

2001-12

Deploying Integrated Web-based Spatial Applications within an Oracle Database Environment

James Carswell

Technological University Dublin, james.carswell@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/dmcccon>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Carswell, J. (2001) Deploying integrated web-based spatial applications within an Oracle database environment. *International Workshop on Web Geographical Information Systems (WGIS2001)*, IEEE CS Press, Kyoto, Japan, December, 2001.

This Conference Paper is brought to you for free and open access by the Digital Media Centre at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, vera.kilshaw@tudublin.ie.



2001-12-01

Deploying integrated web-based spatial applications within an Oracle database environment

James D. Carswell

Dublin Institute of Technology, jcarswell@dit.ie

Recommended Citation

Carswell, James D.: Deploying integrated web-based spatial applications within an Oracle database environment. International Workshop on Web Geographical Information Systems (WGIS2001); IEEE CS Press; Kyoto, Japan; December, 2001.

This Conference Paper is brought to you for free and open access by the Digital Media Centre at ARROW@DIT. It has been accepted for inclusion in Articles by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.



iSMART™+i-Spatial™ Information Server:

Deploying Integrated Web-Based Spatial Applications Within an Oracle Database Environment

Michela Bertolotto
Department of Computer Science
University College Dublin
Ireland
michela.bertolotto@ucd.ie

James Carswell
Digital Media Centre
Dublin Institute of Technology
Ireland
jcarswell@dmc.dit.ie

Liam McGeown, John McMahon
Peter Thijs
e-Spatial Solutions Limited
Dublin, Ireland
info@e-spatialolutions.com

Abstract

In this paper, we describe the architectural and functional characteristics of e-Spatial™ technology, comprising an innovative software package that represents a timely alternative to traditional and complex proprietary GIS application packages. The two main components of the package, developed by e-Spatial Solutions, are the iSMART™ database development technology and the i-Spatial™ Information Server (iSIS), both implemented within an Oracle 9i Spatial database environment. This technology allows users to build and deploy spatially enabled or standard Internet applications without requiring any application-specific source code. It can be deployed on any Oracle supported hardware platform and on any device that supports the Java Virtual Machine, thus providing full support for hand-held and mobile devices.

1. Introduction

Traditional GIS applications require the use of proprietary application-specific software. 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.

Besides reducing the complexity of system management by eliminating the hybrid architecture of typical GIS data models, additional specific benefits (from the point of view of data management) of using this unified approach include:

- The possibility for users to access full function spatial information systems based on industry standards with an open interface to all data (e.g., SQL);
- The capability of storing spatial data in enterprise-wide database management systems thereby enabling the spatial

development of several more enterprise applications;

- The complete integration of management information and spatial data repositories that allow for the development of geo-spatial applications providing additional analysis and reporting functionality.

e-Spatial's™ unique technology has been developed within this unified environment. iSMART™ and its Java plug-in component iSIS dramatically enhance the functionality of an Oracle Spatial database by allowing real-time display, collection, editing, manipulation and query of spatial data using a standard Web browser.

Therefore, by utilising e-Spatial™ technology, Oracle Spatial users can enjoy a full range of advanced spatial data handling and management functions that are typically provided by specialized proprietary application packages. These functions, together with all data stored in the enterprise database, can be accessed from any PC or mobile device using functionality built into the database itself. Indeed, one of the greatest advantages of this software is the fact that it is seamlessly integrated 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.

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 visualize, query and manipulate spatial data without requiring any specific programming or database management background. To this purpose, additional tools and user-friendly environments to build and customize graphic user interfaces that facilitate interaction with the iSMART™+iSIS platform have also been developed.

Finally, an important concern for the design and development of contemporary and next-generation (web-based) information systems relates to interoperability issues. Interoperability refers to the capability of autonomous systems to exchange data and to handle processing requests by means of a common understanding of data and requests [9]. In GIS, data modelling is an important issue within the context of interoperability: agreement at the representation level is essential for exchanging spatial data. Specifications on the conceptualisation of spatial entities and the space/time reference systems where they reside have been provided by the OpenGIS Consortium [7,8].

