GEOSPATIAL DATA HANDLING FOR WEB-BASED AND MOBILE APPLICATIONS

M. Bertolotto [a, *], J. D. Carswell [b], L. McGeown [c], P. Thijs [c]

[a] Dept. of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland - michela.bertolotto@ucd.ie
[b] Digital Media Centre, Dublin Institute of Technology, Aungier St., Dublin, Ireland - jcarswell@dit.ie
[c] e-Spatial Solution, Dublin, Ireland - info@e-spatialsolutions.com

Commission IV, WG IV/4

KEY WORDS: Spatial, Database, Web based, GIS, Real-time, Query.

ABSTRACT:

One of the main bottlenecks for web-based GIS usability is in the transmission and handling of the vast amounts of geospatial data involved. The shear volume of raster, vector, attribute, and other data to be analysed, queried, transmitted and displayed over the WWW in real-time is prohibited by connection speeds. In this paper we introduce e-Spatial™ technology, a fully integrated open-standard web-based solution to manage multiple spatial relationships directly in an Oracle Spatial database and to deploy in real-time spatially enabled (or standard) applications completely independent of data volume. Indeed, no application code is ever installed on the client side. With the database generating all requested information on the fly, only the minimum data required is downloaded to the client, which leads to the most economical usage of bandwidth and resources for real-time response. This technology dramatically extends the functionality of Oracle Spatial by allowing real-time spatial data display, collection, editing, manipulation, and query using a standard web browser. Therefore, applications can be deployed on any device that supports the Java Virtual Machine, thus providing full support for wireless, PDAs and other mobile devices.

1. INTRODUCTION

Many GIS applications (such as those developed by ESRI, MapInfo, Autodesk, Intergraph) require the use of proprietary application-specific software (East et al., 2001). Furthermore, they store spatial and non-spatial data attributes separately: typically spatial properties of objects are stored in files that are managed by a file management system while attribute data is stored in a commercial database (e.g., a relational database). This split design presents the difficulty of maintaining data integrity between spatial and attribute data as the two types of data are not managed by the same engine. Oracle Spatial provides the information infrastructure that includes a single database system for managing both types of data, together with a data structure that is independent of the particular application.

From a data management point of view, using such a unified approach has an important advantage: it allows users to access full function spatial information systems based on industry standards, i.e. with an open interface to all data (e.g., SQL). Furthermore, the capability of storing spatial data in existing enterprise-wide database management systems enables the spatial deployment of several more enterprise applications that may have otherwise remained unexploited. Analysis and reporting functionality is also facilitated by the complete integration of management information and spatial data repositories.

e-Spatial™ technology (www.e-Spatial.com) has been developed within the Oracle Spatial unified environment. Its web-enabling application architecture component iSMART™ and its Java plug-in iSIS dramatically extend the functionality of an Oracle Spatial database by allowing real-time spatial data display, collection, editing, manipulation and query using a standard web browser (Bertolotto et al., 2001). By utilising e-Spatial™ technology, Oracle Spatial users can exploit a full range of advanced spatial data handling and management functions that are typically provided by specialised proprietary application packages. These functions, together with all data stored in the enterprise database, are available for access using functionality built into the database itself. Indeed, one of the greatest advantages of this software solution is its seamless integration into the database instead of being developed as a separate application.

This has been achieved by implementing all management information and spatial analysis functions as stored procedures. This characteristic makes the package specifically suited for interactive real-time analysis for mobile location applications such as utility and government mapping, navigation systems, GPS applications and the emerging location-based applications.

One of the main bottlenecks for web-based GIS usability is in the transmission and handling of the vast amounts of geospatial data involved. The shear volume of raster, vector, attribute, and other data to be analysed, queried, transmitted and displayed over the WWW in real-time is prohibited by connection speeds. e-Spatial™ technology provides a fully integrated open-standard web-based solution to manage multiple spatial relationships directly in the database and deploy in real-time spatially enabled (or standard) applications (in a two or three tier architecture) completely independent of data volume. Indeed, no application code is ever installed on the client side resulting in a thin client applet. With the database generating all requested application information on the fly, only the minimum data required is ever downloaded to the client, which leads to the most economical usage of bandwidth and resources using standard browser technology on any device that supports the Java Virtual Machine, thus providing full support for wireless, PDAs and other mobile devices.

