Journal of Geographic Information System, 2021, 13, 452-465
https://www.scirp.org/journal/jgis
ISSN Online: 2151-1969
ISSN Print: 2151-1950

Creation of a Meta-Model for the Generation of a Webmapping Application

Paul Franck Hery Antenaina[1,2,3]*, Aimé Richard Hajalalaina[1,2,3], Hasina Rakotonirainy[1,2,3], Reziky Zafimarina[4]

[1]School of Management and Technological Innovation, University of Fianarantsoa, Fianarantsoa, Madagascar
[2]Laboratory of Computer Science and Mathematics Applied to Development, University of Fianarantsoa, Fianarantsoa, Madagascar
[3]Computer Science, Geomatics, Mathematics and Application, University of Fianarantsoa, Fianarantsoa, Madagascar
[4]University of Antananarivo, Antananarivo, Madagascar
Email: *heryantenaina@gmail.com

How to cite this paper: Antenaina, P.F.H., Hajalalaina, A.R., Rakotonirainy, H. and Zafimarina, R. (2021) Creation of a Meta-Model for the Generation of a Webmapping Application. *Journal of Geographic Information System*, **13**, 452-465.
https://doi.org/10.4236/jgis.2021.134025

Received: June 21, 2021
Accepted: July 27, 2021
Published: July 30, 2021

Abstract

Our research focuses on creating a meta-model for generating a web mapping application. It was difficult for non-geomatics developers to implement a webmapping application. Indeed, this type of application uses geospatial data that require geomatics skills. For this reason, in order to help non-geomatics developers to set up a webmapping application, we have designed a meta-model that automatically generates a webmapping application using model-driven engineering. The created meta-model is used by non-geomatics developers to explicitly write the concrete syntax specific to the webmapping application using the xtext tool. This concrete syntax is automatically converted into source code using the xtend tool without the intervention of the non-geomatics developers.

Keywords

Webmapping, Model Driven Engineering, Model Driven Architecture, Meta-Model, WebGis, GIS

1. Introduction

The technological evolution always brings more powerful tools especially in the field of information processing and more precisely in the processing of spatio-temporal data. Among the fields of use of the spatial data processing technique, a very elaborate tool lends itself best to use: the geographic information system and more precisely the online mapping or webmapping [1].

Each time a user, more precisely a decision-maker, wished to have spatial information concerning a specific domain, he was obliged to call upon a geomatician, *i.e.* both a developer and a specialist of geographic information systems. Thus, the user, whether he is a computer scientist, a geographer, a topographer, or a geographer, even if he is not a geomatics specialist, needs a tool to develop or automatically generate a webmapping application. Therefore, this research is in continuity with the previous work on the generation of a WebGis platform by Di Martino [2] and Ceri [3]. In other words, we need a unique model to implement to generate any webmapping application. It is in this context that this article applies, we are interested in the creation of a meta-model for the generation of a webmapping application.

For the creation of a meta-model, we think particularly of model-driven engineering, and more precisely of MDA or Model Driven Architecture [4]. Therefore, we will first see the state of the art of model-driven engineering, then the state of the art of webmapping. Then, we will see some experiments that we have conducted using various tools and frameworks, followed by a solution proposal. Finally, we will present the results obtained in the framework of this research.

2. Context and Issues

From the late 20th century until today, it was difficult for a non-geomatics developer or a simple user to set up a domain-specific online mapping application to visualize spatial data online. As soon as a domain needed an application, the decision maker was obliged to hire a geomatics specialist who knew both the geographic information system technology for processing spatial data and ensuring the integrity of the data to be implemented and then he had to know the computer development with many webmapping implementation tools.

As illustrated in **Figure 1**, many specialists had difficulties implementing a webmapping application because it requires knowledge of geographic information systems and a good background in computer science, more specifically in application development.

In these two different contexts, on the one hand, the redevelopment of a webmapping application and on the other hand, the choice of implementation tools, we can formulate the problem: How to model the development of a webmapping application taking into account all the concepts? In other words, how to create a

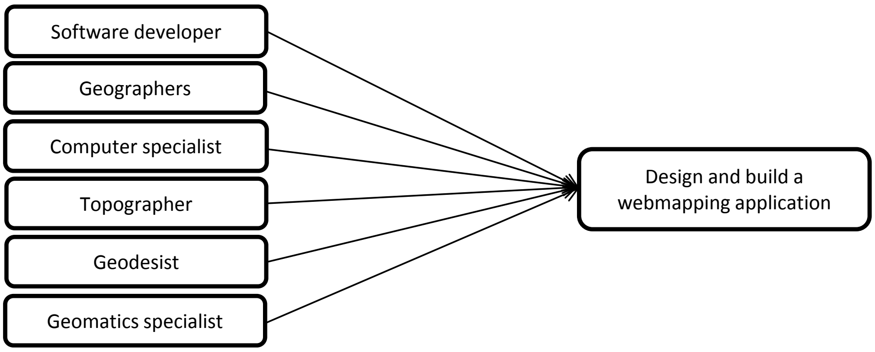

Figure 1. Users of a webmapping application.

dedicated language with a meta-model so that all users can create and generate a webmapping application easily by implementing a model.

