Integrating catalog and GIS tools: access to resources from *CatMDEdit* thanks to *gvSIG* ⋆

Javier Nogueras Iso[1], Jesús Barrera[1], Fernando Gracia Crespo[2] , Silvia Laiglesia[2], Pedro R. Muro Medrano[1]

[1] Computer Science and Systems Engineering Dept.,University of Zaragoza
C/María de Luna, 1, 50018 - Zaragoza (Spain)
{jnog,jesusb,prmuro}@unizar.es
[2] GeoSpatiumLab S.L.
C/Carlos Marx, 4, local der., 50015 - Zaragoza (Spain)
{fgcrespo,silvialm}@geoslab.com

Abstract. *CatMDEdit* is a metadata editor tool that facilitates the management of resources through their metadata, paying special attention to geographic information resources. The objective is to promote the metadata creation as an instrument that facilitates the processing of data in an effective way. Therefore, important efforts have been made towards the integration of *CatMDEdit* with GIS tools. This work presents the mechanism for connecting *CatMDEdit* with *gvSIG* for the manipulation of geographic information. Once the metadata of a resource are generated (by hand or via existent semi-automatic mechanisms for some GI specific formats), *CatMDEdit* autogenerates a *gvSIG* project and invokes *gvSIG* to have access to the resource with the adequate parameters.

Key words: metadata, Spatial Data Infrastructures, GIS, resource management

1 Introduction

According to the Global Spatial Data Infrastructure Association Cookbook (Nebert, 2004), "the term Spatial Data Infrastructure is often used to denote the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data". SDIs provide the framework for the optimization of the creation, maintenance and distribution of geographical information at different organization levels (e.g., regional, national, or global level) and involving both public and private institutions.

One of the key elements for the success of these information infrastructures facilitating the distribution and access to resources is the adequate documentation

⋆ This work has been partially supported by: the Spanish Ministry of Education and Science through the project TIN2007-65341 from the National Plan for Scientific Research, Development and Technology Innovation; the University of Zaragoza though the project UZ2007-Ref. 200700024; and GeoSpatiumLab S.L.. The work of Silvia Laiglesia has been partially supported by the Spanish Science and Technology Ministry (Torres Quevedo program ref. PTQ06-2_0798).

of these resources. Metadata constitute the mechanism to characterize data and services in order to enable other users and applications to make use of such data and services. They offer description of the content, quality and any other characteristics of a resource. Knowing details such as the coordinate reference system, the spatial representation model (e.g., raster, vector, object oriented) or the spatial resolution is a basic prerequisite for viewing and processing data. Thus, organizations and standardizations bodies such as FGDC (Federal Geographic Data Committee)[3], OGC (Open Geospatial Consortium)[4] or ISO/TC211 (ISO technical committee for goegraphic information and geomatics)[5] have focused on standardising metadata vocabularies (e.g., "Content Standard for Digital Geospatial Metadata - CSDGM" or "ISO 19115 - Geographic Information Metadata").

This work analyzes the role of *CatMDEdit* [6] as a tool to facilitate the management of geographic information resources through standardized metadata (Zarazaga-Soria et al., 2003). *CatMDEdit* is an Open Source cross-platform desktop tool, developed in Java and internationalized to six different languages, whose main aim is to promote metadata creation as a mechanism that facilitates the processing of data in a more effective way. Data and metadata management cannot be understood as separate tasks. Thus, *CatMDEdit* facilitates semi-automatic mechanisms for the generation of metadata in order to minimize the interaction with users and hide the complexity of metadata standards. Additionally, it provides alternative mechanisms for integrating Geographic Information System (GIS) tools that enable the access to datasets through their associated metadata. In the same way as other content management tools do, metadata are the means that allow users to describe a resource and launch associated tools for its visualization or modification.

Making emphasis on the idea of unifying data and metadata management, this paper presents the mechanism for connecting *CatMDEdit* with the open source tool *gvSIG*[7] (Alfaro, 2005b, Alfaro, 2005a) for the manipulation of geographic information (GI). Once metadata of a resource are generated (manually or via existent semi-automatic mechanism for some GI specific formats), *CatMDEdit* autogenerates the configuration file of a *gvSIG* project and directly opens this tool to have access to the resource with the adequate parameters.

