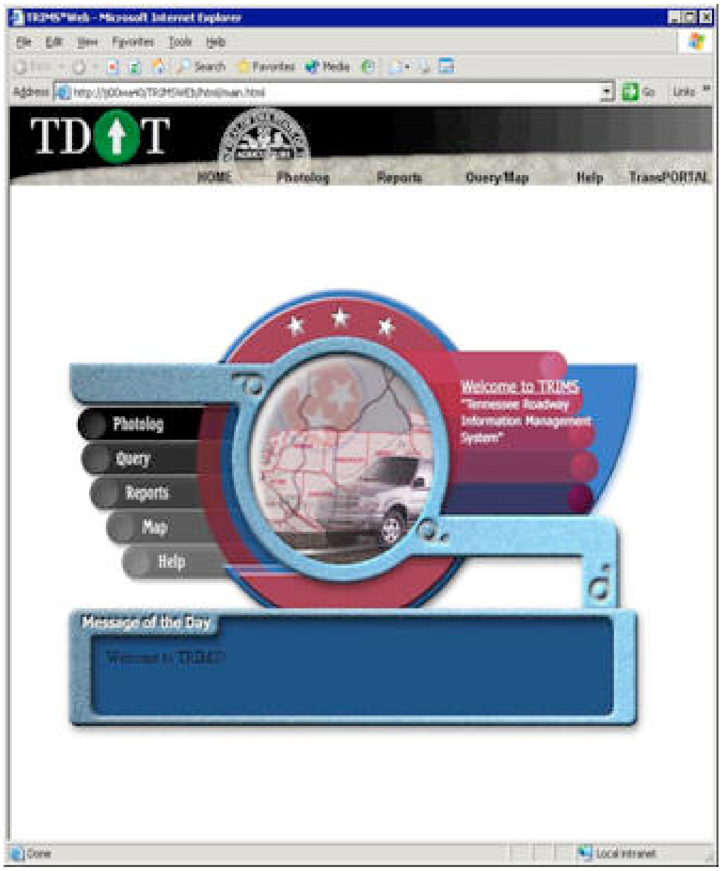
Tennessee Roadway Information Management System Web - TRIMSWeb

By Radhaji Mani

Directions Magazine (Apr 22, 2005)

Overview

TRIMSWeb is the Web-based Tennessee Roadway Information Management System (TRIMS) developed for the Tennessee Department of Transportation (TDOT) and managed by the TDOT GIS. This system is intended for all TDOT Intranet users and is specifically important to the TDOT planners and engineers. It is based on the requirements for the TRIMS client-server application, which has been in production for some time. TRIMSWeb is scalable, although it primarily focuses on delivering the core functionality of TRIMS in the web environment. The architecture of TRIMSWeb is built with three key user requirements in mind: a) ease of administration; b) performance; and c) scalability.



Welcome to TRIMS
"Tennessee Roadway Information Management System"

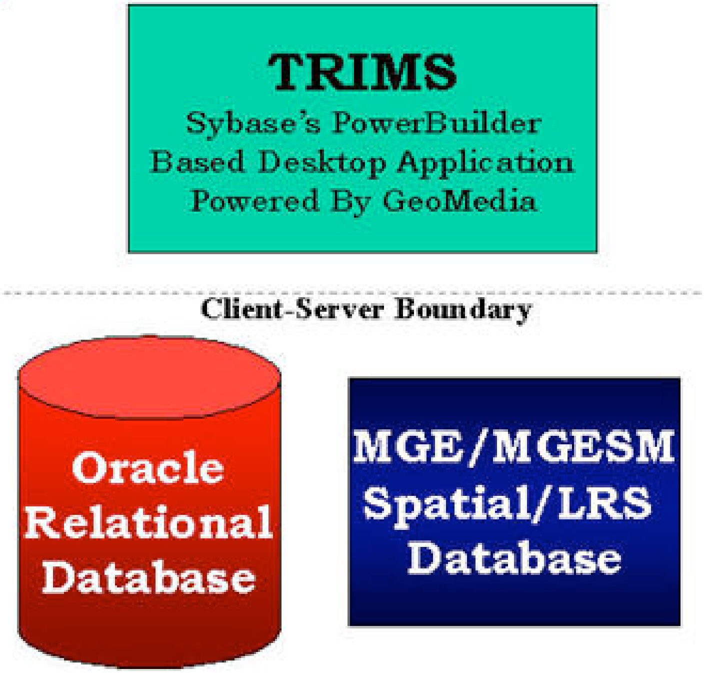
Welcome to TRIMS!

(Source: GISbiz, Inc.)

TRIMS

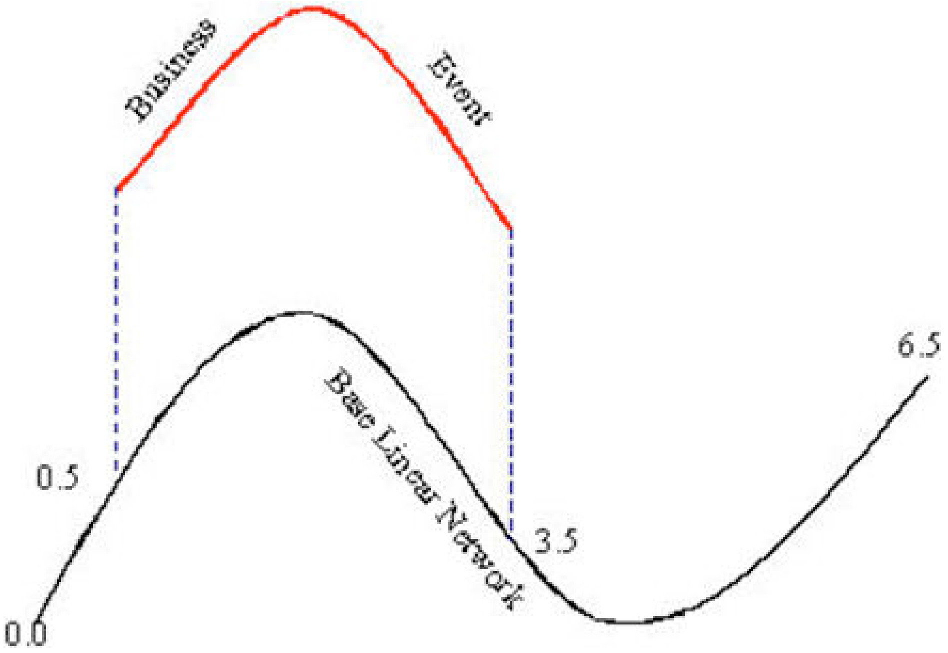
To understand the benefits of TRIMSWeb, you need to understand the functionality offered by TRIMS and its underlying architecture.

