Chapter 1

Introduction

1.1 Problem Statement:

To optimize the traffic and signal timings at the major intersections of a state highway. New Jersey Department of Transportation (NJDOT) personnel uses Synchro [4]®[1], traffic software for the optimization. The goal of optimization is so inefficient and error prone if the location of the intersection and the routes are not physically shown. In Synchro [4], we cannot locate the routes and intersections, actually on the map. The idea is so rigorous on client's sight of view. The challenge is to develop a map, show intersections, show traffic data on GIS map and data handling.

1.2 Objective:

The objective is to establish an interface between Synchro [4] and Geographic Information System such that, the traffic data from Synchro [4] for the intersection can be displayed in GIS map. The data can be updated; updated data can be exported back to Synchro [4], does traffic optimisation and again displayed in GIS map.

1.3 Motivation:

The traffic consultants and the New Jersey Department of Transportation (NJDOT) personnel find the difficulties in optimizing traffic at the major intersections of state highways. Synchro [4] is used to optimize traffic and signal timing at the intersections. The motivation of doing this is the drawbacks of the traffic software, Synchro [4].

In Synchro [4], we cannot locate the route actually on the map. The idea is rigorous on

client's sight of view, in spite of the efficient functionality of the traffic data. But, for the clients, it is always be more understandable, if the route is shown with respect to the actual map. That affects into accurate results of the traffic network problems and efficient solutions.

The future proposed work of the project is to implement the application on the web. So, the client does not required to have Synchro [4] on his computer.

- **Related Work:**

This is a research project of New Jersey Department of Transportation. Not anyone has developed an interface between GIS and traffic software. The key advantages are to use the positive features of both GIS and traffic software and make it functional software, which overcomes the disadvantages of the traffic software, if used standalone otherwise. For signal timing optimization consultants use TRANSYT. But TRANSYT is not as user friendly as Synchro [4] is. Synchro [4] has easy input window, which allows user to work easily. The comparison of TRANSYT and Synchro [4] [14] is as follows.

- The delays shown in TRANSYT are roughly equivalent to Synchro [4]'s signal delays. The signal delay shown in Synchro [4]'s MOE reports will equal the delay per vehicle in TRANSYT when the following is true. Pretimed signals, no permitted left turns, TRANSYT volumes are adjusted for PHF and lane utilization, and volume less than capacity.
- TRANSYT uses different assumptions for actuated signal operation, permitted left turn operation, and congested situations. For these applications, TRANSYT's delay

[1] Synchro ® is a traffic optimizing software developed by Trafficware Corporation, "www.trafficware.com"

will vary from Synchro [4]. Synchro [4] adjusts volumes for lane utilization and PHF while

- TRANSYT doesn't. It may be possible to get the delay per vehicle to match but not the total delay because
- TRANSYT is not aware of these volume adjustments.

The comparison of Synchro [4] vs. CORSIM [13] or SimTraffic [13][14] is as follows. Microscopic models such as SimTraffic and CORSIM [13] actually measure the delay to individual vehicles. The methods are different and in many cases the delay per vehicle will be noticeably different. If delays vary, here are some things to look for:

1. One or more lane groups has v/c greater than 1 in Synchro [4].
2. Closely spaced intersections cause blocking in SimTraffic or CORSIM [13]
3. Closely spaced intersections cause uneven lane utilization in SimTraffic or CORSIM [13]
4. Gridlocked conditions cause abnormally high delays in SimTraffic or CORSIM [13].
5. Short storage space cause blocking in SimTraffic or CORSIM [13]
6. Forced lane changes cause blocking in SimTraffic or CORSIM [13]

1.4 Overview of the System:

The system works fine with NJ 18, NJ 10 and NJ 23. Eventually we can make a whole network of routes in the state of New Jersey. The user must have a Synchro [4] license to run Synchro [4]. The basic components which make this system useful are: Synchro [4], Geomedia Professional [3][12]®[2], Database Access, ODBC, Operating System, Customized Visual Basic code for importing and exporting data. In my

[2] Geomedia Professional ® is Geographic Information Systems environment developed by Intergraph Corporation, "www.intergraph.com"

application, the client's interface is a user interface called "nj.gws", which is called the geoworkspace in terms of Geographical Information Systems (GIS). Geomedia Professional [3][12] is a platform to develop GIS applications. To develop a customized geomedia application, I wrote a script in VB and generated a customized geomedia command.

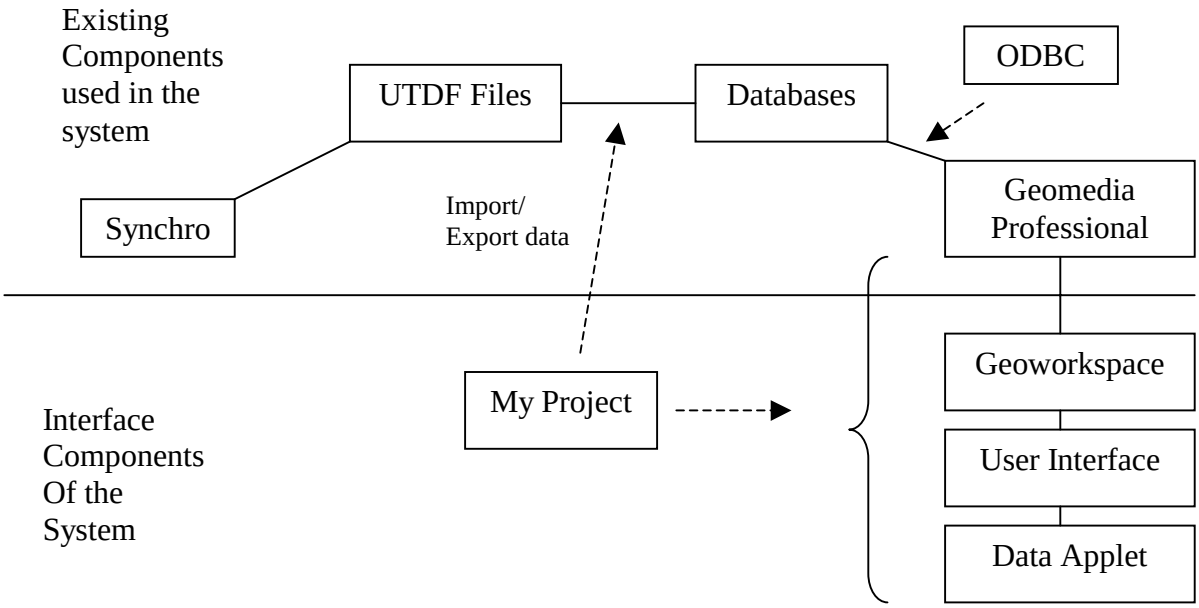


Figure 1: Global view of the system

Once data is being input in Synchro [4], we import that data in our map application. The map is developed such a way that a visual basic form is loaded when we click an intersection, showing the same details which are in the similar format as Synchro [4]. Therefore, it is not difficult to operate for the traffic consultants, who are used to work with Synchro [4]. It required accurate and demanding work. For importing data from Synchro [4] to Geomedia Professional [3][12], we use another Visual Basic program, which imports to Access database, and from that Geomedia Professional [3][12] is reading that data. The purpose of using Database is to maintain the records.

We may think that whether the application of the Synchro [4] in actual GIS environment, will be as user friendly as Synchro [4] itself. It is, because the input data we can update in the loaded Visual Basic form, and that the data is automatically updated in the database. So, before running the Synchro [4], the updated data must be read back by Synchro [4]. By UTDF, Synchro [4] can easily read the data. Later we run the program, and give the output back to the Geomedia Professional [3][12].

1.5 Technical approach:

I use Synchro [4] to carry out optimization of the signal timings and traffic at the intersections. At the same time, for the client's perspective, we show the Synchro [4] data on actual GIS (Geographic Information Systems) map. As a base GIS map display, we are using Geomedia Professional [3][12], GIS software. The real challenge is to develop a GIS map and interfacing it to the Synchro [4]. I develop a map in Geomedia Professional [3][12] using co-ordinates database and run geo spatial query to locate the intersections on the map.

In the coming chapters, I describe our system explaining the functionality of Synchro [4] and the systems architecture explaining systems requirements, sequence diagram, architecture description, sequence diagram and various use cases. How map objects in the user interface are developed using Modular GIS environment (MGE) and MGE segment manager, showing intersections are explained further. In this research application, it required detailed work to "flow" data from one piece of software to the other. Data management is possible with Universal Traffic Data Format. The detailed data flow diagram is shown in the "Data Flow" chapter. For the user, it is necessary that he

understand the concept of the user interface, which is described as a separate chapter. I also propose the web architecture for the system to make it universal.

- **Summary of Results:**

Interface between Geographical Information Systems and Synchro [4] is established and working as per NJDOT's requirements. For now, we added NJ 10, NJ 18 and NJ 23 route to the network. Likewise, we can add multiple busy routes in the state of New Jersey. I also propose an efficient way to make the interface run as a web based application in concluding chapter.

1.6 Contributions:

My contribution towards the project is the architecture design, software development and the documentation. I also wrote a brief manual for installing the software and a manual for troubleshooting tips; those are attached as appendices.

Chapter 2

Systems Architecture

2.1 Systems requirements:

- **Functional Requirements:**

The user can import the data from Synchro [4] to the system's databases.

The system should provide the up to date information about the selected intersection in the similar fashion to Synchro [4].

The authorized user can make the change in the database and can export the data back to Synchro [4].

Concurrent Data: The data in the database of the system should be concurrent. I.e. data in the database should be clean. For particular intersection, if the user has changed the data and we did not write it back, it may result into "dirty" data.

The users are should allow running query on the developed map to learn the relationships among the different attributes of the base mapping database.

Usability: This software is usable to all the employees in New Jersey Department of Transportation. This greatly helps in analyzing the traffic data of any route, with respect to the graphical representation.

2.2 Overview of Synchro:

Synchro [4] is a complete software package for modeling and optimizing traffic signal timings.

Capacity Analysis

Synchro [4] provides a Windows based, easy-to-use solution for single intersection capacity analysis and timing optimization.

In addition to calculating capacity, Synchro [4] can also optimize cycle lengths, splits, eliminating the need to try multiple timing plans in search of the optimum.

All values are entered in easy-to-use forms. Calculations and intermediate results are shown on the same forms.

If the intersection is coordinated, Synchro [4] explicitly calculates the progression factor. With the Highway Capacity Software (HCS), it is necessary to guess about the effects of coordination. Synchro [4] calculates the effects of coordination automatically and accurately.

Coordination

Synchro [4] allows you to quickly generate optimum timing plans. Synchro [4] optimizes the split, cycle length, and offsets.

Synchro [4] optimizes to reduce delays. This makes Synchro [4]'s timing plans similar to TRANSYT, which optimizes to reduce stops and delays.

Unlike other coordination software, Synchro [4] is fully interactive. When you change input values, the results are updated automatically. Timing plans are shown on easy to comprehend timing diagrams.

Actuated Signals

Synchro [4] is the only interactive software package to model actuated signals. Synchro [4] can model skipping and gapping behavior and apply this information to delay modeling.

Time-Space Diagram

Synchro [4] has colorful, informative Time-Space Diagrams. Splits and offsets can be changed directly on the diagram.

Synchro [4] features two styles of time-space diagrams. The bandwidth style shows how traffic might be able to travel down an entire arterial without stopping. The vehicle flow style shows individual vehicles that stop, queue up, and then go. The traffic flow style gives a much clearer picture of what the traffic flow actually looks like.

Time-space diagrams can be printed using any Windows compatible printer.

CORSIM, TRANSYT-7F, HCS

Synchro [4] features preprocessors to these three popular software analysis packages.

Enter data once with easy-to-use Synchro [4], and then perform analyses with the industry standard, legacy software.

Synchro [4] is available in two different levels with different features and price points.

Synchro [4] Light has a limit of 10 intersections per file. Optimizations and reports cannot be created for files with more than 10 intersections. Signalized and unsignalized intersections count towards this limit but bends and external nodes do not.

Synchro [4] has no limit of intersections. Synchro [4] has been tested with up to 300 intersections per file. Synchro [4] contains preprocessors for TRANSYT 7F and CORSIM [13].

All versions of Synchro [4] 5 contain preprocessors for HCS Signal, HCS Unsignal, and SimTraffic.

2.3 Synchro Data:

Volume Data:

The code generated for volume file, worked good for the volume files only.

As the code was coded keeping volume data only in mind, Volume data file has a nice format so; it's kind of easy to code it. The first attribute is the 'date', and I declared date as a text, rest all the attributes were the integers. For all the intersections the format is the same, so, idealizing such condition, code for reading a data from .DAT file and exporting that into Access database, was prepared.

Volume data includes, volume data for an intersection for North Bound Left (NBL), North bound through (NBT), North bound right (NBR), SBL, SBT, SBR, EBL, EBT, EBR, WBL, WBT, WBR. For one intersection, the # of records for volume data is one.

Phasing Data:

The code prepared for volume will not work for the phasing data exporting to Access database. As the # of attributes will vary, the attributes are varying, the number of records are varying from the Volume. As, I stated earlier that in Volume data, there is only one record per intersection. While in this case, there are several records like walk phase, minimum green time phase etc., unlike volume data. In this case, the record name attribute is a text, but the length of each record name is different, unlike date attribute in volume data.

Timing Data:

Timing Data contains signal timing, at the intersection. Timing data, like Volume data contains only one record per intersection.

Lane Data:

Lanes data contains information about, lanes, how many of them are shared, width of the lanes, grade, speed, first detect, last detect, ideal flow, lost time, saturated

flow, headway factor, volume, peds, bicycles, PHF, growth, Bus stop details, distance, travel time.

2.4 Modeling with Synchro [4]:

Step 1. Create the Network

To begin a new network, open Synchro [4] and select the command File -> New. From the MAP window (press ‘F2’ if not already in the MAP window), create the network shown below (see the topic on Mapping out Links and Intersections). The length of the links is 1000 feet.

To add a link to the map:

A. Select the Add Link button or hold down the ‘A’ key.

B. Position the mouse cursor on the map window where you want the link to start, and click the left mouse button. The status indicators, at the lower-right corner of the window, show the East and South coordinates in feet (meters). Note: To cancel adding a link, press ‘Esc’.

C. Release the mouse button and move the cursor to the position on the window where you want the link to end. Click the left mouse button again. Refer to the status bar at the bottom of the window to see the length and direction of the link

To create the intersection, simply create another link that crosses the link created in steps A through C above. The intersection will be placed where the links cross and will be shown as a circle.

Use the default Link Settings for all approaches.

Step 2. Enter Lane and Volume Data

From the MAP window, click on the intersection of Main Street & 1st Street and activate the LANE window by pressing the Lane Window button or the ‘F3’ key. Enter the lane data that is shown below:

	INTID	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
Lanes and Sharing	3	1	1	1	1	1	1	1	1	0	1	1	0
Shared Lanes *	3	0	0		0	0		0	2		0	2	
Lane Width (ft)	3	12	12	12	12	12	12	12	12	12	12	12	12
Storage Length (ft)	3	250		250	250		250	200			200		
Storage Lanes (#)	3	1		1	1		1	1			1		
Grade (%)	3		0			0			0			0	
Leading Detector (ft)	3	0	0	0	0	0	0	0	0		0	0	
Trailing Detector (ft)	3	0	0	0	0	0	0	0	0		0	0	

* For Shared Lanes, 0 = None, 1 = Left, 2 = Right, 3 = Both (Shared with Through)

To add lanes, enter the number of lanes for that lane group. For each lane group, enter the number of lanes as a value between 0 and 5, or select the lane configuration from the drop down list.

For the through lane group, specify whether it shares with left or right traffic by pressing ‘R’ or ‘L’ selecting the appropriate configuration from the list.

Switch to the volume window by pressing the Volume Window button or the ‘F4’ key. Enter the volume data shown below:

INTID	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
3	100	600	100	100	500	100	100	600	100	100	500	100

Use the defaults for all other values in the VOLUME window.

Step 3. Enter the Timing Data

For this example, the left and right turn phases are permitted. The controller type is pretimed. For further information, see the topic on Turn Type. Use the default values for the remaining fields shown on the timing window.

The basic information required to perform an analysis of a basic two-phase intersection is now entered. To set phase specific parameters, such as the minimum split, yellow and red times, and pedestrian interval settings, see the phasing window or this example, use the default values assigned in the phasing window.

Step 4. Optimize Intersection Cycle Length

Now that the basic data is entered, the next step is to find the best timing plan for this isolated pre-timed intersection. Use the Optimize Intersection-Cycle-Length command to set the intersection to the Natural Cycle Length. The Natural Cycle length is the lowest acceptable cycle length for an intersection operating independently. Synchro [4] will automatically optimize the intersection splits when you perform this step. The resulting cycle length is shown in the Current Cycle Length field show on the left of the timing window.