The rest of the paper is structured as follows. Section 2 describes the general architecture of the tool. Section 3 describes the mechanisms used in *CatMDEdit* for integrating GIS tools, being *gvSIG* one of the most powerful GIS tools currently integrated. Section 4 reviews the related work in metadata editors with advanced GI capabilities. Finally, this papers ends with some conclusions and some discussions about future work.

[3] http://www.fgdc.gov/
[4] http://www.opengeospatial.org/
[5] http://www.isotc211.org/
[6] http://catmdedit.sourceforge.net
[7] http://www.gvsig.gva.es/

2 Architecture of *CatMDEdit*

Before describing the architecture of *CatMDEdit*, it is important to understand two core concepts managed by this application: resources and repositories. A resource is the basic information unit managed by the tool and it represents a dataset and its associated metadata. On the other hand, resources are grouped into repositories. These repositories represent a storage source (folder, databases,), where digital datasets and metadata are stored jointly.

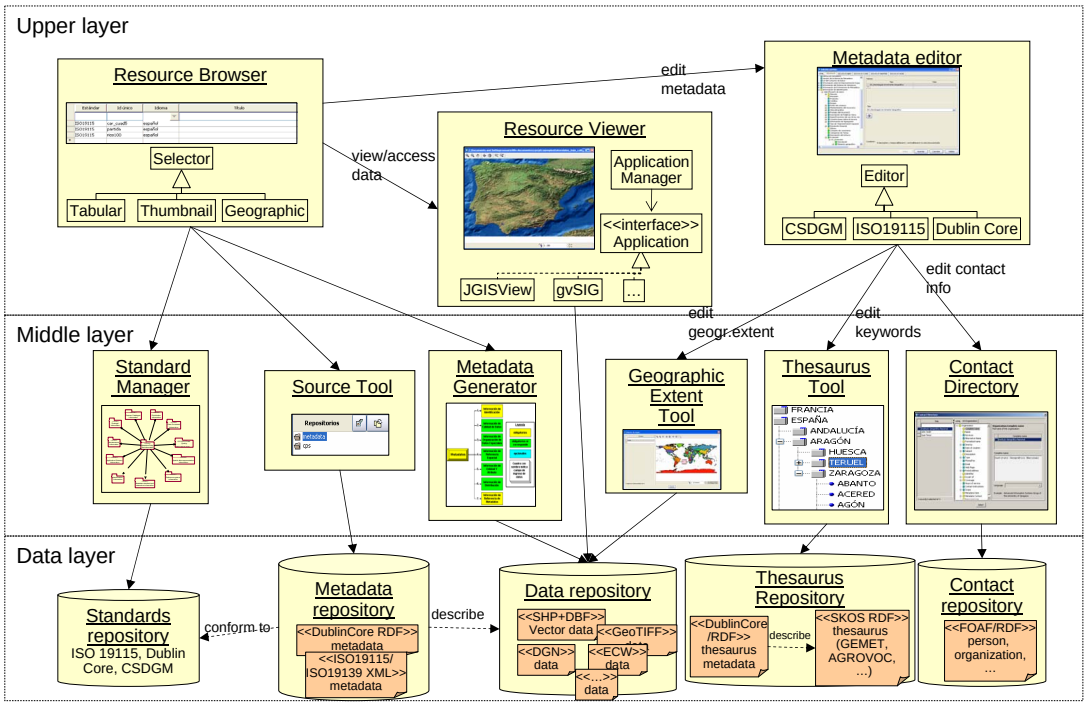

Fig. 1. High level architecture of *CatMDEdit* components

As shown in figure 1, *CatMDEdit* has been designed using a three-layered architecture. The upper-layer contains the three main functional components of the application: a *Resource Browser*, a *Metadata Editor*, and a *Resource Viewer*. The second layer contains the middleware software libraries that support the development of the functional components. And the data layer includes the different repositories needed for the configuration of the tool and the storage of data and metadata, which are managed or produced by the application. We will describe the specific components in each layer in parallel with its interaction with the three main functional components of the upper layer.

Resource Browser is the main functional component within the upper layer. It is in charge of browsing the resources organized in different repositories, and

invoking the rest of components of the application. For the browsing of resources in a repository, different selection mechanisms are allowed: tabular display of resources, each row shows the main features of a resource; selection by means of thumbnails representing the contents of the resource; or graphical display of resources on a map where each resource is represented by its geographic extent. Within the upper layer, this component interacts with the *Metadata Editor* for the edition of metadata describing a resource, and with the *Resource Viewer* for connecting with applications that enable the access and modification of data (e.g. GIS tools).

