Towards the Automatic Generation of Web GIS
University of Salerno - DMI
Via Ponte Don Melillo, I-84084 Fisciano (SA), Italy
2nd line of address
{sdimartino, fferrucci, lpaolino, msebillo, gvitiello}@unisa.it

ABSTRACT
The goal of disseminating, manipulating, and exploiting spatial knowledge over the Internet, has led to the development of Web applications, known as Web Geographic Information Systems (Web GIS). The growing need for tools that allow to rapidly develop Web GIS has motivated the extension of existing approaches for traditional web applications in the GIS domain. Among the different solutions, automatic generation tools deserve special attention thanks to the higher level of abstraction required during development. Indeed, developers have only to provide a visual specification of the target application that is automatically constructed by the tool.

In the present paper, we propose an extension to the GIS context of the approach based on WebML, a high-level, formal visual language specifically conceived to design data-intensive Web applications. The proposal is motivated by the observation that Web GIS can be considered as a particular class of data-intensive Web applications and by the fact that WebML is gaining an increasing success for designing and developing traditional data-intensive web applications. In the paper, we describe the proposed extension of the visual formalism for modeling relevant interaction and navigation operations typical of Web GIS, and provide an example of its application within a prototype of the tool supporting the automatic generation of Web GIS.

Categories and Subject Descriptors
H.5.4 [Hypertext/Hypermedia], D.2.2 [Design Tools and Techniques]Computer-aided software engineering (CASE)

General Terms
Design, Human Factors, Languages, Management.

Keywords
Web GIS, Design Models, Web Applications.

1. INTRODUCTION
Recently, the increasing popularity of the Internet has led to the development of Web applications, known as Web GIS, intended for dissemination and manipulation of spatial knowledge in specific domains. In fact, Web GIS hold the potential to make geographic information available to a worldwide audience, allow Internet users to access GIS applications from their browsers without purchasing proprietary GIS software, and make it possible to add GIS features to a wide range of network-based applications in business, government, education, etc... [19].

With the increased time-to-market pressure, it is no longer possible to deal with low-level issues, and create Web GIS from scratch. Thus, there is a growing need for tools and methodologies that allow us to rapidly develop this kind of Web applications and to rapidly modify them to meet the ever-changing business needs [12]. Recently, in order to assist the classic developing methodology based on API, some frameworks have been developed by companies and research laboratories. To date, solutions like ArcIMS by ESRI [7], MapGuide by AUTODESK [1], WebMap by INTERGRAPH [11] and Mapacell [14] are becoming very popular because they enable users to develop and deploy WebGIS applications in a fast and easy way by simply selecting functionalities a user needs. However, the easiness of producing Web GIS applications is limited to generate interfaces where layouts are predefined. Further customization requires expertise in programming code. The outcome of this situation is a large quantity of WebGIS which are very similar both in form and functionalities, that is to say the easiness has been paid in terms of customizability.

Other solutions allow developers to customize interfaces through WYSIWYG environments where widgets are selected from specialized palettes and then located into panels representing web pages [17], [21], [1]. This approach guarantees a flexible layout composition, but the customization of functionalities continues to be strictly deferred to the implementation level. As for traditional Web applications, many solutions are currently available meant to increase productivity and to support rapid development by allowing users to work at a higher level of abstraction. These can be grouped into two major approaches, one based on frameworks and the other based on automatic generation starting from visual specifications

A Web framework is software that makes easy to build Web applications by abstracting frequent problems of Web development and providing shortcuts for common programming tasks (e.g. to display, and validate HTML forms, to convert user-submitted input into data structures that can be suitably manipulated, etc.), such as Django, Ruby on Rails, .Net, or J2EE. Automatic generation tools allow developers to go “up” one further level of abstraction. Indeed, they are based on a suitable visual language that is used to model the target application. Starting from the supplied visual specification, these tools are able to automatically generate the corresponding application (e.g. WebRatio, W2000, VisualWade). They differ in several aspects
due to the characteristics of the underlying visual languages and the typology of web applications each visual language was originally conceived for (data-intensive, hypermedia, workflow, etc.).

In the context of Web frameworks, much attention is currently being devoted to provide developers with specific support to the implementation of Web GIS applications (see for example the plug-in GeoRuby [9] or GeoDjango [10]).