Step 5. Interpreting Results

The final step is to interpret the measures of effectiveness (MOE) for the example. The most common MOEs are shown in the timing window rows for each lane group.

The intersection wide MOE's for volume to capacity (v/c) ratio, delay and level of service (LOS) are shown in the left side of the timing window.

The most commonly reported MOEs are the v/c ratio, the delay and the level of service.

The v/c Ratio is the v/c ratio using actuated green times and cycle lengths. The v/c ratio indicates the amount of congestion for each lane group. Any v/c Ratio greater than or equal to 1 indicates that the approach is operating at above capacity.

The delay is a measure of the total control delay, in seconds per vehicle, experienced for the given lane group.

The LOS is a means of describing the operational efficiency of a given intersection based on the calculated delay. The range of service quality has been defined in terms of six LOS ranges (A to F). The range of LOS is shown in Table 6-1. LOS A represents free flowing conditions with insignificant delays. LOS F represents forced flows (jammed conditions) with excessive delays. Under LOS F, queues may block upstream intersections.

To quickly print these results, use the command File -> Print-Window. For other more detail reports, see the topic on Intersection Reports.

2.5 Use Cases [8][11]:

1. Opening the Geoworkspace map

User Action	**System Responsibilities**
Makes sure the system has Geomedia Professional version 4.0. If it does not have then installs it.	
The user runs setup program to install MGSM Project.	Setup program installs MGSM Project
Opens the nj18.gws file.	Geomedia Professional [3][12] loads the geoworkspace map.
Activates the Geoworkspace.	

Contracts [8]:

Responsibilities: Opens the geoworkspace to allow user to operate. Activates the geoworkspace.

Type: User

Exceptions: MGSM Project or Geomedia Professional [3][12] is not installed.

Problems in ODBC connectivity.

Dynamic link library for myproject is missing.

Pre-conditions: MGSM Project and Geomedia Professional [3][12] are installed on the system. ODBC connectivity is properly defined and all the files to run the program are located in their respective path.

Post-conditions: The user can click on the intersection to load the user interface.

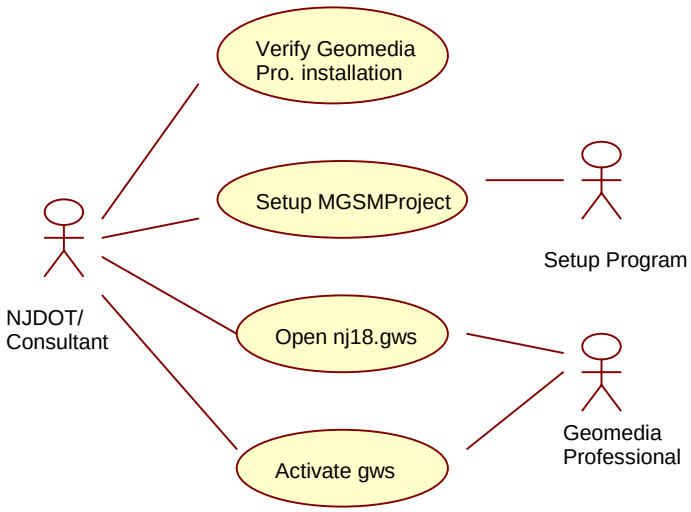


Figure 2: Use Case: Opening the Geoworkspace map

2. Opening the user window

User Action	**System Responsibilities**
Makes sure the geoworkspace map is activated.	
If not activated, then user activates the geoworkspace.	Geomedia Professional [3][12] activates the geoworkspace map.
Finds the intersections of route 18	
Clicks the particular intersection whose Data he wants to analyze.	Geomedia Professional [3][12] loads the user window.

Contracts:

Responsibilities: Loads the user window of the application, so the user can load the data he wants from Synchro [4].

Type: System

Exceptions: Dynamic Segmentation is not being performed for the intersections.

Dynamic link library is missing.

Pre-conditions: The geoworkspace is opened and is activated. Dynamic segmentation is being performed on the base map.

Post-conditions: The user can display the data of types Lane Data, Layout Data, Volume Data, Timing Data or Phasing Data as an individual applet.

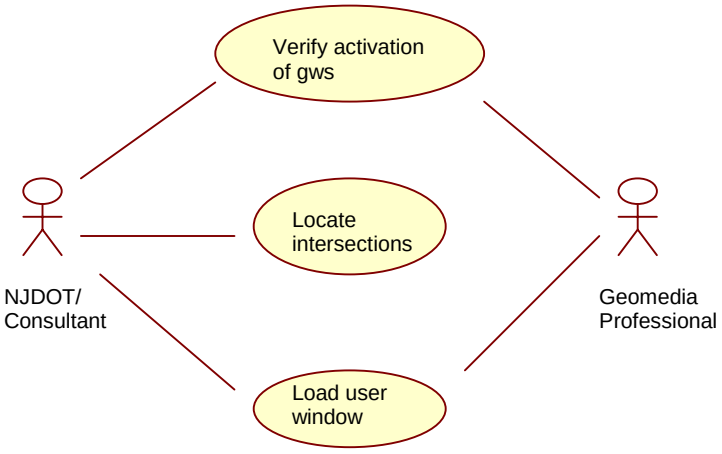


Figure 3: Use Case: Opening User Window

3. Importing data from Synchro [4]

User Action	System Responsibilities
Selects any intersection of route 18 map in Synchro.	
Exports data in Universal Traffic Data Format as a .csv file, and saves.	
User imports the data	VB myproject imports the data from UTDF files.

Contracts:

Responsibilities: User saves the Synchro [4] data into various files as Universal Traffic Data format. VB myproject loads the UTDF data into the database.

Type: System

Exceptions: UTDF Files are not stored in the predefined path. Customised program is erroneous.

Pre-conditions: UTDF files are stored in the predefined path.

Post-conditions: The user can see the Synchro [4] data into the VB applet.

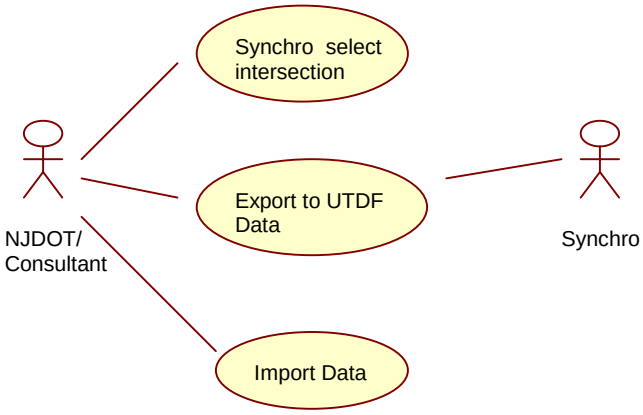


Figure 4: Use case: Importing Data from Synchro

4. Exporting data to Synchro [4]

User Action	**System Responsibilities**
Changes the data in the user window.	Respective traffic data database updates the data changed by the user
User exports the data	VB myproject exports the data from the traffic data database into UTDF.
In Synchro, user imports UTDF Data and saves it.	Synchro displays the updated data.

Contracts:

Responsibilities: Saves the changes made by the user into the traffic data database. VB myproject writes the updated data of the database into UTDF files. Synchro [4] reads the updated UTDF files.

Type: System

Exceptions: The changes made by the user are not saved to the database. Updated UTDF files are not stored in proper path.

Pre-conditions: Geoworkspace is activated. The applet is loaded properly.

Post-conditions: The user can see the updated data into Synchro [4] and utilizes those for the analysis.

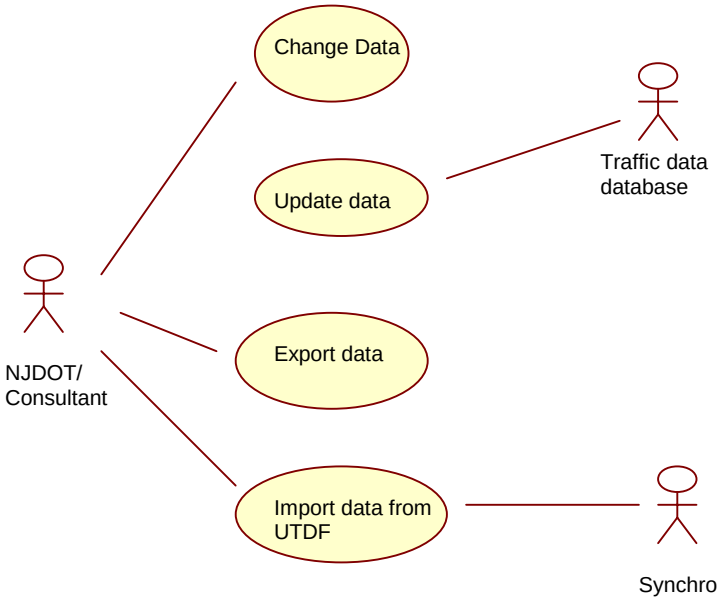


Figure 5: Use Case: Exporting Data to Synchro

2.6 **Collaboration Diagram [8]:**

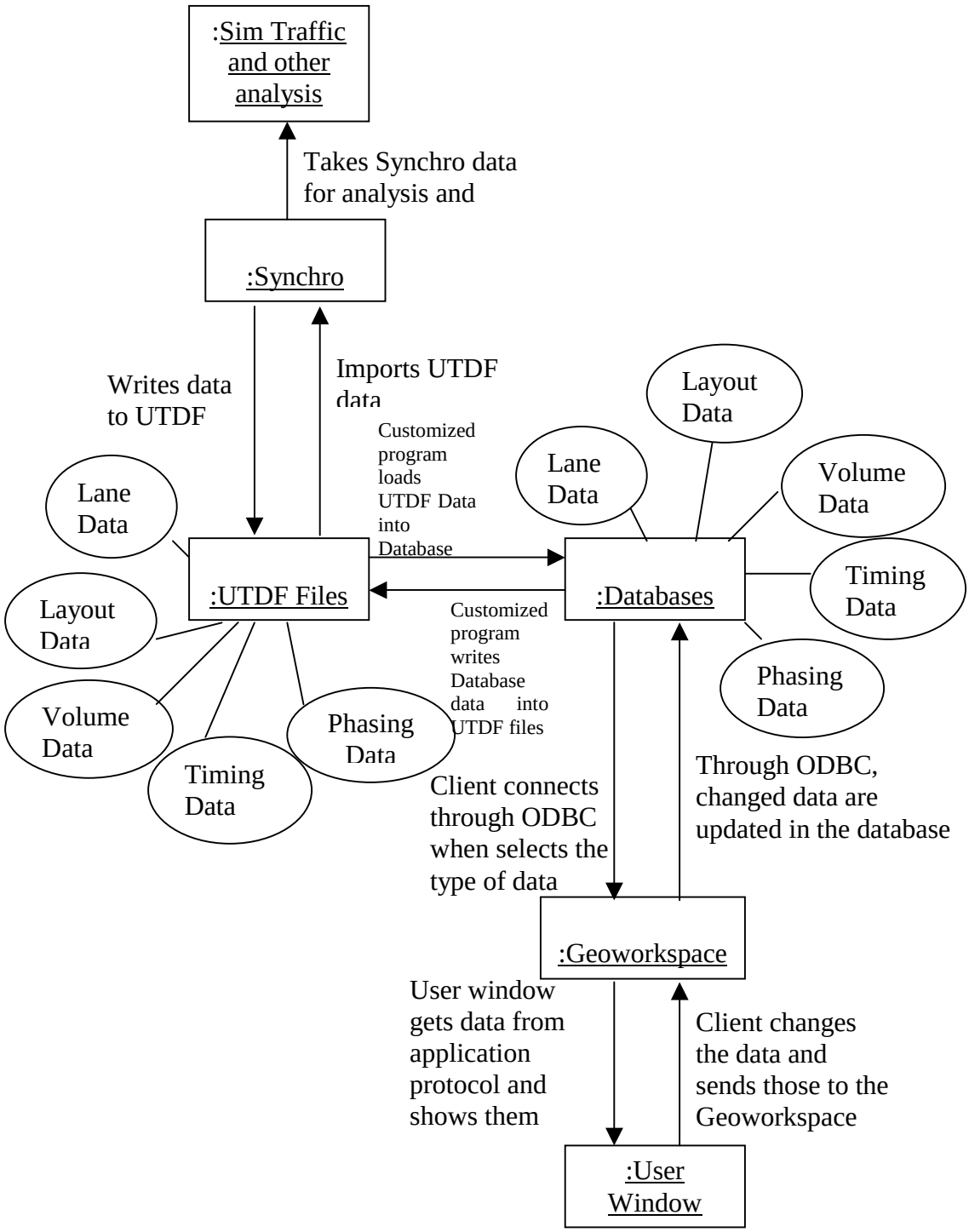


Figure 6: Collaboration diagram

2.7 Description of Architecture:

Synchro [4] is traffic analysis software. Synchro [4] supports Universal traffic data format (UTDF). The client writes Synchro [4] data to the UTDF files. Here we write all the data of same type write into one file. So, there will be five types of files: Lane Data, Layout Data, Volume Data, Timing Data and Phasing Data. On our application we have “Import Data” button. When the client clicks on that button, the customized VB program runs in the background. Basically, it opens the UTDF data file, reads record by record and inserts into the database. We have separate database for each type of data. I.e. we have five databases: Lane Data database, Layout Data database, Volume Data database, Timing Data database and Phasing Data database. The user selects the option button for the data, and clicks “Show Data” button on the user interface. As soon as user clicks that button, the application is connected to the respective type’s database through ODBC (Open Database Connectivity). At the same time, one graphical user interface applet appears, that shows the selected data exactly in the similar fashion as Synchro [4]. The applet gets the data from the application protocol, which is one layer below than applet in the protocol stack.

The client wants to change the data and export the updated data back to Synchro [4] to see the effect in the analysis. Our applet is very user friendly. The client can change the data in the applet and can save those changes in the database. The user clicks “save data” button and the updated data are being sent to the lower protocol, which is application layer. The application is connected to the database through ODBC. So, can easily update the database. On the user interface we have “Export Data” button. By clicking that, one customized program runs in the background, which loads updates into

those UTDF files. Synchro [4] not only writes data to UTDF files, but also can read the data too. We see the updates in the Synchro [4]. The client can run simulation in Simtraffic or any other software to see the effects of the updates.

The Abstract Machine Model:

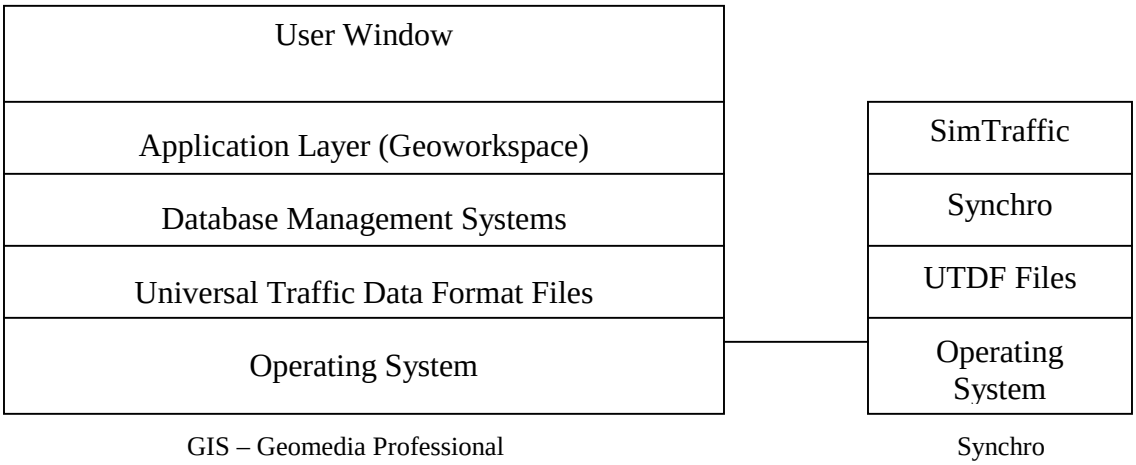


Figure 7: Abstract Machine model

The abstract machine model of architecture models the interfacing of sub systems. It organizes a system into a series of layers each of which provides a set of services and communicates with the immediate neighboring layers. Operating System and Universal Traffic Data Format files are the common protocols in these two protocol stacks. Database management systems take the data from UTDF layer and pass them to application layer. When user selects the type of data, graphical user interface layer gets the data from the application layer. On Synchro [4] side, Synchro [4] gets the data from UTDF files and passes them to Simtraffic or other analysis tools.