While in the past spatial information was utilised within specific applications and exclusively by expert high-end users, recently, thanks to the diffusion of desktop GIS and the Internet, its integration within the widest range of information systems is becoming a common requirement. Within this context, a pressing need relates to providing non-expert users with easy-to-use environments where they can visualise, query and manipulate spatial data without requiring any specific programming or database management background. To this purpose, additional tools to build and customise graphic user interfaces that facilitate interaction with the iSMART™+iSIS platform have also been developed.

In this paper we focus on the geospatial data handling characteristics of i-Spatial™ Information Server (iSIS) developed by e-Spatial™ Solutions. The data model utilised is the Oracle Spatial object-relational model. Furthermore, we discuss the problem of guaranteeing the preservation of topological consistency as it represents a critical issue from the point of view of data management (Egenhofer and Franzosa, 1991). We describe how the integrated Oracle 9i topology management functionality completely controls topology within the iSMART™+iSIS platform: if a spatial element is updated, all spatially related elements are automatically changed accordingly to guarantee consistency.

Finally, existing systems have limited ability to switch on/off specific subsets of spatial data. They are restricted to using all the data contained within each data layer. Therefore the data content is fixed. However, using e-Spatial™ technology not only are all datasets maintained in a single Oracle database, but the users can also select the particular data they wish to view and query. For example, in the case of a routing query the user may only wish to see the road network from their starting location, and the buildings only within 500 metres radius of their final destination. They have no requirement to view buildings data along the actual route itself. This ability to define exactly what the users need to query greatly improves the speed with which the individual query is completed. This characteristic conforms to the information-on-demand approach discussed by several authors in the context of web-based vector map generali sation (Bertolotto and Egenhofer, 2001; Buttenfield, 1991; Cecconi and Weibel, 2000).

The remainder of this paper is organised as follows. Section 2 presents the architectural components of e-Spatial™ technology. In section 3 we discuss the management of the topological structure of geo-spatial data within web-based systems developed using such a technology. Additional utilities for improved user-friendliness are described in section 4. Section 5 reports some concluding remarks and future developments.

2. e-SPATIAL™ ARCHITECTURE

In this section we briefly describe the two major components of e-Spatial™ technology, namely the iSMART™ database development technology and the iSpatial™ Information Server (iSIS), together with their architectural characteristics. More details on the e-Spatial™ architecture can be found in (Bertolotto et al., 2001).

iSMART™ is an intelligent Oracle application database development technology that allows database administrators and software engineers to design, build and deploy spatially enabled (as well as standard) Internet applications without requiring any proprietary or application specific source code. It therefore allows to reduce overall system development and deployment costs. All iSMART™ applications are automatically generated on the fly directly from the Oracle database and can be used and modified on-line without having to log-off existing Internet applications or writing any client application code.

The iSpatial™ Information Server is a Java plug-in component of iSMART™. It consists of a database management application program that enhances the spatial functionality of an Oracle Spatial database.

2.1 iSMART™ Architecture

The iSMART™ development environment relies on a three-tier architecture comprising three main layers, namely the Client Layer, the Application Server Layer, and the Database Layer (see Figure 1).

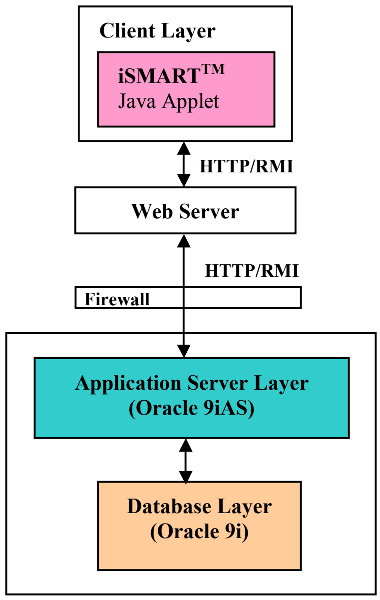

Figure 1. The iSMART™ architecture components.