In fact, this paper is made to produce a tool for automatic generation of a webmap-ping application by taking into account all the necessary concepts. The result of this research should allow all the users mentioned in **Figure 1** to develop or even generate easily a dynamic mapping application.

3. Methodology

The objective of this document is to set up a meta-model allowing to automatically generate a webmapping application without worrying about development languages and the various existing frameworks. The method used is based on experimentation and model-driven engineering. Our approach starts with some experiments in developing a webmapping application that we have experienced. From these experiments, we will go back and generalize the development of these applications and identify the concepts needed to create our meta-model. But before we start these experiments, we will see what a webmapping is.

3.1. Webmapping

Webmapping is a generic term defining both the process of distributing maps via a network such as the Internet or Intranet and viewing them in a browser. In other words, we can call it a web GIS.

Cartography refers to the creation and study of maps. It mobilizes a set of techniques used to produce maps [5]. Cartography constitutes one of the privileged means for the analysis and the communication in geography. It is used to better understand space, territories and landscapes. It is also used in related sciences, demography and economics in order to propose a spatialized reading of phenomena.

Online cartography meets real needs for rapid dissemination of information and remote updating of data [6]. Although the cartographic result allows to facilitate the understanding of the surrounding space, the implementation of such platforms requires transversal skills in both computer science and geography. **Figure 2** shows the basic principle of internet mapping.

Architecture of the Webmapping Application

The architecture of the webmapping application is almost similar to the architecture of a web application. Apart from the web server, data server, and the client the only difference is the presence of the map server along with the wms, wfs and wcs connectors as shown in **Figure 3**.

3.2. Experimentations

Our approach is based on the experience we had on the development of webmapping applications, that's why we will present some specific experiments on the development of webmapping applications.

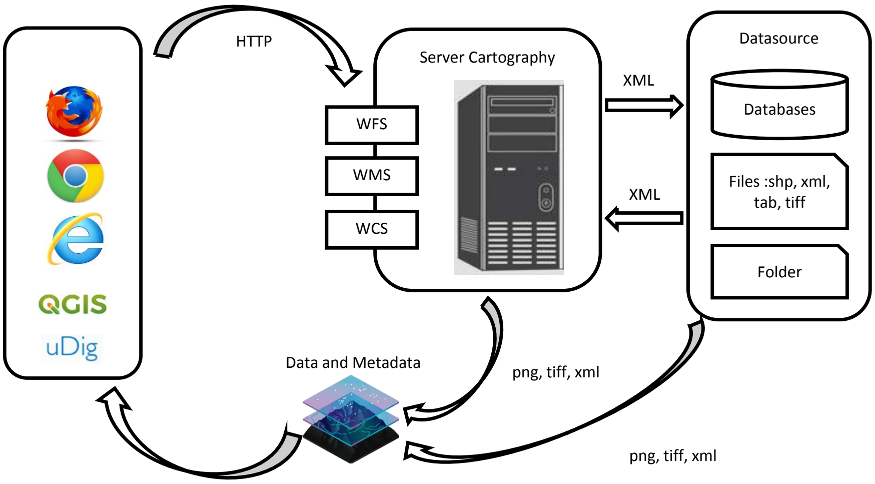

Figure 2. Principle of Internet mapping.

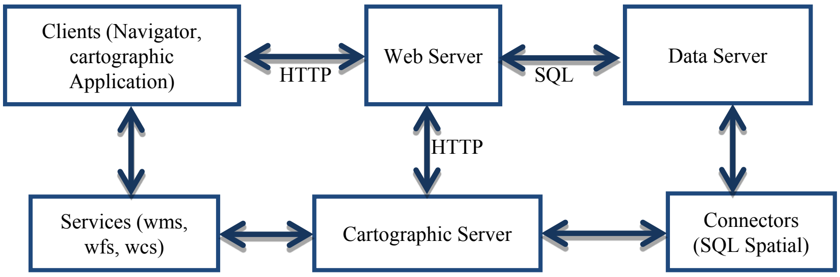

Figure 3. Architecture of the webmapping application.

3.2.1. ROAD-M Webmapping

This online mapping application system was set up to manage and represent road and structure data after a field survey, and also to analyze this mapping data to better facilitate decision making. Thus, this mapping project shown in **Figure 4** was born to contribute to the processing of road networks in Madagascar [7].

3.2.2. Webmapping Cenit

Cenit webmapping is an application that was used to process election results in Madagascar through the webmapping decision support system. This work has resulted in an application that allows the decision to collect election results in polling stations in relation to the District Electoral Center. **Figure 5** represents the home page of the cenit webmapping application.

3.2.3. Aerial Thermography

The aerial thermography in **Figure 6** was a diagnostic tool that allows to visualize the heat loss through the roofs. The objective is to monitor energy losses through roofs in order to raise awareness among individuals, businesses and condominiums of the urgent need to insulate their homes or buildings.

In principle, to develop a webmapping application, the application files are all provided with all the modules. The work of the developers is therefore to set up and configure the three main files, namely: Mapfile, Mapbook and framework.

