, and *hyponymy3* supports another kind of metadata for pivots: the *essence* specifies if the proper name belongs to a religious, historical or fictional domain.

2.2 Monolingual Part

The monolingual part is specific to each language. It consists in a *linguistic* description (there are big divergences between languages on morphological mechanisms applying to proper names) and a set of words (the *instances*, which are all inflected forms of proper names) associated to their morphology.

The major concept in the *linguistic* layer of Figure 1 is the proper name: we use the term *prolexeme* to refer to the lemma of all the instances of a proper name. Proper names can have *aliases*, which are different variants of a prolexeme, uppercase or lowercase, diacritics, acronyms, abbreviations or transcriptions. Moreover proper names and their aliases may have *derivatives*: the lemma of the prolexeme allows to replace a word in a specific language by another one during translation. For example, in order to translate *It is the car of an inhabitant of Belgrade* in Serbian, we will have *То је Београдјанинов ауто* where the proper name *Београђанинов* is in fact a derivative (more exactly a possessive adjective). The other concepts represented by entities associated to the prolexeme describe features of proper names. A *BLARK* (Basic Language Resources Kit) [6] is an indicator of fame which depends on different factors (the country, the period, etc.). An *extended context* points to a local grammar describing a context where the proper name can occur: it is useful in translation (as it varies from one language to another).

More information on proper names, not detailed in Figure 1, is supported: we allow to indicate if a proper name is linked to an *antonomasia*, a rhetoric device that indicates if we can substitute a phrase for a proper name or vice versa. For example, in English the proper name *biro* has become a common name for a ballpoint pen, whereas in French we use *bic*. We can store *Idiomatic expressions*: for example, *not for all the tea in China* in English will be translated into French by *pour rien au monde* (i.e. for nothing in the world). We have associated to every proper name information about its *sorting*. In most dictionaries, some multiword proper names are classified by permuting their units. For instance, in a French dictionary we will find *Mer d'Aral* under letter A. It is also sometimes useful to indicate whether a proper name may have an article (*determination*): *e.g.* the proper name *Spain* takes an article in French (*l'Espagne*). Finally, every prolexeme, alias or derivative is linked to an *inflection* paradigm.

3 The XML Representation

In the following, we first report on our experiment in translating the E/R (Entities/ Relationships) conceptual model into an XML schema (with constraints), then we present the resulting schema and discuss some of its features.

3.1 From Conceptual Model to XML Schema

There are surprisingly few works on methods of XML schema design, either from scratch or from conceptual models. Derivations from relational models [9] and from UML models [16] have been investigated, but compared to the vast amount of publications about the design of relational databases, this domain still lacks contributions. As we were dealing with an E/R model, we tried to follow steps described in [11] when it was possible. In particular, we use a grammatical notation of XML schemas similar to the one used in [11].

Schema Notations. There are several languages for describing schema of XML documents, and the choice between them is not obvious. DTDs are historically the first means to specify the structure of XML documents, and they are still widely used, even for specifying standards such as the Lexical Markup Framework (ISO standard [7]). But DTDs have shortcomings: in particular, in order to use XML as a logical model from a database point of view, it lacks means to define integrity constraints such as primary keys and foreign keys. In fact, dealing with these constraints in XML document is also a research area in itself ([5], [4]). The W3C consortium has proposed a formalism called XML Schema (or XSD) [3], which offers a variety of new constructs. But recent studies ([2]) tend to demonstrate that current schemas written in XSD only sparingly use these new features for structural specifications: most of them can be expressed by DTDs (the study does not address integrity constraints).

In this paper, we choose to use a high level schema notation which is coupled with a notation for integrity constraints: it is a tree grammar such as in [11]. Any schema written in any existing schema language can be easily translated into such a grammar.