Conversely, as for the second major approach, at the best of our knowledge, no web application automatic generation tool has been extended so far in order to specifically support the development of Web GIS. Among the existing automatic generation tools, WebRatio seems to be especially appropriate for an extension to the GIS domain [6]. Indeed, it exploits a high-level, formal visual language, WebML (Web Modeling Language), especially conceived to design data-intensive Web applications [5], [4], [22]. The observation that Web GIS can be considered as a particular class of data-intensive Web applications and the increasing success gained by WebRatio [6], has motivated us to extend the visual formalism so as to model relevant interaction and navigation operations typical of Web GIS. The underlying WebML methodology has been adapted to deal with the Web GIS context by considering the two orthogonal perspectives of Geodata and Metadata Conceptual Model and of Hypertext and Web Mapping Model. In particular, for the first aspect the Spatial E-R model by Calkins [3] has been exploited to design the conceptual schema of (spatial) data. While, for the second aspect the WebML Hypertext Model has been extended to include new notations specifically tailored for GIS concepts and tasks. The corresponding extension of the WebRatio tool can provide potential Web GIS developers with a Rapid Application Development environment, where he/she is able to design in a visual fashion the application that will be automatically generated by the tool.

The remainder of the paper is organized as follows: in Section 2 we recall the main concepts of WebML, while in Section 3 we describe the proposed modeling concepts and notations which extend WebML for Web GIS. In Section 4 we present the architecture of the WebGIS applications generated by the WebRatio tool, while an example of use of the proposed methodology is described in Section 5. Some remarks conclude the paper.

2. THE WEBML METHODOLOGY FOR DATA-INTENSIVE WEB APPLICATIONS

Graphical and diagrammatic representations play a central role in the field of software engineering [8]. They are widely employed to support many activities of the software development process, such as specification, analysis and design, since they provide powerful aids to describe and understand complex systems, by means of abstractions and different views. Moreover, they help to identify and describe design patterns that can be reused in different applications, thus avoiding to create from scratch. These represent also the main reasons for the success of the WebML method and motivated our proposal to adopt the WebML approaches for the development of Web GIS applications.

WebML is a modeling language suited to support users in designing data-intensive Web applications [6], by providing a set of visual notations to model the content structure, the data, and the navigational aspects of a Web application.

The design of a Web application is based on two orthogonal perspectives: data and navigation. The former is usually described by means of the well-known E-R data model, to represent all the relevant entities and relationships. The latter issue is modelled by the WebML Hypertext Model, which describes how the previously defined contents will be arranged and rendered within the Web application. In particular, the application structure and appearance is defined through four main levels of abstraction: site views, areas, pages and content units.

The site view is the highest level. It is used to model a part of the web site that is relevant for a user group. A site view is composed of many areas, modelling the sections of the application. Each area can contain both other areas (in a recursive way), both pages, that are the containers of the information. Each page is composed by some content units, the elementary pieces of information taken from the data schema. To bind a unit with the corresponding data, the WebML Hypertext Model requires to specify a source and a selector. The former is an entity of the database, while the latter is a condition over this entity, to filter the rendered data. These objects can be interconnected by links, to model the resulting navigational aspects. Links can be also used to represent data movements, or parameter passing, between units. As an example, information gathered from the user via a web form, can be used in the selector condition for extracting the data instances to be displayed. Many different types of content units are provided. For instance, it is possible to model units to update the data, or to delete them. Other visual primitives allow the designer to deal with other key aspects of a Web application, such as getters and setters for Session Parameters and other global variables.

The WebML methodology is centered on their modeling concepts which help to keep distinct the two orthogonal perspectives of data and navigation. Moreover, the well defined semantics and syntax of the proposed notations allowed for the construction of a specifically conceived CASE tool, WebRatio, which automatically generates the Web application starting from the corresponding design models, thus effectively reducing time and costs of software development. Within this tool, each stereotype has both a visual and a XML-based textual representation. This is used to specify additional detailed properties, such as the grammar for the combination rules. The availability of an XML specification has two main advantages: it enables the deployment of the same design into multiple rendering formats, and it supports the definition of custom extensions of the language.