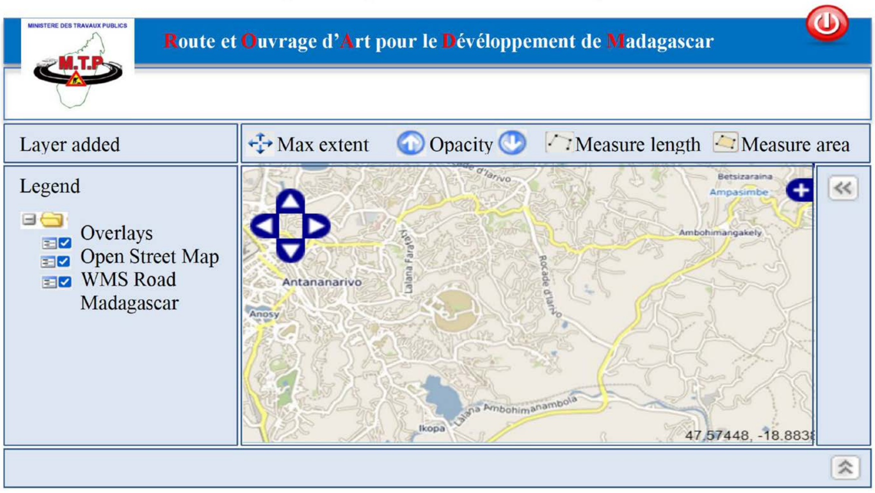

Figure 4. ROAD-M webmapping home page.

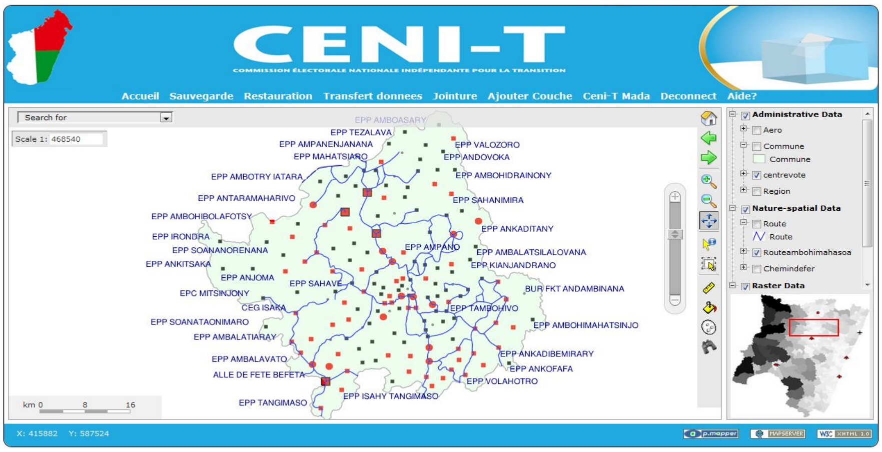

Figure 5. Webmapping CENI-T, case District Ambohimahasoa Madagascar.

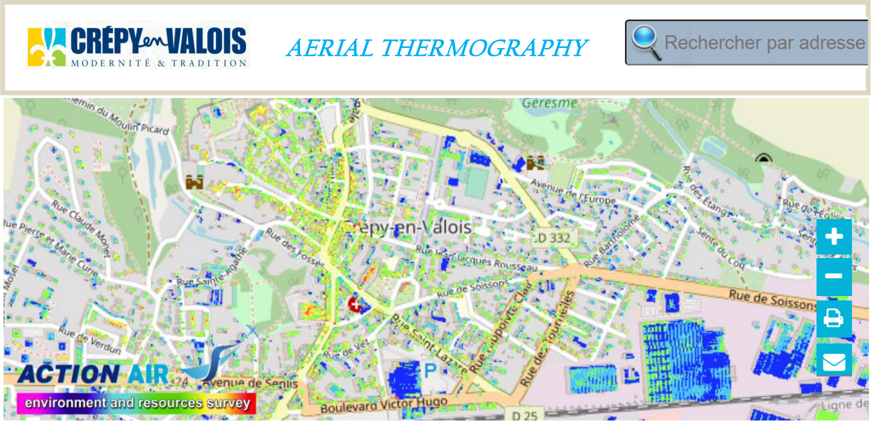

Figure 6. Aerial thermography of Crepy en Valois.

The mapfile is the heart of the mapping application, it defines the relationships between objects, tells the server where the data is located and defines how things should be drawn [8].

The mapbook document is an XML file that is used as a configuration file for an application. It is used to configure items such as map sources, layers, services, and tools for the application.

The framework file is used to configure the installed modules, languages, methods and interface components.

We should therefore model each of these 3 concepts. To do this, we will use model-driven engineering [9], more precisely the model-driven architecture approach [4].

3.3. Model Driven Engineering

Model Driven Engineering proposes to model applications at a high level of abstraction and places the model at the heart of the design process then automatically generates the application code from the models.

In continuity of the principle in the 80's which is "everything is object", the creation of software is moving towards model driven engineering [8] and the principle of "everything is model". This new approach can be considered both in continuity and in rupture with previous works [9] [10].

Currently, the most used model driven engineering approaches are MDA [11], EMP [12], Software factorial [13], AToM3 [14], KM3 [15] and Kermeta [16]. Among these IDM approaches, we will take a closer look at the MDA approach [17].

