GEOGRAPHIC
DIGITAL CONTENT COMPONENTS *

André Santanchè[1], Claudia Bauzer Medeiros[1]

[1] *Institute of Computing – University of Campinas – UNICAMP*
CP 6176, 13084-971 Campinas, SP, Brazil
santanch@ic.unicamp.br, cmbm@ic.unicamp.br

Abstract Projects using geographic information tools involve a large variety of data objects, represented in different formats. Many efforts pursue standards to represent each kind of data object, and the interoperability between geographic information tools. The proliferation of data and tools raises the need for their reuse. This need can be extended to project reuse. This work presents a proposal to reuse geographic information projects based on a model called *digital content component*. This model can represent all elements involved in a project – including software components – and their relationship in a open homogeneous format.

Keywords: Digital content component, GIS interoperability, reuse, Semantic Web

1. Introduction

The production and processing of geographic data often involves a collection of different data objects represented in distinct formats. Besides format variety inherent to the kind of represented object (e.g. raster images, vector data and tables), there are divergences of representation of the same kind of object across systems.

To face this problem there are two main approaches. First, there are initiatives to define standards for data representation, like ESRI Shapefile (ESRI, 1998) and GeoTIFF (Ritter and Ruth, 2000). Second, Open GIS Consortium (OGC) (`www.opengis.org`) has proposed standardized APIs for GIS, to foster interoperability between software modules developed by different institutions. It promotes a component-based software development approach.

*This work was partially financed by UNIFACS, by grants from CNPq and FAPESP, and projects MCT-PRONEX SAI and CNPq WebMaps and AgroFlow.

The component-based approach opens the possibility for integrating specialized tools to deal with specific kinds of data objects, like editor systems, drawing tools, and so on. If OGC standards enable to relate these tools at execution time, the challenge remains on how to represent the relations between these data objects for storage and reuse purposes.

For example, if a user wants to share an entire software project including all used data objects and the combinations or relationships between these objects, he could face two problems. First, if another user wants to reproduce or modify this project he may not be able to access the same collection of components used to develop it. Second, how to represent the relationship between different kinds of data objects, managed by distinct specialized software components?

This work proposes a solution to this problem, which consists of a homogeneous model to represent software components and data used in a geographic information project. This model represents not only program code but any kind of data object by means of a component. This extended notion of component is named *digital content component* (Santanchè and Medeiros, 2004) and can encapsulate any kind of digital data: program code, raster data, vectorial images, tables, and so on. Like software engineering components, digital content components components encapsulates specific kinds of data objects inside a standard package. This package hides the data representation specificities and presents a public standard interface to guide the connection of components.

This approach enables representing a geographic information project as a network of interconnected data and program components. It adopts open Web standards to represent component structure, interface and metadata. Moreover, components' interfaces and metadata are related to ontologies which follow Semantic Web standards.

The remainder of the text is organized as follows. Section 2 introduces related work. Section 3 presents the digital content component model applied to the geographic information domain. Section 4 presents two case studies using digital content components in the GIS context. Finally, Section 5 presents concluding remarks and the present stage of this work.

2. Reuse and Interoperability

Reuse and interoperability are closely related dimensions, since the increase in interoperability expands the opportunities to reuse. Both reuse and interoperability in the geographic information domain constitute the kernel of our work, hence this section will explore these topics.

2.1 Data Reuse

In despite of the many kinds of data objects manipulated by geographic information systems, they are based on a fundamental kind of concept: spatial relationships visualized through maps. Therefore, the primary efforts to create open standards representations are concentrated on maps.

For raster data, the GeoTIFF (Ritter and Ruth, 2000) – based on the popular TIFF format – constitutes an improvement to the traditional standards for raster data, since it embeds georeferencing capabilities in the image file.

ESRI Shapefile (ESRI, 1998) format is a standard for vector data. It is divided in three sections, stored in three files. The first two – file extensions are `shp` (shapes) and `shx` (index of shapes) – describe the polygonal shapes in a map. However, since most maps relate shapes with a list of attributes, the shapefile format establishes a third file, which consists of a table (represented in dBase format – `dbf`), where each row defines a set of attribute values related with a shape.

An open format that represents distinctly the pieces that compose a data object, and the relationship between these pieces allows the user to reuse and modify the whole object, or each part individually. For example, the user can use a DBMS to add some attributes to the table file – which will be automatically related with the shapes – without modify the shape files.