2.8 System Sequence Diagram [8]

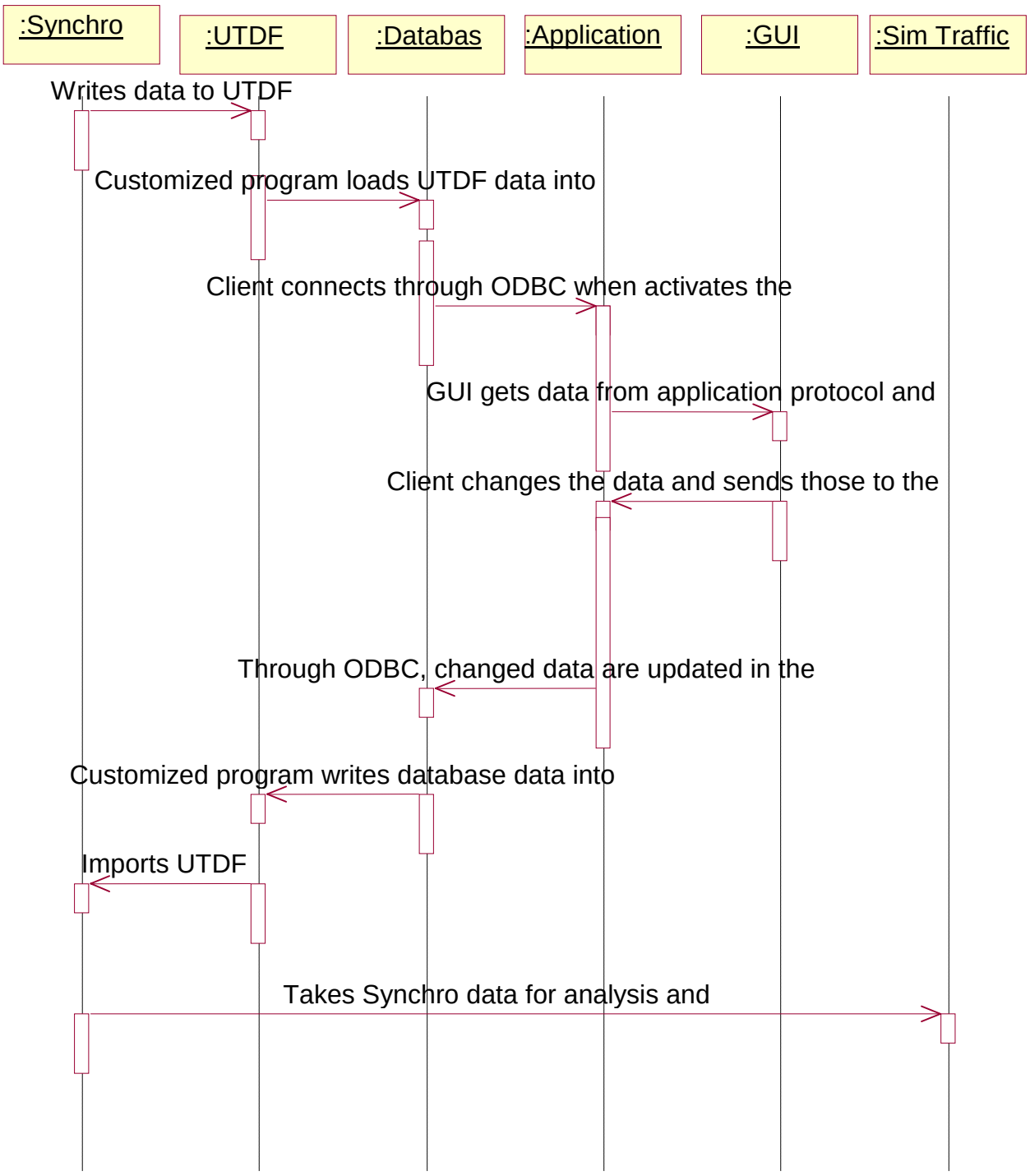


Figure 8: System Sequence diagram

2.9 Design Details

- **Run Time Model:**

Execution orderness: The system is an event driven system, where the execution depends on the user's actions. The user can generate the list of actions in a non-linear fashion, in any order.

Time Dependency: There are no time constraints on the system, since it depends on the user actions, and events. It is real time system since the current data in the database is given to the user whenever he clicks on the intersection. It is not as of this time periodic.

- **Concurrency:** At any time only one user can access the system to view and change the data.

- **Algorithms:**

Locking and Commit algorithm: The algorithm can be used when any re-order use-case is carried out in a network supported application. At any time, it allows only one user thread to access the database to place a re-order query for a particular product. This ensures that re-order takes place only once which keeps the consistency of data.

If any other user wants access the database, a queue mechanism is implemented to allow user access.

Chapter 3

Developing map and map objects

3.1 Developing a map

Here, the Geomedia-command VB program is developing the New Jersey map. The program creates the Geoworkspace (Geomedia Professional [3][12] terminology), defines the co-ordinate systems for the GeoWorkspace, creates warehouse connections, and displays data in the map window.

GeoWorkspace [3][12]:

A GeoWorkspace is the container for the work in Geomedia Professional [3][12]. Within its confines are the warehouse connections to the data, map and data windows, toolbars, coordinate system information and queries, which are built. When all the design files and the overlaying layers are showed on the map, the GeoWorkspace is saved and the environmental system variable is set and the warehouse connection is saved. So, next time, we do not have to develop all the design files and overlaying layers again on top of the map, and we can directly proceed to our application. By that means, it becomes user-friendlier.

Coordinate systems:

The co-ordinate system provides the mathematical basis for relating the features of the map to their real world positions. Because the shape of the earth's surface varies from one geographic area to another, the software interprets coordinates with reference to a network of geodetic control points. The default coordinate system in Geomedia Professional [3][12] contains the following settings.

- Base Storage type - Geographic

- Horizontal resolution -1 degree
- Projection algorithm -Cylindrical Equi-rectangular
- Projection parameters -Centered at the equator and the prime meridian
- Geodetic datum and ellipsoid -WGS84
- Paper Space - 1:50,000

One can change the coordinate system settings in the individual GeoWorkspace or in an empty read/write warehouse.

Warehouses [12]:

We display feature geometries and attribute data in a GeoWorkspace through connections to warehouses where the data are stored. Each warehouse connection uses a data server to convert the data into a format that Geomedia Professional [3][12] can display. If one wants only to display the data in Geomedia Professional [3][12] from one or more warehouses, one simply create one or more warehouse connections and then use map windows and data windows to display the data.

3.2 MGE (Modular GIS Environment)[1]:

MGE has a multipurpose structure that allows accessing common utilities, application software, graphic data and the databases. This system creates and maintains GIS (Geographic Information Systems) or mapping databases. MGE provides a flexible set of tools for production and planning environments. MGE combines graphics for mapping and relational databases for data, which offers a flexible working and development environment for end users of our system such as municipalities, system integrators, consultants and third party developers.

It is very easy to run GIS applications on top of MGE environment provided both runs on common platform. That allows performing topological queries and analysis, photogrammetry, surveying, digital terrain modeling, image processing, land information management and cartographic output tasks.

Here is MGE system protocol stack.

<table>
<tr><td>MGE Basic Administrator (MGAD)</td><td>MGE Base Mapper (MGMAP)</td><td>Other Intergraph/ MGE Compatible Applications</td><td>User designed Applications</td></tr>
<tr><td colspan="4">MGE Basic Nucleus (MGNUC)</td></tr>
<tr><td>RIS</td><td>ODBC</td><td colspan="2" rowspan="2">Microstation</td></tr>
<tr><td colspan="2">Relational Database Management System (RDBMS)</td></tr>
<tr><td colspan="4">Operating System (Windows 2000/NT/98/95)</td></tr>
</table>

Figure 9: MGE System protocol stack

MGNUC (MGE Basic Nucleus) [1] defines projects and performs query and reviews operations. MGNUC provides a map coordinate system through MGE Coordinate System Operations (MCSO), which is delivered with MGNUC. MGE Basic Administrator (MGAD) defines the structure of the database. MGE Base Mapper (MGMAP) captures, generates, cleans, validates and manipulates GIS data.

MGNUC and other MGE applications use microstation platform and either Relational Interface System (RIS) or Open Database Connectivity (ODBC) to interface to a relational database management system (RDBMS).

MGE Project [1]: MGE Project consists of the configuration files, command files, data files, design files, fence definition files, geographic index files, saved query files, report files, seed files, setup and configuration files and list files.

Project Data set: Before running a project, it is necessary to understand the different components of it. The project data set consists of all the data, system files and support files necessary for the project as described below.

Map Files: A map file is a design file that contains digitized map features. Each feature is an element with at least one linkage (feature linkage) to the features table and optionally linkages to user-defined attribute data tables. It contains an MGE coordinate system element.

Geographic Index Files [1]: A geographic index file is a design file that shows the spatial relationships between the maps in a project. Each level in the index file contains shapes that represent the spatial bounds of individual map design files for the category.

Database: The database is a relational database, which contains feature, category, maps and user defined attribute tables that contain information for the project. The following are the different tables required for the MGE project to generate the map from the database.

Attribute_catalog	label	user_defined tables
Category	list_domain	view_catalog
Domain_catalog	maps	view_content
Feature	mscatalog	view_join
Join_catalog	range_domain	

Creating the project schema:

MGE interacts with one of several relational database systems through the RIS (Relational Interface System) or an ODBC (Open Database Connectivity) driver. We use ODBC driver to connect MGE project with our database. We associate our database MGE_Gis_Export.mdb with the ODBC data source, rutgers.

Support Files: List files, seed design files and other files are required for the project. MGE uses seed files to ensure that all maps created in a project are cartographically compatible. A seed file, which serves as a template for the creation of new design files, is a blank design file containing mapping parameters. Seed file defines the primary coordinate system and sets up the working units.

Actual digitizing the map

After creating a project, seed file and database tables, we are set to populate the project by adding graphic data and populating the database. To add the graphic data to the project we must create a design file.

We can convert existing or entering new graphic data by digitizing or tracing an existing hard copy map. Before digitizing map into a graphic file, make an association between the digitizing surface and the design file with digitizer setup. Digitizer setup is required for entering map features accurately. By connecting the digitizer to the workstation, setup is completed successfully.

3.3 Developing Features with MGE Segment manager (MGSM)[2]:

We already discussed protocol stack of Modular GIS environment. MGE segment manager (MGSM) can run on top of MGE in the hierarchy of protocols. MGSM is a slottable application in the Modular GIS environment. It is helpful in performing

dynamic segmentation of linearly referenced data, including roads, railroads, rivers and pipelines. Dynamic segmentation is the geographic overlay and display of attributes describing conditions along a linearly referenced network. It builds the control network and analyzes the distributed attributes provided by the application developer. Input data can be from multiple referencing systems such as mileposts, reference marker and latitude/longitude systems. We may obtain input data from ODBC supported relational database systems. MGSM requires the following.

A digital map that models the network, which is prepared in Modular GIS environment.

Attribution for the digital network that creates an intelligent, base reference system for relating all distributed attributes back to the network.

Distributed attributes describing the network by linear or geographic location for analysis and display.

With the use of MGE segment manager, a query functionality of the distributed attribute information can be provided by building design files while providing full control of symbology and attribution. This can automatically place legends and generate reports from the output database tables. MGSM is used to manage systems where primary set of data is in reference to a linear network.

MGSM terminology [2]:

Linear Element: A design file graphic element (line or line string)

Network Linear Feature (NLF) – One or more linear elements that represent a single linear object.

Network Linear Feature ID – A name used to identify a particular NLF.

Control Point – A graphic element and its associated MGE feature and attribute information that represent a known position on a NLF from which distributed attribute information can be referenced.

Attributes – Information stored in an MGE relational database table that is directly linked to graphic elements.

Distributed Attributes – Information stored in relational database tables having values that change along the length of a network linear feature. These are the attributed that are to be dynamically segmented.

Segments – Graphic elements output from dynamic segmentation. In MGE segment manager, segments can be either point or linear elements.

Distributed Attribute Reference Systems – these systems specify the columns in the distributed attribute tables that are used to determine the position along the Network Linear Features. Following are the systems which support for referencing distributed attributes.

1. Known Marker – Distributed Attribute Referencing System in which the beginning and ending points of distributed attributes are recorded as offsets from known points along the network linear features.
2. Distance – Distributed Attribute Referencing System that used distances from the origin of the network linear feature.
3. Distance Length – Distributed Attribute Referencing System that uses a distance from the origin to specify where the distributed attribute starts and a length form the point to specify where the distributed attribute ends.

4. Geographic XY – The beginning and ending points of distributed attributes are recorded in the primary coordinate system's geographic units.
5. Projected XY – The beginning and ending points of the distributed attributes are recorded in the primary coordinate system's projected units.

Secondary ID – A value that can be used to ensure that the control point or distance is unique.

District – An area in the data set in which control point ID and Network Linear Feature ID are guaranteed to be unique and in which the 3 D distances do not overlap.

District ID – This ID is used in Control Network Builder and Coordinate File Builder to specify the Districts that will be processed.

Noncontiguous Network Linear Features – A network linear feature that consists of multiple components.

Component – One or more linear elements that make up one contiguous portion of a Network Linear Feature.

2 D Distance – The map distance, the distance that is measured from one point to another along a network linear feature in a 2 D map design file, does not take into account changes in elevation.

3 D Distance – The true distance along the real world Network Linear Feature, does take into account changes in elevation.

2-D/3-D ratio – Ratio that allows accurate placement of a distributed attribute value specified in 3-D distance onto a 2-D map.

Fixed Offset – This is the offset value used to display the distributed attributes at an offset distance from the centerline.

Scaled Offset – This offset value is a ratio of the stored database value used to display the distributed attributes at a scaled offset distance from the centerline.

Concurrent Network Linear Features – When one Network Linear Feature is intended to represent multiple Network Linear Feature identifiers. MGSM handles concurrent Network Linear Features if multiple linkages pointing to the different Network Linear Feature names are attached to a single linear element.

MGSM workflow:

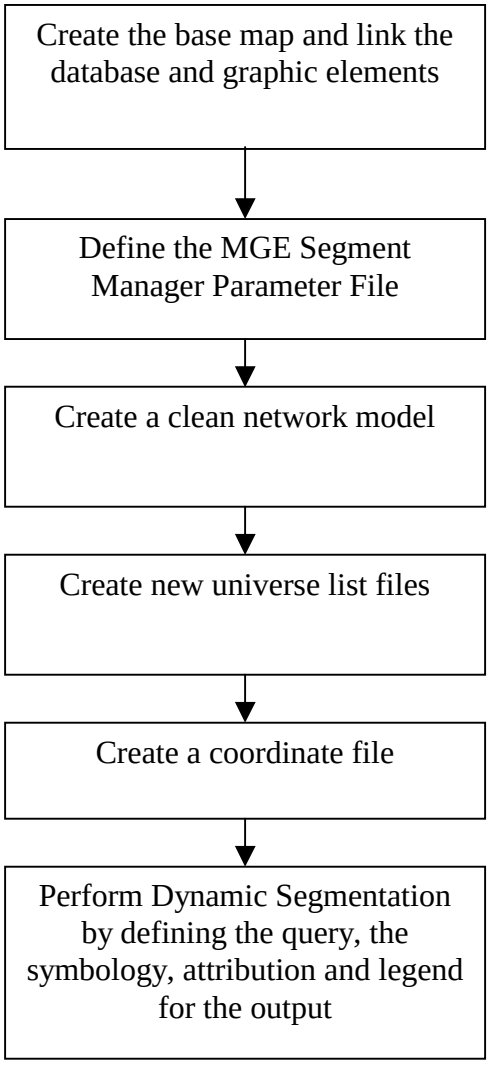


Figure 10: MGSM work flow

Create the Base map and link the database and features:

Creating the map is the first task in the project. We can create an attribute base-map using MGE or MGSM to link the graphics and the database tables.

Define a parameter file [2]:

A parameter file is used to describe the structure of the database tables, and column names to other MGE segment manager options. The parameter file lets MGSM what the table names and column names are in the database that represent the network linear features, the control points, and the distributed attributes that will be used by MGSM.

Create a clean network:

We have to verify a linear referencing system information into a representation that is compatible with MGSM. It may require to use existing control points that we enter to determine the distances along the network linear features or it may be required to create control points from linear distances stored on the network linear features.

Universe List Files (ULF)[2]: Once we have the clean network, we can create new universe list files (ULF) using the control-point design file and linear-element design file.

Create a coordinate file [2]: The coordinate file is used when we perform dynamic segmentation. To build this we use a tool of MGE coordinate file builder. That will create an output coordinate file that contains all of the coordinates of every vertex of every element in every Network Linear Feature and control point. The purpose of creating a

coordinate file is to make the process of dynamic segmentation very fast. The time consuming process of ordering the control network information is performed.

Perform Dynamic Segmentation [2]: Dynamic segmentation is a great tool of displaying the user-defined queries on to the actual map. Example, select all the segments where the road conditions are poor and the number of accidents is greater than 15. By performing dynamic segmentation, the user has a great idea about the current conditions of the problem on the local area. The physical display of the query on to the map as an overlay is a great benefit of performing dynamic segmentation. This supports the definition of a database query and outputs segments (linear or point) to a design file. Segments can be overlaid to create new segments. The resulting segments can be feature tagged, attributed to existing distributed attribute tables or attributed to a new table. The query can be from one or more distributed attribute tables with selections based on values stored in the tables. The output segments can be based on output feature.

