2010-01-01

Modeling Geographic Awareness of Road Networks for Consistency Verification

Ari Kassin
University of Texas at El Paso, ari.kassin@gmail.com

MODELING GEOGRAPHIC AWARENESS OF ROAD NETWORKS

FOR CONSISTENCY VERIFICATION

ARI KASSIN FUENTES

Department of Computer Science

APPROVED:

Rodrigo Romero, Chair, Ph.D.

Paulo Pinheiro da Silva, Ph.D.

Raed K. Aldouri, Ph.D.

Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

to my

FAMILY

with love

MODELING GEOGRAPHIC AWARENESS OF ROAD NETWORKS FOR CONSISTENCY VERIFICATION

by

ARI KASSIN FUENTES

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2010

Acknowledgements

I want to thank my advisor Dr. Rodrigo Romero from the Department of Computer Science and the Cyber-ShARE Center of The University of Texas at El Paso, for his complete faith and support in me. I met him while taking Advanced Computer Architecture with him and was drawn by his passion to teach and ability to explain complicated topics. He has showed me how to do research properly, from looking for topics of interest, problem solving techniques and writing style, to how to persevere and the meaning of the word Mentor. Throughout my experience in Grad School I faced many obstacles of different kinds but his encouragement and advises were the constant that made me work hard in order to beat each obstacle. I always felt that he was on my side and that with his approval I could accomplish any goal.

I also want to thank the other members of my committee, Dr. Paulo Pinheiro da Silva of the Computer Science Department and Dr. Raed K. Aldouri of the Regional Geospatial Service Center (RGSC), both at The University of Texas at El Paso. Their comments and additional guidance were vital to my research. I must mention that my position as a Research Assistant with Dr. Aldouri gave me the necessary tools and knowledge to complete my work because I was able to join together two disciplines I love, Computer Science and Geographic Information Systems. During my time in the RGSC, Dr. Aldouri's patience and support gave me the opportunity to continue and finish my Master's program, for which I am deeply grateful.

Additionally, I want to thank both The University of Texas at El Paso Computer Science Department professors and the Regional Geospatial Service Center staff and fellow students there. I spent so many good moments and experiences in those buildings that it's impossible to thank everyone here however, my gratitude and respect will always go to both departments. The following individuals played an important role in my formation as a researcher:

Dr. Olac Fuentes

> He gave me the opportunity to be his Teaching Assistant and have a taste of what it takes to teach at a College level. My interaction with other students as their TA left me with immense joy and Dr. Fuentes' guidance was invaluable. Also, as a Grad student I took my first class with him and although he might not know this, his influence was important in the start of my career at UTEP.

I want to mention my friends for listening and being there to support me, specially Luis Enrique Morales whose philosophy on life has helped me many times. His friendship and passion for sports have turned a lot of bad times into great times.

And finally, I must thank my family. My parents and my sister are the foundation of my life, they have given me values that will stay with me forever and that played a key role in completing this Thesis. All the love and support I got from them is beyond measure and I can only pay them back by loving them and making them proud. I should also thank the rest of my family which grows bigger each year, I am eager to see the younger members of my family take on quests of their own. To all of you, thank you.

This material is based upon work supported in part by the National Science Foundation (NSF) under CREST Grant No. HRD-0734825 and Grant No. 0923442 and by the Department of Homeland Security (DHS) under Grant No. 2008-ST-062-000007. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or DHS.

NOTE: This thesis was submitted to my Supervising Committee on November 29, 2010.

Abstract

Problems related to transportation and inspection of valuable or sensitive assets such as commercial products and materials, cultural items and works of art, and hazardous materials share similarities and can be modeled by a core set of abstract entities including a payload, a vehicle, a driver, and an inspector. To make the load-handling capabilities of security monitoring and inspection systems more scalable, security can be increased by reducing the variability of transportation routes to a finite set of authorized routes between trip origin and destination locations. Then trip anomalies, which are unexpected trip variations, can be used in inspection targeting criteria. In addition, the effectiveness of the inspection sampling rate can be increased by always inspecting trips with unjustifiable anomalies and fulfilling the remainder of the sampling rate with other trips. While some anomalies can be automatically detected by vehicle tracking systems, other anomalies can be detected by inspectors based on information provided by drivers. This creates a need for an independent objective verification of the quality of the information provided by the driver, as this information can be used to detect transportation anomalies and, in some cases, justify the anomalies to avoid costly inspections. This thesis proposes the use of a geographic awareness modeling (GAM) system to determine whether route-following anomalies of monitored vehicles can be justified by driver detection of adverse road conditions. While current GIS applications provide geographically accurate representations of a road network, they do not provide mechanisms or information to corroborate or contradict scenarios describing trip conditions along a given route. This thesis also presents a design and one implementation of a GAM system which can provide independent, reliable verification of route-related information provided by vehicle drivers. A benefit of implementing such a system would be to complement the expertise and the experience of inspectors to increase the effectiveness of inspection targeting policies and sampling rates.

Table of Contents

List of Figures

Chapter 1

Introduction

Logistic monitoring environments in which vehicles must follow preassigned routes due to concerns such as scheduling control, fuel efficiency, environmental protection, and vehicle or cargo security, must detect and deal with route deviations, as they may represent increased operational costs or risks. In addition, cases in which deviations can be justified may be handled differently from the way cases without a justification are handled. In the former case, driver road awareness modeling can be used to determine if route deviations of monitored vehicles are justified by driver detection of adverse road conditions. In this case, road conditions of interest would be detected by the driver of the monitored vehicle in a route region ahead of the monitored vehicle current location. For instance, conditions such as traffic accidents, traffic congestions, road blockage due to maintenance, and weather-affected road segments could be justifications for route deviations, provided that the vehicle driver can see such conditions in time to divert from an authorized to another authorized route.

This thesis presents a geographic awareness modeling (GAM) system which can analyze the effects on a road network of events related to weather conditions, road traffic conditions, and other conditions on a trip instance in order to determine if a route deviation may be justified. While current GIS applications provide geographically accurate representations of the road network, they do not provide mechanisms or information to verify the consistency of trip descriptions along a given route. In addition, other sources of information, such as weather reports and road traffic reports, only provide aggregate data which identifies affected zones, but such sources do analyze the effect that such zones can have on a specific trip instance description. The GAM system is based on a stratified representation model

where the base stratum is provided by standard GIS information and the upper strata provide ancillary information that helps to create a complete analysis context for a trip instance.

The GAM system models a monitoring environment by using trip descriptions as independent analysis targets. A trip description, which may include information about the vehicle, the driver, the cargo, the source, the destination, and so on, is instantiated in the system with default or preassigned information for all of the instance attributes. Once the vehicle arrives at a trip inspection point, information collected while the vehicle is in route is compared to information captured during trip instantiation to assess the risk or threat level associated with the trip, i.e., actual data is compared to expected data. The architecture of the GAM system is based on stratified semantics where every stratum models and makes available information for upper strata. Models in a given stratum have independent semantics from those on other strata but they also provide the building blocks for upper strata models. While the lower strata information is modeled through a geographic information system (GIS), information in the upper strata is modeled in a knowledge base. Between the lowest stratum and the highest one, models transition from points in longitude-latitude coordinates to road networks and trip routes and from trip data to trip descriptions that can be analyzed to assess the trip threat level.

The interdisciplinary study of statistics, mathematics and computer science provide the foundation for geographic information science which, merging database and software technology with cartography, has produced geographic information systems. A GIS enables the user to represent spatial data of geographic objects (e.g. mountains and rivers in a region), maps (e.g. political boundaries) and demographic data (e.g. city population and road networks) in the same space while keeping relationships between objects in the data. The user can query and analyze the system to obtain information concerning both regular data and geographic data. For instance, a simple query would be the following: find all the gas stations where gas price is lower than 2.95 dlls per gallon within a radius of 20 miles from a specific geographic location. Query results could appear in a map for the

user to visualize. Applications of GIS include urban planning, archeology, and resource managing and logistics. A problem with GIS is that beginning users face a steep learning curve to learn how to operate such systems, as many GISs require the end user to become familiar with technical details of data representations and formal ways to query data and get reports.

The GAM system is focused on solving a problem which requires the strong spatial analysis included in a typical GIS and using the resulting data to infer potential risks or threats associated with transportation of mission critical assets. For instance, transportation of hazardous materials, cultural assets, unique objects, or expensive merchandize requires close monitoring. A logistic monitoring environment which must track the shipped cargo can rely on precise timing to satisfy its monitoring goals. However, the monitoring system must also take into consideration delays and route deviations in order to account for the impact of allowable events in the road network. For instance, a cargo vehicle driver might be forced to deviate from a preassigned route due to obstacles such as road work or accidents. A monitoring system must detect and deal with route deviations through strategies such as recommending cargo inspections or justifying the deviations to reduce or eliminate the need to perform cargo or vehicle inspections.