2.2 Project Reuse

Any product of a GIS project can be viewed as a collection of related data pieces (e.g., maps, graphs or tables), composed and processed in a specific sequence. Open standards to represent geographic data objects are useful to share the resources used in a project and its results. However, in order to share the whole process followed to achieve a result, it is necessary to devise a mechanism to record, store and reproduce: the used data objects, their relationships and the sequence of combinations and processing operations made over these data objects.

Process recording allows its reproducibility. The user can subsequently modify some aspects and input data, obtaining new results. This originated the notion of scientific workflows (e.g. (Ailamaki et al., 1998)), where a workflow is used to represent a sequence of activities made in some experiment, and the resources used by it in each stage.

WOODSS (Seffino et al., 1999) is a system developed by us which enables the capture of activities in a GIS, and the related data resources used in each stage, to be stored as scientific workflows, which can be later edited, composed and re-executed.

2.3 Reuse Standards and the Semantic Web

The Semantic Web open standards promote interoperability at syntactic and semantic level in many areas. In the geographic information domain, OGC leads an initiative to create standards which promote interoperability for technologies involving spatial information and location.

OGC defines an XML based language – Geography Markup Language (GML) (Buehler et al., 2003) – to enable data interoperability, and an architecture that uses Web Services technology – OGC Web Services (OWS) (Buehler et al., 2003) – to enable inter-process interoperability. Following W3C Consortium (`www.w3.org`) proposed standards for Web Services (Chinnici et al., 2004), OWS specifies its service interfaces through WSDL.

These standards provide interoperability of both data and processes at syntactic level. Furthermore, the Semantic Web initiative defines standards to enable interoperability at the semantic level, involving languages like RDF (Manola and Miller, 2003) used to describe resources and, based on it, OWL (Smith et al., 2003) to describe and relate ontologies. The OWL Services Coalition adopted OWL to describe Web Services through OWL-S (The OWL Services Coalition, 2003). OWL-S uses WSDL as a basis to its syntax interface representation.

Combined, those standards promote interoperability of both data and processes at syntactic and semantic levels. Ontologies enable sharing domain knowledge, and the construction of top-level domain ontologies e.g., the Semantic Web for Earth and Environmental Terminology (SWEET) (Raskin, 2003), which has an OWL representation.

3. Component-based Geo-Information Processing

3.1 Digital Content Component

This work is based on a confluence of two previous works: Anima (Santanchè and Teixeira, 2001) and WOODSS (Seffino et al., 1999). Anima is an infrastructure for software components, originally meant to support the construction of educational applications but whose principles can be extended to other domains. The shaded part of Fig. 1 shows how Anima implements application construction via composition of software components.

Initially the developer builds a software component in a programming language, and uses a module named *packager* to pack a component into a package structure (left side of the figure). Components can be built in different program languages, but the external package format is the same. Anima uses RDF to represent component packages, including interface specification and component metadata.

The right side of Fig. 1 (shaded area) illustrates application construction based on component composition. An application is represented via a network of components whose configurations and connections are stored in an XML file. To allow interoperation of local or distributed components implemented in different languages, Anima provides support for component intercommunication via an XML based protocol.

This has been extended in three ways: (i) the component model was improved to encapsulate any kind of digital content; (ii) software component libraries for geographic information processing have been adapted to this extended infrastructure. (iii) WOODSS will be extended to represent workflows and geographic data inside content components.

The diagram of Fig. 1 shows the new cycle for component production/storage/use, reusing part of Anima implementation. The left side shows the component production. The user can pack inside content components any kinds of digital content alike, including software and data. This is done via the packager tool, which can take a form of a plugin attached to a third party system or a independent module.

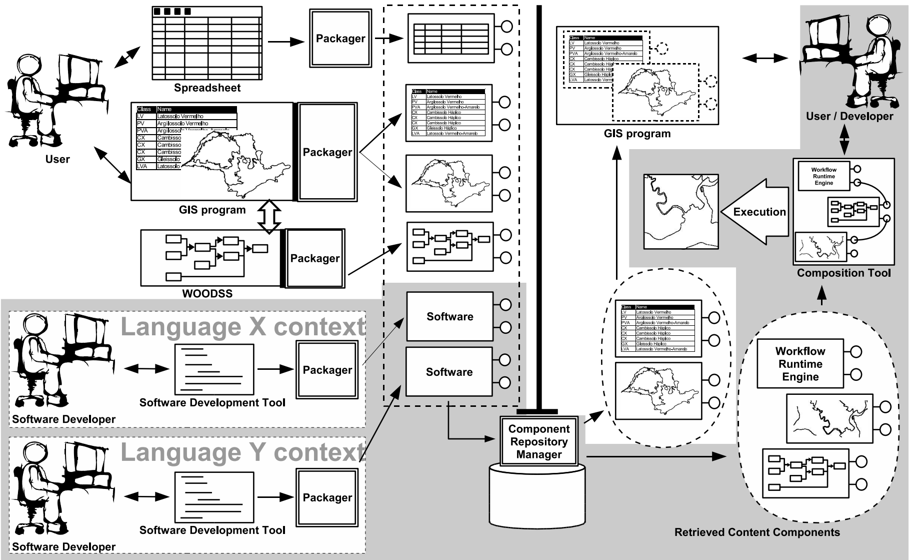


Figure 1. Diagram of digital content component component cycle for production/storage/use.