Definition 1. Grammar for schema: The *grammar representing a schema* is denoted by a 6-tuple $\Gamma = (N, E, A, S, P, C)$, where

- N is a finite set of non-terminal symbols (called *types*).
- E is a finite set of element names.
- A is a finite set of attribute names.
- S is a set of start symbols, $S \subseteq N$.
- P is a set of production rules of the form $X \to x(RE)$, where $X \in N$, $x \in E$ and RE is a regular expression:
  $RE ::= \epsilon | \tau | @a | Y | (RE + RE) | (RE, RE) | (RE)? | (RE) * | (RE)+$
  where τ is an atomic data type, ϵ denotes the empty regular expression, $a \in A$, $Y \in N$.
- C is a set of integrity constraints. □

Such a grammar offers a wide expressive power, but we will restrict ourselves to features that can be translated either into a DTD or into an XSD specification, for instance we consider only the atomic data type *string* (for τ), we do not define regular expressions on attributes, etc. Attribute names are preceded by an @: in our schema (as in DTDs or XSD schemas) attributes are parts of element descriptions and contain only values.

From a database point of view, constraints are of fundamental importance, and specially primary and foreign keys: primary keys are a means of locating specific elements of the document and foreign keys allow to reference an element from another element (relationships). In particular, such information is used to maintain the connection from the concept in the real world to its representation when the system that is modeled evolves. As usual to define integrity constraints for XML, we use a subset of XPath expressions [8], precisely we use paths of the form $p ::= x | @a | p/p$, where $x \in E$, $a \in A$. Let PE denote the set of such path expressions. We define the following notations for primary keys and foreign keys: primary keys can be absolute or relative, and foreign keys are defined in the scope of the primary key they refer to.

Definition 2. Integrity constraint specifications:

- An *absolute primary key constraint* is specified as $pkey(X) = (p_1, ..., p_n)$, where $X \in N$ and $p_i \in PE$, $1 \leq i \leq n$. Paths must end with a data node, *ie* a node having a data value. The set $(p_1, ..., p_n)$ represents items composing the key for the type X. Notice that, as keys are specified for types, the schema must define an unambiguous type assignment.
- A *relative -primary- key constraint* is specified as $key(X)relative(Y) = (p_1, ..., p_n)$, where $X, Y \in N$ and $p_i \in PE$, $1 \leq i \leq n$. Such a specification indicates that *inside an element of type Y*, elements of type X are uniquely represented by the items in $(p_1, ..., p_n)$.
- A *foreign key constraint* is specified as $fkey(X, Y) = (p_1, ..., p_n)$, where $X, Y \in N$ and $p_i \in PE, 1 \leq i \leq n$. Such a specification indicates that items in $(p_1, ..., p_n)$, defined for type X, reference items in a key for type Y. □

In the schema for proper names and their relationships, we will use only *unary* (absolute and relative) primary keys (and thus unary foreign keys too).

Design of the Target XML Schema. Due to the lack of space, we could not analyze functional dependencies in section 2: therefore we can not detail here the translation steps. Although recommendations in [11] have been useful for first stages (to decide how to translate some relationships), it was not obvious to systematically derive an XML schema from the conceptual model. We departed from the method in [11] mainly in two points: we did not consider ID/IDREF(S) (special attribute types proposed in DTDs) as a useful way to express integrity constraints and we have had to strongly reorganize root's subelements.

Clearly, the design has been an iterative process: in fact, we even came back to the ontological level, refining the E/R conceptual model, in order to obtain a realistic XML schema to represent the *dictionary of proper names and their relationships.*

3.2 Schema for Proper Names

The schema grammar is $\Gamma = (N, E, A, S, P, C)$: we present it through its set of production rules P (Figure 2) and its set of constraints C (Figure 3). Items in N, E and A are introduced with production rules where they appear. The unique initial symbol in S is *Root*. The first production rule **p1** specifies that a document containing proper names and their relationships is composed of two parts: *(i)* the paradigmatic *Relationships* part, shared by all natural languages (*i.e.* the *Conceptual* and *Metaconceptual* levels of E/R model in Figure 1), and *(ii)* the *Languages* description part, composed of one description for each language, containing proper names and their features (*i.e.* the *Linguistic* and *Instances* levels of E/R model in Figure 1). Notice that elements *relationships* and *languages* are compulsory but their content may be empty, in order to enhance partial descriptions.