MDA was born because of the increasing complexity of information systems and the high costs of technological migration. One of the major initiatives through which the Object Management Group accomplishes this goal is the promotion of Model Driven Architecture (MDA) as an architectural framework for software development. This framework is built around a number of detailed OMG specifications, which are widely used by the development community [4].

In 2001, the OMG adopted Model Driven Architecture as an approach for using models in software development. Its three main goals are portability, interoperability and reusability through architectural separation of concerns. One of the fundamental aspects of MDA is its ability to address the entire development life cycle, covering analysis and design, programming, testing, component assembly, and deployment and maintenance [18] [19].

The OMG defines four standard models in its guide on MDA as shown in **Figure 7** [20]:

- CIM: CIM is the abbreviation for Computation Independent Model;
- PIM: The term PIM is the acronym of Platform Independent Model;
- PSM: PSM stands for Platform Specific Model;
- PDM: The acronym PDM stands for Platform Description Model.

4. Modeling and Results

To reach our goal, we will first model the concepts necessary to generate a web-mapping application and then we will propose the final meta-model.

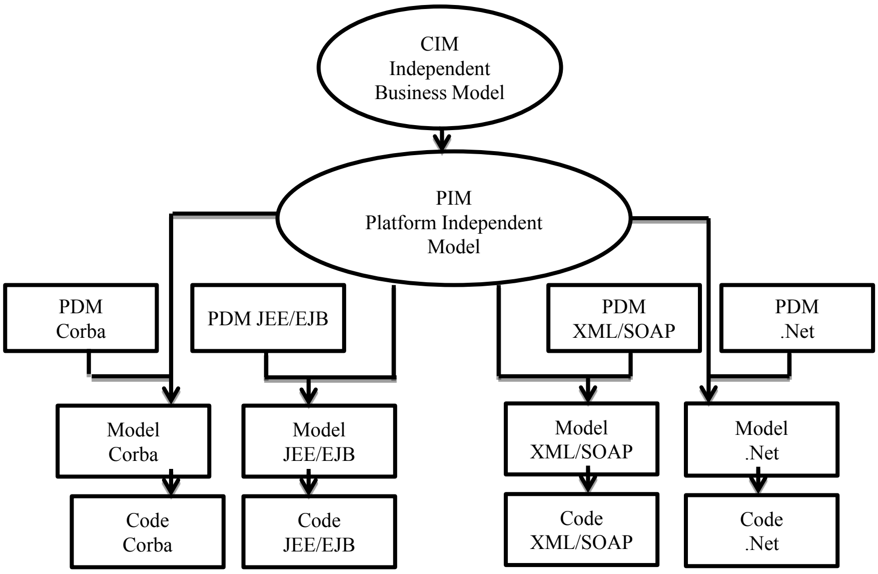

Figure 7. Example of using MDA models to build an application [12].

4.1. Modeling of the Three Concepts Based on Mda

Based on the MDA approach, the three concepts to be presented are:

- A meta-model for the generation of a mapbook file (Figure 8);
- A meta-model for the generation of a mapfile (Figure 9);
- A meta-model for the generation of a framework file (Figure 10);

The relationship between these metamodels is when we generate a mapfile, we have to generate a mapbook file to read this file, and the generation of a framework file is independent of the generation of these two files but we have to make the content of the framework file synchronous with the wms data of a server.

The proposal is thus represented by Figure 11.

The first one (1) corresponds to the meta-model we have set up for the generation of the mapbook file (Figure 8), then the second one (2) is the meta-model for the generation of the mapfile files (Figure 9) and finally the third one (3) is necessary for the generation of the framework and configuration files (Figure 10). They are linked together by the layers.

4.2. Application Example

The created meta-model is used to explicitly write the concrete syntax specific to the webmapping application using the xtext tool:

```
Model:
   (appli+=Appli)?;

Appli:
{Appli} 'Generate' (mapbook+=Mapbook)?
        (mapafile+=Mapfile)?
        (geoext+=Geoext)?
;
```