With respect to the components used by the *Resource Browser* in the middle layer, it interacts with the following components: *Source Tool*, *Standard Manager* and M*etadata Generator*. The *Source Tool* component mediates the interaction between the *Resource Browser* and the *Data Repository*, facilitating services for the management, query and presentation of metadata items. The *Standard Manager* provides the knowledge of the metadata standards understood by the tool. The configuration of metadata is stored at the *Standards Repository*. Finally, the *Metadata Generator* component facilitates methods for the automatic generation of metadata based on the analysis of some specific GI formats.

The *Metadata Editor* component is the component in charge of editing the metadata records according to different standards. It facilitates the dynamic configuration of metadata edition forms on the basis of the metadata standard required for each metadata record. For the specific edition of some metadata elements, this functional component invokes three different components in the middle layer: the *Geographic Extent Tool*, the *Thesaurus Tool* and the *Contact Directory*. The *Geographic Extent Tool* allows the graphical definition of the geographic extent according to different types of geometries (rectangles, polygons). The user may select geometric objects from a vector data layer or define these objects herself. The *Thesaurus Tool* is invoked whenever the user needs to select the values of a metadata element from a controlled vocabulary such as a codelist, a taxonomy, or a thesaurus. The technology of this component is based on the *ThManager* tool (Lacasta et al., 2007), an open source tool for creating and visualizing SKOS RDF vocabularies, a W3C initiative for the representation of knowledge organization systems such as thesauri, classification schemes, subject heading lists, taxonomies, and other types of controlled vocabulary.

The *Resource Viewer* is the functional component in charge of integrating data management tools for the access and further processing of data. This component consists in a data application manager that analyzes the type of resource and establishes the connection with the application that is able to access the contents of each resource. Nowadays, in the case of GI resources, this application manager facilitates the connection with a GIS library called *JGISView* (Béjar et al., 2001) and *gvSIG*. This component is explained in more detail in the following section.

3 Access to resources through *gvSIG*

The *Resource Viewer* functional component includes an *Application Manager* that allows identifying the most appropriate application for accessing and modifying the

contents of each resource type. Figure 2 shows a class diagram with the design of this *Application Manager*. As it can be observed, this *Application Manager* follows the *Abstract Factory* design pattern (Gamma et al., 1996) and depending on the resource type, it provides a component that implements the *IF_DataApplicationModule* interface. This interface defines the methods needed at an abstract level to interact with the expert application enabling the access to the resource contents.

CatMDEdit provides implementations of the *IF_DataApplicationModule* interface for different types of resources. The *MapModule* provides an implementation that links *CatMDEdit* with expert applications in GI resources. But there are implementations for other more general media types such as images (implemented by *ImageModule*), MP3 (implemented by *MP3Module*), PDF (implemented by *PDFModule*) or office documents (implemented by *OfficeModule* and *OpenOfficeModule*).