p 1 ***Root*** $\rightarrow root(Relationships, Languages)$

p 2 ***Relationships*** $\rightarrow relationships(Pivot*, Predication*, Type*, WordNet)$

p 3 ***Predication*** $\rightarrow predication(@pivot1, @pivot2, PReference+)$

p 4 ***PReference*** $\rightarrow pReference(@language, @predicate)$

p 5 ***Type*** $\rightarrow type(@name, Type*)$

p 6 ***Pivot*** $\rightarrow pivot(@num, @essence, @type, @wordNet?, MeronymOf*, Canonical*,$
˜ $Concept+)$

p 7 ***MeronymOf*** $\rightarrow meronymOf(@pivot)$

p 8 ***Canonical*** $\rightarrow canonical(@pivot, @register)$

p 9 ***Concept*** $\rightarrow concept(@language, @prolexeme)$

p 10 ***WordNet*** $\rightarrow wordNet(Ili*)$ ***; Ili*** $\rightarrow ili(@num)$

p 11 ***Languages*** $\rightarrow languages(Language+)$

p 12 ***Language*** $\rightarrow language(@name, Prolexemes, ExtendedContexts, Predicates,$
˜ $Idioms, Blarks, Statictics, Phonetics, Structures, Grammars, Inflections)$

p 13 ***ExtendedContexts*** $\rightarrow extendedContexts(ExtendedContext*)$
˜ ***ExtendedContext*** $\rightarrow extendedContext(@num, @name, @grammar)$

p 14 ***Predicates*** $\rightarrow predicates(Predicate*)$
˜ ***Predicate*** $\rightarrow predicate(@num, @name, @grammar)$

p 15 ***Statistics*** $\rightarrow statistics(Stat*)$
˜ ***Stat*** $\rightarrow stat(@num, @description, @weight)$

p 16 ***Idioms*** $\rightarrow idioms(idiom*)$
˜ ***Idiom*** $\rightarrow idiom(@num, @description)$
[...]
p 17 ***Prolexemes*** $\rightarrow prolexemes(Prolexeme*)$
˜ ***Prolexeme*** $\rightarrow prolexeme(@num, @name, @inflection, @pivot,$
˜ $@determination?, @sorting?, @structure?, @IliAntonomasia?,$
˜ $RIdiom*, RExtendedContext*, RBlark*, RStatistic*, RPhonetic*,$
˜ $Aliases, Derivatives, Instances)$

p 18 ***RIdiom*** $\rightarrow rIdiom(@idiom)$
˜ ***RExtendedContext*** $\rightarrow rExtendedContext(@extendedContext)$
˜ ***RBlark*** $\rightarrow rBlark(@blark)$
˜ ***RStatistic*** $\rightarrow rStatistic(@statistic)$
˜ ***RPhonetic*** $\rightarrow rPhonetic(@phonetic)$

p 19 ***Aliases*** $\rightarrow aliases(Alias*)$
˜ ***Alias*** $\rightarrow alias(@name, @category, @inflection, Instances, Derivatives?)$

p 20 ***Derivatives*** $\rightarrow derivatives(Derivative*)$
˜ ***Derivative*** $\rightarrow derivative(@name, @category, @inflection, Instances, Derivatives?)$

p 21 ***Instances*** $\rightarrow instances(instance*)$
˜ ***Instance*** $\rightarrow instance(@name, @morphology)$

Fig. 2. The production rules of the schema.

c 1 $key(Pivot) =< @num >$

c 2 $key(Predicate) =< @num >$

c 3 $key(PReference) relative(Predication) =< @language >$
// For a given predication element, there can not be two predicates in the same language.

c 4 $fkey(PReference) =< @predicate > REFERENCES(Predicate) < @num >$

c 5 $fkey(Predication) =< @pivot1 > REFERENCES(Pivot) < @num >$

c 6 $fkey(Predication) =< @pivot2 > REFERENCES(Pivot) < @num >$

Fig. 3. Examples of constraints.