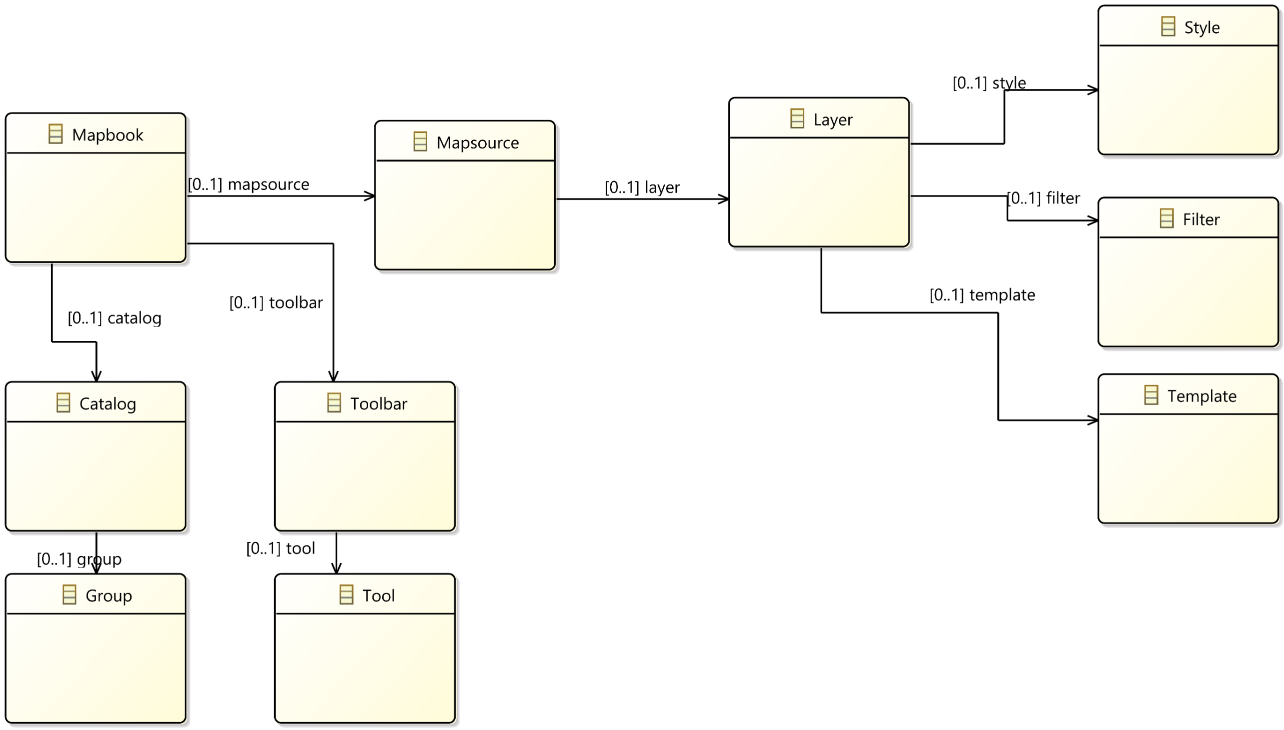

Figure 8. Mapbook concept.

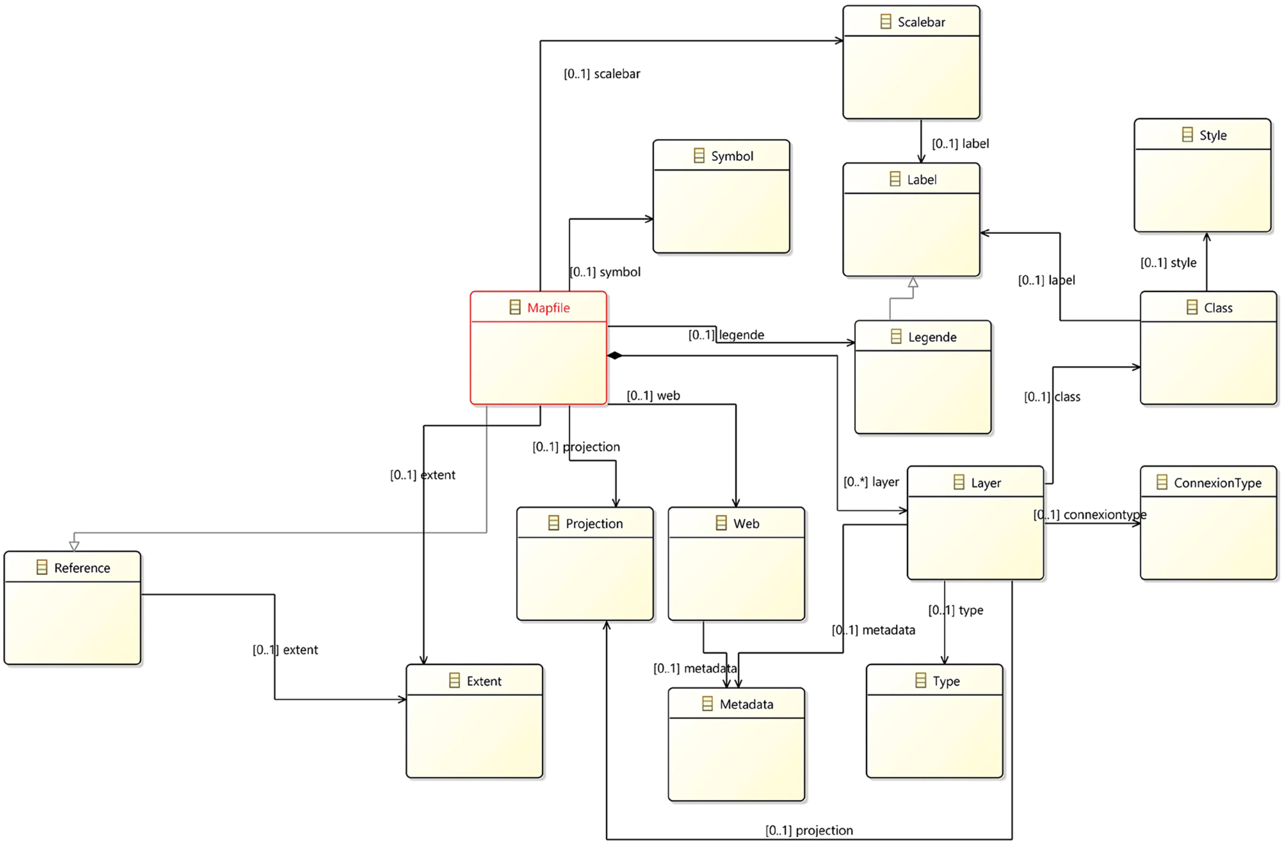

Figure 9. Mapfile concept.