In the example illustrated in Fig. 1 the user interacts with a GIS program; actions are captured by WOODSS in a workflow. Both GIS and WOODSS store data and workflows in content components using packager plugins.

There are two kinds of content component: process and passive. A process component encapsulates any kind of process description (sequences of instructions or plans) that can be executed by a computer. In the example, the spreadsheet, workflow and software components are process components. Process components usually define an input interface, and their results change according to different input values. Passive components are non-process components, and contain data that can be used by a process component. In the example, maps and table components are passive components. As illustrated in the figure, all components packed in a standard structure are stored in a component repository.

The right side of Fig 1 illustrates two typical situations of content component use. Initially the user selects the desired components from the repository; next these components can be read by an application prepared to deal with them, or can be combined in a composition that results in an application.

The advantage for an application (e.g., a GIS) of using content components instead data files is the extra semantic information provided by component interface and metadata.

Each component representation is divided in four parts: (i) The content itself, in its original format; (ii) an XML specification of the internal structure used for component organization; (iii) an adapted WSDL/OWL-S specification of component interfaces; (iv) RDF/OWL metadata to describe functionality, applicability, use restrictions, etc. Components can be recursively constructed from composition of other components, each of which is structured by the same four parts.

3.2 Digital Content Geo-components

There are many projects to provide open software libraries to support development of applications in many domains. Therefore, the task of producing digital content components is mainly an adaptation activity.

Two libraries can be cited for geo-information processing: TerraLib (`www.dpi.inpe.br/terralib/home.html`) and GeoTools (`www.geotools.org`). TerraLib is a C++ component-based library. Their developers have profited from their previous experience in the construction of the Spring GIS (`www.dpi.inpe.br/spring/`), a mature GIS product. GeoTools is a Java based library that seeks to implement OGC specifications.

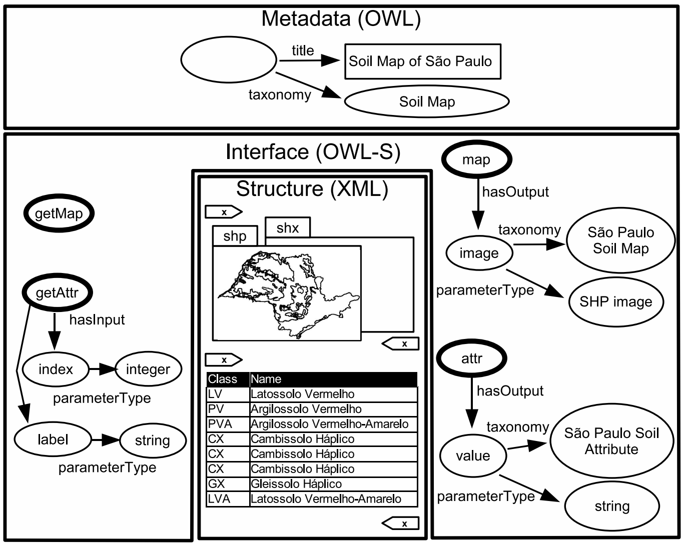
Both libraries can be adapted to our content component structure. However, since Anima infrastructure is mainly implemented in Java, GeoTools was the first choice. Additionally, since our infrastructure for digital content components adopts an adapted version of WSDL to interface specification, OGC based interfaces can be directly adopted.

4. Case Study

This section presents two case studies in agro-environmental planning. They will help understand the model and the infrastructure for digital content components.

4.1 The Soil Map Content Component

Fig. 2 shows an example of a partial representation of a content component. This component encapsulates a soil map of `São Paulo` state and a related table containing soil characteristics. The core of the figure shows the internal structure which follows the shapefile (ESRI, 1998) model, divided in three subcomponents: the polygons that delimit soil type regions (`shp`), the polygons index (`shx`) and the table (`dbf`) with one row of feature attributes for each polygon. XML structures the three subcomponents.