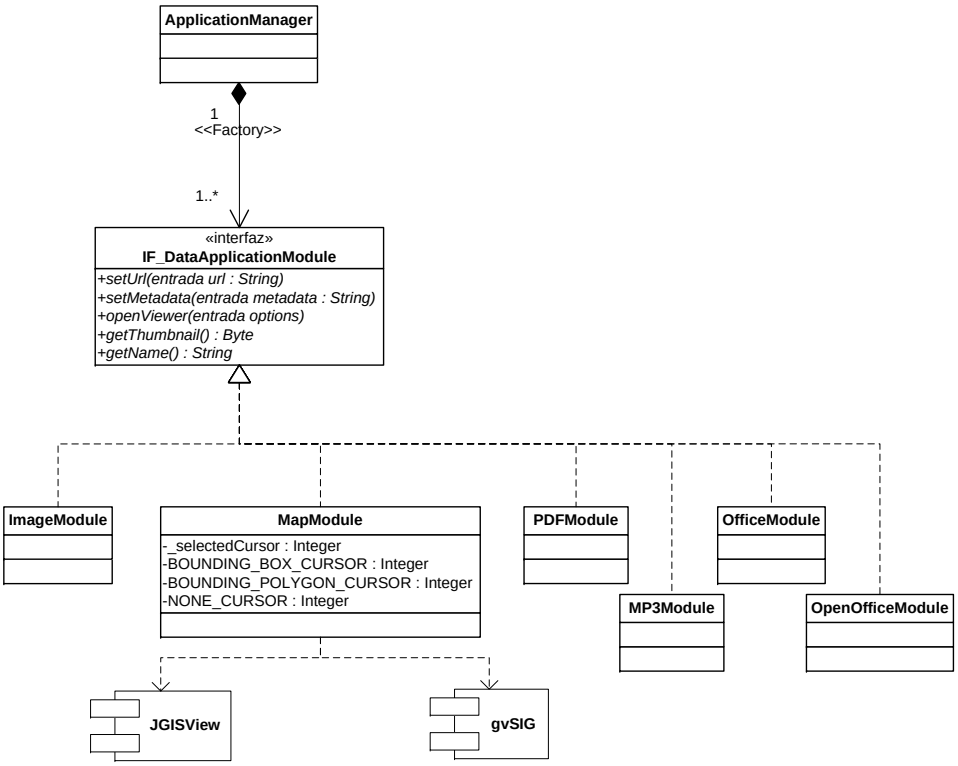

Fig. 2. Use of the factory pattern to implement the Application Manager

In the case of GI resources, the *MapModule* of the new version of *CatMDEdit* to be released in 2009 will integrate the *JGISView* library and the connection to *gvSIG*.

JGISView is a software library developed by the Advanced Information Systems Group of the University of Zaragoza that complies with the JavaBeans specification, a Java specification for building plug-in components (Béjar et al., 2001). This library

provides viewing components for different GI formats that can be easily integrated in Java applications. *JGISView* is oriented towards applications that need to display GI in the form of multi-layer maps, providing capabilities for browsing and display of associated attribute information. In fact, this library is used to generate the thumbnails view shown within the *Resource Browser* or to define the geometries that correspond with the geographic extent of a resource in the *Geographic Extent Tool* used by the *Metadata Editor*. However, although *JGISView* is a lightweight library that can be easily integrated, it is not oriented towards the modification of data and it does not give support for some specific GI formats.

Therefore, the new version of *CatMDEdit* aims at integrating more powerful GIS tools, which are able to facilitate the modification and exchange of data in different formats and spatial representation models. An example of this integration with more powerful GIS tools is the integration with *gvSIG* (Alfaro, 2005b, Alfaro, 2005a). *gvSIG* is an Open Source GIS tool that arose in 2004 as a GIS client tool to cover the needs of GI users working at Generalitat Valenciana (government of the autonomous community of Valencia, Spain). Since then, it has evolved into a cross-platform desktop tool accessible by a worldwide audience. It provides a quick access to the most usual raster and vector formats. The storage devices for GI formats may be accessed either locally or remotely through standardized interfaces such as the Web Map Service (WMS), Web Coverage Service (WFS) or Web Feature Service (WFS) specifications of the Open Geospatial Consortium (OGC). Among its main features, it includes the following ones: viewing functionalities (zoom, pan, selection of features according to attribute information,); applicability of different types of symbology and legends; different import and export capabilities (e.g., export of maps as PDF, Postscript, or image formats); geoprocessing operations (clip, dissolve, merge, intersection, spatial join, convex hull,); management of different spatial reference systems; georeferencing of coverage data; or field calculators to perform different types of calculations on the attribute information.

As a first approach to establish the link between *CatMDEdit* and *gvSIG*, the *MapModule* component launches *gvSIG* as an external application accessible through the runtime environment. As both tools have been developed using the same programming language (Java), the ideal integration should have been by means of application plug-ins. However, because the two tools have used different application frameworks for the development of functional components, we decided at a first stage to test the integration using a lightweight connection.