Conceptual and Metaconceptual Level. The first part, rooted at element *relationships*, is specified by rules **p2** through **p10** in Figure 2. It is composed of (see rule **p2**):

- A list of elements of type *Pivot*: recall that the pivot is an abstract notion used to define general relationships between proper names.
- A list of elements of type *Predication*: such element links two pivots via a predicate of a given language.
- A list of elements of type *Type*: each type is the root of a hierarchically structured group of types. The hierarchy (recursive rule **p5**) reflects the relation of hyponymy.
- An element of type *WordNet*: it records links to WordNet ILIs.

Notice again that all lists can be empty (as well as the content of *wordNet* element), so *partial* views of the database can be valid with respect to the schema.

Before describing the type *Pivot*, we first give indications about the other sub-elements appearing in the content of an element tagged *relationships*. The relation of predication (rules **p3** and **p4**) links two pivots through several predicates, each predicate belonging to one language. In this way, having one pivot we can get the pivots it is linked with, and for each one the list of predicates (one predicate for one language, but one predication can exist in several languages, see for instance *brother of*, *frère de*, *hermano de*, etc.). In the same way, from a given predicate (in one language) we can obtain the two pivots and their related prolexemes (either in the same language or in other languages). We use keys and foreign keys (shown in Figure 3) in order to express these links (and to automatically verify them when updating documents, as usual in relational databases). Notice that key **c3** is relative: it is to ensure that *within one predication* there is at most one predicate for one language. The example in Figure 4 contains a predication indicating that *Paris* is the capital of *France*. Indeed, this instance of document contains two pivots in its *relationships* part which correspond to lemmas *Paris* and *France* in its *english language* part, and one predication corresponding to the predicate *capital*.

A *Pivot* (rules **p6** through **p10**) has a unique identifier, an *essence*, a *type* (notice that from that type we can get hyperonym types). An element *pivot* can refer to an entry in WordNet, it can be a meronym for a set of other pivots, it can reference canonical synonyms (in precise registers). Last, an element *pivot*

```
<root>
        <relationships>
                <pivot @num="400", @essence="historical", @type="city", @wordNet="05558236n">
                        <canonical @pivot="410" @register="diachronic"/>
                        <concept @language="english", @prolexeme="500" />
                </pivot>
                <pivot @num="600", @essence="historical", @type="country", @wordNet="05557178n">
                        <concept @language="english", @prolexeme="800" />
                </pivot>
                <predication @pivot1="400", @pivot2="600"> <pReference @language="english", @predicate="500"/>
                </predication>
                <type @name="Toponym">          <type @name="Country"/>          <type @name="City"/> </type>
                …………
                <wordNet >        <Ili @num="05558236n"/>          <Ili @num="05557178n"/>            </wordNet>
        </relationships>
        <languages>
                <language @name="english">
                <prolexemes>
                        <prolexeme @num="500", @name="Paris", @determination="no", inflection="89", @pivot="400">
                                <derivatives>
                                        <derivative @name="Parisian", @category="3", @inflection="96" >
                                                <instances>
                                                        <instance @name="Parisian", @morphology="S" />
                                                        <instance @name="Parisians", @morphology="P" />
                                                </instances>
                                        </derivative>
                                        …………
                                </derivatives>
                                <instances>        <instance @name="Paris", @morphology="S" />         </instances>
                        </prolexeme>
                        <prolexeme @num="800", @name="France", @determination="no", inflection="89", @pivot="600">
                                <derivatives>
                                        <derivative @name="French", @category="3", @inflection="96" >
                                                <instances>        <instance @name="French", @morphology="S"/> … </instances>
                                        </derivative>
                                        …………
                                </derivatives>
                                <instances>        <instance @name="France", @morphology="S"/> </instances>
                        </prolexeme>
                </prolexemes>
                <predicates>        <predicate @num="500", @name="capital", @grammar="12"/> … </predicates>
                …………
                </language>
        </languages>
</root>
```

Fig. 4. An example of proper names in XML: Paris and France.

represents a *concept* which exists in at least one language: for that reason, it is linked with one prolexeme of at least one language, and only one prolexeme per language. Obviously, it can be linked with several prolexemes, each one belonging to a different language. This is the same situation as for the predication relation (with predicates in languages). Therefore, we modelize it in the same way: *concept* elements are for *pivot* what *pReference* elements are for *predication*. Notice that the *language* in element *concept* as well as in element *pReference* is useful for translation applications. Indeed, in this way the access from one prolexeme (or one predicate) to corresponding prolexemes (or predicates) in other languages is immediate (via the pivot or via the predication).