3. AN EXTENSION OF WEBML FOR WEB GIS

Web GIS applications are a special case of data-intensive Web systems, meant to deal with complex geographic data and share them across several users for different business goals. Geographic information is usually distributed across different layers, which a Web GIS user should be able to handle separately or in overlay modality. Thus, besides common Web navigation and composition tasks, far more complex functionalities are needed in Web GIS for visualization and content management. As a matter of fact, with respect to traditional Web applications, this kind of systems require special focus on spatial data which may be
acquired from different sources and stored in different formats, for all of which the user should be offered direct support.

From a design point of view, Web GIS present many specific characteristics, making them different from traditional data-intensive Web applications. Among them, two are of fundamental relevance: one is related to the data model and the other is concerned with the navigational model. As for the first aspect, the complex nature of geographic data, where the two components, descriptive and spatial, should be analyzed and managed in a joint manner requires the use of a different modelling approach, named Geodata and Metadata Conceptual Model. In our proposed methodology, we suggest the adoption of the Spatial E-R model [3], where each set of geodata is described as a spatial entity characterized by a set of attributes, a geometry and a couple of coordinates. Moreover, spatial associations among entities are expressed in terms of topological relationships. The expected output from this geodata design task consists of a set of logical models, which organize data in a relational way and associate output from this geodata design task consists of a set of logical models, which organize data in a relational way and associate the space they occupy (e.g., a land parcel, a timber stand, a wetland, a soil type, etc.). Each set of geodata is described as a spatial entity characterized by the following properties:

- **Spatial Object**
- **Geometry**
- **Type**
- **XY Coordinate Indicator**
- **Topological Indicator**
- **Regular Object Name**

A similar extension has also been defined for relationships, so as to model entities which are spatially related. According to this model two new symbols should be involved, besides the traditional diamond symbol. An elongated hexagon is used to model the topological relationships connectivity and contiguity. A double bounded elongated hexagon instead represents spatial relationships verified by means of x, y coordinates (coincidence, containment and proximity).

![Figure 1: The entity internal structure](image)

### 3.2 The WebML extension for Web Mapping

In order to exploit WebML for modeling Web GIS, we have considered new interaction metaphors able to capture the particular characteristics of this kind of applications. Besides the common interactions a user performs when navigating and executing a web application, we have taken into account the actions directly performed on a map, when carrying out web mapping or spatial query tasks. Such actions usually produce data, which are not presented in a new web page but are rather rendered onto the same map. Thus, new WebML units are needed to model specific Web GIS interaction tasks, such as map visualization and navigation. In the following we will describe some of them.

#### 3.2.1 The MultiMap Unit

Basically, the MultiMap Unit is a visual metaphor for a map viewer, i.e. a graphic component within a Web browser, able to render both vector and raster data arranged in layers. It is characterized by the following properties:

- **Name**: the name chosen by the Application Architect for the MultiMap Unit.
- **Raster Sources**: the reference to raster entities. This source is optional, because a Web GIS application may not require raster layers.
- **Raster Selector [optional]**: predicate useful to determine which objects belonging to the raster entities share in the unit content.
- **Vector Sources**: the reference to vector entities. This source is optional, because a Web GIS application may not require vector layers.
- **Vector Selector [optional]**: predicate useful to determine which objects from the vector entities share the unit content.

Moreover, to correctly render source data, the MultiMap Unit requires a global parameter, named *Extension*, which contains coordinates of the upper-left and lower-right corners of the presented area. Each time this parameter is modified by an external operation, the area rendered in the MultiMap Unit is updated accordingly.
Figure 2 depicts the visual notation of a MultiMap Unit instance, namely TheMap. Since a Web GIS application is expected to allow the activation and display of layers from a Table of Contents on users’ demand, two Multi-choice Index Units are linked to represent the Table of Contents which in turn is made up of two different legends (Vector and Raster). Depending on aesthetic/usability factors, design layouts can also be provided where legends are integrated into a homogeneous format.

3.2.2 The Zoom In/Out Units
These two units refer to the conventional zoom in and zoom out operations. Both take three input values: the global parameter with the current extension of the map, the global parameter with a magnification/diminishing factor and the instance Point of Geometry Entry Unit, which is used to identify the area on which the operation should be applied. As a result, if the operation succeeds, it sets a new value for the Extension global parameter, according to the zoom factor. This arrangement is graphically depicted in Figure 3 where an OK link leads to a new value for the Extension, while a KO redirects towards a generic error page.