With independence of the type of connection used (lightweight or tight), the main point for accessing a resource from *gvSIG* is to generate the configuration file of a *gvSIG* project with the adequate parameters. In *gvSIG* the access to resources is managed through projects, where a project represents the specific settings of the digital map that the user wants to produce (e.g., a map composed of several vector and raster layers with a specific set of legends and symbology). Therefore, the main task performed by the *MapModule* is to derive the settings of a *gvSIG* project from the contents of the metadata describing a resource. That is to say, the *MapModule* will generate an XML file in compliance with the syntax established by the *gvSIG* tool. Then, *CatMDEdit* will transfer this file (whose name uses the extension *.gvp*)

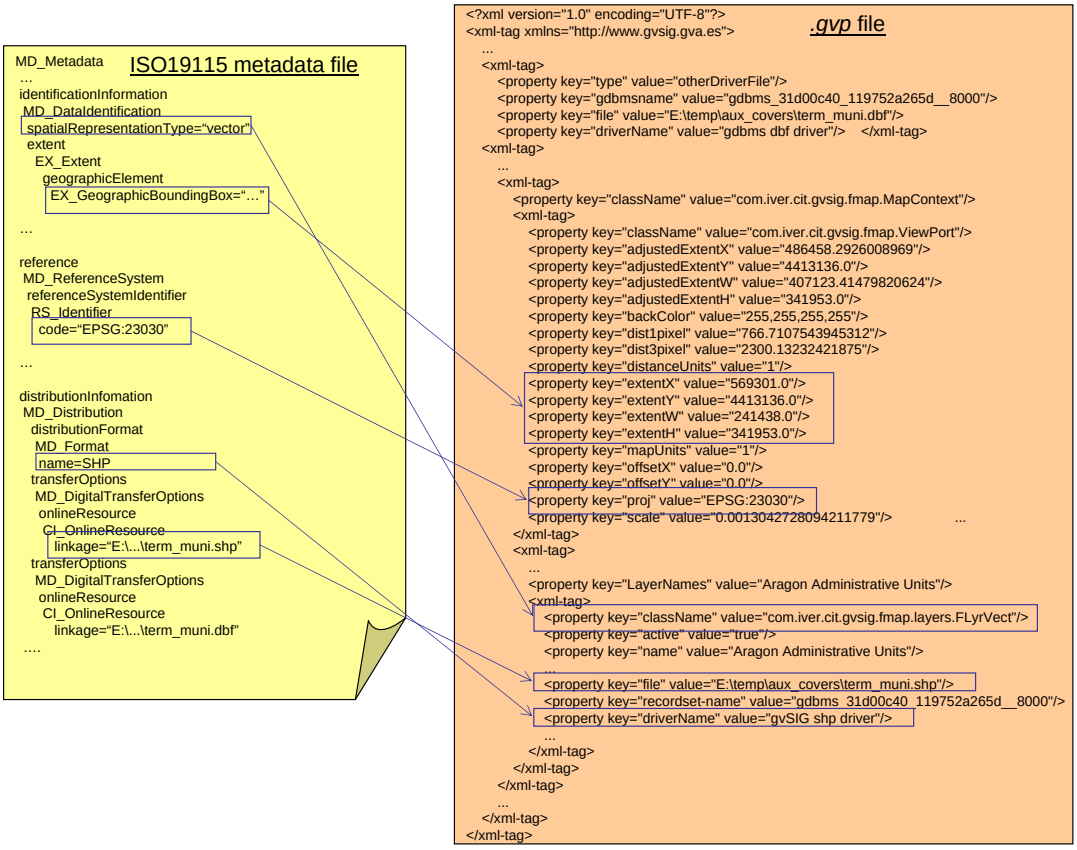

Fig. 3. An example of a *.gvp* project generated by *CatMDEdit*

to the *gvSIG* tool. And in the last term, *gvSIG* will read the features of this project to open the resource with the correct viewing parameters (e.g., coordinate reference system, geographic extent, ...).

Thanks to the metadata describing each resource, the *MapModule* can identify: the storage device (e.g., the path of a file, or the URL of a database accessible through JDBC), the encoding format, the spatial reference system (e.g., the specific geographic or projected coordinate system); and some viewing parameters (e.g., the geographic extent). Upon this information, *MapModule* will generate a *.gvp* project with all the details about the layers to be included and their viewing parameters. Figure 3 sketches the procedure used to extract information from an ISO19115 metadata file describing a GI resource encoded in Shapefile format, and how this information is used to complete the information required within a *.gvp* file.