Linguistic and Instances Levels. The second part of a document of proper names is rooted at element tagged *languages* (see rule **p11**). It contains information about at least one language, each language having a name (which is its key) and containing a set of proper names and their descriptions (rule **p12**). A language can also have a list of *idioms*, useful for translation tasks.

Data about proper names (types in rule **p12**, except *Prolexemes*), are sets of information expressed in a standard $num - description$ shape: see rules **p13** through **p16**. The types *Blarks*, *Phonetics*, *Structures*, *Grammars* and *Inflections* are defined in the same way as *Idioms*. Notice that, as a *grammar* can be the same for several *predicate* elements, we chose to have a set of grammar descriptions and to reference one of these grammars from the *predicate* element: this reference is supported by a foreign key. The prolexemes themselves are under the element tagged *prolexemes*, whose type is described by rule **p17**: each *prolexeme* has a unique identifier *num*, a *name*, an $inflection$ code and a reference to its *pivot* (in order to address easily translation tasks for instance). Moreover, it can have information about its *determination* (in French it is *yes* or *no*) and about how to take its components into account in a *sorting* operation: for instance *2, 1* for *Jacques Chirac* indicates that the sorting must be done on *Chirac* first. The prolexeme can also have a reference to an internal *structure* (for compound proper names) and it can correspond to an *antonomasia*: in that case we allow to refer to the ILI WordNet of the corresponding common name, for translation purposes. The prolexeme can also refer to a set of *idioms* and a set of *extendedcontexts* in which it appears. It can be described by BLARKs, statistics and phonetics, too. For one prolexeme one can have sets of *aliases* and *derivatives* (these sets can be empty). Lastly, there is the set of *instances* (*values*) directly linked to the proper name.

Notice in rule **p18** that elements $rExtendedContext$ encountered in a *prolexeme* contain just a reference to an *extendedContext* tagged element (which contains the description of the extended context). This is the same for BLARKs, statistics and phonetics, whereas *aliases*, *derivatives* and *instances* are fully described inside the prolexeme, as they are never shared by two distinct prolexemes (rules **p19** through **p21**).

Of course, all references are specified using keys and foreign keys: we do not describe every constraints for the sake of succinctness.

3.3 Discussion

The schema designed to represent proper names and their relationships takes advantage of XML nesting capabilities (*e.g.* defining recursive types), while avoiding much redundancies by following normalisation recommendations ([1]).

We have not found any need for union type (*i.e.* a type defined by a disjunctive regular expression), although it is a classical type of XML elements content, which denotes that the element is described either by some features or by other features. For instance, we can specify that a paper in a bibliography is either a presentation in a conference or an article in a journal. We have considered this capability in several places, *e.g.* when dealing with aliases and derivatives, but these two notions play different roles for a prolexeme, and one given prolexeme can have both aliases and derivatives... Hence, it seems that the target we modelize (proper names and their relationships) does not need union types.

The aim of modeling proper names and their relationships is to be as exhaustive as possible. Then, all details in the descriptions are present in the model. But we have carefully designed the schema in order that it can be usable even for partial descriptions (using optional contents every time it was possible).

The proper name description can be embedded in more general frameworks for modeling linguistic information. For instance the LMF (Lexical Markup Framework (ISO standard [7])) describes a high level model for representing data in lexical resources used in multilingual computer applications, including multilingual natural language processing lexicons. It is intended as a general framework, in which specialized vocabularies may be embedded without much difficulties. For that purpose, it provides a method for using *Feature Structures* and *Feature Values* to identify components of the lexical resource described. For instance, we could have: $< fname = numBlark+ > 000221 < /f >$ as an element part of description of a *prolexeme*.