The data model used by *e-Spatial*TM technology is the Oracle Spatial object-relational model. Such a model conforms to interoperability and standardization 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.

The remainder of this paper is organized as follows. Section 2 describes the components of the *iSMART*TM and *iSIS* architectures. Main advantages of the architecture, including scalability and reliability are discussed in Sections 3. Section 4 gives an overview of the additional tools that allow users to build and customise their applications using *e-Spatial*TM technology. To provide an example of application, we dedicate Section 5 to the description of the land information management system that has been developed for the Irish Department of Agriculture. Finally, Section 6 presents some concluding remarks.

2. Architecture components

In this section we describe the two major components of *e-Spatial*TM technology, namely the *iSMART*TM database development technology and the *i-Spatial*TM Information Server (*iSIS*), together with their architectural characteristics (Sections 2.1 and 2.2).

*iSMART*TM 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 significantly reduces the overall system development and deployment costs.

All *iSMART*TM applications are automatically built in real-time 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 *i-Spatial*TM Information Server is a Java plug-in component of *iSMART*TM. It consists of a database management application program that adds spatial functionality to an Oracle Spatial database.

2.1 *iSMART*TM architecture

The *iSMART*TM 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).

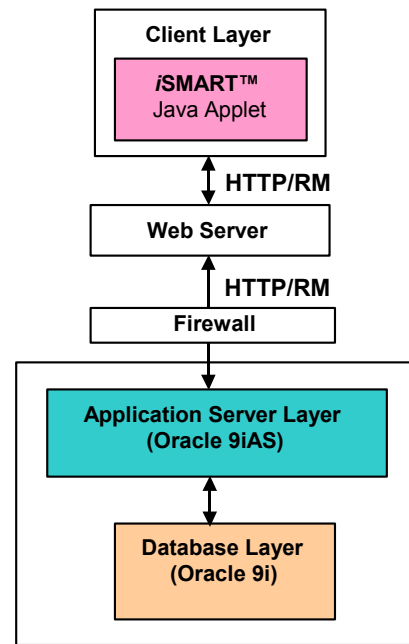


Figure 1. The *iSMART*TM architecture components

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 on a standard web browser. The applet communicates with the application server using the existing HTTP or RMI networking protocols.

2.1.1. Client layer. The client layer is a light-weight client machine running a Java Virtual Machine (JVM). The *iSMART*TM Java applet is a micro thin applet that executes all commands received from a Panel EJB. Upon initialisation the *iSMART*TM applet asks for a username, password, application server URL and port number. The connection information can be defined as start-up parameters for the applet and prevent the login dialog from appearing. For security reasons, however, this start-up method is not recommended.

2.1.2. Application server layer. The application server currently used by *iSMART*TM is Oracle 9iAS. This layer contains several Enterprise Java Beans (EJBs) illustrated in Figure 2.

The *iSMART*TM session EJB is 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*TM Applet before passing it to the entity beans.

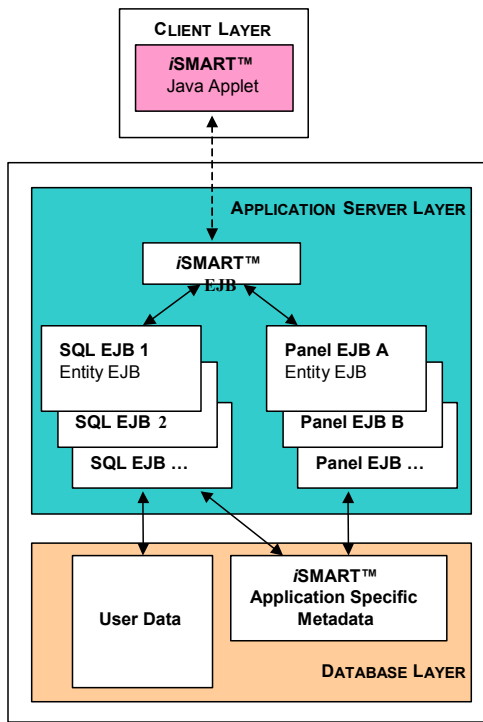


Figure 2. Application server layer EJBs and communications.