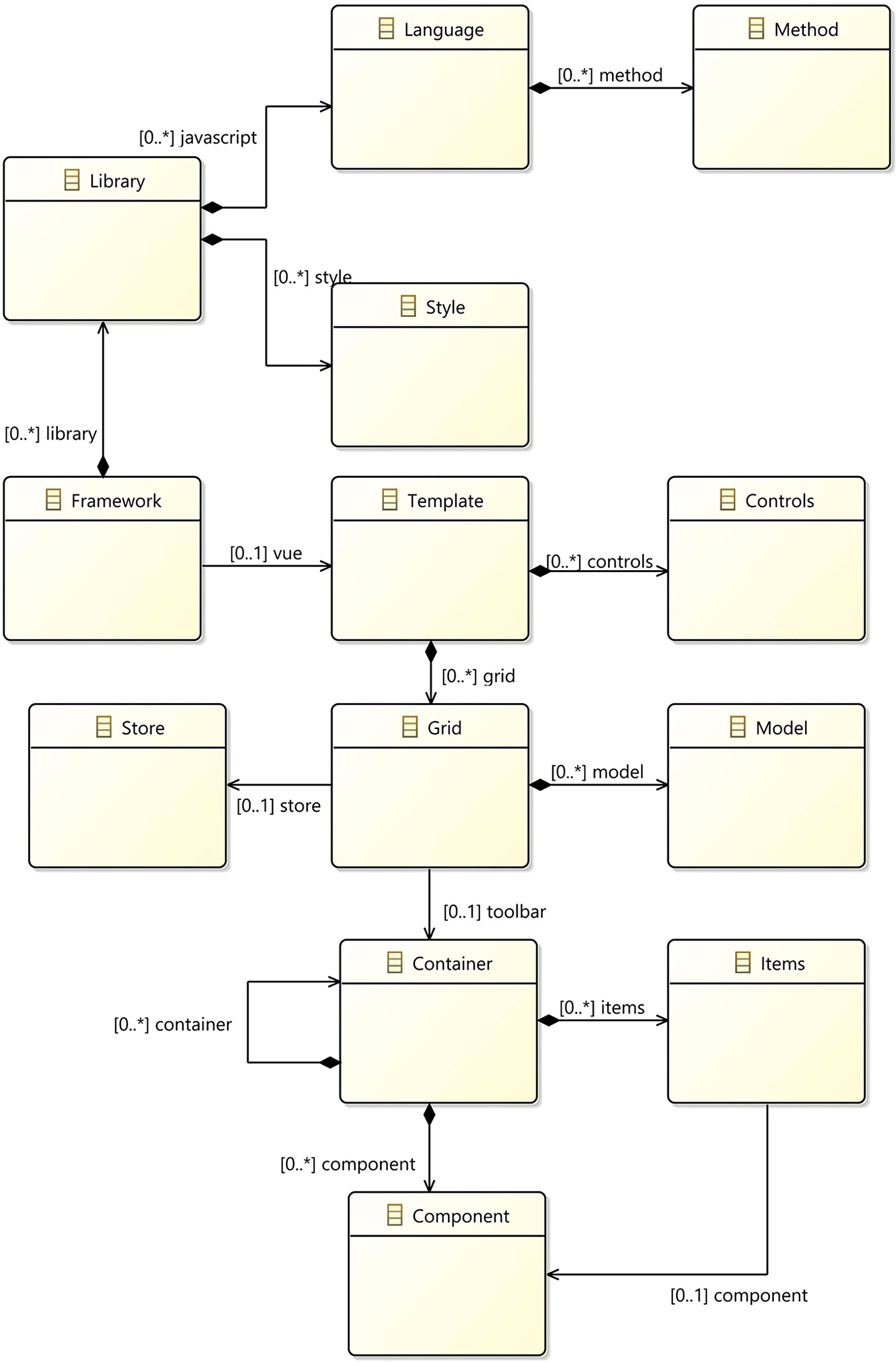

Figure 10. Concept of configuration framework.

```
Mapbook:
  'MapBook' (nom=STRING) '{'
        (mapsource+=Mapsource)*
        (catalog+=Catalog)*
        (toolbar+=Toolbar)* '}'
;
```