Class	Name
LV	Latossolo Vermelho
PV	Argilossolo Vermelho
PVA	Argilossolo Vermelho-Amarelo
CX	Cambissolo Háplico
CX	Cambissolo Háplico
CX	Cambissolo Háplico
GX	Gleissolo Háplico
LVA	Latossolo Vermelho-Amarelo

Figure 2. Soil shapefile content component representation.

The figure shows four interface operations: `map`, `attr`, `getMap` and `getAttr` which enable access to the component content. To simplify the component presentation in this and the following examples, we choose the left side of the interface to display the input operations and the right side to display the output operations.

An input operation can eventually trigger the execution of an output operation. In the example of Fig. 2 `getMap` triggers the output operation `map`, which returns a polygon representation of the map. The `getMap` operation has no parameters. The output polygon map is characterized by a data type (`parameterType`) and a classification following a `taxonomy`. The `parameterType` informs that the output map will be formated as a collection of polygons, in shapefile format. The `taxonomy` associates the parameter with two domain ontologies, illustrated in Fig. 3, through the concept "`São Paulo`

Soil Map". SWEET is used to specify the SoilLayer realm and the Map data product, and the POESIA spatial ontology (Fileto et al., 2003) is used to specify the $São\ Paulo$ state spatial coverage.

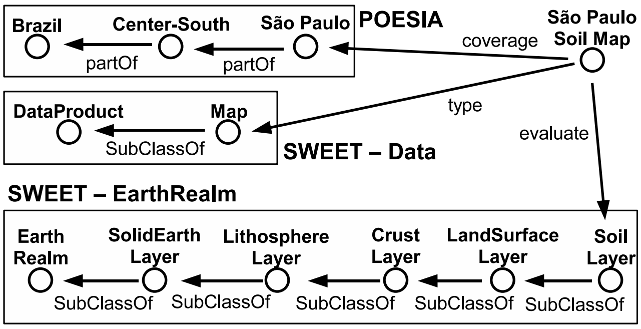


Figure 3. Ontology used by soil map component.

Analogously, the getAttr operation triggers the attr operation. The getAttr requests from the component a value of an attribute for a specified shape. It receives two parameters: the index of the shape and the label of the desired attribute. Operation attr returns the value of the asked attribute.

The component illustrated in the Fig. 2 is passive, therefore it does not embed the program code to execute these operations. In the absence of code, interface operations are implemented in another component named *companion component*. The companion component lends its operations to a passive component in way that is transparent to end users. The choice of the appropriate companion component for a passive component is determined by the application designer, and is sensitive to the context. This allows a homogeneous treatment of passive and active components from the user's perspective.

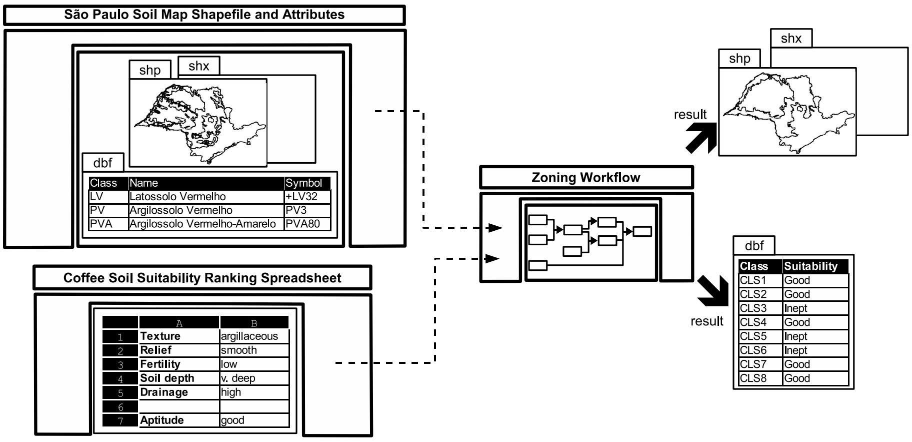
The GeoTools library has support to process shapefiles, including the relationship between shapes and tables. Therefore it will be used to produce the companion component for this kind of passive component.

4.2 Pedological Zoning for Coffee Crop

Fig. 4 illustrates an application constructed via composition of content components to build a pedological zoning for coffee crops. Some details of components representation are omitted to simplify the explanation.

The component presented at the top left side is a map component of the same type of the previous example. It contains a soil map of São Paulo state, and attributes describing soil characteristics for each region.