Any route deviation should be justified and the validity of the justification provided must be verified. The ability to prove if a driver could see a possible obstruction on the road ahead plays an important role in the justification validation.

This thesis presents a novel approach to solve the logistic monitoring problem described above using a GAM system based on GIS technology, analysis of visibility on a road network through line of sight computations, detection of route deviations and validation of deviation justifications through logical inferences, and visualization of inferences by a proof visualization tool.

This thesis is organized as follows: Chapter 2 presents theoretical concepts used as a basis for the work presented in this thesis. Chapter 3 discusses related research. Chapter 4 presents the solution approach and algorithms introduced by the GAM system. Chap-

ter 5 provides GAM system implementation details and results. And chapter 6 presents conclusions and future work.

Chapter 2

Theoretical Concepts

The work presented in this thesis is based on concepts from geography and computer science. From the former, a Geographical Information System (GIS) enables creation and manipulation of various layers of spatial data and the relationships between them. From the latter, logical foundations are used to assess threats and justify route deviations if possible. In addition, this thesis models geographic awareness using a line of sight (LOS) algorithm within a GIS environment to support reasoning about threats and justifications. The following sections cover basic concepts in detail.

2.1 Geographical Information System

A GIS has various definitions which depend on the areas where a GIS is applied to deal with spatial analysis problems. Dueker's definition [10] centers on the use of a computer-based system with database capabilities as a toolbox with the purpose of capturing, storing, retrieving, and displaying spatial data. However, that definition also implies that a GIS application can be related to storage and manipulation of temporal and social information such as activities and events. GIS linkages among distinct types of data provide powerful means to analyze geographic information.

From a spatial analysis viewpoint, Goodchild[16] states that a GIS should adopt a model consisting of modules for input, storage, output, and analysis. The input module takes care of the digitalization of map documents by setting standards that allow a computer system to correctly display them on a screen. The storage module has built-in data structures to manipulate spatial data and perform operations on it. The output module is concerned

with presenting data and map documents to the user in a legible and useful way. Finally, the analysis module enables the user to perform queries on spatial data to support decision-making tasks.

Since the GIS acronym is also associated with geographical information science, throughout this document geographical information science will be referred to GIScience to reserve GIS for a geographical information systems. GIScience provides the theoretical foundation for building a GIS, i.e., GIScience is concerned with the implementation and application of GIS environments. However, there are other areas of interest of GISience which include spacial perception, representation of spatial concepts, and human psychology effects on spatial reasoning.

In general, a GIS is focused on:

- capturing, converting and storing spatial or geographic data (also referred to as geospatial data) of real-world entities that may lay in, over, or above the Earth
- analyzing geospatial data
- integrating geospatial data with other data types, i.e., textual data representing attributes of real-world entities
- presenting geospatial data as maps, tables, and charts in order to support a decision-making process

A GIS contains a considerable amount of spatial data referenced into a map-base data structure where each classifiable data is called a feature. A map has any number of features in it, e.g., all rivers in the surroundings, interstate highways, or a group of buildings in an area. Features can show topography, roads, parks, urban areas, and many other types of geographic data. A map can be seen as the *where* representation of data. However, the map may not be complete without additional information about the map-base data structure. This additional information along with spatial data enables a GIS to create relationships

between the two data types as shown in figure 2.1. ¡RR Note: Replace fig. 2.1 with your own 2.1 figure. /RR¿

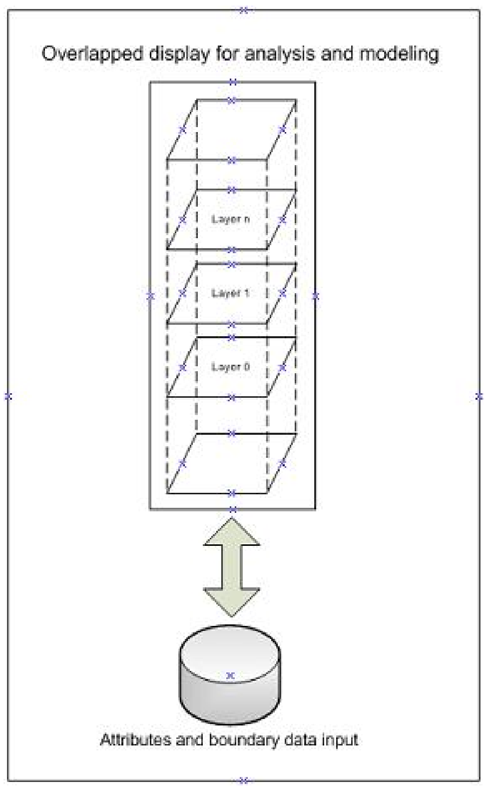

Figure 2.1: Generic design of a GIS with attribute data and digital boundary data.

A GIS supports tasks such as:

- Select and visualize a map or parts of it
- Count, group, reclassify, and quantify features and their patterns on a map
- Measure features, i.e., get the width, distance, height, etc. of objects in the map
- Combine maps by overlaying them and compare features to form new maps
- Slice, interpolate, and analyze surfaces
- Track, predict, and exploit patterns in spatial data

- Locate potential customers, find shortest/fastest paths, and locate businesses

Finally, a GIS enables users to answer questions such as: "Show all the water pipes that have failed more than twice in the last 2 years," "Show the spatial distribution of car ownership compared with bus route locations for these suburbs," and "Show all the contamination within 100 meters of this watercourse."

2.2 Line of Sight Computation

A GIS may support visibility analysis which would help to determine whether target points in a region are visible from specific observer points. Visibility analysis can be performed using a line of sight (LOS) algorithm. Several LOS algorithms have been developed to directly examine the visibility of one point from another point or to create a visibility viewshed, which is a region visible from an observer point.

The information gathered by a LOS analysis is used in applications such as urban planning, archeology, and military science. An urban planer may use LOS analyses to identify an optimum distribution of buildings for a given urban project. An archeologist can use LOS analysis to identify the potential meaning and functionality of ruins and old buildings. A military unit may greatly benefit from visibility analysis of the terrain and road networks in a battle region.

Visibility can be determined using the elevation values of a topographic surface. There are two different formats for representing a surface: digital elevation models (DEM) and triangulated irregular network (TIN). While a DEM representation of a surface as a raster image is well suited for analysis of the surface slope, a vector-based representation of the physical land that is utilized by the TIN format is preferable for visibility analysis due to the TIN format ability to portray terrain in three dimensions [17].

Lee [21] states that two points on a topographic surface are inter-visible if there is a straight line connecting the points without intersecting any part of the surface except at the two points. In this case, visibility analysis consists in finding the area of visible regions,

R_v, for a given set of viewpoints P_o. To define this problem more precisely, let A_j represent a subregion of the entire region that is under consideration and P_i a point in the region. The visibility from P_i on A_j is defined as

$$V_{ij} = 1 \text{ if } A_j \text{ is visible from the point } P_i \text{ and 0 otherwise.}$$

Moreover, the membership c_i of the point P_i in the set V_p is defined as

$$c_i = 1 \text{ if } P_i \text{ belongs to } V_p \text{ and 0 otherwise.}$$

The goal of visibility analysis is to find all subregions A_j that are visible from a point in the set V_p:

$$\bigcup_{j \in R_v} A_j \tag{2.1}$$

where

$$R_v = \left\{ j | \sum_i V_{ij} c_i \geq 1 \right\}, \tag{2.2}$$

which implies that a subregion A_j belongs to the set of visible subregions if and only if both of the following conditions are satisfied:

- $V_{ij} = 1$ (the subregion A_j is visible from the point P_i) and
- $c_i = 1$ (the point P_i belongs to the set P_o).

In the algorithm above, each point of the surface should be considered individually to determine its visibility from the observer point(s), which might result in a large computational complexity. To reduce the number of calculations, an improved LOS algorithm was presented in [24]. The main idea of this algorithm is to consider elevation of points and the geometric relationships between the slopes defined by the start and the end points. The line of sight is determined by taking and comparing the elevation values of two points, P_a and P_b. If the points form an uphill slope (i.e., $P_a < P_b$), the points are recorded, starting at P_a until we find the point P_c such that P_c initiates a downhill slope (i.e., $P_c > P_d$ for some point P_d that immediately follows). When a downhill slope is encountered, the

slope between the points P_a and P_c is used to “draw” a line whose intersection with the surface determines the next visible point. Following this method until reaching the point P_b allows identification of segments along the line of sight that can be discarded and the segments that should be recorded as visible. From previous point computations, segments ahead should not or should be visible, respectively. This algorithm improves the efficiency of LOS calculation by drastically reducing the number of calculations, but it also reduces the precision of results due to approximations of points at which the visible segments start and end.

To allow for precise LOS computation while saving computational resources, a LOS algorithm which utilizes the characteristics of the R3-Tree data structure was designed [34] to allow early filtering of spatial areas without needing an extensive LOS analysis. The algorithm has a multiple step architecture in which the first step is the filtering phase to quickly discard areas that are surely not visible from the observer point. Subsequent steps are refinement phases which closely compute LOS visibility in an area that is a candidate to be visible from the observer point. As in the previous algorithms, the main criterion for visibility analysis is the elevation of the terrain.