TRIMS is a complex two tier application, whose client component is written using Sybase's PowerBuilder. The server component is the relational and spatial data repository. On the server, Oracle is used as the relational database management system and Intergraph's Modular GIS Environment (MGE) and MGE Segment Manager (MGESM) are used as the spatial repositories.



(Source: GISbiz, Inc.)

The TRIMS client is a robust desktop application that provides the ability to query the TRIMS database and generate reports and maps. All attribute queries are performed against the Linear Referencing System (LRS) stored in the database, and all maps are generated using Intergraph's MGESM LRS engine.



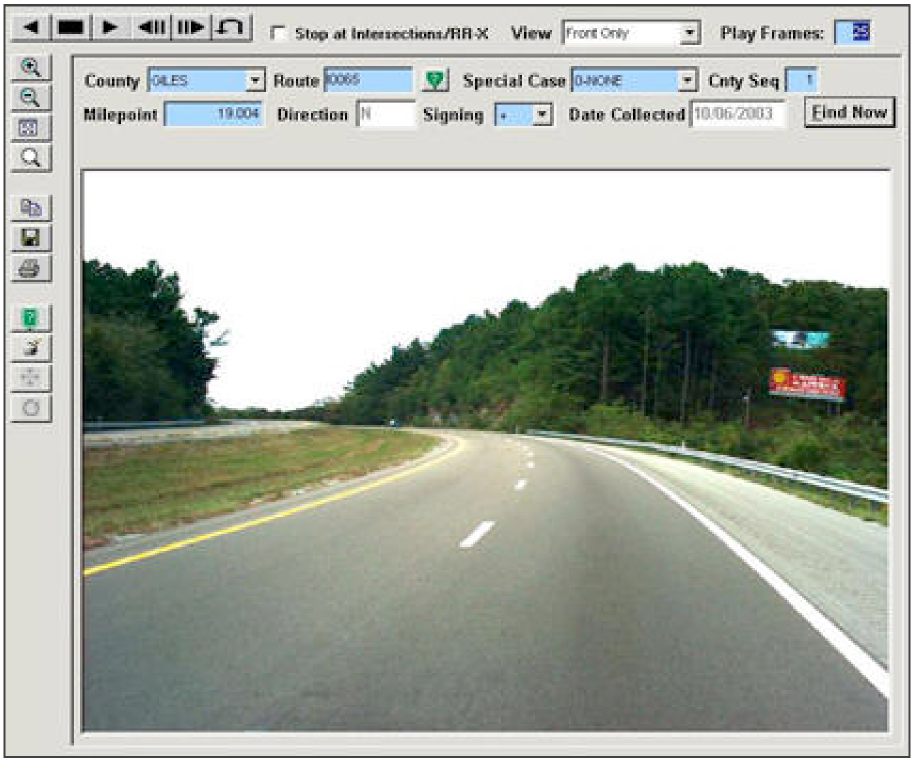
(Source: GISbiz, Inc.)

The TRIMS relational database stores linear events using a unique identifier for the linear segment, as well as its related 'begin' and 'end' log mile values. The unique identifier corresponds to a single base linear network.No spatial geometry is stored in the TRIMS relational database and is derived entirely on-the-fly using the MGESM LRS engine during map generation. This scheme utilizes the base linear network stored within MGESM, and the dynamic event geometry is generated using the unique event identifier and its related begin/end log mile values.

Data gathering for the TRIMS base linear network begins in the field. Data is gathered by physically driving the roads to capture "real world" coordinates. The captured data is then post processed before being stored inside MGESM.Incremental updates are performed to this network on a periodic basis to reflect the physical changes in the base linear network.

Transportation events that occur are captured into the relational database and referenced using the base linear network identifier and the appropriate begin/end log miles where the events occurred. This is also an incremental process and periodic inserts and updates are performed into the event database as new events occur or old events are updated respectively.

The TRIMS client application provides the ability to query and generate maps on a single event table as well as multiple event tables. The key advantage of this application is its ability to query multiple event tables and generate maps by overlaying the results of these events. This is performed using the dynamic segmentation capability of MGESM, as each event table may be segmented differently based on the base linear network.



□ Stop at Intersections/RR-X

View: Front Only

Play Frames: [illegible]

County: GILES

Route: 0065

Special Case: 0-NONE

Cnty Seq: 1

Milepoint: 19.004

Direction: N

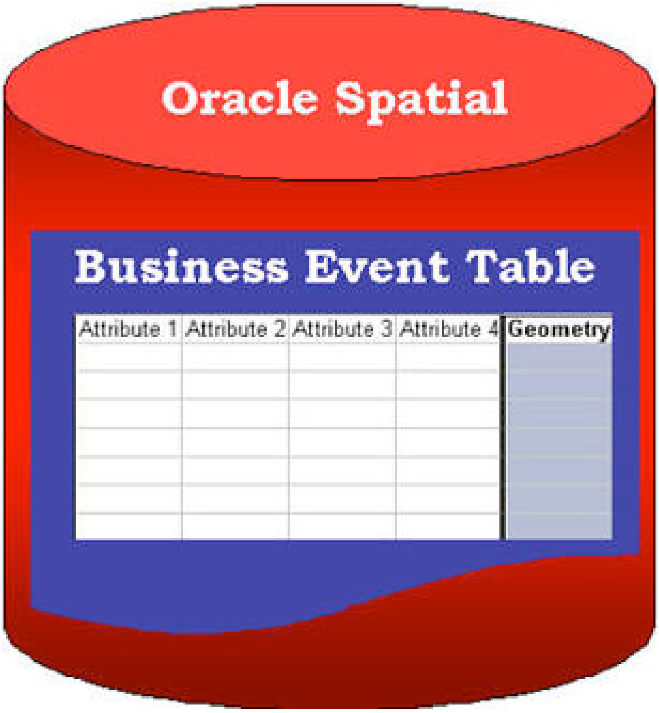
Signing: +

Date Collected: 10/06/2003

(Source: GISbiz, Inc.)