Panel EJBs are entity beans that, when initialised by the *iSMART™* EJB, query the *iSMART™* Application Specific Metadata for all information about the relevant panels (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 and contains details only about its objects and immediate sub-panels. Each panel in the user application corresponds to a Panel EJB in the hierarchy. In Figures 3, 4, and 5 an example of panel hierarchy is shown as it appears in the Visuali environment described in Section 4.

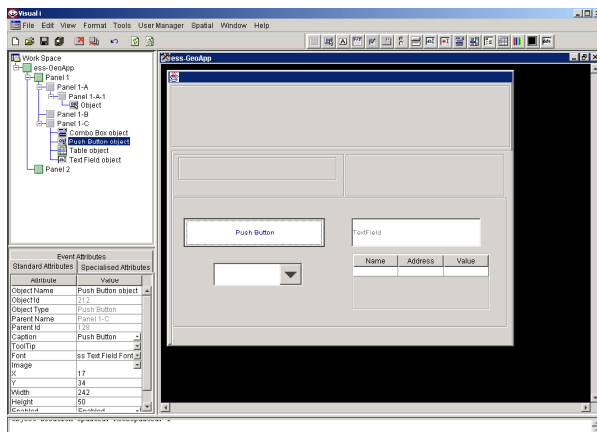


Figure 3. Screenshot of a user application under construction.

Figure 3 shows a screenshot of the user application. On the right hand side of the screen, the user can build panels in a workspace (Figure 4); while on the left hand side, the panel hierarchy is displayed (Figure 5).

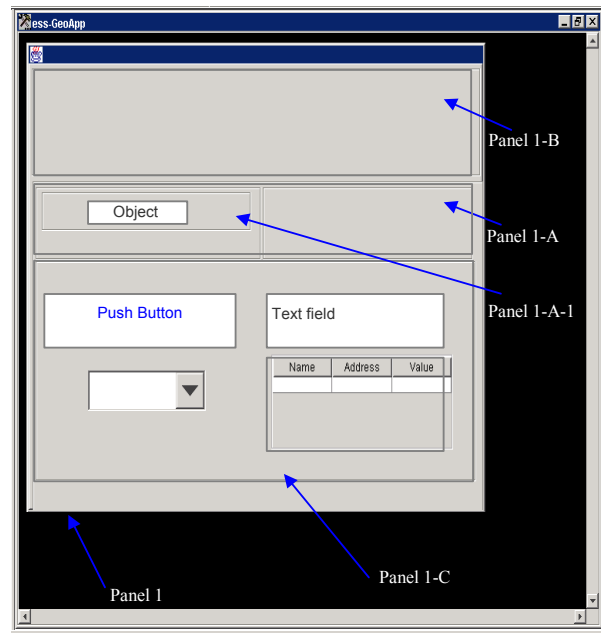


Figure 4. Workspace where users build panels

SQL statements are needed to retrieve from the user database the information to be displayed on the client site. For example, when the user presses a button, some values may need to be inserted or updated in the user database. In this case, an event is associated with that particular button to call an SQL statement. Pre-defined SQL commands are stored in the *iSMART™* Metadata part of the database. The entity beans responsible for the execution of SQL statements are called SQL EJBs.

When an event is triggered by a user action, the *iSMART™* EJB retrieves the SQL identifier (id) of the command that is to be executed and instantiates an SQL EJB passing such an id to the corresponding constructor method. The SQL EJB then retrieves the correct command from the *iSMART™* Metadata table and in turn sends it to the Oracle database engine for execution.