LOS algorithms have been modified to suit applications which require visibility analysis such as urban simulations [37], combat activities [36], and autonomous vehicle movement [26]. To allow for real-time rendering in urban simulations, an online occlusion culling system was created [37]. While visibility computations can be parallelized to shorten rendering time, dynamic changes to the environment can not be handled by this algorithm.

Tuft presented a method for computing LOS visibility in simulations with moving entities [36]. Using region-based visibility, the environment can be divided into smaller areas allowing region to region comparisons to determine if a closer LOS operation is needed. Discarding unnecessary LOS operations speeds up the visibility analysis, but variations in object positions might represent a problem in such approach because the dynamic nature of the environment may reduce its visibility efficiency.

Terrain elevation is the starting point for the LOS implemented in ESRI’s software [14],

which is mainly used as a part of a commercial software for GIS analysis. This software uses slope angle comparisons between two points to compute visibility. The result is presented as a Boolean figure within a polynomial line of alternating visible and invisible segments. Obstruction points are also identified on the terrain. Besides the terrain elevation, this algorithm uses criteria such as Earth curvature and light refraction to aid in determining the visibility region. Finally, ESRI's ArcMap software enables the user to perform spatial queries about visibility that are handled by the ESRI's GIS environment.

Chapter 3

Related Work

The focus of this thesis is on geographic awareness modeling which can be used in a logistics monitoring system to detect and attempt to justify route deviations within a road network. While GIS technology is utilized to perform visibility computations, an inference engine is utilized to detect, verify, and attempt to justify each route deviation instance. A proof visualizer can be used to learn which logic rules and inferences were made to obtain the final threat assessment result. A line of sight algorithm implemented within a GIS environment is used to perform visibility analysis over a road network to determine what onward parts of predefined transportation routes can be seen from previous points within a given route. Within each route, visibility information can be used to check whether a driver's justification for a route deviation can be validated. Precomputed visibility is used to speed up analysis of a transportation trip instance. Rules and assumptions are modeled with a logical programming language to use its inference engine for analyzing visibility of problem scenarios. Proofs are output to allow analysis of facts and rule applications to help advanced users to understand and audit inference results. The following sections describe work related to this thesis based on the problem being addressed, on solution approaches used, or on related applications of tools and environments used for the GAM system.

3.1 Related Applications of GIS

A GIS has been used as the key component to find solutions for goods and materials transportation problems. For instance, the transportation of hazardous materials (hazmat) within an urban area is a very complex problem because a disaster involving hazmat can

pose a significant threat to the population in the affected area and because such a disaster could also cause costly damages to the urban infrastructure, i.e., buildings, highways, parks, etc. To analyze and solve hazmat transportation problems, it is necessary to create systems to asses the risks of transporting hazmat in a given area, simulate the damage caused by a disaster in the area, monitor the transportation operations, assist in an emergency response situation, and select routes that can prevent a hazmat disaster. As an example, the work of[23] demonstrates the use of a GIS to simulate a plume disaster by gathering and using real-time data. Such simulations display affected areas in a map and predict the extent of the problem. Another example is presented by [2] which shows how network problems such as finding the shortest path and route planning can be modeled in a GIS. These techniques have been used together for planning, response, recovery, and disaster mitigation related to hazmat transportation.

Zhang [38] explores using GIS with risk assessment models to select routes which would have the smallest consequences in a hazmat accident. While other risk assessment techniques model disastrous events through probability distributions or wind dispersion of airborne material as in the Gaussian Plume Model (GPM), the authors proposed placing all available routes in a network which can be represented in a GIS environment. GIS can use rasters which have the advantage of discretizing the geographical space in order to reduce the computations needed by complex techniques such as GPM.

Huang [20] presents a solution for hazmat transportation problems based on proper route planning using a GIS and genetic algorithms (GA). Criteria for defining the characteristics of a route include the following: 1. Exposure - the population that is exposed in a hazmat disaster. 2. Socioeconomic impact - infrastructure costs. 3. Risks of hijacking - a route in an unpopulated area represents a greater opportunity for a transport to get hijacked. 4. Traffic conditions - traffic speed and flow affect travel time. And finally, 5. Emergency response - the capabilities of the emergency response teams to contain a hazmat disaster. The following expression for cost uses GA to compute the weights that minimize the cost of R:

$$R = \left(\sum_{c=1}^{n_c} \left(w_c \sum_{cf=1}^{n_{cf}} w_{cf} s_{cf} \right) \right)$$

where R =cost; c =criteria; n_c =number of criteria; w_c =weight of criteria c; cf =factor under criteria c; n_{cf} =number of factors under criteria c; w_{cf} =weight of factor f under criteria c; s_{cf} =score of factor f under criteria c.

Govindan [18] utilizes a GIS to monitor a maritime transportation network. A spatial sensor network is created with different hand-held devices which, like personal digital assistants or global positioning systems, which send real-time information to a GIS environment which can process received data and provide a user with actionable information. For instance, the user can deploy emergency response teams to a disaster area or contact law-enforcement agencies in case of a security threat along the maritime network.

3.2 Naive Geography and Geographic Context Representation

Problems like the one introduced in this thesis expose limitations in GIS environments as many rules needed for a solution to the problem can not be modeled through a GIS without expert analysis, representation, and manipulation of the data. One aspect of this analysis refers to the representation of geographical concepts in a manner that humans understand them. One approach to dealing with this situation is offered by naive geography [12], which deals with the knowledge that people have about their geographic environment. *Naive geographic reasoning* joins spatial and temporal computations available in a GIS with intuitive operations to guide the design of GIS applications which are based on human geographic perception. By adding a naive geographic context to GIS applications, work described below models human assumptions to enhance modeling beyond capabilities offered by GIS environments.

This author [35] models prepositions such as *near*, *between* and *in front of* for inclusion in a GIS to provide the context in which humans use them. For instance, *near* refers to

proximity between objects and it also conveys an intuitive geographical awareness about user preferences. For example, when a user who happens to be a student in a given university queries a GIS about all the restaurants *near* that university, the user expects to get restaurants close to that university which fellow students favor. However, a user who is a tourist visiting the university, would expect to get restaurants suitable for tourists in the same area. These authors proposed adding criteria such as popularity to GIS search tasks and using a search engine to categorize results as well. Once a GIS produces a set of results, logical rules implemented in Prolog can evaluate the set to rearrange it. This can be implemented using an importance index computed from popularity data in a search engine to look for each result in the set. A similar approach in [11] and by[19] proposes the creation and use of a semantic geospatial web instead of using the results of regular search engines. These authors propose enriching the data found on the Internet with metadata and management rules to create a framework where geographical locations can be modeled and stored with more than just geographic coordinates. For example, a query like *Find data sets which allow me to analyze **Lakes in Maine*** would get every data set related to lakes in Maine showing the results categorized by counties or by how many counties a lake touches. The authors' proposed semantic geospatial web is important and a clear example of context-awareness modeling techniques which improve a GIS to solve complex problems.

3.3 Line of Sight and PML on Geographic Applications

Visibility analysis within a GIS can be useful for applications in civil engineering, landscape engineering, and urban planning. Line-of-sight algorithms (LOS) compute visibility between an observer point and a target point (see additional details about LOS in chapter 2). The work by[34] incorporates LOS into a problem about combat units in military exercises. The authors simulate terrain visibility in a GIS by computing LOS between two

combat positions in order to decide the best course of action, which could be anything from choosing an alternative route to deploying more combat units into the area. Chapter 4 of this thesis includes details about the application of LOS computations to model geographic awareness in the GAM system.

Proof Markup Language (PML) is based on proof theory, which allows it to encode justifications as formal proofs. Each proof is a sequence of rule applications which generate an answer. A proof may include formal logic derivations, natural deduction derivations, information retrieval operations, and natural language processing. For instance, [29] illustrates the use of Probe-It!, a PML application which can be used for explanation visualizations. Additional details about Probe-It! and PML can be found in chapter 2. The next chapters discuss how these tools are utilized in the approach to solve the transportation monitoring problems addressed in this thesis.

3.4 Related Transportation Problems

Problems related to transportation of assets which are valuable because of their commercial value, uniqueness, toxicity, scarcity, strategic importance, and so on, share similarities. This section discusses approaches for solving problems related to transportation. While highlighting their applicability to work presented in this thesis, this section identifies disadvantages of blindly implementing such approaches into the solution model used in this work.