The TRIMS client application uses industry standard database connectivity for attribute queries, and the GeoMedia engine for spatial queries. The GeoMedia mapping engine is used to display base maps, as well as spatial data, generated after dynamic segmentation.

In addition to performing queries for reports and maps, the TRIMS client application provides the ability to generate themes for such queries, and also integrates the results of such queries with the TRIMS photo log. Themes are generated using specific event attributes.



(Source: GISbiz, Inc.)

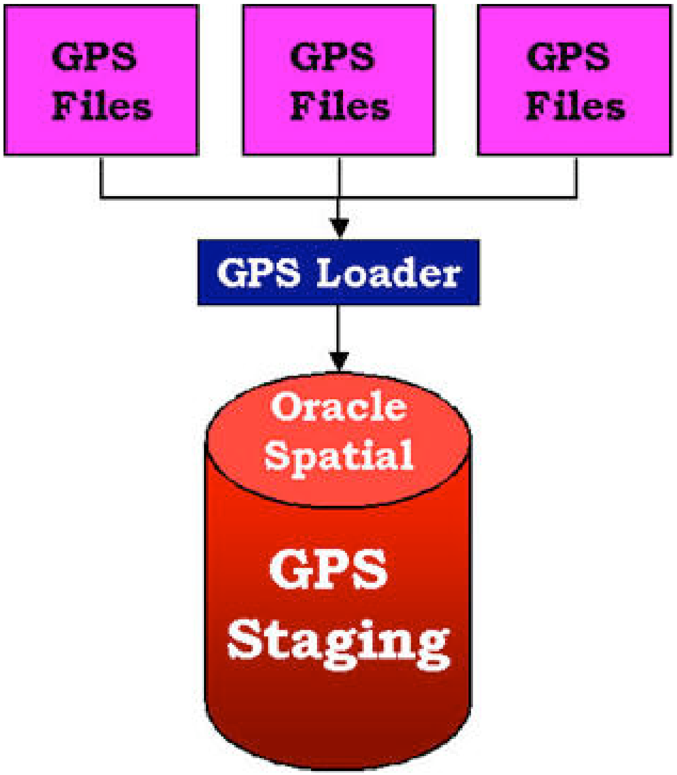
The TRIMS photo log capability provides the ability to capture and display photographs of the linear network.Linear network photographs are captured in stereo (bi-directional) and are cataloged based on the linear network identifier and begin/end log mile values. The TRIMS photo log viewer can therefore display photographs of any section of the linear network and it also includes the capability to play these photographs sequentially from the starting point to the end of the linear network.

TRIMSWeb

The core requirement for TRIMSWeb was two-fold.The primary requirement was the integration of the TRIMS attribute and spatial events data into a single repository. The second requirement for TRIMSWeb was to provide the core functionality of TRIMS in a Web-based environment.

Data Loading

The primary requirement for TRIMSWeb was divided into two sections. The first task was to integrate the attribute and spatial data for the base linear network in order to create the base LRS. The second task was to integrate the attribute and spatial data for all the TDOT transportation business events using the base LRS.Each of these will be described further below.

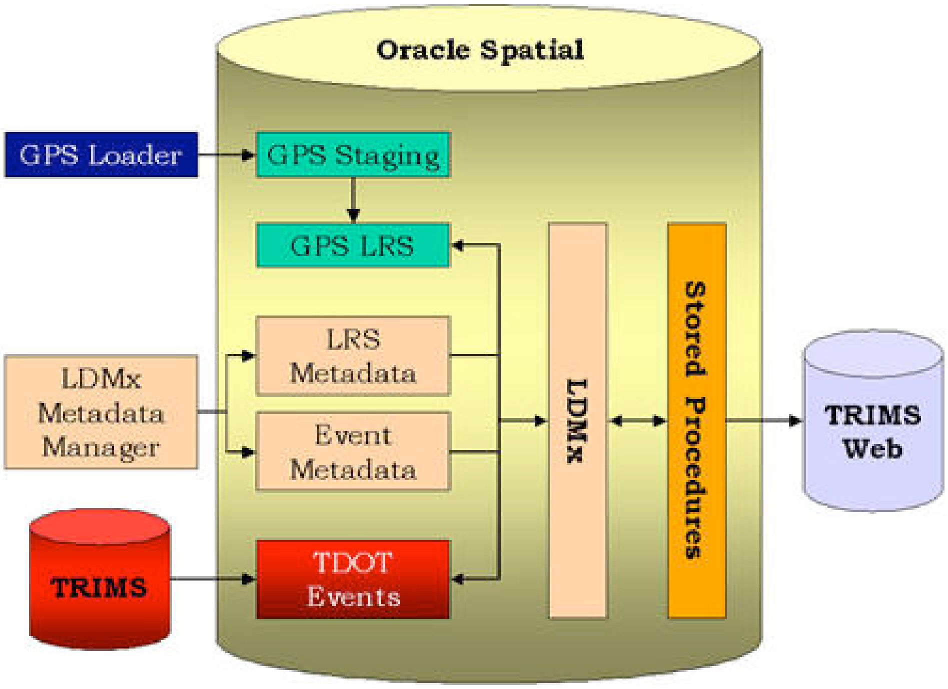


(Source: GISbiz, Inc.)

Base Linear Network

Oracle Spatial repository was chosen to satisfy the primary requirement for TRIMSWeb.To satisfy the first task of the primary requirement, all attribute and spatial data gathered in the field for the base linear network was loaded directly into the Oracle Spatial database after post processing was completed. The time tested and proven methods of the TRIMS data gathering and post processing techniques were retained to populate the Oracle Spatial base linear network (GPSLRS). A custom data loader provided by Bentley Systems was used to load the post-processed field data and populate GPSLRS.This custom data loader created and loaded GPSLRS, which is then maintained on an incremental basis to reflect any changes in the base network.

Business Event Segmentation



(Source: GISbiz, Inc.)

The second task of the primary requirement was the integration of attribute and spatial data for each of the TDOT business events. Bentley's LDMx tool was chosen to perform this integration.Dynamic geometry was generated for each event using LDMx, and stored along with its attributes in an Oracle Spatial column related to the event. This was accomplished by providing LDMx the core business event data, which originated directly from the TRIMS event tables, and the spatial base linear network that was created using the Bentley custom data loader.

Using these inputs, LDMx dynamically segmented event geometries from GPSLRS and stored them into the event tables. LDMx offered excellent capabilities in creating the business event geometries, as it is completely embedded inside the Oracle database and runs as stored procedures. It was possible to automate the process of incremental updates to the event geometries when changes were made to the key event attributes, such as begin/end log mile values, using simple Oracle database triggers. The TRIMS event database is located remotely, and any updates made are reflected in the new TRIMSWeb database by using automated Oracle replication methods.

Web Application

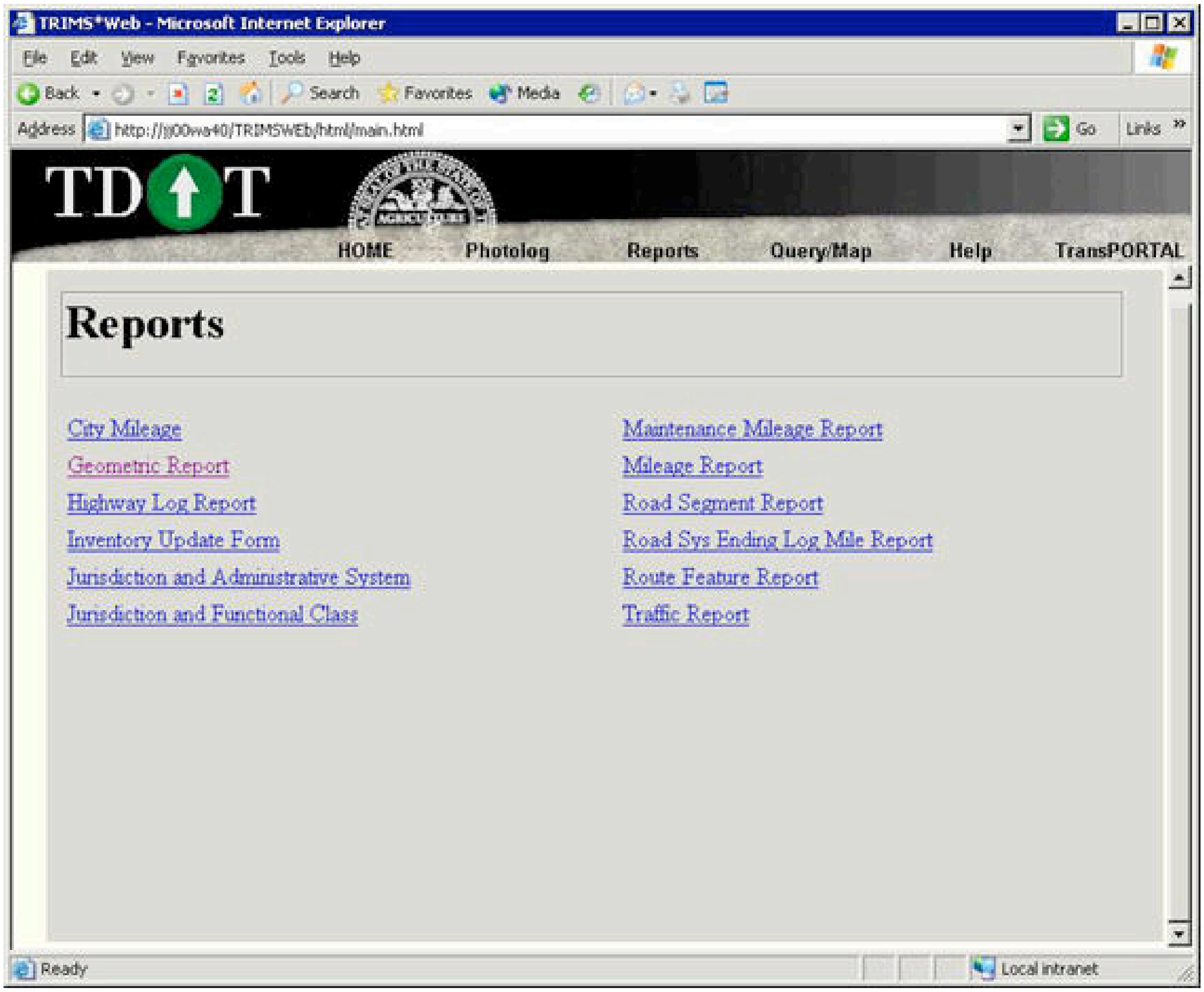
The second requirement of TRIMSWeb was the creation of a Web-based application that provided the core functionality of the TRIMS client-server application. This task was further divided into three major sub-systems:

1. Reporting (Reports)
2. Photo log viewing (Photolog)
3. Dynamic attribute and map queries against event tables (Mapping Component).

All sub-systems were designed and developed in parallel as described below.

Reports

Reports for TDOT transportation events are generated using the live LRS events relational database stored in Oracle. The reporting subsystem was collaboratively designed and developed by TDOT personnel and Intergraph.Several reports used on a day-to-day basis by TDOT personnel were identified for this task.



Reports

City Mileage
Geometric Report
Highway Log Report
Inventory Update Form
Jurisdiction and Administrative System
Jurisdiction and Functional Class

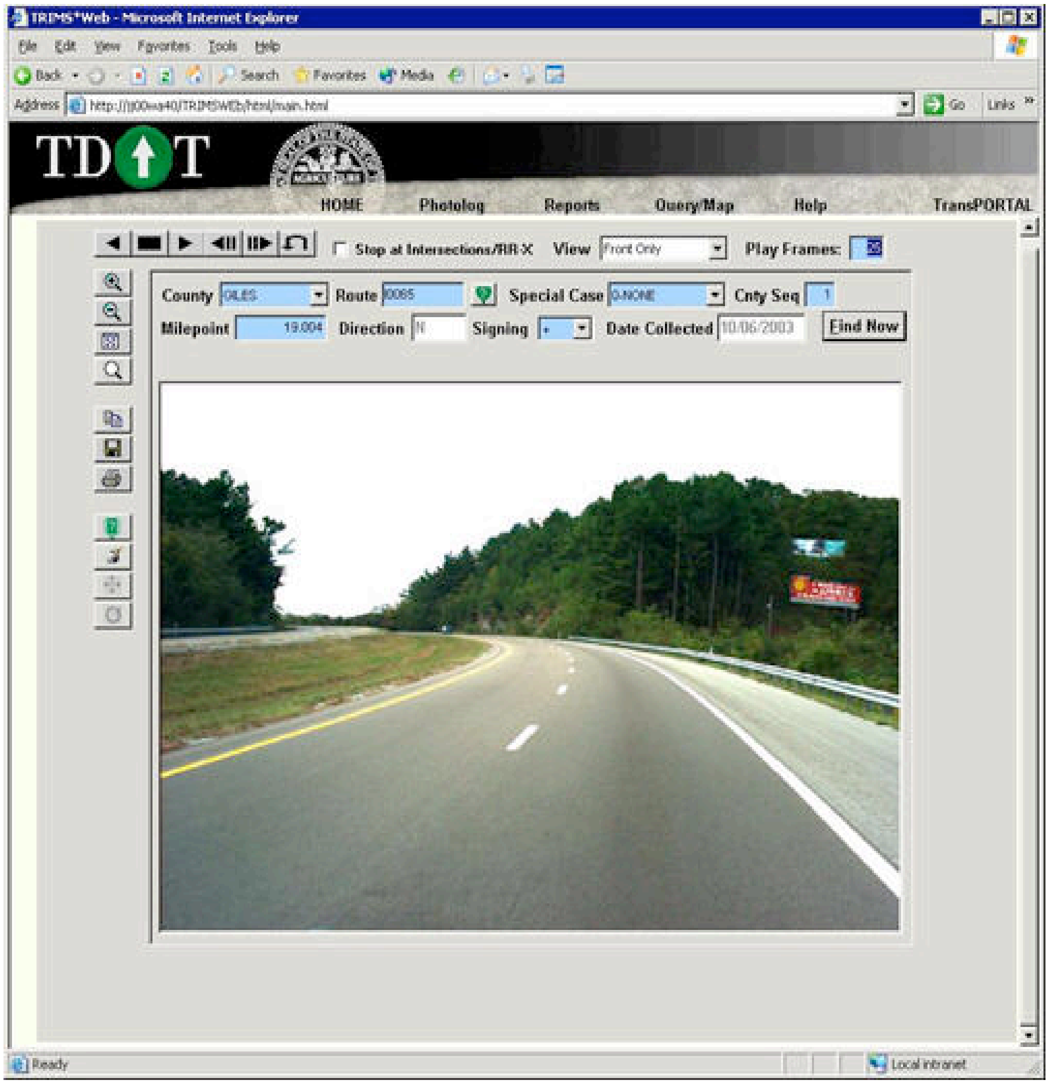
Maintenance Mileage Report
Mileage Report
Road Segment Report
Road Sys Ending Log Mile Report
Route Feature Report
Traffic Report

(Source: GISbiz, Inc.)

Reports are generated by querying the database using user specified criteria. This involves a standard relational query with embedded LRS key field criteria. The results of a query are transformed using custom designed Crystal Reports templates on the server using Crystal Reports Server Edition. The resultant report is exported into PDF format using Crystal Reports Server. As such, all reports are dynamic and provide accurate and up-to-the-minute data. The system is designed such that new reports may be added to the application as required.

Photolog

The photo log sub-system, designed and developed by Intergraph, provides accurate and detailed photographs of the TDOT roadway network. The network is photographed and cataloged using linear network identifiers and mile markers. All images are captured in high resolution, in bi-directional mode and stored by region and county. As such, an image of any mile marker of any linear network, in either direction, is easily accessible.



TRIMS*Web - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://j00wa40/TRIMSWEb/html/main.html
Go Links
TDOT
HOME Photolog Reports Query/Map Help TransPORTAL
Find Now
Ready
Local intranet

☐ Stop at Intersections/RR-X   View Front Only   Play Frames: 30

County GILES   Route 0065   Special Case 0-NONE   Cnty Seq 1

Milepoint 19.004   Direction N   Signing +   Date Collected 10/06/2003

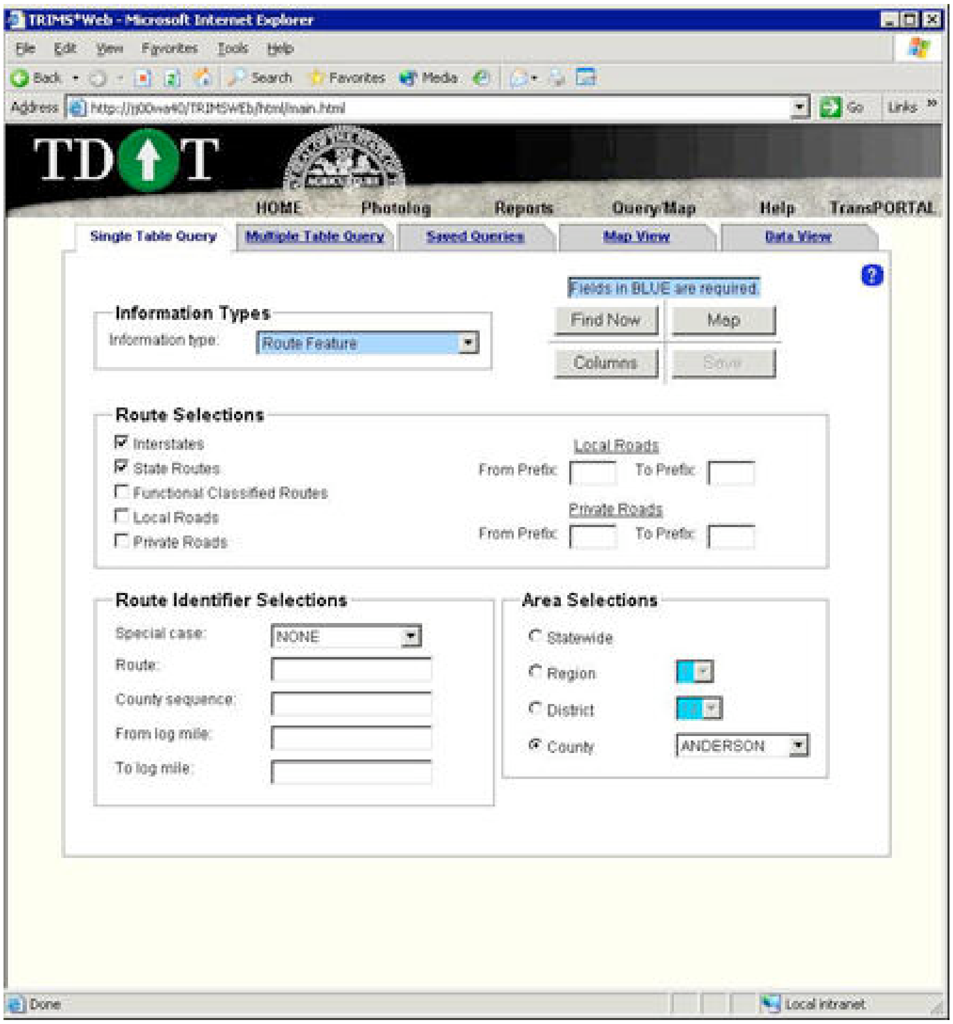
(Source: GISbiz, Inc.)