3.2.3 The Pan Unit
This unit represents the pan operation that shifts the focus of a map toward a specified direction. This unit takes three input values, namely the two points of the map where the user presses and releases the mouse button, and the global parameter with the current extension of the map. If the operation succeeds, a new value for the Extension global parameter is set, i.e. the new area of the map which is visualized. In Figure 4, visual symbols involved in a pan operation are shown.

3.2.4 The CreateOverlay Unit
In Web GIS, a query processing usually outputs a set of geodata which satisfy user’s criteria. In order to display such geodata onto a map, they are organized in a new temporary layer, which may be overlaid to the existing ones. Users may use them to perform other basic operations and derive new geographic information.
In order to describe such functionality through a WebML-like approach, we introduced the `CreateOverlay` Unit, which can be exploited to model scenarios where users pose queries and apply further spatial operations to the corresponding output in a cascade-like fashion. The `CreateOverlay` Unit requires two parameters, namely a spatial filter and a layer on which the filter should be applied. In Figure 5, an instance of the `CreateOverlay` unit is depicted, namely `NewTheme`, where the spatial filter is obtained by a `Geometry Entry` Unit which returns a rectangle, and the layer is the active one. In case the operation succeeds, the OK link indicates that the OID of the newly created layer will be forwarded to the `MultiMap` unit, which will eventually render it.

### 4. A TOOL FOR MODEL-DRIVEN DEVELOPMENT OF WEB GIS APPLICATIONS

In order to support the methodology proposed in Section 2 we have realized a visual environment, which integrates and extends two existing tools, namely `SpaDa Modeler` and `WebRatio`, for the specification and the automatic generation of Web GIS applications.

Figure 7 shows the architecture of the proposed visual environment. In particular, `SpaDa Modeler` allows developers to visually specify the Geodata and Metadata Conceptual Model. Indeed, it allows to design a conceptual spatial schema related to the scenario of interest by defining the abstract description of the entities and relationships, expressed according to the Spatial ER model by Calkins. Starting from the supplied visual specification `SpaDa Modeler` generates:

- the SQL statements corresponding to the defined entities and relationships and the metadata dictionary, which can be used for possible data interoperability. All this information is fed into the adopted GeoDBMS;
- the XML code describing the data structures according to the WebRatio format.

As for the Hypertext and Web Mapping Model we have suitably extended `WebRatio`, the CASE tool supporting the WebML methodology.

By exploiting some of its key features, namely extensibility and flexibility, we defined new content units by specifying the rules to generate the corresponding J2EE-based code for the expected web applications. So, the resulting extension of `WebRatio` supports developers in the visual specification of the application in terms of site views, areas, pages, content units and links. An example of a new unit within the `WebRatio` 4.3 environment is depicted in Figure 6.

Starting from this specification, the tool generates the server side code of the Web GIS application. It is worth noting that the generated application can be connected with any GeoDBMS thanks to JDBC connection provided by `WebRatio`. Figure 7 also depicts a possible architecture of the generated Web GIS applications, where we have indicated PostGIS as GeoDBMS and Geoserver as GIS server.

As we can see, a WebGIS application is based on a multi-tiered architecture and on a set of standard and open technologies such as Apache/Tomcat, Web Feature Service [14] (WFS) and Web Map Service [16] (WMS) protocols, which guarantee the interoperability among server modules and the client browser. Basically, the content units are translated into a JSP/Servlet...
program and successively mounted as a Tomcat/Apache web application. This application will be able to manage requests coming from the Internet and switch them to either the Geoserver or the PostGIS as required.

At run-time actions performed by the user on the client side are translated into HTTP request by JavaScript code and sent to the server where data satisfying the request are selected and sent back to the client as HTML, JavaScript and raster data. In detail, a GIS query is transmitted as a GET or POST request written according to either the WFS or WMS specifications. The request is therefore captured by the generated GIS application and forwarded to Geoserver which, in turn, interprets the query, composes the SQL statement according to the PostGIS DML and sends it to the GeoDBMS.