Security problems associated with transportation of cultural assets are presented by[7]. Museums, curators, and collectors, the authors claim, are often concerned about the risks of material loss associated with transporting their invaluable assets. The authors evaluated use of a Geo-Time Authentication system (GTAs)[27] which takes care of identifying the assets, checking their integrity, and monitoring them throughout the transportation process. According to the results presented in [8], however, GTAs produce such an abundant number of false alarms that the authors decided to look for a solution which would help to

identify and dispose of false alarms. They proposed the use of multi-agent systems which are composed of sensors and an artificial intelligence layer that interprets the sensor values and uses reasoning to produce results. For example, temperature sensors are placed in each package being transported and inside the cargo containment in order to detect any temperature change in case of an illegal package opening. However, when a sensor reads a value higher than a predefined security threshold, the system does not throw an automatic alarm. First, it compares this value to the values measured by the rest of the sensors, including those in packages and in the cargo containment, because it is possible that the environmental temperature is just high and the threshold should just be raised. The implementation of the reasoning module is done with the DALI language [5] [6]. This research shows that a monitoring system including just tracking capabilities and sensors to detect changes may be insufficient to deal with false alarms which are prone to happen during the transportation process.

Work in [25] presents a problem in which waste-carrying trucks travel to the Waste Isolation Pilot Plant (WIPP) to deposit their cargo, which may be sensitive or dangerous material such as chemical waste and nuclear material. Thus, real-time monitoring of each truck is essential. Initially, each truck was being monitored only by its on-board shipment-tracking systems. However, the authors realized that monitoring in that fashion was not enough when a driver took a wrong turn at some point and exposed the system's low degree of monitoring control. The new developed system consists of a more advanced tracking system and a reasoner (i.e., an artificial intelligence application) to detect anomalies along a truck trip. Anomalies are defined as a prolonged stoppage, a communication failure between the truck tracking system and the monitoring headquarters, or route deviations. The system operation is composed of four major elements. First, it establishes the locations and initial points of all the trucks enroute to the WIPP site. Second, it verifies that tracking system location signals are being received properly and continuously. Third, it determines whether a truck has stopped or not. And fourth, it checks if the shipment is on the correct route. This approach does not care about the reason for an alarm, as its only goal is to

identify it for someone to elaborate on its cause. Work presented in this thesis, however, presents an approach that goes one step further because it provides support to analyze a detected trip anomaly.

In the work presented by[15] another transportation problem is discussed. The scenario consists of a road network in which overpasses such as bridges, railways, or other forms of pedestrian crossing between highways are common. In this scenario, overpasses are often used to throw hard objects to transport vehicles below or to carry out similar transgressions in which the advantage of being in the overpass is used by the offending individuals. A simple yet insufficient solution is the implementation of a visual surveillance system that can monitor an area in the road network in real time but can not easily distinguish between a suspicious activity of an individual and a regular crossing of a pedestrian using the overpass. The authors proposed a vision-based system that can automatically classify the different behaviors of unknown objects entering a controlled scene. Their idea is to identify a sequence of events that can lead to an alarming situation such as a person throwing stones to vehicles from the overpass. Their system is composed of two main modules. One includes the image processing software and the architecture responsible for identifying objects and situations in camera feeds. The second is an alarm detection system using logical rules that can classify the pictures into different levels of dangerous situations. For instance, the logical rules define a danger area as a space in the overpass that can be used for a suspicious behavior such as the back of a fence in which a person can hide. Additionally, the time, t, is also important because the system analyzes sequential frames from a camera feed. The following are examples of system rules:

$$\forall t \text{ if } [\exists object(t,i) \wedge \exists object(t+1,j)] :$$
$$Distance(object(t,i), object(t+1,j)) < SpaceThreshold$$
$$\text{then } SlowTransitEvent = TRUE$$

where $object$ represents an entity approaching a danger area. The rule states that for every time instance, if there is an object appearing in two consecutive frames, i.e., t and $t+1$,

then the distance between the two adjacent object positions should be computed to see if it is less than a space threshold representing closeness between them and which translates into a minimum speed threshold. If this is true, a slow-transit event should be triggered by the system. In the following rule:

$\forall(t,k)$ if $\exists j : Distance(DangerArea(k), object(t,j) < SpaceThreshold)$
then $LevelOfWarning(k)$ grows
else $LevelOfWarning(k)$ decreases

where k is a danger area, it is stated that for every danger area and all time, the distance between the area and any object must be computed to adjust the warning level associated with the area. If the distance is less than a safe minimum threshold, the level of warning for the area should increase, otherwise it should decrease. This rule establishes a standard level of warning which means that an event might be suspicious. The next rule elevates an area level of warning to the highest value in the system:

$\forall k \; LevelOfWarning(k) > TimeThreshold$
then $CriticalLevelOfWarningEvent = TRUE$

The system has a user interface that displays color-coded warnings and other helpful data whenever it detects suspicious behavior in a sequence of frames taken from the camera feed. A trained system user can take decisions on whether to act upon alarms thrown by the system or not. Using a similar approach for geographic awareness modeling, which is a key aspect of this thesis, would have some drawbacks. First, installing a surveillance system inside each cargo vehicle cabin to capture the road and recognize objects ahead of the vehicle would be an expensive operation as all the vehicles would have to acquire the new hardware. Second, problems addressed by this thesis do not include defining a specific object along the road network to identify it. Instead, the addressed problem requires visibility analysis to determine how far along a route a driver can see in order to get a better understanding of the circumstances surrounding a trip anomaly such as a route

deviation. Thus, an object recognition module is not suitable or necessary in a solution for the awareness modeling requirements in this thesis.

Ideas and approaches included in research reviewed in this chapter provide a reference for the models created for this thesis. Using a GIS with visibility analysis functionality, i.e., a line of sight algorithm, building a Prolog application with logic rules to model geographic awareness, and finally, using inference visualizer to verify the results, the GAM system presented in chapters 4 and 5 was designed and implemented.

Chapter 4

Problem Definition and Approach

Problems related to transportation of valuable or sensitive assets such as commercial products and materials, cultural assets and works of art, and hazardous materials share similarities. For instance, the elements involved in the transportation process can be abstracted as a core set of entities including the following: a payload, which contains all the assets being transported in a single trip; a vehicle, which carries the payload on the road, through the air, or over a body of water; a driver, who is in charge of controlling the vehicle; and an inspector, who is in charge of ensuring that the payload, the vehicle, the driver, and the transportation process are in compliance with a set of applicable rules and regulations. A particular transportation scenario will include one or more connected trips to transport a payload along a number of locations and a variety of vehicles, drivers, inspectors, rules, regulations, monitoring mechanisms, and procedures to deal with compliant and non-compliant trips. This chapter focuses on land-based transportation of assets, but some of the concepts presented can be extrapolated to other types of transportation, including multi-modal trip combinations. To allow discussion of specific details, this chapter utilizes international transportation of assets as the main target scenario.

Ground transportation of assets across an international border is a complex operation where many entities have to interact in order to achieve their independent goals. For instance, to transport goods and materials between the cities of Juárez, Chihuahua, México and El Paso, Texas, U.S.A., every time a shipper sends a loaded vehicle from Juárez to El Paso, an officer at the U.S. port of entry (POE) will stop the vehicle to verify that the vehicle, the driver, and the payload are in compliance with the importation rules and regulations of a number of U.S. organizations including Immigration and Customs Enforcement

(ICE), the Environmental Protection Agency, the Food and Drug Administration, and many others. While documentation for the vehicle and the driver tends to be fairly constant, which allows a quick and reliable verification of compliance, documentation for the payload and the payload itself will vary for every trip. Not only that, other factors will also vary for a given shipper, vehicle, and driver including the route followed from the shipper location to the POE, the time taken to cover the route, the number, place, and length of vehicle stops while en route to the POE, and so on. Thus, in addition to documentation provided, information provided by the driver can be used by the POE ICE officer to determine whether a vehicle and/or payload inspection is necessary. This creates a need for an independent objective verification of the quality of the information provided by the driver, as this information can be used to detect transportation anomalies.

Some of the detected anomalies can flag the payload as potentially dangerous or illegal, which could result in an expensive and slow inspection of the vehicle and the payload. However, since it is not economically or otherwise feasible to perform a one-hundred percent inspection of vehicles and their payloads, it is important to ensure the effectivity of inspection targeting criteria. This provides additional reasons to verify the quality of information obtained by ICE officers from vehicle drivers. The GAM system presented in this thesis, which models the driver's geographic awareness and supports validation of the driver's justifications for route deviations, can provide independent verification for route-related information provided by the vehicle driver.

4.1 Problem Background

Transporting assets across an international border requires careful planning in order to comply with importation requirements that are verified at the ports of entry. Since commercial traffic has been used for trafficking of illegal substances, goods, and people, the work of inspectors at international ports of entry can be supported by remote monitoring systems which can help in the detection of transportation anomalies. To make the load-handling

capabilities of such monitoring systems more scalable, the variability of transportation trips can be reduced by defining a finite set of authorized pre-established routes between shippers and ports of entry. Trips which must follow such routes can be flagged for anomalies faster and more reliably in an automated way. For instance, if a driver must follow a preassigned route, abandoning the route without justification can be an undisputable reason for inspection. By the same token, taking longer than a predetermined travel time of a route, including some tolerance for varying road conditions, can also be a reason for inspection. The GAM system presented in this thesis addresses detection and justification of transportation anomalies caused by route deviations. By definition, a route deviation is an anomaly in which a trip did not follow its preassigned route entirely. Notice that all authorized pre-established routes for a shipper start at the same origin point and they may share road segments and a U.S. POE, but these two latter points may not be true for all sets of pre-established routes. When a trip fails to follow its assigned route, some segments of the trip actual route are not included in the assigned route, the system should raise and record a route deviation anomaly.