The result of the query is returned to the *iSMART™* EJB, which communicates it to the client. If the same query is run again, the *iSMART™* EJB will re-invoke the SQL EJB which was previously instantiated. The parsed version of the previously executed SQL command will be cached by the SQL EJB thereby eliminating the need to re-interpret the query. Bind variables are used to optimise the execution of the SQL query and to assign values to object variables.

2.1.3. Database layer. The database layer, as described in this paper, refers to a 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.

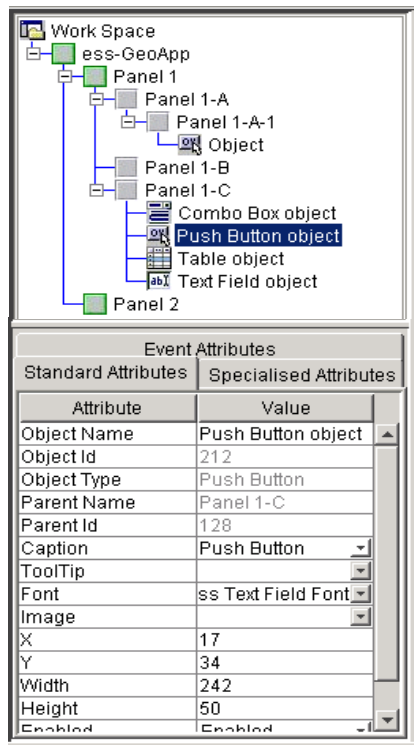


Figure 5. Panel hierarchy corresponding to the user-defined panels displayed in Figure 4

2.2 iSIS architecture

Similarly to the iSMART™ architecture, the iSIS architecture presents three main components: the *Client Layer*, the *Application Server Layer*, and the *Database Layer* (see Figure 6). Each component is described in the following sections.

2.2.1. Client layer. The iSIS client layer bean is a Java plug-in that controls the display and manipulation of the vector and raster data. 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.

2.2.2. Application server layer. 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. When the Dispatcher EJB receives a display request, it hands such a request over to the Sectors responsible for the portion of the map that is being viewed. It then combines the results and sends them back to the client.

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 Source EJB also controls which feature table (storing the feature symbology) is used to represent the source information.

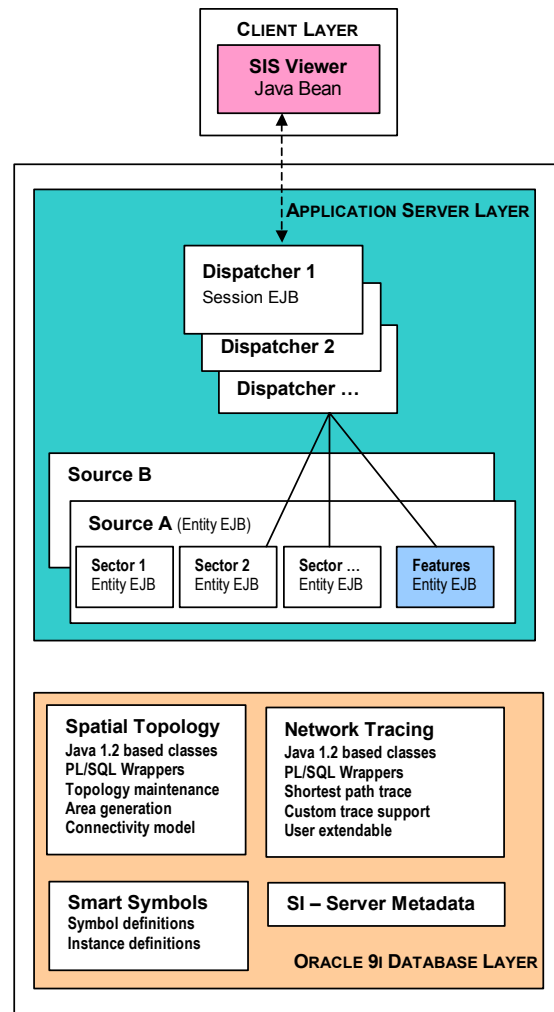


Figure 6. iSIS architecture components

A Feature EJB is an entity bean that controls access to the feature tables in the database. The feature information is used by the Sectors to display object information. The Feature tables can hold, for example, data pertaining to how an object should display at varying zoom levels, colour, style, weight, etc.