MGSM Input Data: A control network is a set of geographic data that has been structured for segmentation. A control network consists of Network Linear Features and control points. Control network can easily be built by selecting feature and attribute data from an MGE project dataset and from relational database tables.

Network Linear Feature is a generic term that refers to a linear feature on which the distributed attributes are to be applied. In MGSM, a network linear feature is composed of one or more linear elements with the same feature tag and attribute table linkage. A unique value in the attribute table record for each element identifies it as a component of a particular NLF.

NLF [2] Database requirements:

1. Each graphic element must have a linkage to both a feature and to its attribute table. The attribute table must have a column containing the Network Linear Feature ID and the record for each feature must have the Network Linear Feature ID properly loaded.

2. There may be one or more attribute tables that contain NLF records. When more than one NLF table is used, it is important that all records for a single NLF be stored in the same NLF table.

3. The record for each element in the attribute table must contain a unique identifier value that matches that of the other elements in the NLF.

4. Segmentation depends on these identifiers for matching control points distance information with the NLF on which the control points are located. MGSM uses control points along the NLF to represent 3-D distance information for the NLFs.

5. A concurrent route, which represents two or more NLFs, must be represented by one graphic element with multiple feature and attribute linkages.

Network Linear Feature Graphic Elements:

Each network linear feature should be represented by a set of linear elements. These elements may be in one or more design files. The end coordinates of these elements should match except for noncontiguous NLFs.

Each element that represents a NLF must have at least one linkage to a NLF table and a corresponding feature linkage. In the case where the element represents more than one NLF, the element will have more the one linkage to the NLF tables.

Network Linear Feature Data requirements:

The following columns may be configured for a NLF table.

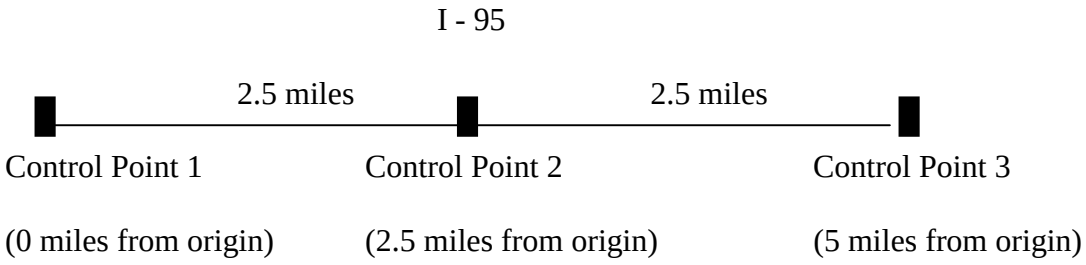
Column Name	Required Column	Supported Types	Suggested Types
Network linear feature id	Yes	Char or int	Char or int
Secondary id	No	Char or int	Char or int
Begin distance	No	Int, real, double	Double
End distance or Length	No	Int, real, double	Double

Control Points [2]: A control point is a digitized point feature that is located at a x, y location on a Network linear feature. Three key attributes are associated with a control point.

1. A unique identifier whose value matches that of the NLF on which the point is located.
2. A value for the 3-D distance from the origin of the NLF to the location of the control point.
3. An optional identifier that uniquely identifies the control point on its associated NLF.

The distance from origin values on control points in a control network is the basis for all dynamic segmentation.

Example: A network linear feature composed of two contiguous linear elements with control points at the start and end of each element. Attribute values on the control points indicate their known locations in terms of 3-D distance from the origin of the network linear feature.



Control Point database requirements:

A single attribute table required for control points.

There must be a sinbgle Mge feature that is linked to the control- points table. This should be a point feature.

The control point table must contain a column that is consistent with the network linear feature ID columns used in NLF table and the distributed attribute table.

The control point table must contain a column for the control points' 3-D distance value.

The control point table must contain a column for the named control point ID if Known Marker System is used.

Control Point Graphic Elements:

The control point graphic elements represent the control points and can be stored in multiple design files.

If control points are provided, they must have a linkage to the control point attribute table and they should reside on a vertex of the network linear feature graphic elements.

Control Point Data requirements:

Column Name	Required Column	Supported Types	Suggested Types
Network linear feature ID	Yes	Char and int	Char and int
Secondary ID	No	Char and int	Char and int
Control point ID	Yes(for known marker system)	Char and int	Char and int
3-D distance from the origin	Yes	Int, real, double	Double
2-D distance from origin	No	Int, real, double	Double

3.4 Showing intersections on the map:

The intersection nodes on the map are the result of the query running to a manually created database for intersection nodes. The intersections are laid on top of the NJ map, as a separate GIS layer using Dynamic Segmentation.

In this case, the intersections are laid on top of the NJ map by performing dynamic segmentation. Dynamic segmentation supports the definition of a database query and outputs segments (linear or point) to a design file. Segments (both linear and point) can be overlaid to create new segments. The resulting segments can be featured tagged, attributed to existing distributed attribute tables, or attributed to a new one. The

query can be from one or more distributed attribute tables with selections based on values stored in the tables.

A query is a request of information. It is a request for the features that meet the conditions that we define and/or a request for certain information about the features. To find the features that meet the condition, one-query feature classes in any open warehouse in the GeoWorkspace or query previously built queries. Queries are stored in the GeoWorkSpace so that, if a warehouse changes, all queries are updates each time they are displayed. If a spatial filter is applied to the warehouse connection at the time the query is defined, the query is limited to the geographic area defined by the spatial filter. Geomedia Professional [3][12] scans the query area for the features that meet the conditions and then displays the results geographically in a map window or in a tabular format in a data window. An entry for the query result is added to the legend, and its display can be manipulated through the legend properties like any other legend entry. Basically, a query can be treated just like a feature class.

Chapter 4

Data Flow

4.1 Universal Traffic Data Format (UTDF):

Universal Traffic Data Format (UTDF) [4] is a standard specification for transferring data between various software packages. UTDF can also be used to share data between software and traffic signal controller hardware. UTDF is useful in storing multiple volume counts and multiple timing plans for the same intersection.

Hardware Related:

- Existing detectors can be refitted to provide traffic counts and be stored in UTDF.
- A library of timing plans can be stored in UTDF and uploaded to the controller on demand.
- A generation 1.5 traffic control system can be developed that automatically performs the above steps in conjunction with the analysis software in real time.

Software Related:

- UTDF allows data to be shared between software packages. It is anticipated that many software developers will support UTDF. In this scenario data is entered once and then used by all the software together.
- It is possible for planning departments to store traffic counts for various scenarios and use them for capacity analysis as well as other purposes. With UTDF compatible software it could be possible for planners to completely automate traffic impact studies for future development and roadway improvements.

UTDF uses text files to store and share data. Both comma delimited (CSV) and column aligned text files are supported. We use comma delimited (CSV) files for UTDF data.

To edit the column aligned files simply open the files with a text editor such as NOTEPAD or MS-WORD. To edit the CSV files simply open the files with a spreadsheet such as MS-EXCEL. Be sure to save the file as comma-aligned and not as Excel format.

4.2 Importing Data from Synchro [4]:

In our application, we want the Synchro [4] data to display on Geo workspace of Geo media Professional. As described earlier, to model and to optimise the traffic signal timing of the section of the NJ ROUTE 18, we use Synchro [4] traffic software only. But our application does is to locate the data on the actual GIS map. To locate the Synchro [4] data on the actual GIS map, it is needed to import the data written in UTDF. To keep track of data at any time and for the records of the data, we save the Synchro [4] data in the database Access. Actually, the data, and number of attributes are different for each intersection of the Route 18. If there is no consistency in number of attributes, we want to create such a program, that creates the database, creates the table, read the attributes from the UTDF file, creates such fields, append the table into the database and finally import all the data to the created database.

I have created a program called Project5, x.frm. This program plays a very important role in the implementation of our software. Here, the logic behind the program is we read the lines from the UTDF file, line by line. Putting those lines into an array, the line, which contains the names of the attributes, is being read. The space character is

considered as a field delimiter. So, we provide such functions, which separate the attributes from the single line, and putting those names into an array. Then, we are creating the database, creating the table, creating the fields where, field names are being assigned from the array into which attribute names are stored after split. The idea behind creating the database automatically in such a way is the inconsistency of the attributes in the same type of data for each intersection.

Once the database is created, we call the function, which imports the UTDF data into the database. The logic behind to split the UTDF data separately is the same as described above. We read line by line from the UTDF data file. Then, since space character is the delimiter, the relevant data are stored into an array for one record at a time. It then inserts the data into the database table and saves the record. Then it reads the next line, splits the data and stores into database. The program does this function until end of file condition occurs. The same thing is repeated for the remaining intersections' data.

In brief, when we run the VB program, it creates the databases for all the intersections, for all the data. Like, it creates separate database for Lane Data and Volume Data for one intersection. It also creates separate database for Lane Data for one intersection and Lane Data for the other intersection. While running the program, if one found that the databases are already created, the program is not creating again, but simply continue importing the data.

4.3 Exporting Data:

We provide the option to update the traffic data of the particular intersection through the displayed form on GIS map. The user also would wish to see the effect of the

changed data in Synchro [4]. For that, to maintain the data consistent, it has to be updated in Synchro [4]. For the user interface, the data are stored in the database. So, when user updates the data, the update takes place only in database and not in Synchro [4]. Therefore, we must export the data from database to Synchro [4]. The user can export data back to Synchro [4] by clicking one command button on main VB form for that intersection. Synchro [4] now, can read the data from the UTDF files and analyses in Synchro [4].

Here, it is noted that UTDF files are .CSV (Comma delimited Files) and comma (,) is used as a delimiter.

4.4 Flow Diagram:

A Flowchart of the interface of Geographic Information System and Synchro [4]:

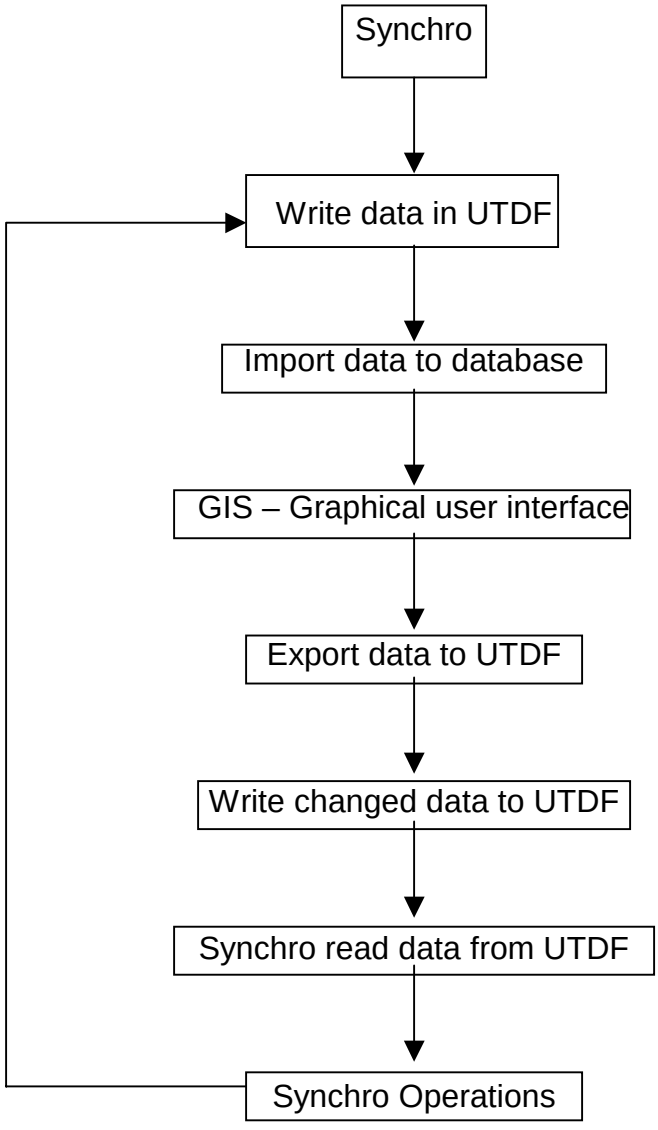


Figure 11: Flow diagram

Chapter 5

User Interface

5.1 Displaying Data to the Form:

This can be done through the visual basic [6][7] form. Here only modal VB forms are loaded in Geomedia Professional [3][12]. So, to display the VB form, it is necessary to make the form modal. The initial form asks to select what kind of data is needed to display. When the user selects the type, press the command “show data”, the other VB modal form is loaded and the data are displayed. The data can be updated in the form itself and immediately, the database data also get updated. To get the output for the changed data, the data are being exported to Synchro [4] again through UTDF. The optimisation is carried out, and again the results are being transferred to the database again to display in Geomedia Professional [3][12].

5.2. User Interface:

User interface on geo server is in terms of geo workspace. The system is built using Intergraph’s Geomedia Professional [3][12]. To run the application on the Internet, we may use Geomedia web map. The interface looks similar to this.

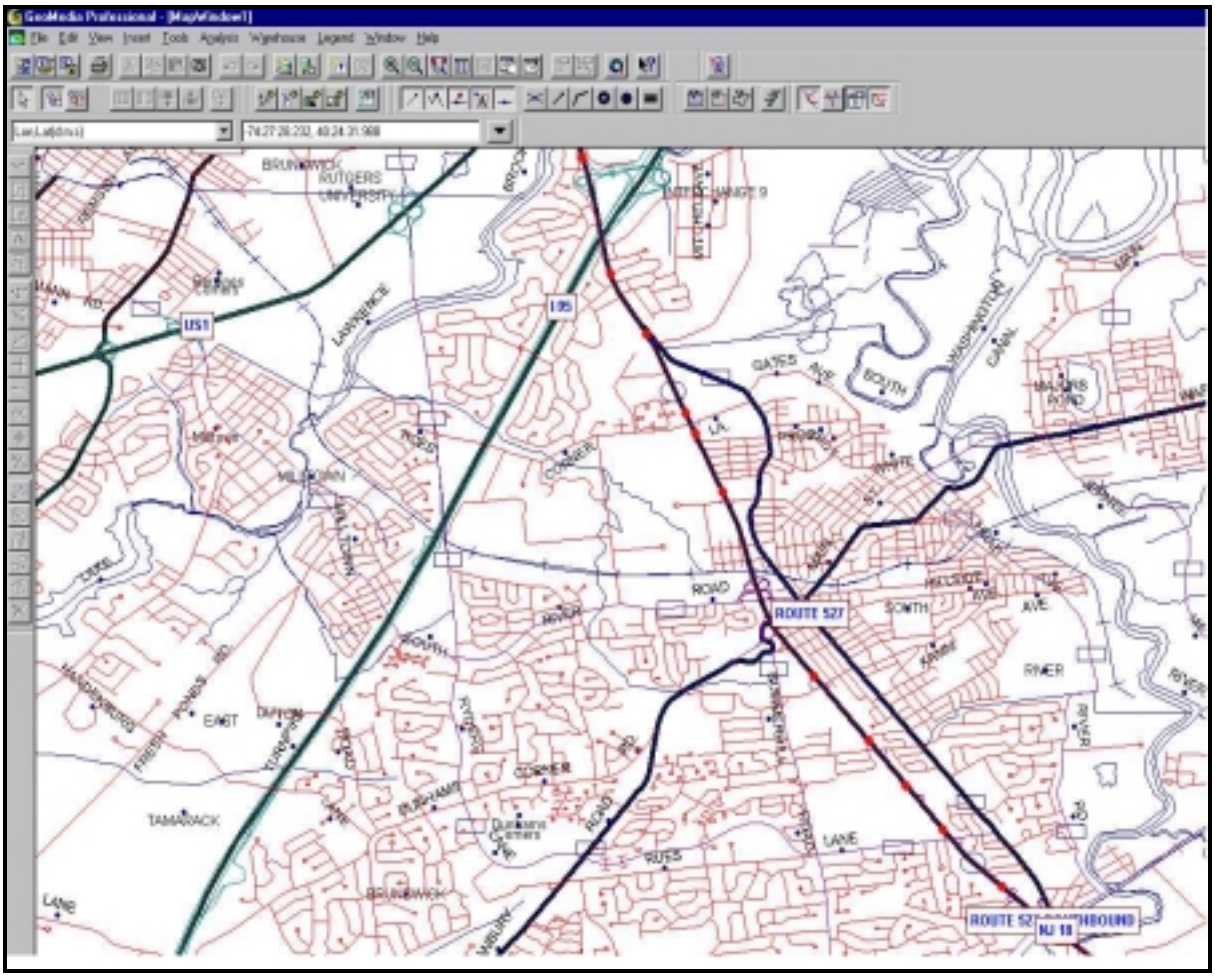


Figure 12: User Interface

User clicks on the intersections, dotted in red. The system provides the data.