Monitoring vehicle movements becomes a very important task because it allows determination of whether authorized pre-established routes were followed consistently throughout a trip. A GIS in combination with a Global Positioning System (GPS) device, as discussed in this thesis, is the perfect tool for such a monitoring task. When a vehicle arrives at a POE, an ICE officer can verify whether trip data indicates that the driver followed the assigned route. Upon finding an anomaly through comparison of the actual trip data against the assigned route data, the officer can ask the driver for information which may justify the change of plans or indicate a need for an inspection. As mentioned in the previous section, validation of the driver justification is critical because false justifications can be used to prevent detection of illegal trafficking and valid justifications can avoid costly detailed vehicle and cargo inspections.

A system which models the driver's geographic awareness, such as the GAM system, can assist the officer to distinguish between valid and invalid justifications and make more

effective inspection targeting decisions, i.e., by deciding to inspect those trips where invalid driver justifications are used as a reason for route deviations and by only sampling those trips with valid justifications or without anomalies. This thesis presents a novel approach to support the POE officer in getting more accurate results at the time of detecting route anomalies and classifying anomaly justifications. The next section includes an example where it is difficult to distinguish between a valid and an invalid justification for a route deviation.

4.2 Problem Scenarios

To show the difficulty in classifying a driver justification for a route anomaly as valid or invalid, the following different scenarios are presented and explanations of the interactions between the driver and the POE officer are included.

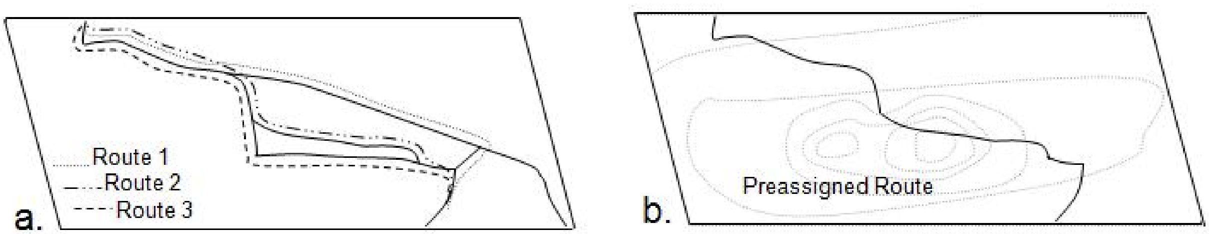

Figure 4.1: a. Authorized routes b. Preassigned route

- Scenario 1: Figure 4.1 illustrates a 2D map with the authorized trip routes and a preassigned trip route. In this scenario, a vehicle is traveling along the preassigned route (from the bottom to the top of the figure) when suddenly the driver spots an obstacle up ahead. In addition, the driver is able to deviate from the preassigned route because there is an intersection of authorized routes between the vehicle and the obstacle locations. The driver decides to avoid the obstacle by deviating to a secondary authorized route. However, this act will be recorded by the GIS monitoring system as an anomaly. When the driver arrives at the POE, an officer will definitely

ask the driver for an explanation for the deviation. Under this scenario, the driver has a valid justification for the raised trip anomaly and the vehicle may be granted a green light to continue the trip into the U.S. without an inspection. The driver's ability to see the obstacle is key to validate the justification for the route deviation.

- Scenario 2: Using the same authorized routes and preassigned route as in scenario 1 and figure 4.1, the driver decides to abandon the preassigned route and follow another route for an arbitrary reason. Because this act will raise an anomaly at the time of inspection, it is possible for the driver to lie about the reason for the deviation. The driver can claim a situation like the one described at scenario 1 and expect the officer to give the trip a green light into the U.S. even if such situation never happened.

In both scenarios, verifying the driver's justification is not straightforward because there is no exact way of validating whether the driver saw an obstacle or there is some other cause for the deviation. For this reason, many ideas have been proposed to protect cargo vehicles from any illegal access, e.g., putting certified seals in the vehicle's container door. The objective behind such protective measures is to lessen the importance of route deviations by focusing on protecting the vehicle's payload. In this case, an ICE officer only needs to check the integrity of the door certified seals to ensure that the payload has not been compromised or lost during the trip. However, the container is not the only place where illegal cargo can be transported. Thus, a way to monitor the route followed by the vehicle is still important despite the use of certified door seals.

4.3 Problem Set Up

The two scenarios above justify the need for an efficient, reliable way to validate the driver's justification for route deviations. This thesis explores the use of visibility analysis based on a line of sight algorithm to create a model that enables the POE officer to make better informed decisions about anomaly justifications and vehicle and payload inspection targets.

By performing a visibility analysis along all the authorized routes, which is the set of all routes that can be preassigned for vehicle trips, visibility between all pairs of points within each route can be determined. For example, considering figure 4.2 and assuming that visibility analysis can be performed, is it possible to prove whether a route obstacle in point P2 is visible from the vehicle driver's position at point P1? The GAM system utilizes a logic component to perform the necessary queries and computations to arrive at a result which an inspection officer can use to conclude whether a driver's justification is valid or not.

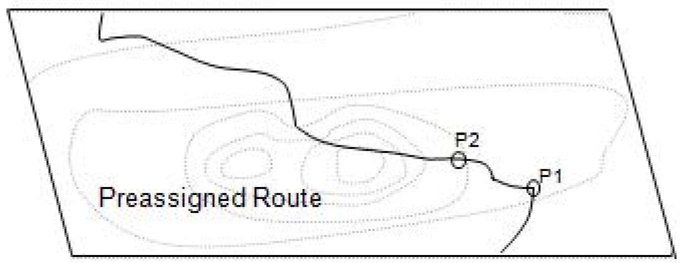

Figure 4.2: Two points along the preassigned route.

4.3.1 Formal Definitions

To support logical processing of scenarios to corroborate or reject deviation justifications, several scenario elements must be formally represented. A transportation vehicle of any type, carrying any type of payload, is represented as $V_0, \ldots, V_n$. In the general case, a vehicle can be an airplain, a ship, a train, or any other means of transportation which can transport a payload. In this thesis, however, the focus is on ground transportation of assets. Thus, a vehicle will most likely represent a loaded truck moving along a road network.

Routes are defined as sets of elements of a road network, including highways, boulevards, avenues, drives, and so on, which are available for a vehicle V_0 to move from a starting point to a destination point. Also, a vehicle V_0 will have one preassigned route R_p, which has been defined as the required route to complete a trip from the starting point to the destination

point. Additionally, there exist more routes that can lead to the destination point called secondary routes $R_{s1}, \ldots, R_{sn}$, these routes are available for the trip planner and vehicle driver, but they are not the preassigned route. Finally, the actual route R_a represents the route taken by a vehicle V_0 and it can be any of all the available authorized routes.

In the GAM system, all routes from R_p to $R_{s1}, \ldots, R_{sn}$ are uniformly discretized into points with the purpose of performing visibility analysis throughout the routes. By sampling a route R_0 to represent it as a set of points $P_0, \ldots, P_n$, it becomes possible to check whether a target point P_t is visible from an observer point P_o. In systems where deviations of route travel time are also processed, each vehicle V_0 will be associated with a time to complete a trip. T_e may represent the expected travel time and T_a may represent the actual travel time.

A trip instance starts when vehicle V_0 is associated with a preassigned route R_p, a set of secondary routes $R_{s1}, \ldots, R_{sn}$, and, if time deviations will be detected, an expected time T_e to complete the trip, i.e., to arrive at an inspection point. V_0 is supposed to follow R_p and reach an inspection point after T_e. An inspection officer or a system at the inspection point could query V_0 for its trip data and check if R_p has been followed in a timely manner. For this to occur, a GIS/GPS tracking device must be installed on V_0 to record all of V_0 movements. If the officer or a tracking system determines that R_p has not been followed or that T_a differs significantly from T_e, the officer will have to ask the driver of V_0 for a justification. The driver of V_0 can justify the route or time deviations with a number of reasons. It will be up to the inspection officer to validate those justifications and decide whether V_0 can continue its trip beyond the inspection point or not.

This thesis focuses on driver justifications that have to do with route deviations caused by obstacles, e.g., accidents, traffic jams, road work, and so on, along R_p which the driver is able to see before running into them. When the driver claims that in order to avoid a delay he or she changed from R_p to R_a, this claim must be validated by the system. The GAM system performs visibility analysis along R_p and $R_{s1}, \ldots, R_{sn}$ in order to prove if it is possible to see a route obstacle from the location where the driver of V_0 claims that

he or she spotted it. GAM also includes a logic application using a set of rules which can be loaded into an inference engine to derive a conclusion about whether the driver's justification for a route deviation is valid or not.