It is clear that our approach is far more, let's say, normative: in fact we have design a schema in the classical spirit of database designers, specifying structures and constraints having in mind that there exist a (database) system to deal with these specifications in order to efficiently manage data, here the XML documents. By *managing* we mean classical tasks of a database system: storing, updating, querying, etc. On the contrary, a resource described in a framework such as ISO LMF could hardly take advantages of current and future XML generic tools, comprising database oriented tools.

4 Conclusions

We have presented a contribution to the Prolex project, recently developed within the RNTL-Technolangue project: the design of an XML schema for proper names and their relationships. XML schemas are useful for integration and/or exchange of data, in particular linguistic data.

During the design process, we tried to apply a method proposed in [11] which is to derive an XML schema directly from an (extended) E/R model. This E/R model is also briefly presented in this paper. Our conclusion was that such a derivation is not really straightforward. Nevertheless, following an iterative process we have obtained a structure (schema), together with integrity constraints, that accurately represent the concepts and relationships of the original E/R model.

Our XML schema is a basis for future work, in particular the specification of semantic tags for text markup. More generally, our aim is to use XML for developing new means of applying the dictionary of proper names in natural language processing tasks such as computer aided translation, information extraction, multilingual alignment text, etc.

References

1. M. Arenas and L. Libkin. A normal form for XML documents. In *ACM Symposium on Principles of Database System*, 2002.

2. G. J. Bex, F. Neven, and J. Van den Bussche. DTDs versus XML schema: A practical study. In *Web and Databases (WebDB)*, 2004.
3. P. Biron and Eds. A. Malhotra. *XML Schema part 2.* http://www.w3.org/TR/xmlschema-2, 2001.
4. B. Bouchou, M. Halfeld Ferrari Alves, and M. Musicante. Tree automata to verify key constraints. In *Web and Databases (WebDB)*, 2003.
5. P. Buneman, S. Davidson, W. Fan, C. Hara, and W. Tan. Reasoning about keys for XML. In *Proceedings of Database and Programmming Languages*, 2001.
6. H. Strik C. Cucchiarini, W. Daelemans. Strengthening the dutch human language technology infrastructure. *http://www.elda.fr/article48.html*, 2000.
7. ISO/TC 37/SC 4 Committee. *Language resource management: Lexical Markup Framework.* ISO WD 24613, 2004(E), 2000.
8. M. Fernandez, A. Malhotra, J. Marsh, M. Nagy, and Eds. N. Walsh. *XQuery 1.0 and XPath 2.0 Data Model.* http://www.w3.org/TR/xpath-datamodel, 2004.
9. D. Lee, M. Mani, F. Chiu, and W. W. Chu. Net & Cot: Translating relational schemas to XML schemas. In *Australasian Database Conference*, 2002.
10. D. Maurel M. Tran, T. Grass. An ontology for multilingual treatment of proper names. In *OntoLex 2004, in Association with LREC2004*, pages 75–78, 2004.
11. M. Mani. Erex: a conceptual model for XML. In *Database and XML Technologies: XSym 2004, LNCS Volume 3186*, 2004.
12. I. Melćuk. Dictionnaire explicatif et combinatoire du français contemporain. *Les presses de l'Université de Montréal*, 1984-I, 1988-II, 1992-III.
13. C. Belleil O. Piton, D. Maurel. The prolex data base : Toponyms and gentiles for nlp. In *NLDB'99*, pages 233–237, 1999.
14. D. Maurel O. Piton. Beijing frowns and washington takes notice : Computer processing of relations between geographical proper names in foreign affairs. In *NLDB'2000*, pages 66–78, 2000.
15. D. Maurel O. Piton, T. Grass. Linguistic resource for nlp: Ask for *Die Drei Musketiere* and meet *Les Trois Mousquetaires*. In *NLDB'2003*, pages 200–213, 2003.
16. N. Routledge, L. Bird, and A. Goodchild. UML and XML schema. In *ACM Conference On Information and Knowledge Management (CIKM)*, 2002.
17. P. Vossen. EuroWordNet: A multilingual database with lexical semantic networks. *Kluwer Academic Publishers*, 1998.