Finally, figure 4 shows the integration with *gvSIG* from the graphic user interface perspective. Once the user has selected a resource through the *Resource Browser* window (see window on the top left corner), the menu option *"File: Open Data"* (or its shortcut button at the toolbar) will open a dialog where the user will confirm whether he is interested in launching the *gvSIG* tool. Then, *CatMDEdit* will automatically generate the configuration file of a *gvSIG* project and will launch the tool as shown in the window on the right bottom corner of figure 4. In case of launching

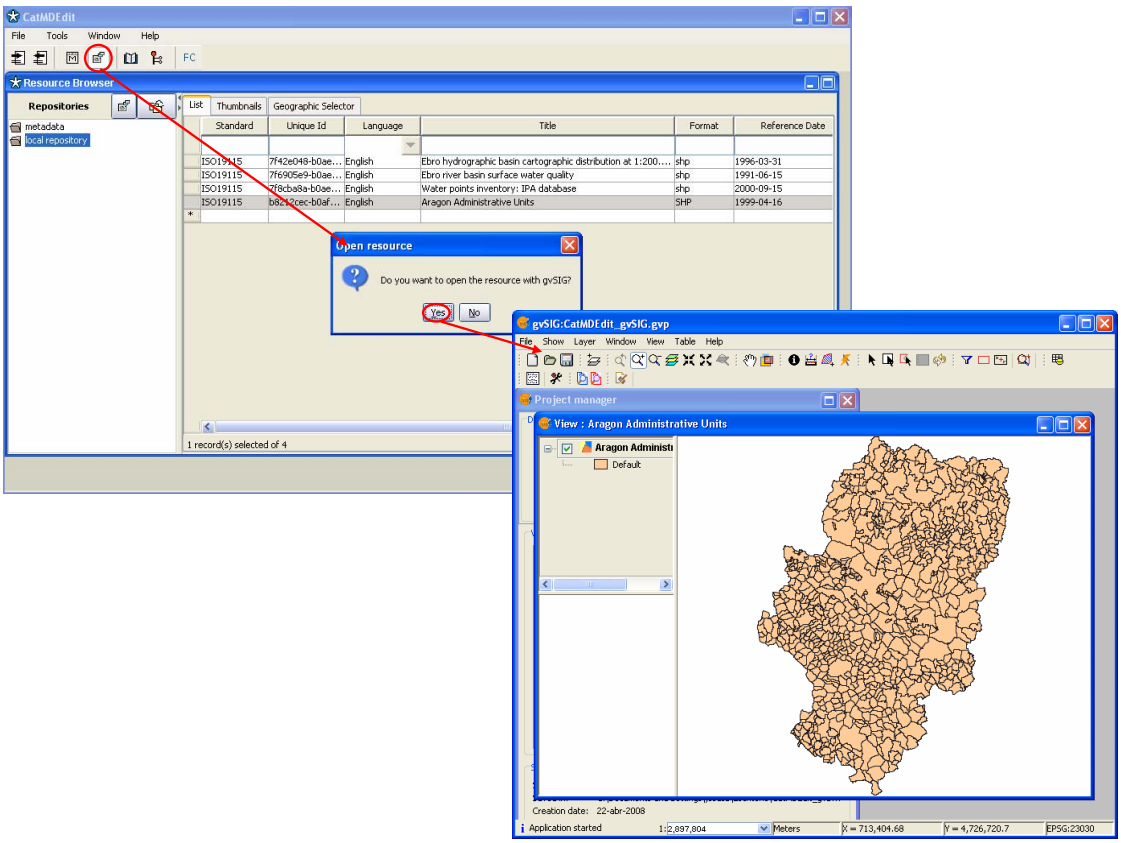

Fig. 4. Connecting to *gvSIG* for accessing a resources

the *gvSIG* tool for the first time, the user will be asked to browse the installation folder of the tool.

4 Related work in metadata editors with advanced GI capabilities

During the last decade different metadata editor tools have been developed to create metadata according different metadata standards such as CSDGM or ISO19115. A detail review of these tools can be found at appendix C of (Nogueras-Iso et al., 2005). However, very few of them have made efforts to facilitate the integration with GIS tools and make profit of the typical GIS functionalities.

After reviewing different metadata editors we have identified some advanced GI capabilities that facilitate the work of metadata creators. These GI capabilities are the following ones: automatic metadata generation (in most cases through the analysis of the encoding format); possibility of defining the geographic extent on a map; automatic generation of thumbnails to provide browse graphics; and direct connection with a GIS tool for accessing/viewing the resource.