Images are rendered on the client using an image viewer, which comes with a robust set of query and display capabilities. The user may query for a particular mile marker of any network on the system using simple and intuitive selection capabilities. Once the image is located, it is downloaded and rendered in the client viewer. The viewer also provides the capability to walk all the images of the selected linear network from the mile point that was queried.Sequential images are dynamically downloaded and displayed on the client as the linear network is traversed. The user is provided several options at any time during image playback. These options include pause and resume, reverse direction playback, incrementing and decrementing playback speed, and aborting playback to query for images of an entirely different network segment.

All the functionality mentioned above is rendered within the browser environment. This is made possible by embedding the image viewer into the browser as an object. The viewer object is dynamically downloaded and installed the first time the server is accessed.Subsequent hits to the server do not require downloads of the viewer object, unless the code base of the image viewer on the server is updated. A new version of the image viewer object is automatically downloaded and updated on the client, should the code base of the viewer object be updated on the server.

Mapping Component

The mapping component of TRIMSWeb is the third and final sub-system of the application. This sub-system provides a robust set of querying capabilities against the TDOT LRS business events data stored on the server. This sub-system was co-developed by TDOT GIS Office personnel and members of GISbiz, Inc., operating as a sub-contractor to TDOT.



TRIMS*Web - Microsoft Internet Explorer

Address: http://jj00wa40/TRIMSWEb/html/main.html

Fields in BLUE are required.

Information Types

Information type: Route Feature

Route Selections

☑ Interstates
☑ State Routes
□ Functional Classified Routes
□ Local Roads
□ Private Roads

Local Roads: From Prefix: To Prefix:

Private Roads: From Prefix: To Prefix:

Route Identifier Selections

Special case: NONE
Route:
County sequence:
From log mile:
To log mile:

Area Selections

○ Statewide
○ Region
○ District
● County: ANDERSON

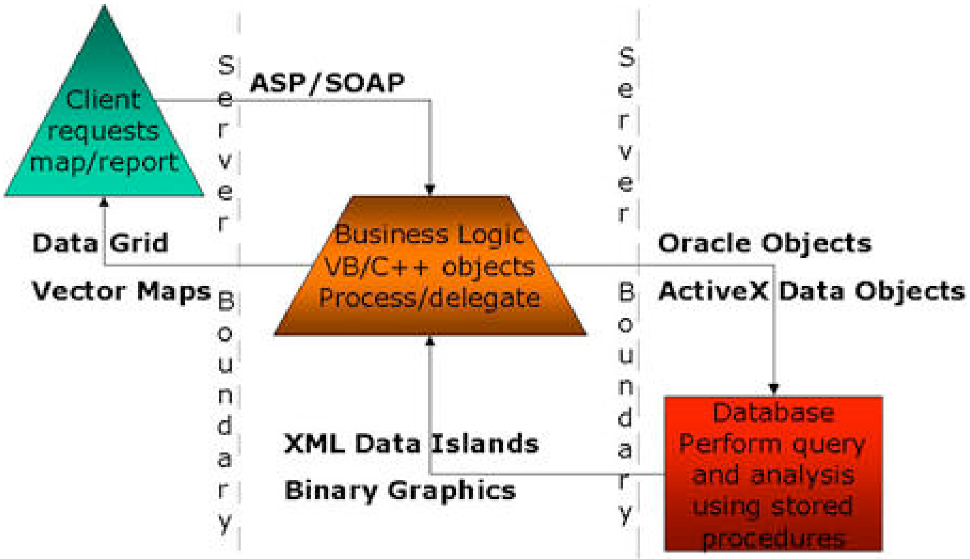
(Source: GISbiz, Inc.)

This sub-system is charted to provide the ability to perform both simple and complex queries against the TDOT LRS business events data for the purposes of displaying data and maps. This sub-system also had to provide the ability to save query definitions for any queries performed such that they may be recalled for future use. Therefore, tasks performed by this sub-system were further divided into five major categories as follows:

1. Single table queries;
2. Multi table queries;
3. Saved queries;
4. Data display; and

5. Map display.

This sub-system integrates all five sub-tasks into a single seamless browser environment using multiple tabbed interfaces.

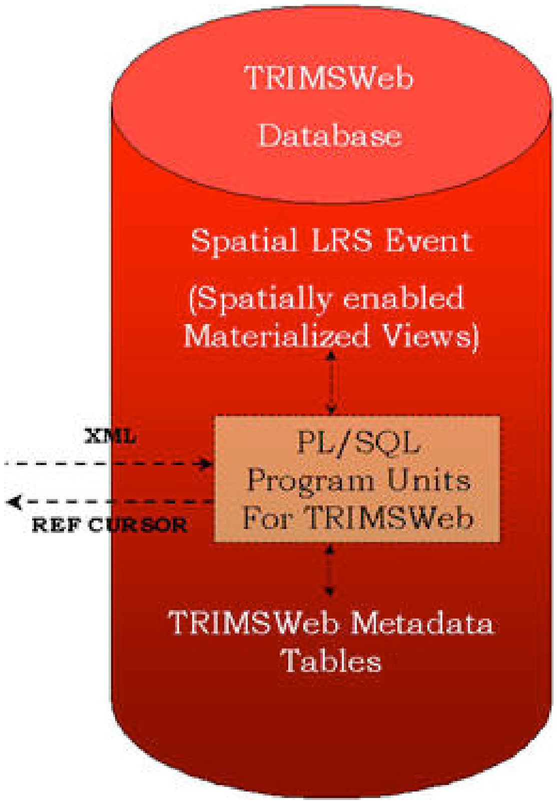

Client requests map/report
Server Boundary
ASP/SOAP
Server Boundary
Business Logic VB/C++ objects Process/delegate
Data Grid
Vector Maps
Oracle Objects
ActiveX Data Objects
Database Perform query and analysis using stored procedures
XML Data Islands
Binary Graphics

(Source: GISbiz, Inc.)