Finally, Sector EJBs are entity beans that handle the retrieval of the data from the database for a given area (called a sector). These Sectors in turn send the information to the Dispatcher which combines all required sectors for return to the client.

2.2.3. Database layer. 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. PL/SQL is the procedural extension of SQL defined by Oracle. 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. Functions include:

- 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.
2. *Smart symbols*: these are representations of network components. These symbol components combine the graphical representation to an internal state representation for the symbol. The internal state consists of the following,
- Port location. A port is a location where outside connections are allowed to the symbol. These ports exist at certain locations on the symbol.
 - Internal connectivity. These connections model the internal connections between the ports on the symbol. These connections have a number of properties such as:
 - Open/closed mask. A flag to indicate the state of the connection
 - Network mask. A mask to indicate what network types are supported by this connection. In a number of applications more than one network is using the same physical wires. TV networks have TV signal, cable modem signals, power etc. all running over the same networks and through the same components. In the connectivity model these components

will act differently depending on the signal you trace.

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 is 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. Architecture scalability and reliability

3.1 Database layer scalability

The *iSMART™* system retrieves operational metadata from a single database, but can access user-based information across a number of different databases. Similarly, *iSIS* transparently supports the display of graphical data from different databases. Each individual database server could thus be optimized for the data it controls. Aside from this, the usual database tuning options are still available since only standard database formats and structures are being used to store the Application Specific Metadata.

3.2 Application server layer scalability

The application server layer can be scaled in a number of ways. In its most advanced form, the application server supports the creation of a single virtual application server that is built across a number of physical machines. The application server then handles all the load balancing and connection issues arising from this set-up. Alternatively a number of parallel application servers could be created and the user connections can be distributed across those machines.

The Dispatchers in *iSIS* can also handle load balancing by sending the process requests to Sectors residing on different machines.

3.3 Client side scalability

The *iSMART™* applet is a micro thin applet. The nature of the applet is to execute commands received from the application server layer and therefore requires no application specific coding on the client for customisation. Whether the Application Specific

Metadata contains information for one or many applications, the client applet never gets any larger. The *iSMART™* applet serves the client layer with exactly what it needs - when it needs it, eliminating the need for the client to interpret compiled application specific source code. This dramatically reduces the storage and memory requirements of the client layer.

3.4 Reliability

To ensure reliability for the database layer, Oracle's fail-safe mechanisms are utilised. Parallel server technologies can also be used.

The Oracle 9iAS application server handles fail-over between nodes transparently and hence controls application server reliability.

4. Additional utilities

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.

4.1 The Pyramid Builder

The Pyramid Builder is a utility designed to load scanned aerial/satellite photos and topographic maps into the database (see Figure 7). The loaded imagery is then available for heads up digitising.

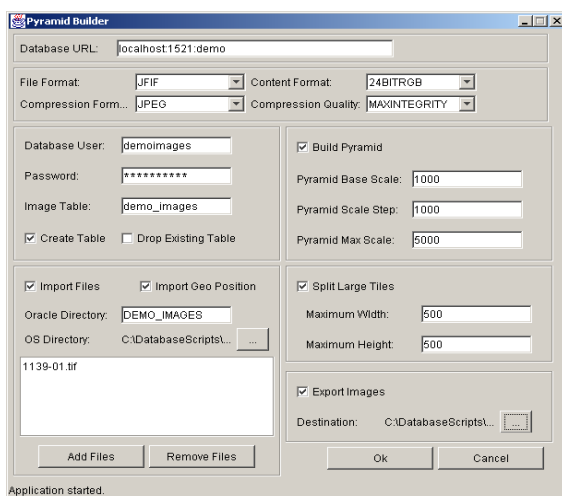


Figure 7. The Pyramid Builder.