5.3 Data forms:

Main VB form: This form is loaded when user clicks the intersection in GIS map. This form is the primary toolbar for all the user actions. There are five option buttons - Lane Data, Phasing Data, Volume Data, Timing Data, Layout Data. There are four command buttons - Import Data, Show Data, Export data, Close. To display the respective data, the user selects one type from the option button and presses the show data button. Depending

on the option button selection, the respective form is loaded. It also shows the name of the intersection on the upper right corner of the form.

Layout Data Form: This form looks similar to the Intersection Properties window in Synchro [4]. It shows X, Y co-ordinates of the intersection, other layout properties.

Lane/Phasing/Timing/Volume Data Form: This form looks similar to the Lane/Phasing/Timing/Volume Data Window in Synchro [4]. It shows all the data, which are shown in Lane/Phasing/Timing/Volume Window in Synchro [4]. Here below it is showed the comparison of the two forms. It is noted that Synchro [4] is optimizing software, and it outputs the data. As in our project also we depend on Synchro [4] too for traffic optimization, we never show output data into our forms in the interface.

These data are taken from UTDF (Universal Traffic Data Format). Synchro [4] writes the data into UTDF, and that data is taken to display in our interface. In some cases, instead of data, we see "N/A (Not Applicable)". This is because; Synchro [4] doesn't share this data through UTDF with any other software. Hence, it is impossible to show them in our interface.

5.4 How does the software function?

Click any of the intersection, main menu form is loaded. Main Menu contains the option button for each type of data. There are four command buttons also: Export data, Import Data, Show Data, Close.

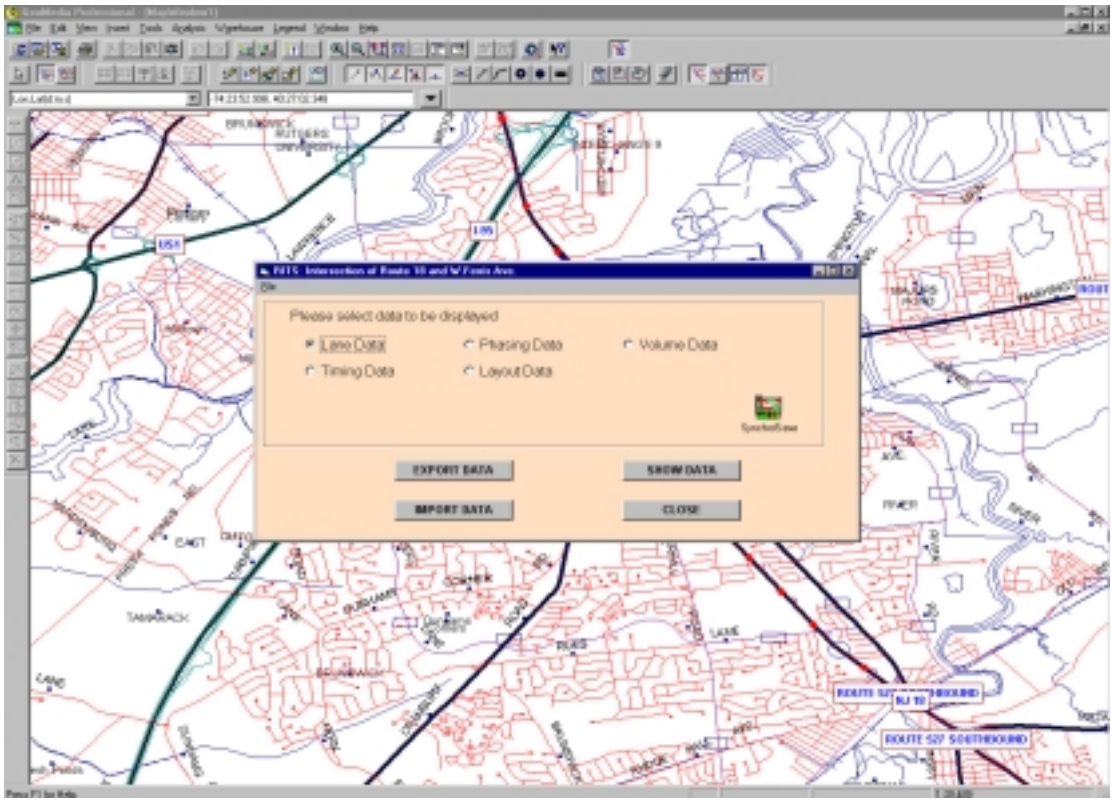


Figure 13: Main Menu Form

Show Data Feature:

If user clicks show data, the respective Synchro [4] data has been showed. Main menu and data display are shown in figure 2 and figure 3 respectively.

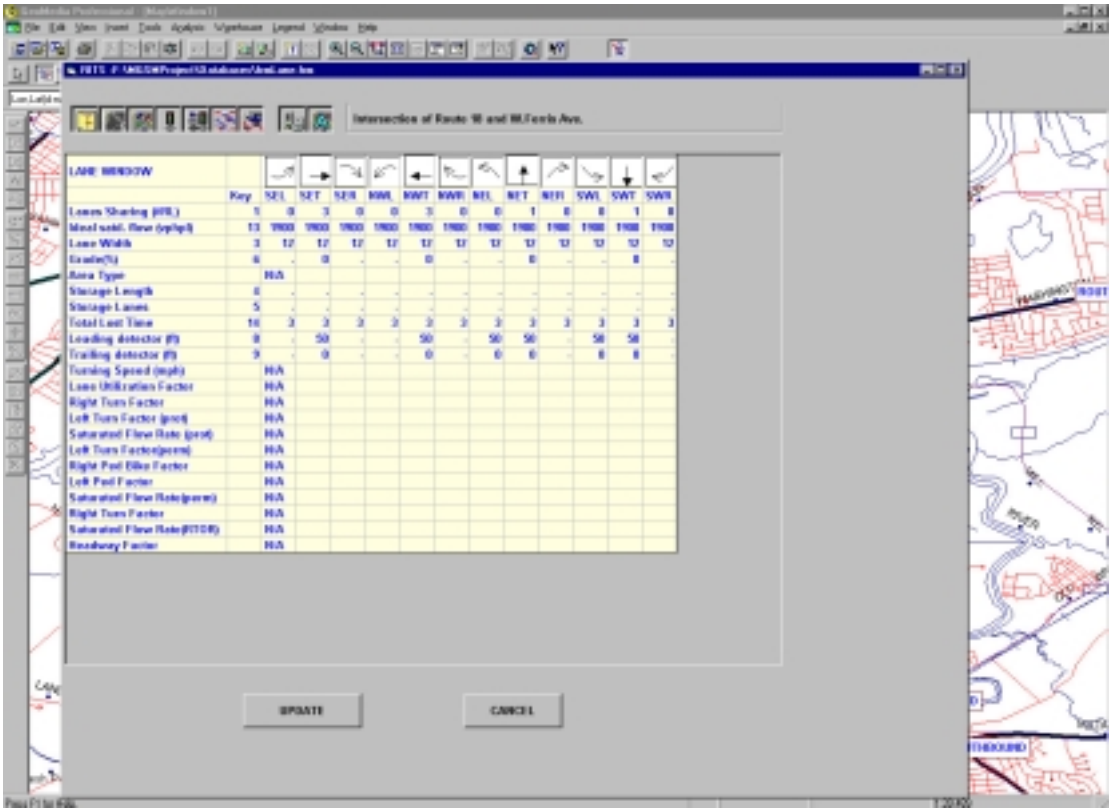


Figure 14: Display of Lane Data

Importing Feature: Explained in the components section.

Exporting Feature:

The user can change the data in the show data form. The user clicks “UPDATE” button, which is located at the bottom of the form, and the respective database is being updated. As soon as, the user clicks the Export data button on the main menu, the software writes the current database data as the UTDF files in the folder MGSMProject\Trafficware. Synchro [4] has a nice feature that can read the data in UTDF format. The Exported data are saved in the MGSMProject\Trafficware folder, as a comma delimited files. User exports that data into Synchro [4].

Chapter 6

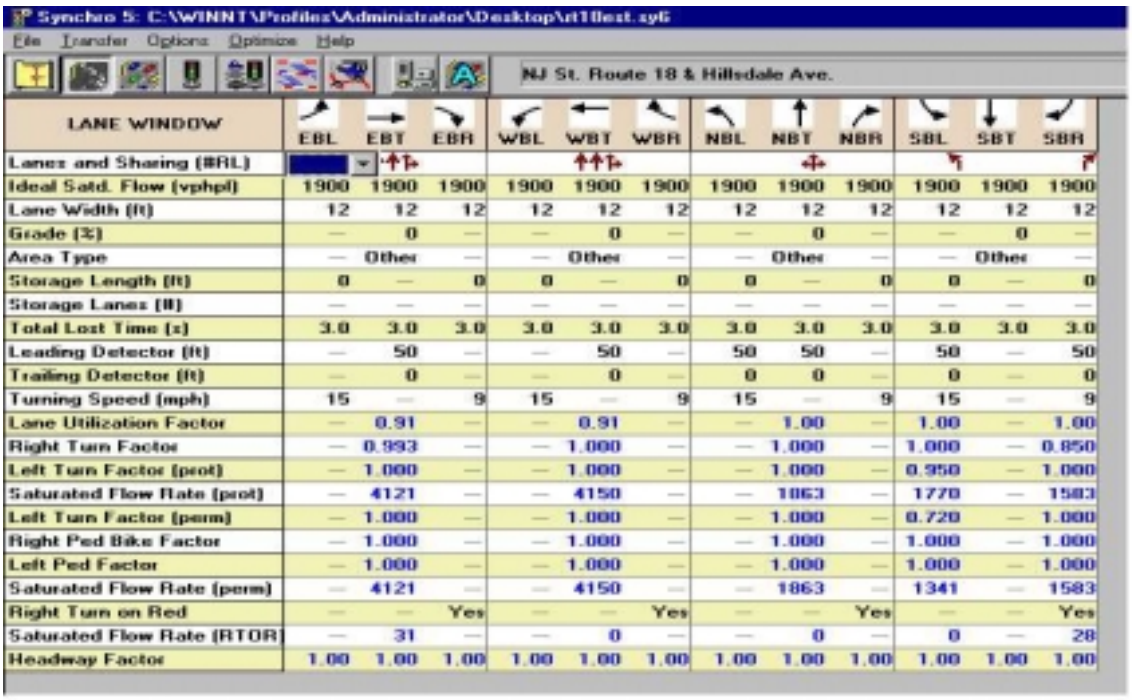
Conclusion

6.1 Results: Form comparison of Synchro vs. our application:

Here below it is showed the comparison of the two forms. One form is from Synchro [4] and the other one is from our interface of Synchro [4] with GIS. It is noted that Synchro [4] is optimizing software, and it outputs the data. As in our project also we depend on Synchro [4] too for traffic optimization, we never show output data into our forms in the interface.

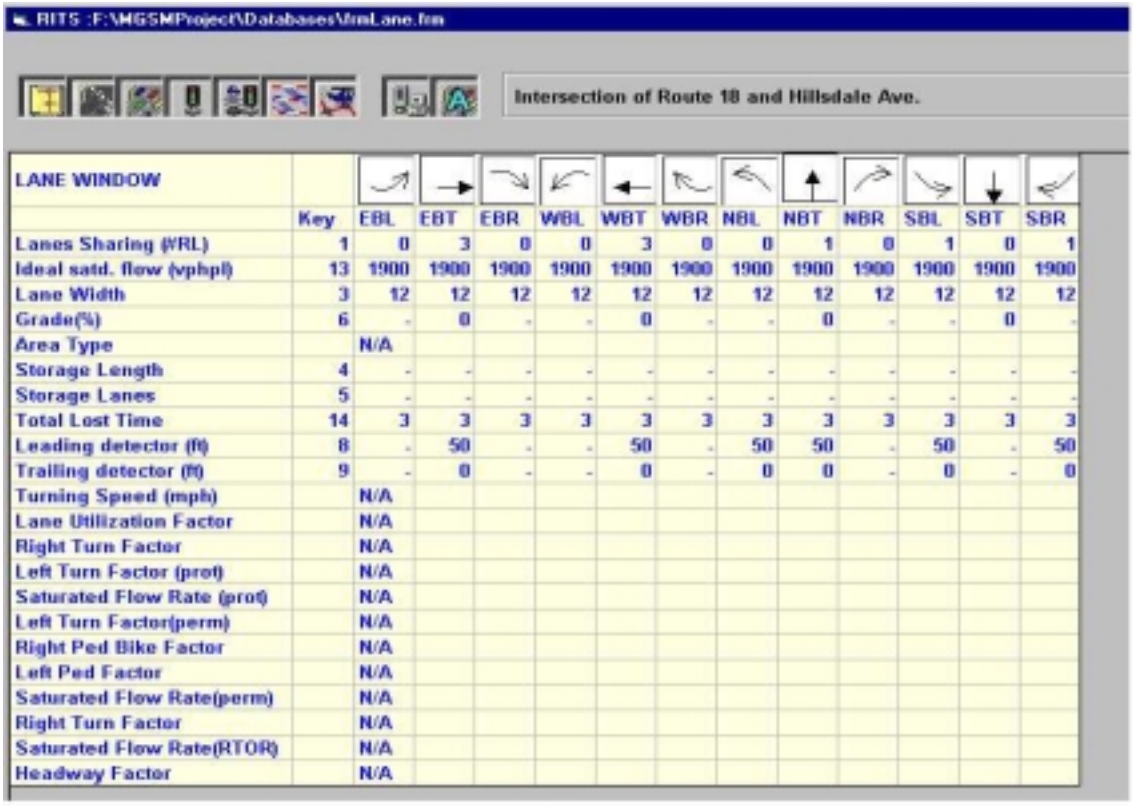
These data are taken from UTDF (Universal Traffic Data Format). Synchro [4] writes the data into UTDF, and that data is taken to display in our interface. In some cases, instead of data, we see “N/A (Not Applicable)”. This is because; Synchro [4] doesn’t share this data through UTDF with any other software. Hence, it is impossible to show them in our interface.

LANE DATA



LANE WINDOW	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
Lanes and Sharing (#RL)												
Ideal Satd. Flow (vphpl)	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Lane Width (ft)	12	12	12	12	12	12	12	12	12	12	12	12
Grade (%)	—	0	—	—	0	—	—	0	—	—	0	—
Area Type	—	Other	—	—	Other	—	—	Other	—	—	Other	—
Storage Length (ft)	0	—	0	0	—	0	0	—	0	0	—	0
Storage Lanes (#)	—	—	—	—	—	—	—	—	—	—	—	—
Total Lost Time (s)	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Leading Detector (ft)	—	50	—	—	50	—	50	50	—	50	—	50
Trailing Detector (ft)	—	0	—	—	0	—	0	0	—	0	—	0
Turning Speed (mph)	15	—	9	15	—	9	15	—	9	15	—	9
Lane Utilization Factor	—	0.91	—	—	0.91	—	—	1.00	—	1.00	—	1.00
Right Turn Factor	—	0.993	—	—	1.000	—	—	1.000	—	1.000	—	0.850
Left Turn Factor (prot)	—	1.000	—	—	1.000	—	—	1.000	—	0.950	—	1.000
Saturated Flow Rate (prot)	—	4121	—	—	4150	—	—	1863	—	1770	—	1583
Left Turn Factor (perm)	—	1.000	—	—	1.000	—	—	1.000	—	0.720	—	1.000
Right Ped Bike Factor	—	1.000	—	—	1.000	—	—	1.000	—	1.000	—	1.000
Left Ped Factor	—	1.000	—	—	1.000	—	—	1.000	—	1.000	—	1.000
Saturated Flow Rate (perm)	—	4121	—	—	4150	—	—	1863	—	1341	—	1583
Right Turn on Red	—	—	Yes	—	—	Yes	—	—	Yes	—	—	Yes
Saturated Flow Rate (RTOR)	—	31	—	—	0	—	—	0	—	0	—	28
Headway Factor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Figure 15: Lane Data from Synchro



LANE WINDOW	Key	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
Lanes Sharing (#RL)	1	0	3	0	0	3	0	0	1	0	1	0	1
Ideal satd. flow (vphpl)	13	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Lane Width	3	12	12	12	12	12	12	12	12	12	12	12	12
Grade(%)	6	-	0	-	-	0	-	-	0	-	-	0	-
Area Type		N/A											
Storage Length	4	-	-	-	-	-	-	-	-	-	-	-	-
Storage Lanes	5	-	-	-	-	-	-	-	-	-	-	-	-
Total Lost Time	14	3	3	3	3	3	3	3	3	3	3	3	3
Leading detector (ft)	8	-	50	-	-	50	-	50	50	-	50	-	50
Trailing detector (ft)	9	-	0	-	-	0	-	0	0	-	0	-	0
Turning Speed (mph)		N/A											
Lane Utilization Factor		N/A											
Right Turn Factor		N/A											
Left Turn Factor (prot)		N/A											
Saturated Flow Rate (prot)		N/A											
Left Turn Factor(perm)		N/A											
Right Ped Bike Factor		N/A											
Left Ped Factor		N/A											
Saturated Flow Rate(perm)		N/A											
Right Turn Factor		N/A											
Saturated Flow Rate(RTOR)		N/A											
Headway Factor		N/A											