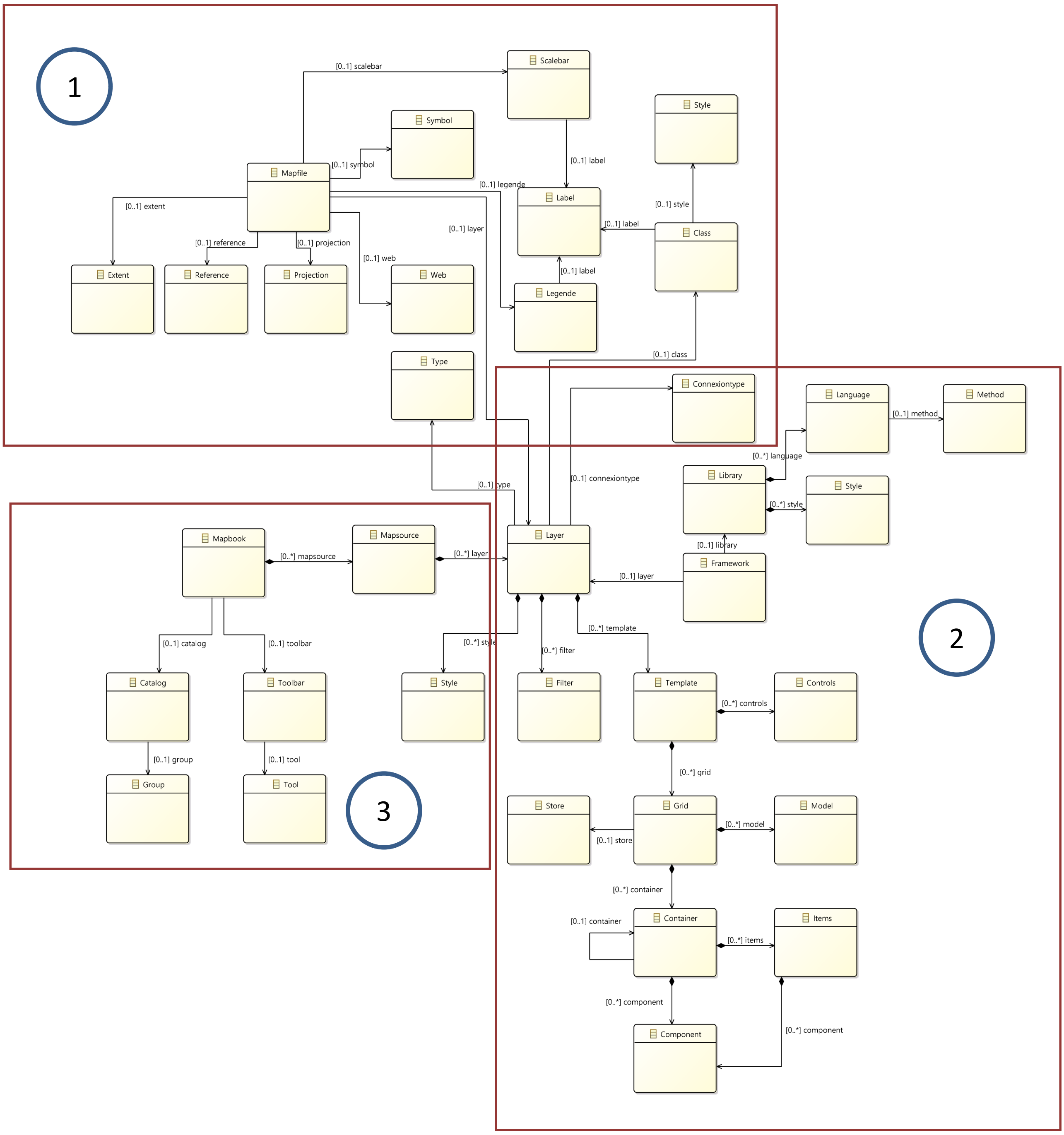

Figure 11. Proposed meta-model.

```
Mapsource:
   'Mapsource'(nom=STRING) '{'
   ('title' title=STRING)?
   ('type' type=STRING)?
   ('file' file=STRING)? ';'
   (param+=Param)*
   (layer+=Layer)* '}'
```

```
;
```

This concrete syntax is automatically converted into source code using the xtend:

```
class MyDslGenerator extends AbstractGenerator {
  override void doGenerate(Resource resource, IFileSystemAccess2 fsa, IGeneratorContext context) {
       for(Mapbook
m:resource.allContents.toIterable.filter(typeof(Mapbook))){
            fsa.generateFile("mapbook.xml",
            ''' <?xml version="1.0"?>
  <mapbook version="3.0">'''+
                 m.bouclMapsource
            )
       }
  }
  def bouclMapsource (Mapbook ms)
  '''
  «FOR cible: ms.mapsource »
………………..
}
```

The XTend generator automatically generates the configuration files for our webmapping application. This test was done on the Road-m webmapping application (experimentation 3.2.1) and we were able to generate all the necessary files for this application. This is an excerpt from the generated source code.

```
<?xml version="1.0"?>
<mapbook version="3.0">
     <map-source name="openstreetmap" type="xyz">
          <layer name="osm_mapnik">
               <legend type="html"><![CDATA[
               &copy; <a href="http://www.openstreetmap.org/copyright"
target="_blank">OpenStreetMap</a> Contributors.
               <br><br>
               ]]></legend>
          </layer>
          <url>https://a.tile.openstreetmap.org/{z}/{x}/{y}.png</url>
          <param name="cross-origin" value="anonymous"/>
     </map-source>
</mapbook>
```

4.3. The Interests of Our Proposal

Firstly, the interest of using our proposal compared to manual developments is the speeds to develop a webmapping application because we just need to build our model to generate all the necessary files with all the parameters in the application, we no longer find difficulties on the implementation of several files and the link between them.

Secondly, the biggest advantage of our proposal is that all users, *i.e.* developers, computer scientists, geomaticians, topographers, simple users who do not know how to develop an application can create their own application in their specific field with a simple click or just by moving graphic palettes. After writing the concrete syntax, our template generates all the files and you just need to launch the url of the application.