GI Capabilities /Tools	*GeoNetwork*	*MetaD*	*disy Preludio*	*ArcCatalog*	*CatMDEdit*
Automatic metadata generation				X	X
Definition of geographic extent on a map		X	X		X
Thumbnails generation	X			X	X
Resource viewing	X	X		X	X

Table 1. Comparison of GI capabilities in metadata editors

Table 1 shows a comparison of the GI capabilities of different metadata editors. The tools compared in the table are the following: *GeoNetwork 2.0* [8] (web-based tool developed with the support of FAO-UN, WFP-UN and UNEP); *MetaD 3.0.4* [9] (desktop tool for metadata creation developed with the support of IDEC, the SDI of the Catalonia autonomous community in Spain); disy Preludio [10] (commercial tool providing a web-based environment to standardize metadata collection, processing and management); *ArcCatalog* (cataloguing tool developed by ESRI [11] and integrated within the ArcGIS tools); and *CatMDEdit* . All these tools provide, at least, one GI capability. For the sake of readability we have excluded those tools not providing GI capabilities.

As it can be observed, *CatMDEdit* is the only tool that aims at providing a solution for all these GI enhanced GI capabilities.

5 Conclusions and further work

This paper has shown the work done within the *CatMDEdit* tool towards the synergy with GIS tools. Up to now most metadata have been created with the main purpose of uploading the contents of catalogs accessible through an SDI initiative. Obviously the visibility of data holdings by means of metadata is very important. But there is no sense in producing metadata if these metadata are not going to facilitate the processing of data in a more effective way. Furthermore, if data processing is combined with metadata management, the quality of metadata will increase and it will be easier to maintain metadata catalogs up-to-date.

Based on the metadata describing a resource, this work has described the mechanism to derive dynamically the information required to create a *gvSIG* project with the adequate viewing parameters. The connection mechanism between *CatMDEdit* and *gvSIG* will appear in the new version of *CatMDEdit* to be released in 2009.

Further work on the integration of *CatMDEdit* and *gvSIG* will focus on establishing a bidirectional communication between both tools. Currently, *gvSIG* is launched from *CatMDEdit* whenever the user is interested in viewing and modifying the data. But in the future we plan a tight integration by means of plug-ins where the modifications performed in *gvSIG* will be notified to *CatMDEdit* and vice versa.

[8] http://geonetwork-opensource.org/
[9] http://www.geoportal-idec.net/
[10] http://www.disy.net/disy_preludio_en.html
[11] http://www.esri.com

References

Alfaro AA (2005a). "gvSIG: Software Libre, Infraestructuras de Datos Espaciales y tendencias del mercado". *Mapping*, 102:86–87.

Alfaro AA (2005b). "gvSIG: soluciones de código abierto para las tecnologías espaciales". *GeoFocus*, 5:38–42.

Béjar R, Fernández P, Álvarez P, Valiño J, Muro-Medrano PR (2001). *Sistemas de Información Geográfica: Una aproximación desde la Ingeniería del Software y las Bases de Datos*, volume 2 of *Monografías y Publicaciones. Colección Ingeniería Informática*, chapter "JGISView: JavaBeans para la integración de utilidades de visualización SIG en aplicaciones y applets", pp 185–190. Madrid, Spain.

Gamma E, Helm R, Johnson R, Vlissides J (1996). *Design Patterns*. Addison-Wesley.

Lacasta J, Nogueras-Iso J, Lopez-Pellicer J, Medrano PM, Zarazaga-Soria FJ (2007). "ThManager: An Open Source Tool for creating and visualizing SKOS". *Information Technology and Libraries (ITAL)*, 26(3):39–51.

Nebert D, (ed) (2004). *Developing Spatial Data Infrastructures: The SDI Cookbook v.2.0*. Global Spatial Data Infrastructure. http://www.gsdi.org.

Nogueras-Iso J, Zarazaga-Soria FJ, Muro-Medrano PR (2005). *Geographic Information Metadata for Spatial Data Infrastructures - Resources, Interoperability and Information Retrieval*. Springer Verlag, Berlin Heidelberg.

Zarazaga-Soria FJ, Lacasta J, Nogueras-Iso J, Torres MP, Muro-Medrano PR (2003). "A Java Tool for Creating ISO/FGDC Geographic Metadata". In *Geodaten- und Geodienste-Infrastrukturen - von der Forschung zur praktischen Anwendung. Beiträge zu den Münsteraner GI-Tagen 26./27. Juni 2003*, volume 18, pp 17–30, Münster, Germany. IFGIprints.