Figure 16: Lane Data from Interface with GIS

PHASING DATA

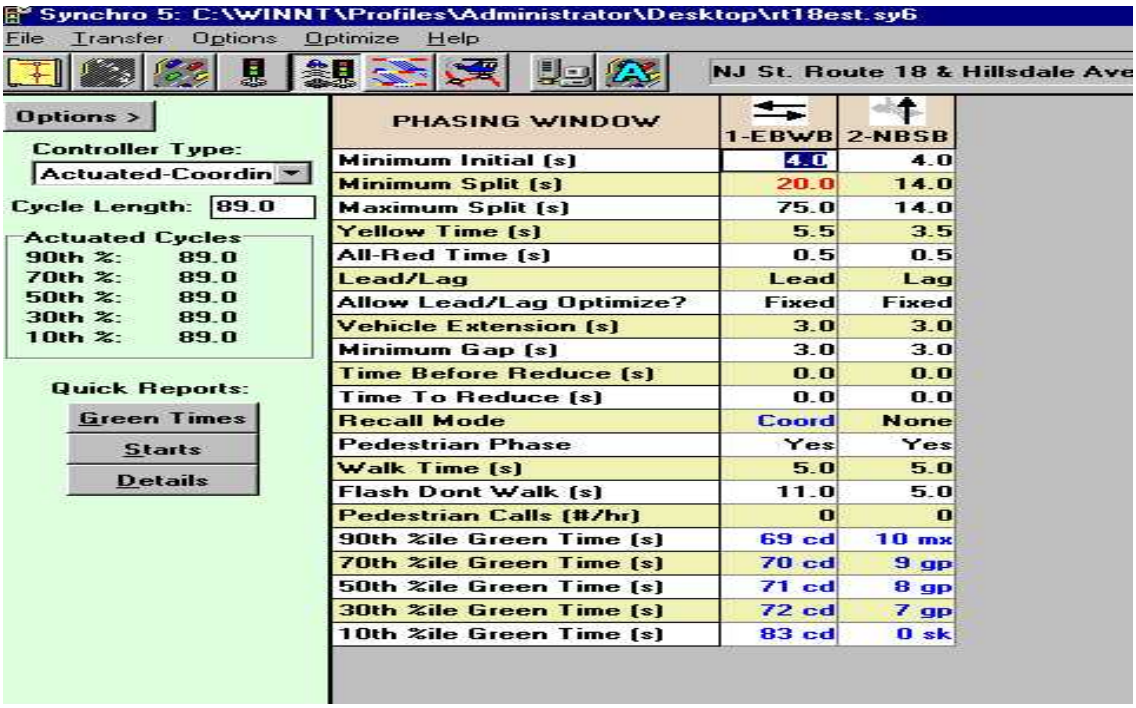


Figure 17: Phasing Data from Synchro

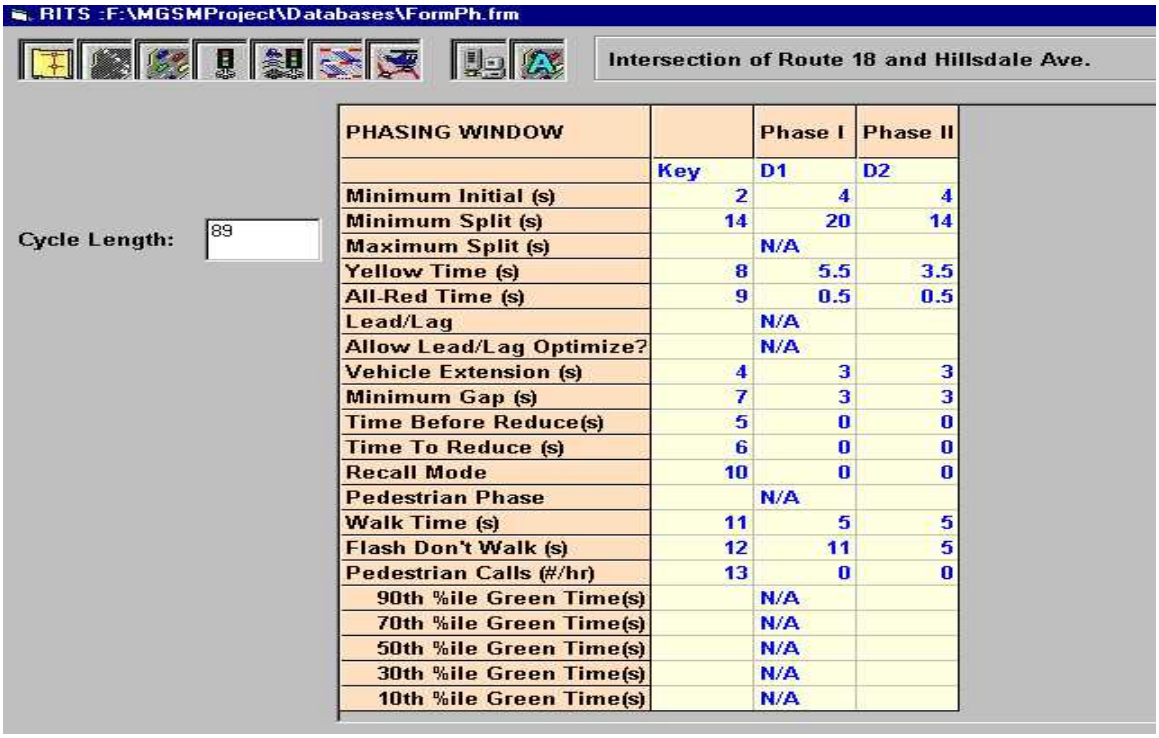


Figure 18: Phasing Data from Interface with GIS

TIMING DATA

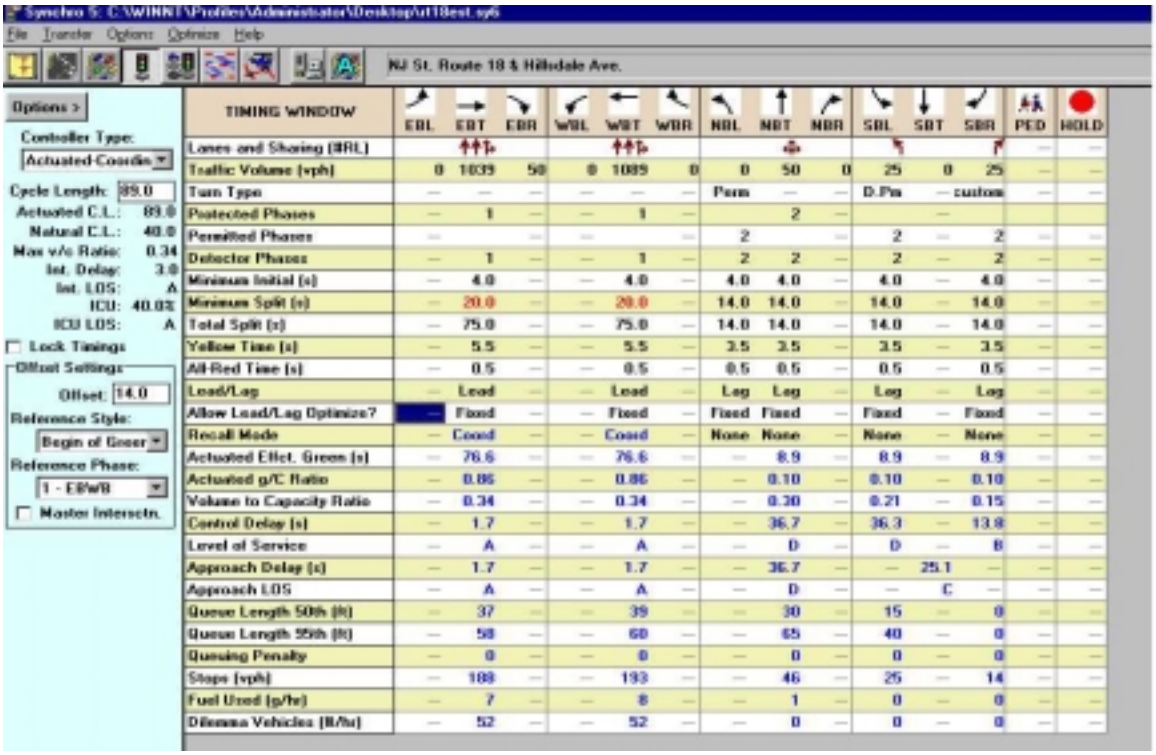


Figure 19: Timing Data from Synchro

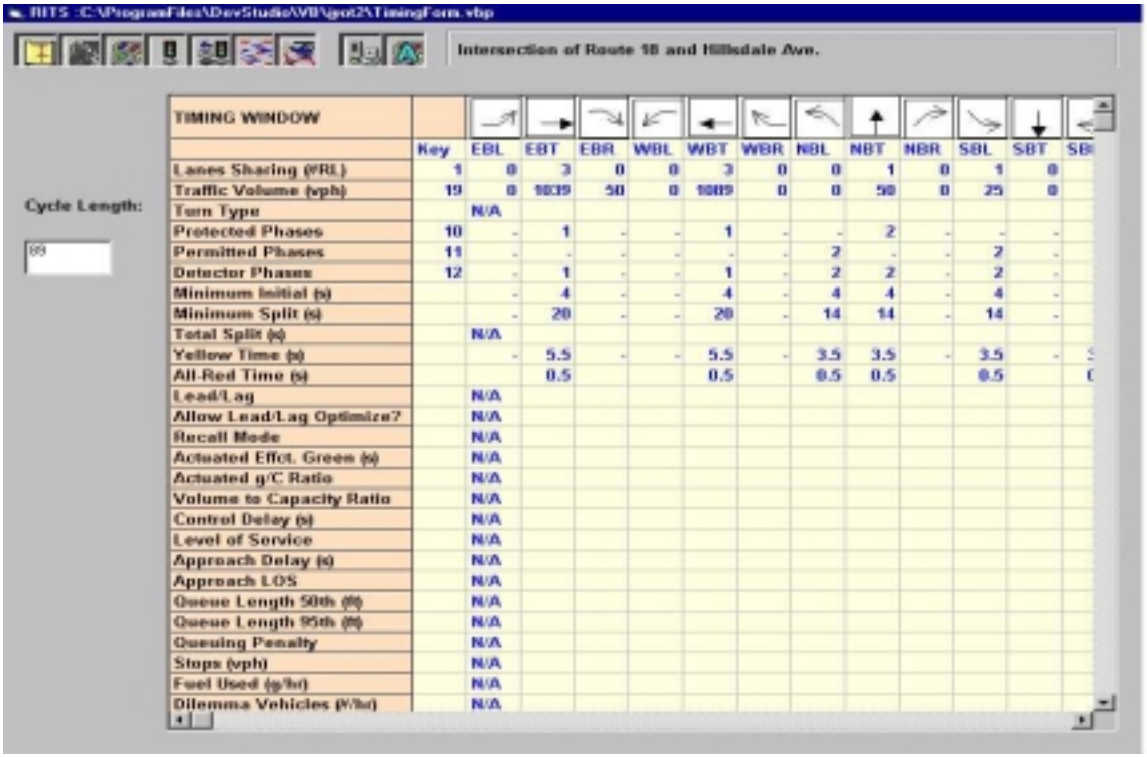


Figure 20: Timing Data from Interface with GIS

VOLUME DATA

Synchro 5: C:\WINNT\Profiles\Administrator\Desktop\rt18est.sy6

File Transfer Options Optimize Help

NJ St. Route 18 & Hillsdale Ave.

VOLUME WINDOW	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
Traffic Volume (vph)	0	1039	50	0	1089	0	0	50	0	25	0	25
Conflicting Peds. (#/hr)	0	—	0	0	—	0	0	—	0	0	—	0
Conflicting Bikes (#/hr)	—	—	0	—	—	0	—	—	0	—	—	0
Peak Hour Factor	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
Growth Factor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Heavy Vehicles (%)	25	25	25	25	25	25	2	2	2	2	2	2
Bus Blockages (#/hr)	0	0	0	0	0	0	0	0	0	0	0	0
Adj. Parking Lane?	No	No	No	No	No	No	No	No	No	No	No	No
Parking Maneuvers (#/hr)	—	—	—	—	—	—	—	—	—	—	—	—
Traffic from mid-block (%)	—	0	—	—	0	—	—	0	—	—	0	—
Link OD Volumes	—	EB	—	—	—	—	—	—	—	—	—	—
Adjusted Flow (vph)	0	1154	56	0	1210	0	0	56	0	28	0	28
Lane Group Flow (vph)	0	1210	0	0	1210	0	0	56	0	28	0	28

Figure 21: Volume Data from Synchro

RITS :C:\ProgramFiles\DevStudio\VB\jpot2\VolumeForm.vbp

Intersection of Route 18 and Hillsdale Ave.

VOLUME WINDOW	Key	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
Traffic Volume(vph)	19	0	1039	50	0	1089	0	0	50	0	25	0	25
Conflicting Peds. (#/hr)		N/A											
Conflicting Bikes (#/hr)		N/A											
Peak Hour Factor	22	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
Growth Factor	23	100	100	100	100	100	100	100	100	100	100	100	100
Heavy Vehicles(%)	24	25	25	25	25	25	25	2	2	2	2	2	2
Bus Blockages (#/hr)	25	0	0	0	0	0	0	0	0	0	0	0	0
Adj. Parking Lane?		N/A											
Parking		N/A											
Traffic from		N/A											
Link OD Volumes		N/A											
Adjusted Flow(vph)		N/A											
Lane Group Flow (vph)		N/A											

Figure 22: Volume Data from Interface with GIS

LAYOUT DATA

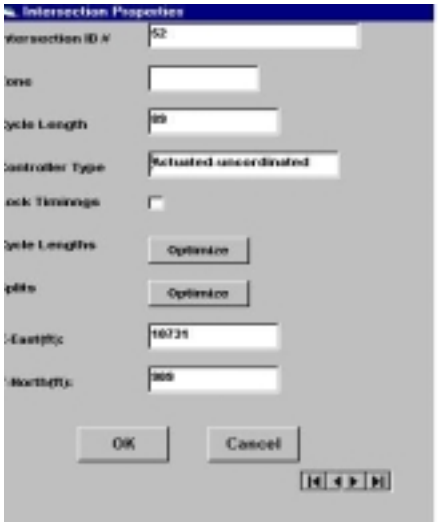


Figure23: Layout Data in Synchro

Figure24: Layout Data in Interface with GIS

6.2 Proposed Web Architecture:

Conceptual Model

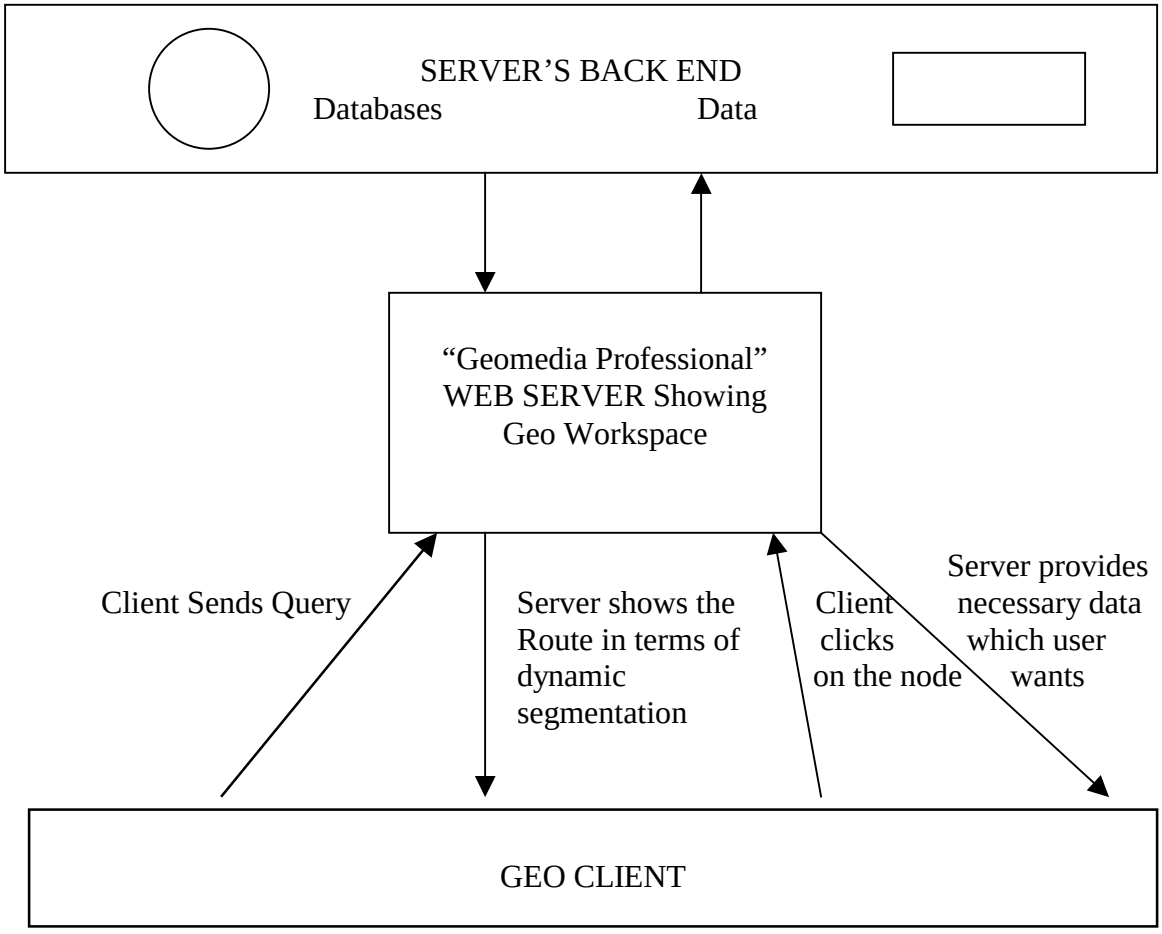


Figure 25: Proposed web architecture

'Geomedia Professional [12]'® is a platform to run a GIS applications supporting web server. The prerequisite on the client side is, the browser has to be ACGM (Active Computer Graphics Metafile) [13] compatible.

Issues to be considered in the architecture:

- **Scalability:** How many web clients are supported by the Geo server at a time? In our project, the clients are not many, and are mostly private consultant agency, government's department of transportation, researchers, etc. Although scalability issue is important, if

the network is not scalable than also it is workable. To achieve higher scalability our application can be built on existing network load balancing software “Windows Load Balancing Service” (WLBS). WLBS dynamically distributes IP traffic across multiple cluster nodes and provides automatic fail over in case of node failure.

- **Availability:** WLBS cluster server sends a signal to the other nodes in the cluster and listens for the signal from the others. If a server in a cluster fails, the remaining nodes adjust and take over the load.
- **Security:** Only authorized users only are allowed to access the geo server. The authorized users, in our system, are the government agency, state’s department of transportation, legal consultants, analysts /researchers for the study of the route.
- **Reliability:** The whole system is reliable, as this system runs on Internet. Thus, for transmission, HTTP, TCP, IP protocols are used. So, we can say that the system is reliable.
- **Concurrent Data / Data consistency:** Data consistency is the prime issue we want to consider for the design of the system. Here, in our case, only legal users are allowed to access the system and thus update the data for the particular route. If one user’s transaction is in process, the “rights” for that intersection has been assigned to the very user. So, no other user can change the data nor can access the data at that time.

Developing a GIS web site:

Developing MGSM project on the World Wide Web has many advantages.

The client does not need to go through the tedious installation procedure of MGSM Project.

MGSM Project is very sensitive in version compatibility of supportive software, such as it requires particular version of Microsoft Access, Geomedia Professional [3][12].

World Wide Web is a universal tool, and need not worry about the stand-alone machine. The client can access the information from any machine connected to the web, provided the browser is ACGM capable.

Need not worry about maintaining the recent copy of the database. In MGSM Project, two clients can change the data independently to view the changes in analysis. In that case, it is difficult to maintain the database on all the machines, which are being worked on the same intersection.

Geomedia Professional [12] [13] is a programming tool kit for web page authors. The map can be created from many different GIS data sources. The software only needs to be loaded on a single Microsoft Windows NT web server that is running Microsoft's Internet Information Server (IIS). It is also possible that the web page author can give clients the ability to dynamically create custom intelligent maps tailored to the user's particular needs. The output map generated on the web page is either ActiveCGM (Active Computer Graphics Metafile) file, JPEG (Joint Photographic Experts Group) or PNG (Portable Network Graphics). The map does not necessarily be static; it can be dynamically created at the moment the client makes the request to the server. Hence, it contains the latest information from the data sources available. An ACGM display has intelligent graphics capability. It contains raster data, vector data tool tips for map features, hotspot actions for the features and programmable action associated with window events on map features.

The key advantage is the client use a standard web browser such as Microsoft Internet Explorer or Netscape Navigator to display maps created directly from the GIS database. It also supports Active Server Pages (ASP) [7]. With ASP script, any event can be executed on the server and display that on to the client. ActiveCGM allows us to create maps that contain intelligent map features that are linked to records in the database. These intelligent features allow the client to click on a feature and display attribute information associated with the feature.

Dispatching requests to the webmap:

Geomedia Professional [12] is composed of several processes; one is the Map server manager and the other is the map server. The map server manager process is the entry point to the map creation process. It handles the access to the available map server processes. It allows as many processes as map server processes. The model can be a FIFO (First In First Out) queue for map requests coming to the web sites. The map server is a component object model that creates the output map. This is done by specifying the inputs necessary to produce a map, such as connections, features, symbology etc. When the map is produced, the map server is released and is ready to service the next request in the queue.

Geomedia web application generator:

The geomedia web application generator [13] is a custom geomedia command that generates a geomedia web map application using information and settings from the currently open geomedia geoworkspace and the currently active map window in geomedia. The initial map presented by the resulting Geomedia web application reflects the feature content and geographic extents of the active geomedia map window. All

feature classes that are visible in the geomedia map window are also visible in the resulting web application, with the same or similar symbology, priority and display range. Queries displayed in the original map window are displayed in the resulting web site, if it is possible to translate them into corresponding Geomedia web map queries, query scripts or markers. In addition to map graphics, the legend, navigation controls and scale bar are also displayed in the resulting web application.

Creating web sites with Internet Information Server (IIS):

Microsoft's Internet Information Server (IIS) [7] has exclusive control of its metadata, allowing it to query, compile and cache each web site as a standalone application. IIS improved scripting performance by implementing web applications isolated web process spaces that compile and cache ASP scripts and application objects. The first time a script in a web application is accessed, IIS compiles it into RAM-resident p-code that is maintained in cache. IIS executes that compiled code whenever subsequent requests are made, removing the overhead of reinterpreting the script. IIS allows the web developer to save global and user-specific variables so that web objects can be created and reused as necessary.

Base Work Flow:

The basic workflow for creating a map is as follows:

1. Returning a Map Server.
2. Assigning a coordinate system.
3. Connecting to a data source.
4. Defining feature sets.
5. Defining display rules for the feature sets.

6. Determining the range for the output map.
7. Generating the map.

6.3 Performance:

The software interface is very useful for traffic analysts. The scope of this project was limited to stand alone application. For the distributed application, I propose web architecture. To optimize the web service, we may use Windows Load Balancing Service (WLBS) to increase the scalability. Hardware side, we may develop a cluster consists of two or more web servers to distribute the load, and thus balances the load. As the number of server increases, the database will become saturated. In this case, we may have to use "mirror" database servers to handle the web load. Load testing is required to accurately determine the ratio of web servers to database servers. Eventually, when we serve from a large database, we may want to consider a private fiber network to directly connect critical devices.

6.4 Future Work:

- Scalability issues in the web architecture
- Oracle SQL Net connection instead of ODBC connection
- Use Oracle or SQL as a database server to improve performance rather than MSAccess
- More routes of the state of New Jersey
- Better Graphical User Interface
- More efficient setup program
- Copy right issues, password protection, license fee

APPENDIX I

Manual

How to install the software?

- **Run the setup program from the CD.**

Select the path where do you want to install MGSMProject in your computer. Install all the files under that path. Setup program will create a folder named MGSMProject under that directory hierarchy.

- **Setup an Environmental variable MGSM_Dir:**

Goto Control panel -> Systems ->Environmental variables

Variable: MGSM_Dir

Value : Path under which MGSMProject has been installed.

Ex : C:\Program Files\MGSMProject

- **Setup ODBC Connection:**

Goto Control panel -> Data Sources (ODBC)

Add new user data source.

Select Microsoft Access Driver (*.mdb)

Data Source name : projInter

Select the database as the whole path to MGE_Gis_Export.mdb database. This database is stored in Path to MGSMProject\MGSMProject\Databases folder.

If MGSMProject folder is stored in C:\Program Files then

Select C:\Program Files\MGSMProject\Databases\MGE_Gis_Export.mdb database.

Data Source name : Rutgers

Select the database as the whole path to MGE_Gis_Export.mdb database. This database is stored in Path to MGSMProject\MGSMProject\Databases folder.

If MGSMProject folder is stored in C:\Program Files then

Select C:\Program Files\MGSMProject\Databases\MGE_Gis_Export.mdb database.

- **Restart the computer to activate the system variables.**
- **Register the Geomedia command.**

Copy, abcd.dll and abcd.ini from MGSMProject\Databases, to the Program folder of Geomedia Professional [3][12].

Open the command prompt. Go to the directory of Program folder of Geomedia Professional [3][12]. (Ex. C:\Program Files\Geomedia Prfessional\Program>)

Install the Geomedia command by

Installusrcmd /prod “Geomedia Professional [3][12]” abcd.dll abcd.ini

Components of the Software:

- **Synchro [4] File:** rt18est.sy6
- **UTDF Files :** saved in the folder MGSMProject/Trafficware

  Lanes.csv, Layout.csv, Phasing.csv, Timing.csv, Volume.csv

- **Customized program to export the UTDF data into Access Database :**

This is a VB program. Running this program, the databases are created for 13 intersections of route 18. If the databases are already been there, this program opens the UTDF files, reads the traffic data i.e. Lane Data, Phasing Data, Layout Data, Timing Data and Volume Data. These data are being exported to the respective database. All these databases are being created in the folder MGSMProject\Databases\.

- **Databases :**

MGSMProject\Databases\MGE_Gis_Export.mdb

Lane Databases (Ex. LaneDB1.mdb for 1st intersection,

MGSMProject\Databases\LaneDB1.mdb)

Phasing Databases (Ex. PhasingDB1.mdb for 1st intersection,

MGSMProject\Databases\PhasingDB1.mdb)

Timing Databases (Ex. TimingDB1.mdb for 1st intersection,

MGSMProject\Databases\TimingDB1.mdb)

Layout Database (Ex. LayoutDB.mdb for all the intersections,

MGSMProject\Databases\Layout.mdb)

Volume Database (Ex. VolumeDB1.mdb for 1st intersection,

MGSMProject\Databases\VolumeDB1.mdb)

APPENDIX II

Read me File

Synchro Data

A Sample Lane Data file in UTDF:

Lane Group Data

```
RECORDNAME,INTID,NBL,NBT,NBR,NBR2,SBL,SBT,SBR,SBR2,EBL,EBT,EBR,WBL2,WBL,
WBT,WBR,NEL2,NEL,NET,NER,NWL,NWT,NWR,SEL,SET,SER,SWL,SWT,SWR

Lanes,15,0,2,0,,0,1,0,,0,3,0,,0,3,0,,,,,,,,,,,,

Shared,15,0,3,,,0,3,,,0,2,,,0,2,,,,,,,,,,,,

Width,15,12,12,12,,12,12,12,,12,12,12,,12,12,12,,,,,,,,,,,,

Storage,15,,,,,,,,,,,,,,,,,,,,,,,,,,,,

StLanes,15,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Grade,15,,0,,,,0,,,,0,,,,0,,,,,,,,,,,,

Speed,15,,30,,,,30,,,,55,,,,55,,,,,,,,,,,,

FirstDetect,15,50,50,,,50,50,,,,50,,,,50,,,,,,,,,,,,

LastDetect,15,0,0,,,0,0,,,,0,,,,0,,,,,,,,,,,,

Phase1,15,,2,,,,2,,,,1,,,,1,,,,,,,,,,,,

PermPhase1,15,2,,,,2,,,,,,,,,,,,,,,,,,,,,,

DetectPhase1,15,2,2,,,2,2,,,,1,,,,1,,,,,,,,,,,,
```

```
IdealFlow,15,1900,1900,1900,,1900,1900,1900,,1900,1900,1900,,1900,1900,1
900,,,,,,,,,,,,

LostTime,15,3,3,3,,3,3,3,,3,3,3,,3,3,3,,,,,,,,,,,,

SatFlow,15,0,3308,0,,0,1750,0,,0,4150,0,,0,4150,0,,,,,,,,,,,,

SatFlowPerm,15,0,3308,0,,0,1750,0,,0,4150,0,,0,4150,0,,,,,,,,,,,,

SatFlowRTOR,15,0,20,0,,0,20,0,,0,0,0,,0,0,0,,,,,,,,,,,,

HeadwayFact,15,1.00,1.00,1.00,,1.00,1.00,1.00,,1.00,1.00,1.00,,1.00,1.00
,1.00,,,,,,,,,,,,

Volume,15,50,50,50,,50,50,50,,0,2031,0,,0,2031,0,,,,,,,,,,,,

Peds,15,0,0,0,,0,0,0,,0,0,0,,0,0,0,,,,,,,,,,,,

Bicycles,15,0,0,0,,0,0,0,,0,0,0,,0,0,0,,,,,,,,,,,,

PHF,15,0.90,0.90,0.90,,0.90,0.90,0.90,,0.90,0.90,0.90,,0.90,0.90,0.90,,,
,,,,,,,,,

Growth,15,100,100,100,,100,100,100,,100,100,100,,100,100,100,,,,,,,,,,,,
,

HeavyVehicles,15,2,2,2,,2,2,2,,25,25,25,,25,25,25,,,,,,,,,,,,

BusStops,15,0,0,0,,0,0,0,,0,0,0,,0,0,0,,,,,,,,,,,,

Midblock,15,,0,,,,0,,,,0,,,,0,,,,,,,,,,,,,

Distance,15,,1125,,,,1051,,,,246,,,,232,,,,,,,,,,,,,

TravelTime,15,,25.6,,,,23.9,,,,3.0,,,,2.9,,,,,,,,,,,,,
```

Field Types of the Lane Data in UTDF :

Field	Example Data	Comments
IntID	12	Number identifying intersection
RecordName	"Lanes", Name of data to follow	
xL	102	data for Left lane group.
xT	487	data for Thru lane group.
xR	34	data for Right lane group.
xU	12	U-turn lane group
xL2	12	Hard left lane group, for intersections with 5 or more legs
xR2	34	Hard right lane group, for intersections with 5 or more legs

Record Types of Lane Data in UTDF :

Lanes: For each lane group, enter the number of lanes. Shared lanes count as through lanes, unless there is no through movement, in which case LR lanes count as left lanes,

Shared: Enter code for sharing of this lane with adjacent movements. This field specifies which movements the through lanes are shared with. Enter 0, 1, 2, 3 for No-sharing,

Shared-with-Left, Shared-with-right, Shared-with-both. Leave this field is normally 0 for turning lane groups. If a left lane shares with left2 or u-turns, the sharing is coded as a '1'.

Width: Enter the average lane width for the group in feet or meters. Decimal meters permitted.

Storage: Enter the length of a turning bay if applicable. Leave this field blank for through lane groups or if the turning lane goes all the way back to the previous intersection.

StLanes: Enter the number of lanes in the storage bay. This can be more or less than the number of turning lanes, but must be at least 1.

Grade: Percent grade, negative is downhill. Enter in primary lane group only.

Speed: Approach speed. Enter in mph. or kph. Enter in primary lane group only

FirstDetect: Distance from the leading extension detector to stop bar in feet or meters. Leave blank if no detectors for lane group.

LastDetect: Distance from the trailing extension detector to stop bar in feet or meters. Can be negative. Do not include calling-only detectors and type-3 detectors. Include extension detectors only. Leave blank if no detectors for lane group.

Phase1, **Phase2**, **Phase3**, **Phase4:** Each lane group can have four primary phases associated with them. Enter phases that give a green to this movement in these fields. Enter -1 for an uncontrolled movement.

PermPhase1, **PermPhase2**, **PermPhase3**, **PermPhase4:** Each lane group can have four permitted phases associated with them. With permitted phases, left turns must yield to oncoming traffic, right turns must yield to pedestrians.

DetectPhase1, DetectPhase2, DetectPhase3, DetectPhase4: Each lane group can have four detector phases associated with them. The detectors for this lane group place a call to the DetectPhase.

Controller: If this intersection uses another intersection's controller, enter the other intersectiond ID# in all fields. For example intersections #6 and #7 makeup a diamond interchange with the controller located at intersection #6. For intersection #7, create a Controller record with all data fields set to 6.

SignControl: This row only applies to unsignalized intersections and indicates the sign control used at the intersection. Codes used are: 0 = free, 1 = stop, 2 = yield.

A Sample Layout Data File in UTDF

Layout Data

```
INTID,INTNAME,TYPE,X,Y,NID,SID,EID,WID,NEID,NWID,SEID,SWID,NNAME,SNAME,E
NAME,WNAME,NENAME,NWNAME,SENAME,SWNAME

1,1: Bend,2,10960,800,,,51,52,,,,,,,NJ St. Route 18,NJ St. Route 18,,,,
```

Field Types of the Layout Data in UTDF

Field	Example Data	Comments
IntID	12	Number identifying intersection
IntName	"1st and Main"	Description of intersection
Type	0	0 for signalized intersection, 1 for external node, 2 for bend, 3 for unsignalized
X	1001540.10	X coordinate of intersection in feet or meters
Y	104986.60	Y coordinate of intersection in feet or meters
NID	14	ID of intersection or node to north
SID	8	ID of intersection or node to south
EID	11	ID of intersection or node to east
WID	13	ID of intersection or node to west
NEID	0	ID of intersection or node to northeast
NWID	0	D of intersection or node to northwest
SEID	0	ID of intersection or node to southeast
SWID	0	ID of intersection or node to southwest
NName	"Main St"	Name of street to north

SName	"Main St"	Name of street to south
EName	"1st Ave"	Name of street to east
WName	"1st Ave"	Name of street to west
NEName		Name of street to northeast
NWName		Name of street to northwest
SEName		Name of street to southeast
SWName		Name of street to southwest

A Sample Phasing Data File in UTDF

```
Phasing Data

RECORDNAME,INTID,D1,D2,D3,D4,D5,D6,D7,D8

BRP,15,111,112,211,212,113,114,213,214

MinGreen,15,4,4,,,,,

MaxGreen,15,69,14,,,,,

VehExt,15,3,3,,,,,

TimeBeforeReduce,15,0,0,,,,,

TimeToReduce,15,0,0,,,,,
```

```
MinGap,15,3,3,,,,,

Yellow,15,4.5,3.5,,,,,

AllRed,15,0.5,0.5,,,,,

Recall,15,0,1,,,,,

Walk,15,5,5,,,,,

DontWalk,15,11,9,,,,,

PedCalls,15,0,0,,,,,

MinSplit,15,20,17,,,,,
```

Field Types of Phasing Data in UTDF :

Field	Example Data	Comments
IntID	12	Number identifying intersection
RecordName	MinGreen	Name of data to follow
D1	8	Data for phase 1 of type named in RecordName
D2	8	Data for phase 2 of type named in RecordName
?/td>		
D32	5	Data for phase 32 of type named in RecordName

Records Type of Phasing Data in UTDF :

Initial Interval Records

MinGreen: Minimum green time (tenths of seconds)

MaxInitial: Maximum initial green time (tenths of seconds)

AddedMin: Added minimum time for each actuation, this record is not to be used if **MaxActuation** is used (tenths of seconds).

MinActuation: Number of actuations before **AddedMin** is applied. This is the number of actuations served by **MinGreen.** This record is not to be used if **MaxActuation** is used (number of actuations).

MaxActuation: This field is used for Computed Initial Interval. The initial interval is calculated as actuation * MaxInitial / MaxActuation. Subject to **MinGreen** and **MaxInitial**. This record is not used when **AddedMin** or **MinActuation** is used (number of actuations).

Synchro [4] does not read or write these records except MinGreen; they are for the use of others only. Synchro [4] assumes that the detectors and initial interval will be setup so that the initial interval will be able to accommodate any vehicles waiting after the last extension detector.

Gap Reduction Records

MaxGreen: This is the maximum time the phase is green. This record is not used if MaxExtension is used (seconds).

MinSplit: Minimum Split is used by Synchro [4] during split optimization. This value does map to a value found in traffic signal controllers.

MaxExtension: This is the maximum green time used after the initial interval. This value is used by some 170 controllers rather than MaxGreen. This record is not used if MaxGreen is used. (seconds)

VehExt: This is the time the signal is held green by each actuation. It is also the maximum gap when using a volume density controller (hundredths of seconds).

TimeBeforeReduce: This is the time before gap reduction starts on volume density controllers (seconds).

ReduceBy: This is the amount of time the gap is reduced for every ReduceEvery interval. If ReduceEvery is omitted, then this is the amount reduced for every second. This value is given in hundredths of seconds. This record is not used if **MinGap** and **TimeToReduce** is used (hundredths of seconds).

ReduceEvery: This is the interval that ReduceBy is applied. This value is given in seconds. This record is not used if **MinGap** and **TimeToReduce** is used (seconds).

TimeToReduce: This is the amount of time to reduce the gap from **VehExt** to **MinGap.** This record is not used if **ReduceBy** and **ReduceEvery** is used (seconds).

MinGap: This is the minimum gap to achieve in hundredths of seconds (hundredths of seconds).

Synchro [4] does not read or write **ReduceBy** and **ReduceEvery,** use **TimeToReduce** and **MinGap** instead. Synchro [4] does not support **MaxExtension**, use **MaxGreen** instead.

Phase Option Records

Yellow: This is the time each phase displays yellow (tenths of seconds). **AllRed** + **Yellow** must equal a whole number of seconds.

AllRed: This is the time each phase displays all-red clearance before the next phase (tenths of seconds). **AllRed** + **Yellow** no longer need to equal a whole number of seconds for Synchro [4].

RedRevert: This is the time each phase displays red before it can display green again. This is used for when a conflicting call is cleared during yellow or this phase receives a preemption during yellow (tenths of seconds).

RedLock: This field is "Y" if actuations occurring during red are remembered.

YellowLock: This field is "Y" if actuations occurring during yellow are remembered.

DoubleEntry: When this field is "Y", the phase is active when a compatible phase is active in another ring and there is no other compatible demand in this ring.

SimGapout: When this field is "Y", the phase(s) require both rings to gap out before crossing a barrier.

Recall: This field can have the following values, 0 = no recall, 1 = minimum recall, 2 = pedestrian recall, 3 = maximum recall, 4 = rest in walk. If recall is not zero, the phase

will be serviced on every cycle for the miminum green time, walk+dont walk time, or maximum green time respectively.

If only one (or two simultaneous phases) are set to recall, the signal will rest on those phases. If these phases are set to RestInWalk, the signal will rest on walk in these phases. Note that with some 170 controllers, if Rest in walk is used, the offsets are referenced to the beginning of Dont Walk.

Pedestrian Options

Walk: This is the time for the Walk indication or 0 for no pedestrian phase (seconds).

DontWalk: This is the time for the Flashing Don't Walk interval (seconds). Note that Flashing dont walk ends at the start of yellow time.

PedCalls: This is the number of pedestrian calls per hour received by this phase. Set this field to blank for no pedestrian phase. Set this field to zero or other number to activate a pedestrian phase. If **PedCalls** is not used, a 0 or blank in **DontWalk** can be used to turn off the pedestrian phase.

A Sample File of Timing Data in UTDF:

Timing Plans

```
PLANID,INTID,S1,S2,S3,S4,S5,S6,S7,S8,CL,OFF,LD,REF,CLR
DEFAULT,15,74,18,,,,,,92,31,1,1,
```

Field Values of Timing Data in UTDF

Field	Example Data	Comments
INTID	12	Number identifying intersection
PLANID	PM1	Number or name of timing plan. (for example PM1)
CL	100	Cycle Length in seconds
S1	15	Split for phase 1in seconds
S2	15	Split for phase 2 in seconds
?		
S32	0	Split for phase 32 in seconds (Normally only split1 to Split 8 are used)
OFF	23	Offset in seconds
LD	1357	Indicates leading phases, "1357" normally or "2457" if phases 1 and 3 lag.
REF	26	Coordinated phase(s) that offsets are referenced to.
CLR	0	Yellow or Don't Walk clearance time used with 170 offsets.

A Sample file of Volume Data in UTDF:

```
Turning Movement Count

60 Minute Counts

DATE,TIME,INTID,NBL,NBT,NBR,NBR2,SBL,SBT,SBR,SBR2,EBL,EBT,EBR,WBL2,WBL,W
BT,WBR,NEL2,NEL,NET,NER,NWL,NWT,NWR,SEL,SET,SER,SWL,SWT,SWR

3/24/2001,1700,15,50,50,50,,50,50,50,,,2031,,,,2031,,,,,,,,,,,,
```

Field Types of Volume Data in UTDF

Field	Example Data	Comments
DATE	12/31/96	Date of data collection, (or future projection)
TIME	0700	Time of day, 24 hour format
COMMENT	"Hand Count"	Comments about source of data
INTID	12	Number identifying intersection
xL	123	Left turn volume
xT	903	Through volume
xR	14	Right volume
xU	5	U-Turn Volume, this column can be omitted.
xL2	5	Hard left volume, for intersections with 5 or more legs

xR2	12	Hard right volume, for intersections with 5 or more legs
XPD	44	Pedestrian count, for crosswalk to right of leg.
XBS	35	Count for bus or carpool lane.
XLO	50	Occupancy time for left lane, percent of time detector is occupied%
XTO, xRO, xR2O, xUO, xL2O, xPDO, xBSO	30	Occupancy time for other lane groups. %

x = prefix of approach. Valid approach prefixes are:

NB	Northbound
SB	Southbound
EB	Eastbound
WB	Westbound
NE	Northeast-bound (diagonal)
NW	Northwest-bound (diagonal)
NE	Southeast-bound (diagonal)
NW	Southwest-bound (diagonal)

(*All data files taken from Synchro [4] 5.0 User Guide, developed by Trafficware Corporation, 1009B Solano Ave., Albany, CA.)

Main VB Program

- Form: frmassgn
  Main form of the program
  User selects from different options

Private Sub Command3_Click()
Exports the database data into .CSV files.

Private Sub form_load()
Establishes MGSM Connection as per ODBC connectivity, Loads the GIS map.

Private Sub EventControl1_Click(ByVal MapviewDispatch As Object, ByVal Button As Long, ByVal Key As Long, ByVal WindowX As Double, ByVal WindowY As Double, ByVal WindowZ As Double, ByVal worldX As Double, ByVal worldY As Double, ByVal worldZ As Double)

Intersections are overlaid on the map using dynamic segmentation. This sub routine, listens to the user's action event. When the user clicks, it uploads the main form of the program according to the Begin_Marker, End_Marker, and sri, sld_name etc. properties of that selected intersection.

Public Sub Command1_Click()
Unload all the forms from the memory.

Private Sub Command2_Click()
As per the option button select, it uploads the form, which user has selected.

Private Sub Form_Unload(Cancel As Integer)
Unloads the form.

- Form: Form1
  Phasing Data
  User views the data, updates or quits the form.

Displays phasing data of the selected intersection.

Private Sub Command1_Click()
Cancels the form, unloads the form. It does not affect the database.

Private Sub Command2_Click()
Updates the database, as per the changes made by user.

Sub form_load()
Connects to the database, forms queries that depends on the intersection, generates record sets and loads the form showing relevant data from the database. The form looks similar to Synchro [4].

Private Sub Form_Unload(Cancel As Integer)
Unloads the form. Closes the record sets and disconnects the connection to the database.

Private Sub MSFlexGrid1_KeyPress(KeyAscii As Integer)
This event allows user to update the cell of the table, while the data are loaded.

- Form: Form2
  Volume Data
  User views the data, updates or quits the form.

Displays volume data of the selected intersection

Private Sub Command1_Click()
Private Sub Command2_Click()
Sub form_load()
Private Sub Form_Unload(Cancel As Integer)
Private Sub MSFlexGrid1_KeyPress(KeyAscii As Integer)

- Form: Form3
  Timing Data
  User views the data, updates or quits the form.

Displays timing data of the selected intersection

Private Sub Command1_Click()
Private Sub Command2_Click()
Sub form_load()
Private Sub Form_Unload(Cancel As Integer)
Private Sub MSFlexGrid1_KeyPress(KeyAscii As Integer)

- Form: Form4
  Lane Data
  User views the data, updates or quits the form.

Displays Lane data of the selected intersection

Private Sub Command1_Click()
Private Sub Command2_Click()
Sub form_load()
Private Sub Form_Unload(Cancel As Integer)
Private Sub MSFlexGrid1_KeyPress(KeyAscii As Integer)

- Form: Form5
  Layout Data
  User views the data, updates or quits the form.

Displays Layout data of the selected intersection

Private Sub Command3_Click()
Updates the database, as per the changes made by user.

Private Sub Command4_Click()
Cancels the form, unloads the form. It does not affect the database.
Sub form_load()
Private Sub Form_Unload(Cancel As Integer)

APPENDIX III

Trouble shooting with installing the interface of Geomedia Professional and Synchro

- **Software to be installed** :

Geomedia Professional 4.0

Microsoft Access 97

Microsoft Visual Basic 5.0

Synchro

MGE

This application works with the combination of all of these versions.

- **Required Files:**

Myproj1.vbp (Main VB program)

Abcd.dll (Required for registering Geomedia Command)

Abcd.ini (Required for registering Geomedia Command)

Project5.vbp (Customized VB program for creating the databases for each intersections)

Rt18est.sy6 (Synchro [4] file for route 18)

MGE_GIS_Export.mdb

Rutgers.mge

Nj18.gws (Geoworkspace for the application)

- **Write UTDF Files:**

Write Synchro [4] data (i.e. Volume Data, Timing Data, Phasing Data, Lane Data, Layout Data) for all intersections at a time in UTDF (Universal Traffic Data Format) one by one.

- **Create Databases for each intersection and exporting UTDF data to them:**

To create database for each intersection of the route and export UTDF data into the respective database needs a customized Visual Basic program: project5.vbp

(User must check the directory path of the input UTDF files and the creation of the database)

- **Files to be needed to run the application:**

Databases: MGE_GIS_Export.mdb (Tables: points, highways)

MGE file: Rutgers.mge

Parameter File: njdot.prm

Coordinate File: intersections.crd

Seed Design File: seed83.dgn

- **ODBC connections:**

Required connecting Geomedia Professional [3][12] with Access Database

Control Panel -> select Data Sources

User DSN, add Microsoft Access Driver.

Select Data source as MGE_GIS_Export.mdb database.

- **Locating Event Control Button on VB form:**

If unable to locate Event Control command on the main VB program, then we have to register event.ocx command. For that,

From Start -> Run ->

Regsvr32 event.ocx

Create .dll file for the VB project. Name it as abcd.dll and it will create abcd.ini

- **Setting up Environment Variable:**

Control panel -> systems -> Environment

Set MGSM_Dir command name and Path is the path of MGSMProject .

Eg. In my computer it is F:\MGSMProject

- **Installing Geomedia Command:**

Copy abcd.dll and abcd.ini in Geomedia Professional [3][12]'s project folder. On a command prompt, go to the path of \Geomedia Profession\Programs

Install Geomedia's command,

Installusrcmd \prod "Geomedia Professional [3][12]" abcd.dll abcd.ini

- **Activating Geomedia Command:**

When, nj18.gws is loaded, activate the registered Geomedia command named "Displays a Link Map".

When u close the Main VB form for the particular intersection and want to load the main form of other intersection, we must have to reactivate the Geomedia command "Displays a Link Map".

References:

[1] "MGE Getting Started", Intergraph Corporation, Huntsville, Alabama (1999).

[2] "MGE Segment Manager", Intergraph Corporation, Huntsville, Alabama (1999).

[3] Geomedia Professional, ver 04.00.22.12, online help, Intergraph Corporation, Huntsville, Alabama (1996).

[4] David Husch, John Albeck, "Synchro 5.0 User Guide", Trafficware Corporation, 1009B Solano Ave., Albany, CA (1993).

[5] Trueblood Michael, "Should I use CORSIM or Sim Traffic?" , HDR Engineering, Omaha, NE.

[6] Michael Halvorson, "Microsoft Visual Basic 6.0 Professional", Microsoft Press, Redmond, WA (1998).

[7] Microsoft Corporation, Microsoft Developer Network, http://msdn.microsoft.com

[8] Craig Larman, "Applying UML and patterns: An Introduction to object oriented analysis and design and the unified process", 2nd edition, Prentice Hall, Upper Saddle River, NJ (2002).

[9] Ian Somerville, "Software Engineering", 5th edition, Addison Wesley, Harlow, UK (1996).

[10] Pressman, R., "Software Engineering: A Practitioner's Approach", 4th edition, Mc. Graw- Hill, New York (1997).

[11] Rational Rose Help, v2002, Release 2 for Windows, Rational Software, IBM Corporation, Armonk, NY.

[12] Geomedia Professional Manual, version 4, Intergraph Corporation, Huntsville, Alabama (1996).

[13] Geomedia Webmap Manual, version 5.1, Intergraph Corporation, Huntsville, Alabma (2002).

[14] Georgia Institute of Technology, CORSIM study, http://traffic.ce.gatech.edu/CORSIM/,

[15] Trafficware Corporation, “How do Synchro delays compare to those in other models?”, www.trafficware.com