For a route deviation the next rules must be checked:

- Ensure that the road obstacle identified by the driver is in an area inside of the preassigned route, in other words, is P_t in R_p? where P_t refers to the visibility analysis target point observed by the driver of V_0.

- Is the vehicle before an intersection of authorized routes that would allow the driver to deviate into a secondary route? This is important because, without visibility before the intersection, the driver must decide to deviate from R_p before seing any obstacle, flagging this behavior as suspicious. Is the vehicle position, observer point, P_o inside R_p and R_a?

- Finally, do P_o and P_t have positive line of sight? If the driver cannot see P_t from P_o then the driver's justification should not hold.

If needed, to detect or justify a time anomaly, compare T_a with T_e including an authorized time tolerance. If the difference between T_a and T_e falls outside of the time deviation tolerance then the driver's justification should not hold either.

A Geographic Information System performing visibility analysis and an inference engine deriving conclusions about route deviations and justifications are the key working components of the GAM system. The next section goes into detail about the system components and the modeling approach.

4.4 Solution Approach

To address the problem of validating or rejecting route deviation justifications in land transportation scenarios like those discussed in the previous sections, this thesis proposes

a system with a stratified architecture where each stratum solves a subset of the problem. Figure 4.3 shows a diagram of the stratified architecture of the proposed GAM system.

Before presenting each stratum in detail, it is important to mention that a simpler solution which would not utilize all of the components used in the presented architecture would fall short of producing the intended results. For instance, a solution based on a model which only uses GIS technology would allow only representation of GIS data layers such as ground elevation data, commercial road networks, and authorized pre-established routes. This solution could also allow performing visibility analysis on the road network layer. However, it would need a trained GIS user to analyze the results and produce an actionable conclusion. Moreover, without a reasoning component for non-GIS items, it would be very difficult to define and execute the rules presented in the problem definition section. Although it would be possible to create a logical component within a GIS application, this task would be better implemented on a language or environment which is suitable for logic programming such as Prolog. Thus leaving the GIS components to perform what GIS does best: spatial analysis.

GAM's stratified architecture encompasses the following components:

- Commercial road network and terrain elevation. The first stratum contains all of the highways, avenues, boulevards, drives, and so on, generically referred to as roads, included in the geographic area where shippers and ports of entry of interest are located. This stratum can be used to monitor each vehicle V_0 more precisely as we can see the very road it is following while in route to a POE. Terrain elevation data is very important because it will be used as one of the inputs for the visibility analysis operations, i.e., for LOS computations. This stratum component is comprised of a series of layers in a GIS map.

- Authorized pre-established trip routes. This is the set of all authorized routes for all shippers in the model. This stratum has very detailed information about all the routes, i.e. R_p and $R_{s1}, \ldots, R_{sn}$ for all trips for all shippers, but it only shows the

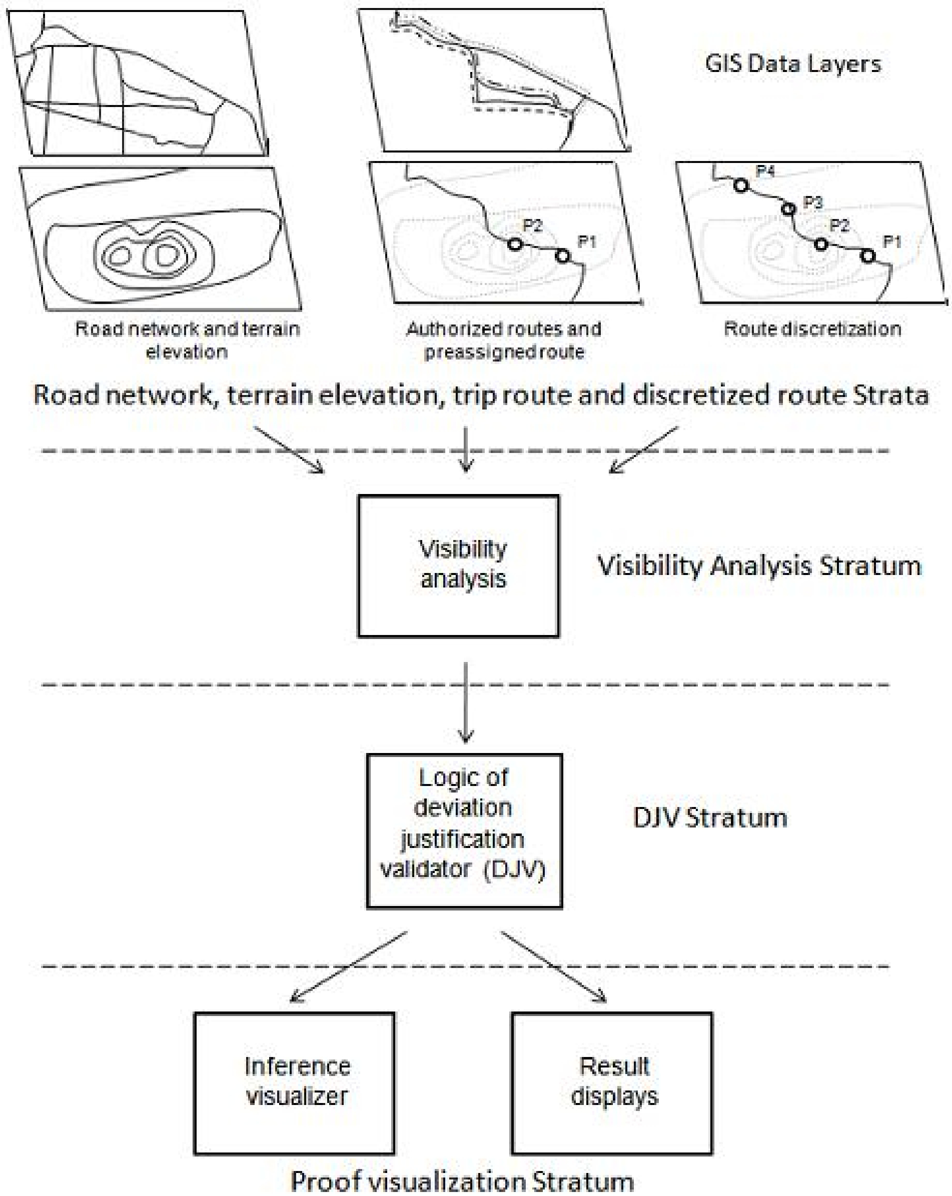

Figure 4.3: Stratified solution architecture of the GAM system.

routes that are available for a given trip instance. Identifying such routes is important because all the logical rules used to detect route deviations and validate justifications are based on known facts about the authorized preassigned and secondary routes. Having these data on a separate layer enables quick access to the information. This stratum component is comprised of a series of layers in a GIS map.

- Discretization of routes. Each authorized route, i.e., R_p and $R_{s1}, \ldots, R_{sn}$ for all trips of all shippers, has to be discretized from its uniformly sampled segments. The resulting route R_0 made out of evenly distributed sample points represents an area of the map over which LOS visibility analysis can be performed. This stratum component uses an algorithm to create and navigate each discretized route which runs as a GIS process.

- Visibility analysis. This stratum performs LOS computations for all the authorized pre-established routes. For each route, the algorithm iterates over all pairs of sampled points along the route to compute LOS visibility in the route origin to destination direction. The utilized line of sight computation has three input parameters: the observer point P_o, the target point P_t, and the terrain elevation data underlying the road network containing the discretized routes. The computed output is a true or false value indicating whether P_t is visible from P_o. This stratum component uses LOS algorithms and it runs as a GIS process.

- Route Deviation and Justification Validator (DJV). This component has all the logic behind the decisions and rules created to validate driver justifications. All the facts and rules were written in Prolog to build an application which could use the results of the route visibility analyses computed by the previous component.

- Proof visualization. The last component is a visualization tool which shows how the inference engine arrived at a result. Since it is hard to follow the inference process when working with numerous logical rules and abundant facts, having a proof

visualization tool simplifies the manual checking of the steps taken by the inference engine to produce each result. Probe-It! is used as the proof visualization tool supporting the DJV Prolog application.

By designing the GAM system based on independent stratified components, components are logically isolated to allow modification and replacement of individual components without inadvertently making unintended changes to other components. For instance, this allows the use of different software tools and technologies with a minimalist framework for interoperability. This allows use of a commercial GIS, the DJV logic program written in Prolog, and an off-the-shelf proof visualization tool in the same solution model. The next section discusses how the objectives of each component are related. The GAM system implementation is discussed in chapter 5.