Thirdly, all those users we mentioned in the previous paragraph don't need to know the relationship between all the implementation tools anymore because everything is generated by our model.

5. Conclusions and Perspectives

To conclude, it is difficult to create a webmapping application either for a user or even a developer, hence the need to create a meta-model for the generation of a webmapping application. To achieve this, we explored model-driven engineering, more precisely the Model Driven Architecture approach, and then we clarified what we need to know about mapping and webmapping.

The experiences and concepts we presented allowed us to go back and generally see the principle of developing a webmapping application in order to propose a meta-model. As reverse engineering is manual to generate specific files, we have created a concrete syntax based on the proposed meta-model using xtext and then for the transformation we have used xtend in order to generate automatically the source codes. We had tested this meta-model with the concrete syntax and the automatic generation xtend and we succeeded in reproducing the application "ROAD-M Webmapping" and the application "WEBMAPPING CENIT".

Although this meta-model is done, these languages are still being validated, being improved and must follow several tests in order to have a "generic and intuitive" language.

Once the optimized meta-model is validated and the created language meets the needs of the users, we will deploy it and put it in service in order to be used by the target audience. Our goal is to have a textual and graphical language usable by all users to generate a webmapping application in a specific domain by implementing the architecture according to their needs and with tools they want to implement using a unique meta-model.

In this research we have proposed a meta-model for the generation of a webmapping application. And currently, we intend to link the models with free software like QGIS, Grass Gis, Udig to facilitate the processing of data to be implemented with the generated application. We can also link this model with a mobile geolocation platform.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Baldé, I. (2008) Mise en place d'une plateforme de cartographie dynamique, Mémoire ingénieur de conception en Génie Informatique.

[2] Di Martino, S., Ferrucci, F., McArdle, G. and Petillo, G. (2009) Automatic Generation of an Adaptive WebGIS. *International Symposium on Web and Wireless Geographical Information Systems*, Maynooth, 7-8 December 2009, 171-186. https://doi.org/10.1007/978-3-642-10601-9_12

[3] Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S. and Matera, M. (2002) Designing Data-Intensive Web Applications. Morgan-Kaufmann Publishers, Burlington.

[4] Moghrabi, X. (2003) L'approche Model-Driven Architecture, crédible pour développer un progiciel de gestion intégré, école doctorale systems.

[5] Coutard, M., Klipfel, J.-P. and Blanc, S. (2005) La cartographie SIG en ligne ou Webmapping: Les outils libres État des lieux des solutions applicatives autour de MapServer. http://mappemonde.mgm.fr

[6] Mappemonde Portails d'information. https://journals.openedition.org/mappemonde/

[7] Antenaina, H. (2016) Mise en place d'une application cartographique sur web baptisee road-m webmapping, au ministère des travaux publics, antananarivo, Madagascar April.

[8] Lime, S. and McKenna, J. (2021) The Mapfile in Mapserver. https://www.mapserver.org/mapfile

[9] Bézivin, J. (2004) Sur les principes de base de l'ingénierie des modèles. *L'Objet*, **10**, 145-157. https://doi.org/10.3166/objet.10.4.145-157

[10] Bézivin, J. (2005) On the Unification Power of Models. *Software and System Modeling*, **4**, 171-188. https://doi.org/10.1007/s10270-005-0079-0

[11] Kleppe, A., Warmer, J. and Bast, W. (2003) MDA Explained: The Model Driven Architecture Practice and Promise. Addison Wesley Professional, London.

[12] Gronback, R.C. (2009) Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit. Pearson Education, London.

[13] Greenfield, J. (2004) Software Factories Assembling Applications with Patterns, Models, Frameworks and Tools. *International Conference on Software Product Lines*, Boston, 30 August-2 September 2004, 304. https://doi.org/10.1007/978-3-540-28630-1_19

[14] Vangheluwe, H. and De Lara, J. (2002) AToM3: A Tool for Multi-Formalism Modelling and Meta-Modelling. *European Conferences on Theory and Practice of Software Engineering ETAPS'*02, Grenoble, 8-12 April 2002, 174-188. https://doi.org/10.1007/3-540-45923-5_12

[15] Bézivin, J. and Jouault, F. (2003) KM3: A DSL for Metamodel Specification.

[16] Muller, P.-A. (2006) De la modélisation objet des logiciels à la metamodélisation des langages informatiques.

[17] Minsky, M. (1968) Matter, Mind, and Models. In: Minsky, M., Ed., *Semantic Information Processing*, MIT Press, Cambridge, 425-432.

[18] Bezivin, J. and Blanc, X. (2002) MDA vers un nouveau paradigme (1). *Développeur Référence*, **2**, 7-11. http://mfworld42.free.fr/cnam/nfe115-informatique_decisionnelle/MDA.Partie1.JBXB.Last.prn.pdf

[19] Bezivin, J. (2002) Les nouveaux défis des systèmes complexes et la réponse MDA de

l'OMG. https://dblp.org/db/conf/jfsma/jfsma2002.html#Bezivin02

[20] OMG (2001) Model-Driven Architecture, a Technical Perspective. https://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf