GeLaBa

A framework to define classes of XML documents and to automatically derive specialized infrastructures (demonstration)

Benoît Pin and Georges-André Silber
Centre de recherche en informatique
Ecole Nationale Supérieure des mines de Paris
35, rue Saint-Honoré
77305 Fontainebleau cedex
{Benoit.Pin,Georges-Andre.Silber}@ensmp.fr

Introduction. GeLaBa has been created when our research center began to participate to a project called `lheo` funded by the french ministry of employment. The goal of `lheo` was to create an XML schema to represent and exchange informations about professional formations in an uniform way in France. To suit the needs of all the users of this schema, several things are provided by `lheo`: a DTD, a W3C XML Schema, a complete documentation, a specialized API, etc. The problem of maintaining coherence between all those components while insuring an easy evolution of the language was very tricky. Because no existing tool suited our needs, we created a new language (GML) based on XML and a set of generic tools to automate the creation and maintenance of `lheo`. Those tools and GML became the core of GeLaBa. `lheo` is now completely managed with GeLaBa, including its web site `http://www.lheo.org`.

GeLaBa[1] is a framework build upon a language called GML[2] used to define XML classes of documents. Starting from a language definition in GML, GeLaBa provides a collection of tools to automatically generate a complete specialized infrastructure to handle this specific class of XML documents.

GML. The foundation of GeLaBa is GML, a class of XML documents that can describe a subset of all the classes of XML documents[3]. It can represent fewer XML structures than the other schema languages (DTD, W3C XML Schema, Relax NG) but it is more regular, leading to simpler content models. This regular structure is sufficient in many cases and leads to simple but powerful tools. The two main constructions that we allow in GML are: 1) an ordered sequence of unique elements that can be repeated several times (in DTD syntax, content of the form `(a,b+,d?,c*)`), and 2) an element chosen in a restricted set of elements (content of the form `(a|b|c)`).

GML has features that are not present in any schema language. Elements and attributes can be typed with usual data types with various sizes (defined by the user). GML also supports the definition of dictionaries, whose keys can be used as enumerated types in elements or attributes. A dictionary is a simple list of entries, where each entry is a couple (`key`, `value`). Documentation is a fundamental aspect of GML: a multilingual documentation can be embedded in every component of the language definition (elements, attributes, dictionaries). Components can also have properties defined by the schema creator, to add extra informations not defined in the GML schema.

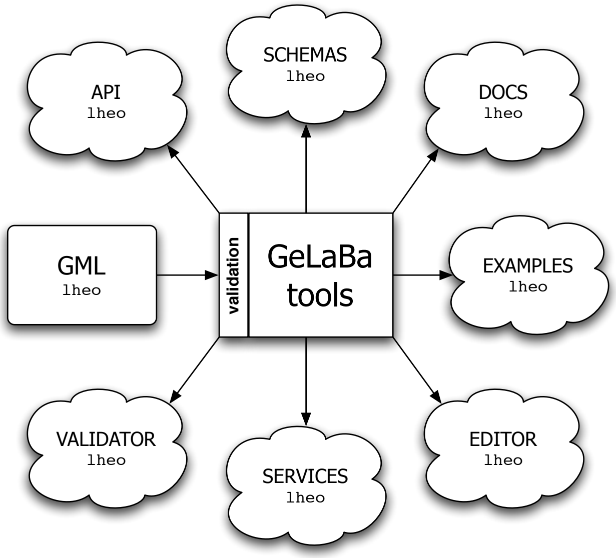

Figure 1: The generation of a complete infrastructure for the `lheo` class of XML documents with the GeLaBa framework.

To sum up, GML is a schema language with a simple content model, embedded multilingual documentations, data types, dictionaries and properties.

[1] GeLaBa means "*Générateur de Langage de Balisage*" (in French), i.e. *Markup Language Generator* in English.