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 [3], hence the name "Pyramid Builder". The idea is 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™ 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. For example, a typical black & white aerial photo scanned at 30µm (850dpi) resolution requires approximately 60Mb of disk space (uncompressed) [6]. Colour imagery, of course 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.

4.2 The Database Topology Engine

The *iSIS* database topology engine builds and maintains the topological relationships of all 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 the Oracle standard and geometry tables). This functionality is unique to *e-Spatial™* technology.

4.3 Topology Wizard

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

4.4 The Feature Builder

The Feature builder is a utility that allows users to define new data sources and the feature classes that the source controls (see Figure 8). 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 digitizing operations. For example, in a land parcel application, when a "Property Centroid" feature first gets digitized, 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.

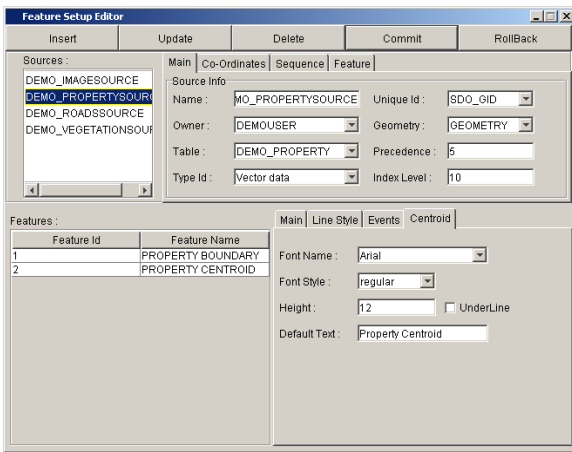


Figure 8. The Feature Builder

4.5 The Application Builder

The *iSMART*TM application builder, called Visuali, 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 Figures 3,4,5 for screen shot examples of a user application under construction).

Visuali 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 subsequently shades all the property polygons according to their “Property Value” attribute.

SQL commands are also created within the Visuali application builder through the use of the SQL Command Editor (see Figure 9).

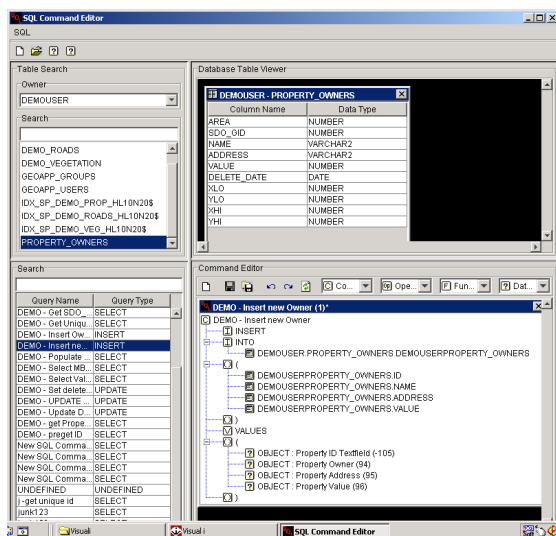


Figure 9. The SQL Command Editor

Using drag-and-drop operations, SQL queries are built where table columns can be dynamically attached to their respective “text field” objects on the

application panels. Thus both retrieving data from the database and inserting/updating data is easily accomplished.

5. Land information management system

In this section, we describe one of the applications that have been implemented using *e-Spatial*TM technology.

The LIMS application, (known as iMAP), developed for the Irish Department of Agriculture, utilises the Spatial Information Server platform to deliver a spatially enabled Internet solution for the tracking and management of land information based on land usage, land classifications and land ownership changes over time. Figures 10 and 11 illustrate screenshots of such an application as viewed on a standard web browser and on a PDA device, respectively.