All communications between the client layer and the database are conducted through the application server layer. The application is executed on the client using an applet that runs in a standard web browser. The applet communicates with the application server using the existing HTTP or RMI networking protocols.

The client layer is a light-weight client machine running a Java Virtual Machine (JVM). The iSMART™ Java applet is a micro thin applet that executes all commands received from a Panel EJB.

The application server currently used by iSMART™ is Oracle 9iAS. This layer contains several Enterprise Java Beans (EJBs) described in the following. Session EJBs are responsible for all communication with the client. As it is a stateful session EJB, an object of this type is instantiated for each user session. This

EJB validates all data submitted by the iSMART™ Applet before passing it to the entity beans.

Panel EJBs are entity beans that query the iSMART™ Application Specific Metadata for all information about the relevant panels (or forms to be displayed on the client applet) from the database and send information back to the iSMART™ EJB which returns it to the client. Each Panel EJB has a hierarchical structure (i.e., it can contain nested sub-panels and objects). Each panel in the user application corresponds to a Panel EJB in the hierarchy. In Figure 2, an example of panel hierarchy corresponding to a user application is shown as it appears in the Visuali development environment (see Section 4). On the right hand side of the screen the user builds panels in a workspace; on the left hand side the panel hierarchy is displayed.

SQL statements are needed to retrieve from the database the information to be displayed on the client site. Pre-defined SQL commands are stored in the iSMART™ Metadata tables of the database. The entity beans responsible for the execution of SQL statements are called SQL EJBs.

The database layer, as described in this paper, can refer to single or multiple database instances. This layer comprises two parts: the iSMART™ application specific metadata and the user data. The first component is a collection of standard database tables. The core behaviour and characteristics as well as the functionality of each user-built application are defined in these metadata tables. All pre-defined SQL statements are also stored in these tables.

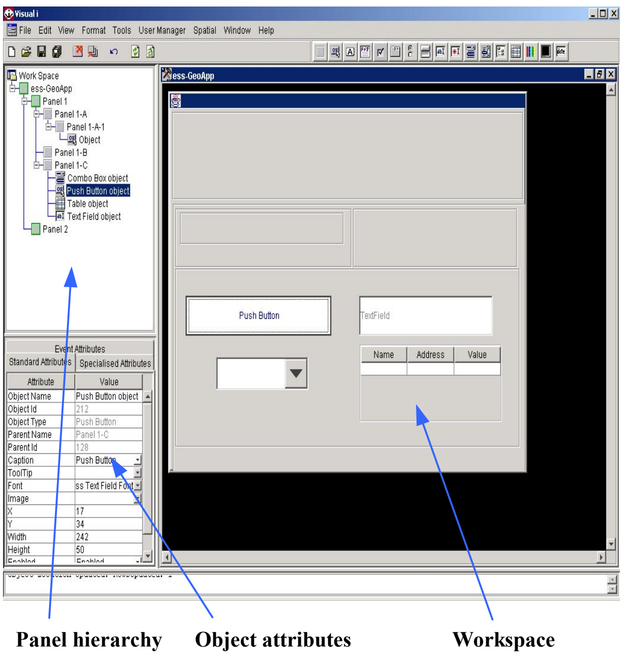

Figure 2. Screenshot of a user application under construction.

2.2 iSIS architecture

Similar to the iSMART™ architecture, the iSIS architecture presents three main components: the Client Layer, the Application Server Layer, and the Database Layer (see Figure 3).