4.5 Algorithms

This section discusses each component of the GAM system in detail. The objectives of each component from the previous section are matched to the algorithms, assumptions, limitations, and configuration settings used. Each major piece of the GAM system is analyzed to show how it solves a subset of the the problems presented in this thesis with an aim to achieve two goals: the first goal is to present a general approach to solve similar problems; and the second goal is to use a proof-of-concept approach to show one way to implement the presented ideas. The next subsections have all the details and specifications for building the GAM system and chapter 5 shows the integrated working implementation.

4.5.1 Route Discretization

Figure 4.4 presents an authorized route, which is R_2 in figure 4.1, divided into sampled segments. Each segment is sampled once in the middle of the segment and the resulting point is used to build a discrete representation of the route for LOS computations and sym-

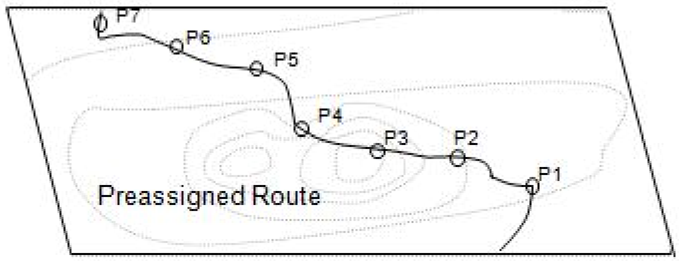

Figure 4.4: Preassigned route with discretized points.

bolic processing in DJV. The discretization of the route provides a way to identify specific points along R_2 which have both a geographical and a logical representation for symbolic processing by the DJV Prolog application. The discretization of R_2 will also control the performance of the visibility analysis algorithm, as the sampling distance determines how many pairs of the points representing each route are used as input for visibility computations. Without route discretization, the visibility computations along each route and the corresponding symbolic representation of their results would have required a different representation and logic processing approach. This is briefly explored in the Future Work section of this thesis.

The route sampling distance may be selected on a case-by-case basis as each road network will have unique characteristics that the modeler could consider. For example, modeling a geographic region that does not have a lot of buildings or an interstate highway on a plain terrain that is mainly a straight line may dictate different values of the sampling distance. For a region with a dense set of buildings along a route, a very small sampling distance may be appropriate. In contrast, a long straight flat highway could be sampled with well-dispersed points with a low risk of losing visibility information. However, to sample a route which includes a highway that goes around a mountain a small sampling distance value may be needed.

4.5.2 Visibility Algorithm

Once each route is discretized, point-to-point visibility can be computed. The visibility algorithm below starts by taking the first point of a discretized route and using it as an observer point P_o, the target point P_t is the next point P_o along the direction from route origin to destination. For each (P_o, P_t) pair, if the computed visibility is true, the pair is added to the KB as a fact. To continue, P_t is dropped and the next point along the route is selected as the new P_t to compute the visibility of the new pair. For each route, each point in the route will be paired with all the points ahead of it in order to compute route visibility.

Algorithm 1: Route visibility analysis algorithm using LOS computations

```
Input: Each route R containing sampled points
Output: A True/False value for each pair of points with positive LOS
foreach R in Routes do
    for i ← 0 to R.length do
        for j ← i + 1 to R.length do
            IsVisible ← ComputePointToPointLOS(i, j);
            if IsVisible = True then
                AddRuleToKnowledgeBase(R, i, j);
```

The visibility analysis is performed with LOS algorithm which uses terrain elevation data to compute if P_t is visible from P_o for all (P_o,P_t) pairs in the model. The GAM system uses ESRI's LOS algorithm, which is located in the Military Analyst Toolbox extension of ArcMap 10. The following assumptions were taken into consideration:

- The LOS algorithm uses terrain elevation and point coordinate data to compute visibility. It does not take into consideration weather conditions, vegetation, time of day, position of the sun, or any other criteria for its analysis. The addition of such criteria is the topic of ongoing research and future related work which started with work in this thesis[22].

- The driver has no sight problems or any other obstacles for visibility along a route. All outdoors environment conditions are optimal and all drivers have the same visibility range.
- Visibility analysis will only be used within each route, no cross-route visibility analysis will be used or necessary. The driver's ability to see an obstacle up ahead in the preassigned route is all that matters because that is the only information used to leave a preassigned route for a secondary route.
- The visibility analysis is pre-computed once for all the routes in the road network and visibility data is added to the KB. Visibility data can be used any number of times by DJV without having to perform on-the-flight visibility analysis.

4.5.3 The Threat Detection System and Deviation Justification Validator

The following logic rules represent the core assumptions, constraints, and criteria modeled in this thesis. The rules below are embedded into GAM's DJV and the Threat Detection System (TDS), a Prolog application which was designed and implemented for the CargoTrust project of The University of Texas at El Paso. CargoTrust was funded by a grant of the U.S. Department of Homeland Security (DHS). TDS models threat detection for land transportation problems of interest to DHS. There are three assumptions considered in the creation of rules for DJV:

- The visibility analysis pre-computed with the LOS algorithm is valid for all times and visibility facts are always present in the KB for rule application.
- The time dimension is not considered in the analysis. This means that any time stamp of a given obstacle in a route is discarded and there is no need to match the time span of an obstacle to the time span of an approaching driver and visibility information of the obstacle. The consideration of the time dimension in the GAM

system is out of the scope of this thesis. However, use of timing criteria is discussed in the future work section.

- The focus of this thesis is on checking whether a driver could see an obstacle ahead in the preassigned route to support a justification to abandon the route for a secondary route. Reasons for choosing an alternative route over another are out of the scope of this thesis, but they are additional opportunities for future research.

The following are the core logic rules used for the DJV Prolog application:

- $hasThreat(V) \leftarrow hasAnomaly(V) \wedge (!) \vee hasUnjustifiableAnomaly(V)$

  $HasThreat$ is the entry point for evaluating threats and deviation justifications. If $HasThreat$ evaluates as true for a vehicle V, there is a problem in the trip instance being evaluated. $HasAnomaly$ detects problems that automatically stop the evaluation because of deviations which cannot be justified such as when the KB lacks monitoring information about V, i.e. a vehicle appeared out of nowhere at a POE. $hasUnjustifiableAnomaly$ is true for anomalies without a valid justification including those representing route deviations.

- $hasUnjustifiableAnomaly(V) \leftarrow hasRouteTimeAnomaly(V) \wedge$ $\neg(hasJustification(V, P_o, P_t))$

  $hasUnjustifiableAnomaly$ becomes True if there is a time or route anomaly i.e. $hasRouteTimeAnomaly$ is True and there is no justification for that anomaly, i.e. the negation of $hasJustification$. $hasJustification$ for the vehicle V, the observer point P_o, and the target point P_t returns the justification computation for those precise points.

- $hasRouteTimeAnomaly(V) \leftarrow hasTimeDeviation(V) \vee hasRouteDeviation(V)$

  This rule is the first half of the previous rule and $hasRouteTimeAnomaly$ is True if either the time or the route are the cause of the anomaly.

- $hasTimeDeviation(V) \leftarrow \neg(timeOK(V))$
  $timeOK(V) \leftarrow hasExpectedTime(V, T_e) \wedge hasActualTime(V, T_a) \wedge onTime(T_e, T_a)$
  $timeTolerance(T) \leftarrow T = 5$
  $onTime(T_e, T_a) \leftarrow timeTolerance(E) \wedge (T_a > T_e - E) \wedge (T_a < T_e + E)$

  hasTimeDeviation checks whether the time data caused a time anomaly or not and whether it has a proper justification. *hasExpectedTime* checks whether there is an expected time T_e for vehicle V in the KB and *hasActualTime* checks whether there is an actual time T_a for V in the KB. *onTime* uses a value of E time units as the deviation tolerance to detect whether a time anomaly is justified. $(T_a > T_e - E)$ and $(T_a < T_e + E)$ are evaluated and if the T_e and T_a are not approximately equal within that range the anomaly is not justified.

- $hasRouteDeviation(V) \leftarrow \neg(routeOK(V))$
  $routeOK(V) \leftarrow hasPreassignedRoute(V, R_p) \wedge hasActualRoute(V, R_a) \wedge (R_p = R_a)$

  hasRouteDeviation becomes True if the V does not have a preassigned route R_p and an actual route R_a in the KB. It also checks whether the routes are equal. Otherwise this rule will flag a route anomaly for the system to justify.

- $hasJustification(V, P_o, P_f) \leftarrow$
  $\neg(invalidPoints(P_o, P_f, allpointsinR_p, allpointsinR_a))$
  $\vee (jamJustify(A) \wedge trafficJam(V)) \vee (accidentJustify(A) \wedge trafficAccident(V))$
  $\vee\ weatherJustify(A) \wedge weatherDelay(V)$

  hasJustification models justification validation for route deviations and other justifications processed by DJV. *invalidPoints* receives P_o, P_t, and all the points in R_p and R_a in order to whether the target point is visible from the observer point and the latter is part of both routes.

- $invalidPoints(P_o, P_f, R_p, R_a) \leftarrow processPointsVisibility(P_o, P_f, R_p, R_a)$

This rule was made to isolate the process of checking the points' visibility. The next rule will go over the detail of such operations.