The developed land information management system provides a seamless Oracle 9i Spatial database environment for the combination of multiple land information datasets. Integrated database topology ensures the integrity of the topology relationships within the spatial database. The normal edit (i.e. create, modify, and delete) and spatial analysis functions associated with a traditional GIS based land management applications, are deployed as Java stored procedures in the Oracle Spatial database.

The implemented functionality includes:

- Vector data manipulation (pan, scroll, zoom, locate, window queries, etc.);
- Vector and attribute data entry and editing;
- Spatial and attribute data analysis (polygon overlay and network tracing);
- Thematic mapping (display of query results with assigned symbology);
- Raster/Vector integration.



Figure 10. The land information management system viewed on a standard web browser.



Figure 11. The land information management system viewed on a PDA device.

6. Concluding remarks

This paper presents a revolutionary technology that extends Oracle Spatial databases by adding spatial information management functionality that can be used within web-enabled applications accessed via any device that supports the Java Virtual Machine, e.g., standard web browsers, PDAs and 3G mobile phones. A major advantage of this technology is that it does not require the use of complex GIS proprietary application specific software.

Furthermore, no application code is ever installed on the client side, thereby eliminating any redundant code. The client applet never exceeds 35k and supports an unlimited number of users and an unlimited number of applications. 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.

Existing systems have no 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, they 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 switch on and off data and also define exactly what the users need to query greatly improves the speed with which the individual query is completed.

This characteristic also conforms to the information-on-demand approach discussed by several authors in the context of web-based vector map generalization

[1,2,4]. The potentialities of the application of e-Spatial™ technology in this context are currently under investigation.

From a data management point of view, a critical issue involves guaranteeing the preservation of topological consistency [5]. 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, another advantage of the e-Spatial™ technology relates to its ease of use. Indeed, it includes a user-friendly environment that allows building and customising the application by means of drop-drag functionality without requiring any particular programming skill. In this way, even non-expert users can easily build their own spatially enabled application.

7. References

- [1] M. Bertolotto and M. Egenhofer, 2001, "Progressive Transmission of Vector Map Data over the World Wide Web", *GeoInformatica* (in press).
- [2] B. Buttenfield, 1999, "Progressive Transmission of Vector Data on the Internet: A Cartographic Solution", *Proceedings 18th International Cartographic Conference*, Ottawa, Canada.
- [3] J. Carswell, F. VdLaan, 1992, "Tools for Large Scale Operational Raster Processing", *Proceedings EGIS92*, Munich Germany.
- [4] A. Cecconi and R. Weibel, 2000, "Map Generalization for On-demand Web Mapping", *GIScience2000*, Savannah, GA.
- [5] M. Egenhofer and R. Franzosa, 1991, "Point-set topological spatial relations", *International Journal of Geographic Information Systems*, 5(2), pp. 161-174.
- [6] P. Kern and J. D. Carswell, 1994, "An Investigation into the Use of JPEG Image Compression for Digital Photogrammetry: Does the Compression of Images Affect Measurement Accuracy", *Proceedings EGIS94*, Paris, France.
- [7] OGC (OpenGIS Consortium), 1996a, *The OpenGIS guide, introduction to interoperable geoprocessing, Part 1*, <http://www.opengis.org/guide/guide1.html>.
- [8] OGC (OpenGIS Consortium), 1996b, *The OpenGIS abstract specification*, <http://www.opengis.org/public/abstract.html>.
- [9] M. Sondheim, K. Gardels, and K. Buehler, 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.

For further information please

e-Spatial Solutions Limited
 Jabil Building
 Clonshaugh Industrial Estate
 Dublin 17, IRELAND
 Tel: +353 1 867 5500
 Fax: +353 1 867 5599
 Email: info@e-spatialolutions.com

The marks e-Spatial, i-Spatial and iSMART are trademarks of e-Spatial Solutions Limited. Use of these trademarks in any form, or in a descriptive or generic context, without the express consent of e-Spatial Solutions Limited is strictly prohibited.