This sub-system employs the industry standard three-tier architecture involving the server, business and client tiers. The server provides TDOT linear network data, the business tier controls all business logic and the client tier provides all user interaction capabilities as expected.Unlike conventional GIS applications however, where the server typically only serves data and all GIS processing is handled at the business tier, this sub-system involves an alternate architecture for GIS data processing as described below.

Database Tier

During analysis and design of this sub-system, it was determined that the TDOT spatial linear data was highly granular. Simply put, the spatial data density was very high, or the number of points per linear segment was very high. This imposed a problem, not unlike in other GIS applications, of performance when large quantities of spatial data were being analyzed to extract the desired results. Moreover, and because of the nature of the queries, dynamic segmentation had to be performed to generate the results of the queries. The results of such segmentation were unique to each query and could not be performed before the exact query criterion was obtained from the client.


TRIMSWeb
Database
Spatial LRS Event
(Spatially enabled Materialized Views)
XML
PL/SQL Program Units For TRIMSWeb
REF CURSOR
TRIMSWeb Metadata Tables

(Source: GISbiz, Inc.)

The standard approach of extracting spatial data from the server and performing the required analysis and segmentation using business logic was discussed. While this approach was feasible, it posed a performance bottleneck when a large number of queries issued by various clients was submitted to the business tier. This approach required large quantities of spatial data to traverse the network, from the database server to the business tier, where the ultimate processing would take place. In order to avoid such data traversal, all spatial data processing had to take place within the database server, and simply output the results of such processing to the business tier.

Pushing all spatial data processing into the database server was easy because all spatial data was stored within Oracle Spatial, which had all required LRS and spatial processing capabilities. As such, all client requests were redirected to the Oracle database from the business tier, where Oracle Stored Procedures performed data retrieval, spatial analysis and dynamic segmentation as needed. The results were then sent back to the business tier to create data grids for attribute queries and maps for mapping queries.

Two key areas had to be addressed in order to embed all query processing capabilities into the database server. As you would expect, the first challenge was informing the database server of the details of the query, and the second was returning the results to the business tier for final delivery to the client. Each of these tasks was investigated separately, which yielded the following unique solutions.

The first challenge was simple and clear.In simplest terms, the server required an SQL statement to perform the attribute only or, attribute and spatial queries involving dynamic segmentation. The client or the business tier typically generates such SQL statements. This option was considered and eliminated for the following reasons. First, the client or business tier logic would become server specific, and second, additional logic was required to save the query definition into the database for future recalls. With respect to the latter, merely storing the SQL statement of the current query was insufficient information to subsequently recall the query and update the client interface.

Conventionally, the state of the client form can be stored and recalled to generate the SQL statement during subsequent use. While this option is sound, this approach was rejected for the following reason. This approach did not provide the means of recalling the saved queries without the use of the TRIMSWeb application and the use of the specific form within the application that generated the query. In other words, you could not recall and perform a query saved from this sub-system in an automated fashion such that the results could be consumed by other applications.

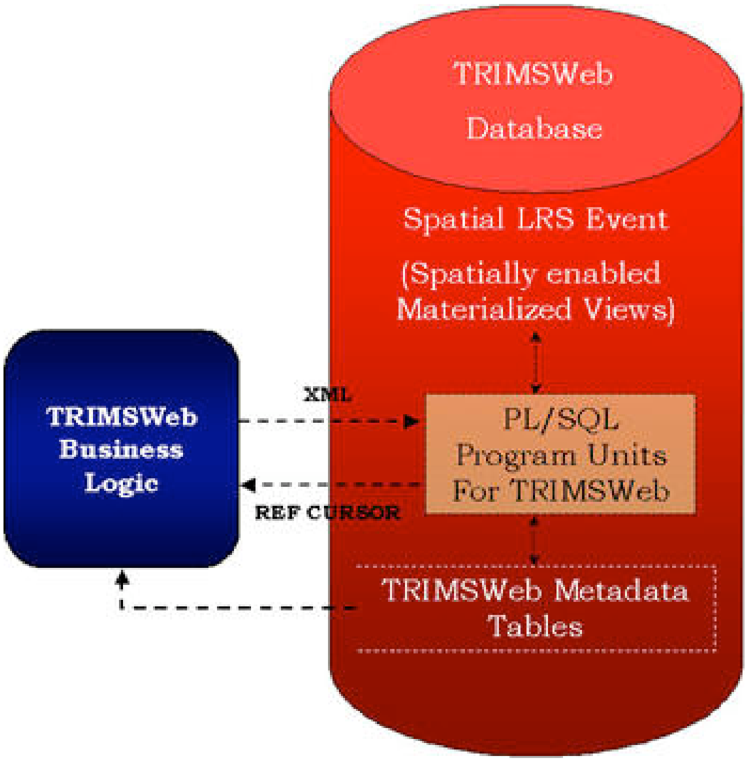
To eliminate creating server specific business logic and to store query definitions in a manner that they may be subsequently recalled without the use of this sub-system, a unique scheme was designed and developed using industry standard XML technology. Simply put, an XML representation of the query definition was generated at the business tier. This query XML was submitted to the database server where the required SQL statement was generated on the fly.

The benefit of this approach was significant. It became possible to submit complex query definitions to the database server, even if such definitions involved the use of multiple SQL statements. In conventional approaches, multiple SQL statements would have to be issued to the server from the client tier or the business tier, involving multiple round trips to the database server. This approach eliminated unnecessary round trips to the database server, which within its stored procedures was capable of generating the required SQL statements based on the query XML. Additionally, the query XML could be saved into the database for future recalls, and transformed using XSL templates to generate the client forms while maintaining the state of the client form.