Once the GeoDBMS computes the query, results are gathered by Geoserver to create the answer. In particular, in case of a WMS request, Geoserver computes a raster map containing the results encoded as a standard picture format (GIF, PNG, SVG, etc.). In case of a WFS request, Geoserver collects data from the GeoDBMS and returns a GML (Geography Markup Language) encoded data to the generated GIS server application. The latter further processes the resulting GML data, by sending it back to the client side in HTML format.

5. MODELING AND GENERATING A WEB GIS APPLICATION FOR FARM HOUSES MONITORING

In this section, we describe the use of the tool in the design of an application meant to monitor farm houses located in Campania, a region of Southern Italy. Here, Health Districts may control phenomena related to distribution of livestock and disease diffusion, by navigating a map, displaying geodata and posing queries about areas of interest. Moreover, the Web GIS may support vets and administrators in making decisions about policy and preventative plans.

Figure 8 depicts the framework for an interface layout for this kind of Web application. It consists of a Map Working Area where geodata are displayed and users may navigate, a Table of Contents, where a legend allows to select and activate layers, a palette of GIS Tools to invoke GIS functionalities, a Query Frame where a query may be posed, and a KeyMap which supports users in orienting themselves.

As for the design phase, in this section we report on two main aspects of the entire process, namely:

- the description of geodata and their relationships by means of the Spatial E-R extension, and
- the application of the proposed WebML extension for visually describing the transitions among pages and the web mapping interaction tasks.

As for the first step, Figure 9 depicts the conceptual schema which models entities and relationships involved in this scenario. Three spatial entities, Region, Health District and City, are used to describe the administrative boundaries of land boards in Campania, while the FarmHouse spatial entity is provided to describe both farm house properties and positions. City and Health District, as well as Health District and Region, are connected by means of two aggregation relationships, which reflect the “part of” association which exists between them. Analogously, farm houses are contained into at least one city. Finally, information about vets is managed through a conventional entity. In this case, a relationship between the Veterinary and Health District has been defined in order to establish that a vet works just in one health district.

As for the Hypertext & Web Mapping design, the Web GIS site is composed by three ANDed pages (see Figure 10):

- a SideButton page that contains buttons to activate the GIS functionality, such as Zoom In, Pan and SelectByLayer;
- a Central page, containing the TheMap instance of the MultiMap Unit, the Multi-choice units to de/select raster and vector layers within the map, an Index Unit to select the active layer where the user computes an operation, and finally some global parameters to store the active layer name, the map Extension and the ZoomFactor;
- an Error Page.

In order to describe the zoom in functionality, we exploited the Zoom In Unit, namely Magnify, which is fed through the ZoomFactor and the Extension global parameters and an instance of a Geometry Entry Unit, which returns a point. The result is a new map extension which will be stored into the Extension parameter. The TheMap Unit will automatically be refreshed.
For the Pan operation, the Move instance of the Pan Unit handles three parameters, namely the present extension, the direction and the distance to move along that direction. Also, the Drag instance of the Geometry Entry unit is modeled which allows users to visually specify direction and distance by means of a rectangle drawn by a drag operation on the map. Such values, together with coordinates received from the Extension global parameter, are fed to the Move unit in order to compute the new map extension. Finally, to visualize the new map boundaries we put the Move result into the Extension parameter.

The last operation to model corresponds to a spatial query aimed to identify all the farm houses within a user drawn rectangle. Then, in order to get graphical results, the following steps are necessary:

- to add a new entry into the TheMap source,
- an instance of the GeometricEntry Unit to draw the rectangle of the area of interest on the map;
- the NewTheme instance of the CreateOverlay Unit in order to manage output rendering.

Figure 11 shows the resulting Web GIS application, where, by following the same approach, also other functions have been considered.

6. CONCLUSIONS

In the present paper, we described a methodology for the design of Web GIS applications based on the WebML approach. The WebML methodology seems to be especially adequate also because it allows us to concentrate on different concepts: data and presentation. In particular, we proposed to exploit the Spatial E-R model to define spatial data and we introduced some visual notations to model relevant interaction and navigation operations that are specific for Web GIS.

The methodology is supported by a prototype we have implemented and tested against a real case study, dealing with the
management of spatial information on farm houses in the Southern of Italy.

As future work, we also plan to consider some other existing visual notations for the specification of web application and study their extension to the GIS domain. This would also allow us to perform a comparative analysis of different approaches to the visual specification and automatic generation of Web GIS.

7. REFERENCES