The bottom left part has a spreadsheet component that specifies rules to rank the soil suitability to coffee, based on soil characteristics. The zoning workflow


São Paulo Soil Map Shapefile and Attributes
shp
shx
dbf
Coffee Soil Suitability Ranking Spreadsheet
Zoning Workflow
result
shx
shp
result
dbf

Class	Name	Symbol
LV	Latossolo Vermelho	+LV32
PV	Argilossolo Vermelho	PV3
PVA	Argilossolo Vermelho-Amarelo	PVA80

	A	B
1	**Texture**	argillaceous
2	**Relief**	smooth
3	**Fertility**	low
4	**Soil depth**	v. deep
5	**Drainage**	high
6		
7	**Aptitude**	good

Class	Suitability
CLS1	Good
CLS2	Good
CLS3	Inept
CLS4	Good
CLS5	Inept
CLS6	Inept
CLS7	Good
CLS8	Good

Figure 4. A composition to produce a soil suitability map.

component is responsible for getting each attribute in the map component, and submitting it to be ranked by the spreadsheet component.

A table containing coffee soil suitability is generated as an intermediate result of workflow execution. Next, the workflow component merges adjacent polygons which contains the same soil suitability value. This will result in the two illustrated results: a map with a pedological zoning for coffee, and a table containing the suitability attributes for each zone.

The use of ontologies in the interface description can enhance component interface matching. For example, the workflow component can specify interface it requires to allow connection with the soil map component, using the same ontology illustrated in Fig. 3, without referring to `São Paulo`. This means that the workflow component accepts a soil map of any geographic region.

Some process components need an extra code layer to be executed, since their process description needs to be interpreted. This is the case of the workflow and the spreadsheet components. Similiar to the mechanism used by passive components, process components can be associated to companion components enabled to interpret its process description or instructions.

5. Concluding Remarks

This project presents a solution for data and program code reuse to face the proliferation of geographic data formats and software tools. The main contribution is a homogeneous model to deal with software components and data objects in the geo-information processing domain, taking advantage of advances in the Semantic Web to boost interoperability.

The work combines two previous experiences: Anima, for component-based applications for the educational domain (Santanchè and Teixeira, 2001), and WOODSS on the use of scientific workflows, for reuse of GIS projects (Seffino et al., 1999).

References

Ailamaki, A., Ioannidis, Y., and Livny, M. (1998). Scientific Workflow Management by Database Management. In *Proc. 10th IEEE International Conf. on Scientific and Statistical Database Management*, pages 190–201.

Buehler, K. et al. (2003). OpenGIS Reference Model. `www.opengis.org/docs/03-040.pdf`, accessed 08/2004.

Chinnici, R. et al. (2004). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language – W3C Working Draft 26 March 2004. `www.w3.org/TR/wsdl20/`, accessed on 06/2004.

ESRI, Environmental Systems Research Institute (1998). ESRI Shapefile Technical Description. `www.esri.com/library/whitepapers/pdfs/shapefile.pdf`, accessed on 08/2004.

Fileto, R., Liu, L., Pu, C., Assad, E. D., and Medeiros, C. B. (2003). POESIA: An ontological workflow approach for composing Web services in agriculture. *The VLDB Journal*, 12(4):352–367.

Manola, F. and Miller, E. (2003). RDF Primer – W3C Working Draft 23 January 2003. `www.w3.org/TR/rdf-primer/`, accessed on 11/2003.

Raskin, R. (2003). Semantic Web for Earth and Environmental Terminology (SWEET). In *Proc. of NASA Earth Science Technology Conference 2003*.

Ritter, N. and Ruth, M. (2000). GeoTIFF Format Specification – GeoTIFF Revision 1.0. `www.remotesensing.org/geotiff/spec/geotiffhome.html`, accessed 08/2004.

Santanchè, A. and Medeiros, C. B. (2004). Managing Dynamic Repositories for Digital Content Components. In *Proc. of the ICDE/EDBT Ph.D. Workshop 2004*.

Santanchè, A. and Teixeira, C. A. C. (2001). Anima: Promoting Component Integration in the Web. In *Proc. of 7th Brasilian Symp. on Multimedia and Hypermedia Systems*, pages 261–268.

Seffino, L. A., B. Medeiros, C., Rocha, J. V., and Yi, B. (1999). WOODSS – A spatial decision support system based on workflows. *Decision Support Systems*, 27(1-2):105–123.

Smith, M. K., Welty, C., and McGuinness, D. L. (2003). OWL Web Ontology Language Guide – W3C Candidate Recommendation 18 August 2003. $www.w3.org/TR/2003/CR - owl - guide - 20030818/$, accessed on 11/2003.

The OWL Services Coalition (2003). OWL-S: Semantic Markup for Web Services. `www.daml.org/services/owl-s/1.0/`, accessed on 05/2004.