The second challenge was returning the results of the query from the database server to the business tier, which could then be appropriately processed for delivery to the client. As stored procedures are employed to perform the query, it was decided to return the results of such processing as Oracle REF CURSORS. Using REF CURSORS proved to be the best approach as it eliminated the need to create temporary tables to hold the results of the query. Since REF CURSORS are pointers to memory locations where the query results are stored, the results are entirely maintained in the server cache, and the cache is cleared as soon as the requester releases the cursor.

Business Tier

The business logic maintained in this tier of the mapping component is crucial for the entire application.It creates and renders all client interfaces dynamically as well as processes all subsequent client requests. All client interfaces in the mapping component are database driven and are described in further detail in the next section. All business logic is contained in business objects written using Microsoft Visual Basic and is accessed as in-process servers during runtime.



(Source: GISbiz, Inc.)

One of the major tasks of the business logic is client request pre-processing. All client requests arrive at the business server via HTTP/SOAP.During pre-processing, the business logic gathers client requests and creates equivalent query XML strings. For modularity purposes, all query operations, regardless of the type of query - single table, multiple table or saved queries - are handled by the same logic. Each type of query however, involves different tables and columns. The challenge was creating logic that was not hard-coded for particular table and column names, as they are subject to change at anytime.

Hence, a metadata management scheme was designed and developed at the database server to facilitate this process. The mapping component metadata scheme is complex and discussing it is not within the scope of this article.It can be summarized as a catalog of all client forms, database tables and their columns, and TDOT specific relationships between such tables and columns. Using such metadata, the business logic dynamically creates an XML representation of the query. Once generated, the XML is self sufficient with all query requirements, and the business tier submits a request to the database server to perform the query.

Another important task of the business logic is database response post-processing. Query responses from the database arrive at the business tier as Oracle REF CURSORS. Upon arrival, the business logic determines the appropriate action to take based on the nature of the query. A data grid is created when the client requests only attribute data related to the query criteria. A map is generated when the client requests a visual representation of the query.

The determination of the type of request, attribute only or map, is made ahead of issuing the query to the database server. The type of request determines the manner in which the request is submitted to the database server. The business logic utilizes Oracle Objects for OLE (OO4O) to submit attribute only requests to the database server. When map requests are involved, the business logic utilizes a custom VC++ based GeoMedia data server to submit the requests to the database server. In either case, the resultant cursor, once it is received from the database server, is converted into a record set for further processing.

The choice of using two disparate vehicles to submit requests to the database server was made for performance reasons. OO4O was employed for attribute requests as it provided excellent capabilities for paging and converting the results into XML format. Once the results are converted into XML format, they are applied against XSL templates to create and render the data grid.

Generating maps using the resultant cursor posed its own challenges. The first step was determining which GIS tool, ESRI's ArcIMS or Intergraph's GeoMedia Web Map (GWM), to use to generate the map. After analyzing the capabilities of both tools, it was determined that GWM was equipped to handle data originating as Oracle REF CURSORS. Using GWM, it was possible to convert the resultant REF CURSOR into a GeoMedia record set that could be consumed by the GWM mapping engine to generate the map. Alternatively stated, the resultant Oracle spatial geometry had to be converted into GeoMedia spatial geometry before a map could be generated. This conversion was handled in the custom VC++ based GeoMedia data server.

The custom GeoMedia data server approach was employed to help improve conversion performance. The query is submitted to the database server stored procedure using the custom data server. The resultant Oracle cursor is then converted into a custom record set within the custom data server. During this conversion, all required attribute and Oracle spatial geometry is retrieved from the Oracle cursor. All Oracle spatial geometry is converted into GeoMedia spatial geometry within the custom data server. Finally, the custom record set is converted into a GeoMedia record set to be handed over to the GeoMedia mapping engine.

Several significant requirements were considered during map generation. These included, and are not limited to, query layer symbology, base layers and corresponding symbology, display scales, informational tool tips and hotspot actions. After careful analysis, we decided not to embed such information into the business logic as they were deemed as flexible elements that are subject to change at anytime. Therefore, such information was pushed into the application's metadata scheme and stored in the metadata tables. Such information is dynamically determined by the business logic using relevant metadata tables before map generation.

Client Tier

This is the final tier of the mapping component sub-system, and provides all user interaction capability for this sub-system. Within this tier, the user interface is divided into five major categories; single table query, multiple tables query, saved query, map view and data view as mentioned earlier.

All user interfaces in this tier are controlled by the database using the application's metadata scheme. Such information includes, and is not limited to, the number of client forms to be generated, the number of elements to be placed in each form and the relationship of the form elements to the database tables. It is therefore possible to alter the contents of the user interface by changing entries in the application's metadata tables.

Using such information, the business logic dynamically creates the application interface during application startup. First, user interface contents are gathered from the metadata schema. Next, an XML data island representation of the user interface is created. Finally, XSL templates are applied to the XML to create the desired HTML user interface. It became easier to generate multiple disparate user interfaces by placing the look and feel of the user interface into XSL templates.

Summary

The first version of TRIMSWeb provides the basic querying and viewing functionality of the TRIMS desktop application in the web environment. It brings together several technologies, which includes and is not limited to, Oracle Spatial Cartridge, Bentley System's LDMx, Intergraph's GeoMedia Web Map and industry standard XML, XSL and SOAP.

All application related data, attribute and spatial, are stored inside the enterprise Oracle database.Spatial data is stored in Oracle's SDO_GEOMETRY format. Query processing in TRIMSWeb is entirely performed inside the database within PL/SQL program units. These program units perform all attribute and spatial queries, including dynamic segmentation of business events using Oracle's native LRS capabilities.

The application's business logic is the bridge between the end-user and the database server. It pre-processes all end-user requests before submitting them to the database server, and post-processes all responses received from the database server to generate data grids and/or maps. All data grids are generated from XML data islands using XSL style sheets, and all maps are generated using GeoMedia Web Map.

The application provides the ability to view and play photographs of the TDOT roadway network using Intergraph's Photo Log Viewer. PDF reports of the queries are also generated using Crystal Reports Server.