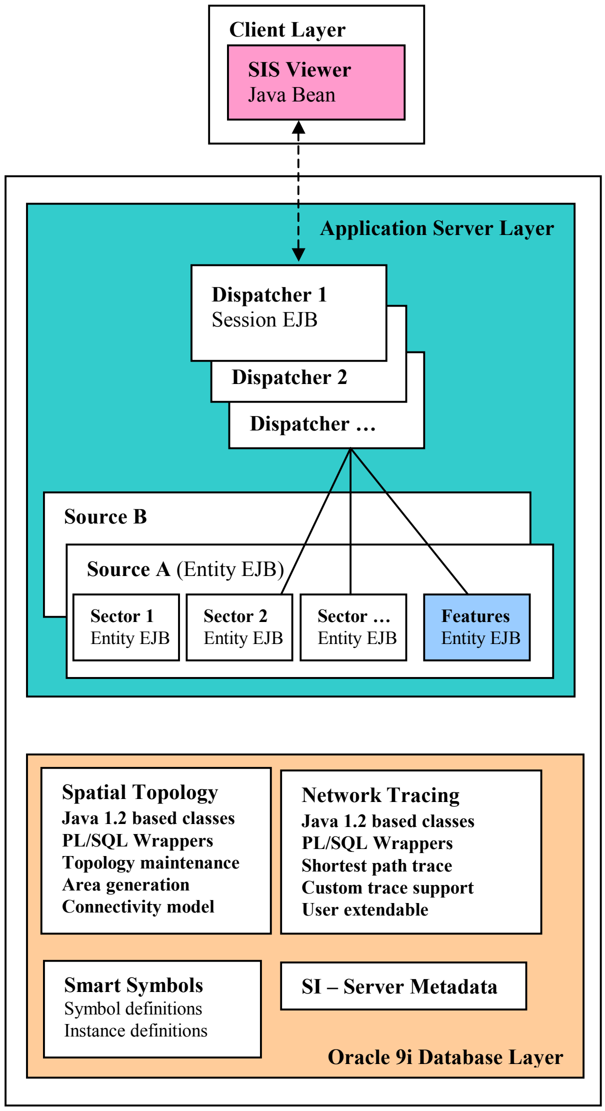

Figure 3. iSIS architecture components.

The iSIS client layer bean is a plug-in that controls the display and manipulation of the vector and raster data within the web browser. The bean requests the specified data from the Dispatcher EJB (on the application server layer described later), which in turn renders all appropriate data for client display using the bean.

The application server layer contains several EJBs. The Dispatcher EJB is a stateless session bean that handles communication with the client and coordinates the display of the different Sector EJBs. These are entity beans that handle the retrieval of the data from the database for a given area (called a sector). Source EJBs are entity beans that control the access to the SI server metadata in the database. This information is used to determine whether the subset of data considered (called source) is a vector or a raster dataset and where the source information can be found. A Feature EJB is an entity bean that controls access to the feature tables (storing the feature symbology) in the database. The feature information is used by the Sectors to display object information.

The iSIS database layer consists of a collection of Oracle database tables and stored procedures written in Java that run inside the Oracle Aurora Virtual Machine. This way, all functions and procedures have fast and efficient access to the information stored inside the database without adding to the network overhead. These functions can be accessed via PL/SQL wrapper code.

The database has four distinct features:

1. *Spatial Topology*: the spatial topology package builds and maintains a topology model in the database for the spatial objects. This topology is built and maintained from the geometry objects in a specified source table (more details on the data model and management of the topological structure are provided in Section 3).

2. *Smart symbols*: they are representations of map symbols that can have an internal connectivity and a number of external ports. The smart symbol definition models the possible states for the internal connections between the ports of the object. The smart symbol instance models the current state of all the internal connections of a given smart symbol. This state information can then be used by the trace modules to conduct a network trace inside the database.

3. *Network Tracing*: the network trace classes are a series of predefined classes that can be extended by the user to perform any type of trace. These classes work with the topology information and the smart symbols to determine connectivity and connection information. The base trace classes use topological connectivity to determine if objects are traceable and connected. Users can easily extend this model to determine if components are traceable by accessing other criteria such as attribute data from secondary tables or other criteria. The definition for the trace algorithms are also stored in a set of metadata tables that can be modified and extended with the use of SQL. This allows the users to create their own trace definition on the fly.

4. *Spatial Information Server Metadata:* since the spatial information server is a generic tool to display spatial information from any table that contains Oracle spatial data columns or Oracle Intermedia images, it requires some information on where it can find these objects and images and how it should access them. This information is stored in the SIS metadata tables.