[2] GML stands for *GeLaBa Markup Language*

[3] The complete DTD of GML can be found here: `http://www.gelaba.org/1.0/gml.dtd`

Demo of the GeLaBa framework. Figure 1 illustrates how GeLaBa works, with the example of `lheo`. Starting from a definition of `lheo` expressed in GML, GeLaBa validates the GML definition of `lheo` and then generates a specialized infrastructure to handle documents in `lheo`. The following description gives a list of what we are going to present during the demonstration. Here are the components of the specialized infrastructure automatically generated by GeLaBa:

API Our tools automatically derive an Application Programming Interface (API) to read, modify and write documents in `lheo`. The API validates the document and ensure that a modified document follows the schema of `lheo` before writing a file. This API is available under the form of Python classes (compatible with the Zope/CMF[4] platform), and Java classes. All classes are build upon SAX and DOM.

SCHEMAS Usual schemas are derived from the schema represented in GML: DTD, W3C XML Schema, Relax NG, Schematron. Those schemas have not the same power, but they can be used for several level of validations. For instance, the generated DTD can only express unbounded repetitions whereas the generated W3C XML Schemas can express a maximal number of repetitions. The validation can also be done by our ad-hoc validator (see below).

DOCS Documentations for `lheo` are generated from the GML definition, in HTML and PDF (using LaTeX).

EXAMPLES A generator of valid random documents in `lheo` is derived from the GML definition. This generator is in C language and very fast, allowing the production of a large amount of XML documents in `lheo`. The content of the elements is a random content that respect the types or is taken from a companion example file we provide. This generator is also available under the form of a CGI file to provide a web service. The purpose of this generator is to propose to the users of `lheo` a bunch of documents to test and stress their infrastructures.

EDITOR GeLaBa generates a complete solution to edit valid documents in `lheo`: an HTML form with embedded JavaScript code that uses AJAX[5] techniques to give flexibility to the user. This form can run on top of a generated server side CGI program which transforms the result into XML or on top of a complete Zope/CMF infrastructure using the generated API (see above) that adds persistent storage for the `lheo` documents.

SERVICES The dictionaries of `lheo` are used to create CGI programs that provide services that can be used with a "RESTful"[6] approach. For instance, consider a dictionary `countries` with two letter keys coding the country and values containing the names of the countries (ISO 3166). With this dictionary, GeLaBa generates a service `entry`. This service can be used with the URL[7]:

```
http://lheo.org/countries/entry?key=FR
```

In this case, the service returns a fragment of GML dictionary corresponding to the key `FR`, with the value in English:

```
<entry><key val="FR"/>
  <value xml:lang="en" val="FRANCE"/>
</entry>
```

This service can also be used with the URL:

```
http://lheo.org/countries/entry?value=ANG
```

In this case, the service returns a fragment of GML dictionary with entries where the value in french contains the string `ANG`:

```
<entries>
  <entry><key val="AO"/>
    <value xml:lang="en" val="ANGOLA"/>
  </entry>
  <entry><key val="AI"/>
    <value xml:lang="en" val="ANGUILLA"/>
  </entry>
  <entry><key val="BD"/>
    <value xml:lang="en" val="BANGLADESH"/>
  </entry>
</entries>
```

Note that the generated editor (previous topic) use these services with the AJAX approach.

VALIDATOR An validator in XSLT is derived from the GML definition of `lheo`. This validator is more powerful and more strict than the generated schemas (see above), and it can be run in a simple web browser as a stylesheet for a `lheo` document. We also generate a web validation service for remote validation. Note that the API described above also gives a validation tool: the specialized parser validates the file it reads.

Conclusion. GeLaBa has been implemented in XSLT and in Python for some parts. An interesting observation is that GML is reflexive, i.e. GML can be defined in GML. It means that all the tools of GeLaBa can be used to manipulate GML definitions of new languages: for instance, the web editor generated from the GML definition in GML can be used to create new languages in GML.
GeLaBa is an open source project under a GPL licence. Its web site is `http://www.gelaba.org`.

Acknowledgments. This work has been funded in part by DGEFP (French Ministry of Employment) for the `lheo` project. The authors would like to thanks DGEFP people (Jean-Michel Mazouth, Alain de Lorgeril, Stéphane Akondé) for their support. The authors would also like to thank Bertrand Blanchard from Items Media Concept for fruitful discussions about GeLaBa.

[4] `http://www.zope.org/Products/CMF`

[5] `http://en.wikipedia.org/wiki/AJAX`

[6] `http://www.xfront.com/REST-Web-Services.html`

[7] The actual service is available at `http://test.lheo.org/1.1/services/pays`.