3. TOPOLOGICAL DATA MODEL

In this paper we describe the spatial data model on which e-Spatial™ technology is based together with the mechanism for guaranteeing the preservation of topological consistency, an essential property for data usability.

The spatial data model used by e-Spatial™ technology is the Oracle Spatial object-relational model. Object-relational models extend classical relational models by introducing object oriented concepts. The advantage of using object-relational systems (instead of object-oriented systems) is that they provide object orientation features while remaining compatible with previous relational applications. Therefore they minimise the need for complex migration of existing data, re-training of personnel and re-writing of application programs.

The basic object features of the Oracle object-relational model are type definition services that extend significantly the domain definition services of relational systems. Oracle Spatial offers the option to use both the relational and the object-relational models. Such a model consists of a schema, a spatial indexing mechanism and a set of operators and functions that facilitate the storage and retrieval of spatial data in an Oracle Database. Spatial queries such as window and range queries as well as spatial joins can be performed using the predefined spatial functions and operators provided.

The spatial attribute of an object (i.e., its geometric representation in a coordinate system) is stored in a single column. A single record (row) is used per geometry instance. Oracle Spatial allows to store vector data in the form of points, lines and polygons together with their topological relationships such as disjoint, overlap, touch, contains, etc. (Egenhofer and Franzosa, 1991). The topological consistency checks (e.g., check if a given object is self intersecting) and topological queries (e.g., are two objects disjoint) are based on pre-defined spatial operators and functions.

Within the iSMART™+iSIS platform topology is completely controlled by the Oracle 9i topology management functionality: if a spatial element is updated, all spatially related elements are automatically changed accordingly to guarantee consistency.

Such functionality is used within the spatial topology component of the iSIS database layer (described in Section 2): the iSIS database *topology engine* builds and maintains the topological relationships of all the geometries (node, arc, boundary, area, region, etc.) within the database itself. The topology engine supports the automatic building of the topology inside the Oracle Spatial database as data (geometries) are bulk loaded into the database (i.e., as vector data is bulk loaded the topological relationships are automatically created inside Oracle geometry tables).

Specific characteristics of this component are:

- Bulk topology creation from existing tables containing Oracle Spatial objects. This function allows topology to be built (rebuilt) from existing data.
- Topology maintenance functions handle changes in the topology when elements in the base table are created, modified or deleted. To ensure data consistency this is usually handled via database triggers, but other custom approaches are possible, depending on the application's needs. The topology system can be configured to combine multiple sources or parts of different sources into a topology layer. Examples include building topologies from certain features from a combination of source tables.
- Multiple topology layers. The system is not limited to just one topology. The system supports an unlimited number of topologies.
- Object connectivity model. The topology can be used as a connectivity model for linear topologies. When other applications change the source tables, the

connectivity is maintained automatically since it is dependant on the topology, which in turn is automatically maintained.

- Client independent. Regardless what desktop software is being used to edit the spatial data, topology and connectivity are maintained automatically inside the database.
- Dynamic generation of areas with support for holes from the boundaries in the topology. Will act differently depending on the signal you trace.

A topology wizard within iSIS allows users to assemble, build and maintain topology. iSMART^{TM} also provides a set of generic tools to extend and customise the topology wizards to each organisations specific requirements.

4. DEVELOPMENT TOOLS

Additional tools are provided for facilitating the building of applications that exploit spatially enabled datasets. These tools are particularly useful for non-expert programmers as they allow the creation of features together with their symbology, the importing of digital imagery and incorporate the drag-and-drop methodology to graphical user interface design.

The *Pyramid Builder* is a utility designed to load scanned aerial/satellite photos and topographic maps into the database. The loaded imagery is then available for heads up digitising. This image loading utility follows the long standing approach to digital image handling within existing dedicated image processing systems by pre-processing the image data into multi-resolution representations (Carswell and VdLaan, 1992), hence the name "Pyramid Builder". The idea being to eliminate the need to resample the image at run time by loading the best fit resolution image to the zoom factor currently active within the web browser.

e-Spatial^{TM} technology advances this trend by further subdividing the individual reduced resolution image files into tiles of a pre-defined pixel size. Thus, when a web browser is viewing a particular geographic location, only those individual tiles at the requested zoom factor will be physically sent over the network, and not the entire image file. This of course is essential to real-time viewing of digital imagery due to their inherent size restrictions: a typical black&white aerial photo scanned at 30um (850dpi) resolution requires approximately 60mb of disk space (uncompressed) (Kern and Carswell, 1994), while colour imagery at the same scanning resolution will require three times as much space, i.e. 60mb for each of the three colour bands; red, green, and blue.

The *Feature builder* is a utility that allows users to define new data sources and the feature classes that the source controls. All feature characteristics (e.g., colour, weight, pattern, etc.) are defined using this utility. Also incorporated within the Feature Builder is the capability to add events to specific feature digiti sing operations. For example, in a land parcel application, when a "Property Centroid" feature first gets digiti sed, an event can be associated with this operation that calls a pre-defined SQL query to populate the "Property ID" column in the user attribute "Property" table and subsequently display this attribute value as the centroid text in the graphics. Alternatively, if a Property Centroid gets deleted from the graphics, an SQL query previously attached as an event to this type of an operation (on this type of feature) can delete all the user attributes associated with this centroid feature.

The iSMART^{TM} application builder, called Visual*i*, has been designed to allow any user to build a customised graphic user interface for both standard MIS and spatial applications without writing any source code. This application development incorporates drag-and-drop functionality to add objects, such as buttons, tables, combo boxes, etc. to panels and sub-panels in a hierarchical fashion (see Figure 3 for a screen shot example of a user application under construction).

Figure 4. A LIMS application developed using e-Spatial^{TM} technology and viewed on a PDA.

Visual*i* allows the building of "business rules" that are attached as events to the buttons and other GUI objects contained within the application panels. For example, a business rule can be created and attached as a "mouse pressed" event on a "Property Value" button object that, when pressed, subsequently shades all the property polygons according to their "Property Value" attribute.

SQL commands are also created within the Visual*i* application builder through the use of the SQL Command Editor. Using drag-and-drop operations, SQL queries are built where table columns can be dynamically attached to their respective "textfield" objects on the application panels. Thus both retrieving data from the database and inserting/updating data is easily accomplished.

5. CONCLUSIONS

In this paper we presented the advantages of the geospatial data handling approach of e-Spatial^{TM} technology that allows for effective development of web-based and mobile spatial applications within an Oracle database environment. Such a technology utilises the Oracle Spatial object-relational model for on-line storage, retrieval and query of spatial data. This data model presents the great advantage of automatically preserving topological consistency, an essential property for data exchange and usability between remote sites (such as in web-based and mobile applications). Therefore, if a spatial element is updated (for example, by a remote user), all spatially related elements are automatically changed accordingly to guarantee consistency.

From a data management point of view, the greater advantage of using Oracle Spatial is that it handles spatial and non-spatial data in a completely unified manner. Both types of data are stored in the same database: the geometric representation of an object (e.g., in the form of a point, line or polygon) shares the same table with the other attributes of the object. This contrasts typical spatial handling systems (such as commercial GIS) in which spatial properties of objects are stored in files that are managed by a file management system (such as shape files, design files, etc.) while attribute data is stored in a commercial database.

An important concern for the design and development of contemporary and next-generation information systems relates to interoperability, i.e., the capability of autonomous systems to exchange data and to handle processing requests by means of a common understanding of data and requests (Sondheim et al., 1999). Specifications on the conceptualisation of spatial entities have been provided by the OpenGIS Consortium. The Oracle Spatial object-relational model (upon which the e-Spatial™ data model relies) conforms to interoperability and standardisation requirements as it corresponds to the "SQL with Geometry Types" implementation of spatial feature tables described in the OpenGIS ODBC/SQL specification for geospatial features (OGC, 1996).

e-Spatial™ technology has been developed within the Oracle Spatial unified environment. Its web-enabling application architecture component iSMART™ and its Java plug-in iSIS represent a dramatic extension of the functionality inherent to an Oracle Spatial database. Indeed, it allows for real-time spatial data display, collection, editing, manipulation and query simply using a standard web browser. Advanced spatial data handling and management functions, typically provided by specialised proprietary application packages, can now be exploited by Oracle Spatial users without having to install any application-specific source code. This has been achieved by implementing all management information and spatial analysis functions as stored Java procedures. Therefore, applications can be deployed using standard browser technology on any device that supports the Java Virtual Machine, thus providing full support for wireless, PDAs and other mobile devices.

An example of web-based application developed using Spatial™ technology is the on-line Land Information Management System (LIMS) serving 125,000 farmers in the territory of the Republic of Ireland (developed for the Irish Department of Agriculture). The implementation and testing to make such a system available on hand-held mobile devices (Figure 4) are currently being finalised (Bertolotto et al., 2002).

Finally, existing systems have limited ability to switch on/off specific subsets of spatial data. Using e-Spatial™ technology users can select the particular data they wish to view and query. This ability to define exactly what the users need to query greatly improves the speed with which the individual query is completed, a critical issue for mobile applications. This characteristic conforms to the information-on-demand approach discussed by several authors in the context of web-based vector map generali sation (Bertolotto and Egenhofer, 2001; Buttenfield, 1991; Cecconi and Weibel, 2000). The potentialities of the application of e-Spatial™ technology in this regard are currently under consideration.

Furthermore, greater flexibility presents greater opportunities for personalisation: for example, users could build personal profiles and the system could automatically push personalised context-aware information – a topic of future work we are also investigating with this technology.

6. REFERENCES

Bertolotto, M., Carswell, J. D., McGeown, L., McMahon, J., 2002. Web-Based Spatial Information Management Systems. In: *Proceedings 95th Annual CIG Conference*, Ottawa, Canada.

Bertolotto, M., Carswell, J. D., McGeown, L., McMahon, J., 2001. iSMART™+i-Spatial™ Information Server: Deploying Integrated Web-Based Spatial Applications Within an Oracle Database Environment. In: *Proceedings of the International Workshop on Web Geographical Information Systems (WGIS2001)*, IEEE CS Press, Kyoto, Japan.

Bertolotto, M., Egenhofer, M., 2001. Progressive Transmission of Vector Map Data over the World Wide Web. *GeoInformatica*, 5(4), pp. 345-373.

Buttenfield, B., 1999. Progressive Transmission of Vector Data on the Internet: A Cartographic Solution. In: *Proceedings 18th International Cartographic Conference*, Ottawa, Canada.

Carswell, J., VdLaan, F., 1992. Tools for Large Scale Operational Raster Processing. In: *Proceedings EGIS92*, Munich Germany.

Cecconi, A., Weibel, R., 2000. Map Generalization for On-demand Web Mapping. In: *Proceedings GIScience2000*, Savannah, GA.

East, R., Goyal, R, Konovalov, A., Rosso, A. , Tait, M., Theodore, J., 2001. The Architecture of ArcIMS, a Distributed Internet Map Server. In: *Proceedings SSTD'01*, *Lecture Notes in Computer Science n. 2121*, Springer-Verlag, pp. 387-403.

Egenhofer, M., Franzosa, R., 1991. Point-set topological spatial relations. *International Journal of Geographic Information Systems*, 5(2), pp. 161-174.

Kern, P., Carswell, J. D., 1994. An Investigation into the Use of JPEG Image Compression for Digital Photogrammetry: Does the Compression of Images Affect Measurement Accuracy. In: *Proceedings EGIS94,* Paris, France.

OGC (OpenGIS Consortium), 1996. The OpenGIS abstract specification. http://www.opengis.org/public/abstract.html (accessed March 2002)

Sondheim, M., Gardels, K., Buehler, K., 1999. GIS Interoperability. In: *Geographical Information Systems: Principles, Techniques, Management and Applications,* P. Longley, M.F. Goodchild, D.J. Maguire and D.W.Rhind (eds.), Second Edition, Cambridge, GeoInformation International, pp. 347-358.