- $processPointsVisibility(P_o, P_f, R_p, R_a) \leftarrow \neg(isValidPoint(P_o)) \vee \neg(isValidPoint(P_f)) \vee \neg(isFinalPointInPreassignedRouteOnly(P_f, R_p, R_a)) \vee \neg(isInitialPointInBothRoutes(P_o, R_p, R_a)) \vee \neg(isPositiveLOS(P_o, P_f))$

  $processPointsVisibility$ evaluates symbolic point-to-point visibility and route position rules for two points. $isValidPoint$ checks whether P_o and P_t are part of the KB by determining whether those points belong to any route. *isFinalPointInPreassignedRouteOnly* becomes true if P_t belongs to R_p. This is important because in order to justify a deviation, the driver would have to be observing a point along R_p. *isInitialPointInBothRoutes* makes sure that the driver is before an R_p,R_a intersection, i.e., it is possible to abandon the preassigned route for a secondary route. To do this, it checks if P_o is inside both R_p and R_a and that the driver is in condition to see an obstacle and deviate to another route if needed. Finally, *isPositiveLOS* provides the visibility data of the two points (i.e. P_o and P_t), if there is a positive LOS the rule becomes True. LOS values are pre-computed by the GIS and added to the KB.

- $isValidPoint(P) \leftarrow discretePoint(P)$

  This rule goes through the KB searching for the given point P. If it is found the rule becomes True.

- $isPositiveLOS(P_o, P_f) \leftarrow positiveLOS(P_o, P_f)$

  This rule goes through the KB searching for a fact stating that points P_o and P_f have positive LOS.

- $isPointInRoute(P, R) \leftarrow pointInRoute(P, R)$

  This is an important rule because it takes a point P and checks whether it belongs to the set of points representing route R.

- $isFinalPointInPreassignedRouteOnly(P_f, R_p) \leftarrow isPointInRoute(P_f, R_p)$

  $isFinalPointInPreassignedRouteOnly$ becomes True if the target point P_t belongs to the preassigned route R_p.

- $isInitialPointInBothRoutes(P, R_p, R_a) \leftarrow$
  $isPointInRoute(P, R_p) \wedge isPointInRoute(P, R_a)$

  This rule checks whether point P is in both routes R_a and R_p.

4.5.4 Results and Proof Visualizer

The rules in the previous section, which are implemented in the DJV Prolog application, produce a logic value indicating whether the driver's justification is valid or not. If the novel work presented in this thesis is implemented in a production system, there will be a need to check specific results for system auditing and testing. Probe-It! can be used to see exactly how DJV arrived at a conclusion. This part of the GAM system would only be accessible when there is the need to check the reasoning of a specific instance, as during audits and debugging sessions. Moreover, the day to day use the GAM system would not have the option to use Probe-It! This functionality would only be available for advanced users and system developers.

Chapter 5

Implementation Details and Results

The algorithms and ideas presented in chapter 4 were implemented in the GAM system. The implementation architecture is based on several software tools including applications developed as part of this thesis, off-the-shelf tools, and commercial software. Figure 5.1 depicts the system architecture and software applications which will be discussed in this chapter.

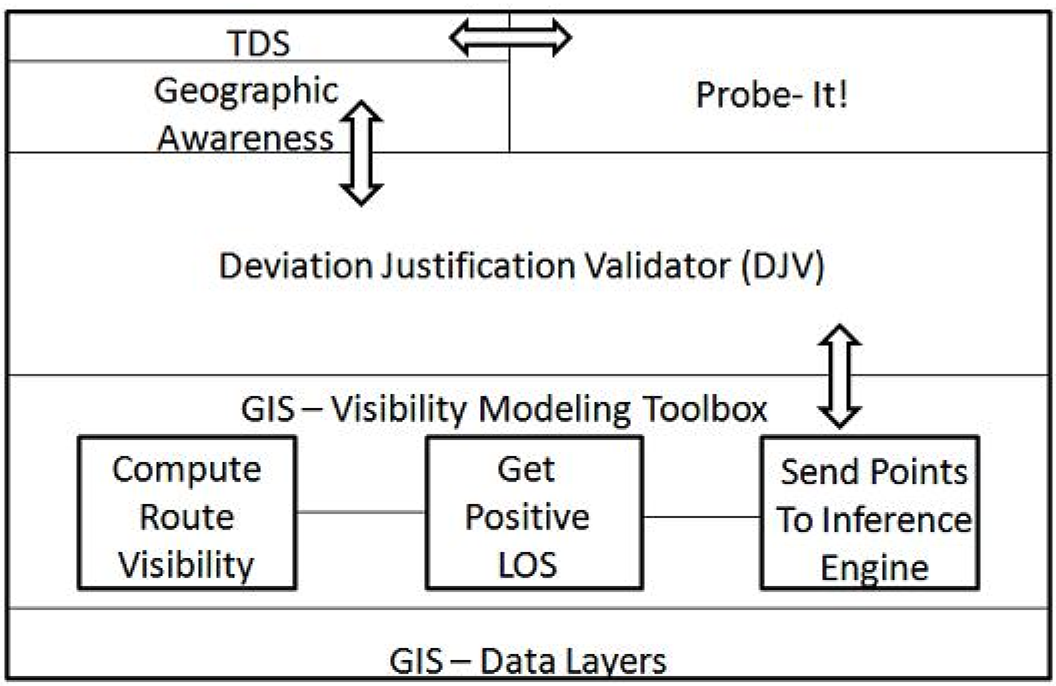

Figure 5.1: The GAM system architecture

At the bottom, the GIS data layers model the underlying road network, terrain elevation, and route definitions used in this thesis. All the layers are based on the same coor-

dinate system: **GCS_North_American_1983**, projected coordinate system **NAD_1983-_StatePlane_Texas_Central_FIPS_4203_Feet** and projection: **Lambert_Conformal-_Conic**. A coordinate system is a three-dimensional reference system that locates points on the Earth's surface, the projected coordinate system is a two-dimensional planar surface used to locate points in a computer system with a 2D view. A projection is a mathematical transformation to convert three-dimensional coordinates into two-dimensional coordinates.

Terrain elevation is defined by a raster layer which is the preferred layer type for elevation data. The commercial road network layer has all major highways in the area utilized for visibility analysis. These two layers represent the first stratified component in the Solution Approach section of chapter 4. The second stratified component are the available pre-established trip routes, i.e. the authorized preassigned and secondary routes, which are also represented in the road network layer. The third stratified component is the discretization of routes and is represented by a point layer on top of the road network layer.

In figure 5.1, there is a Geographic Information System on top of the GIS Data which contains both the data and the necessary programming tools to compute the visibility analysis. The GIS software used for this thesis is ArcGIS ArcMap 10[13] and it represents the fourth stratified component introduced in the Solution Approach section of chapter 4. ArcMap 10 offers functionality to perform spatial operations on data layers. These operations are grouped into Toolboxes and can be accessed in the Toolbox section of ArcMap 10. Moreover, it is possible to create customized toolboxes and perform specialized operations on data. Thus, a Visibility Modeling Toolbox was created in order to carry out the work being presented.

A GAM Visibility Modeling Toolbox for LOS computations over each route of a road network was written using ArcMap. The GAM Toolbox includes three elements: The model named **Generate LOS Layers** and two scripts named **Get Positive LOS** and **Get Positive LOS DATA**, respectively. A diagram and detailed specifications of the **Generate LOS Layers** model can be found in Appendix B of this document. However,

it is worth mentioning that its main component is the *Linear Line of Sight from Features* tool available in the Military Analyst Tools extension of ArcMap 10. The model receives the discretized points of each route and computes point to point visibility with the line of sight tool. Figure 5.2 shows the on-screen results after running the model; green line segments indicate positive visibility along a route. When the only goal is to find the target point's visibility from the observer point, that is, to calculate whether a target point P_t is visible from an observer point P_o, only the visibility information adjacent to P_t has to be considered.

The task of extracting the visibility information on P_t is done by the **Get Positive LOS** script. It is written in Python[32] and the source code can be found in appendix B. This script also identifies a visible point pair and it records it in two places: A table inside the GeoDatabase where all layers are saved and in a Prolog file (i.e., .pl extension file) which is uploaded into the KB. The last script, **Get Positive LOS DATA** has variable names and other information used to run the first script, its source code can also be found in appendix B.

The next application shown in figure 5.1 is the DJV application written in Prolog and it can be mapped to the fifth stratified component mentioned in the Solution Approach section of chapter 4. DJV is based on the Thread Detection System, an application designed to model problems related to transportation of valuable and sensitive assets. Every logical rule from the Logic Rules section in chapter 4 was implemented in DJV. Appendix A contains the source code for the rules. Geographic awareness is modeled through the combined application of DJV logic rules, the visibility facts for points in discretized routes produced by the GAM Visibility Modeling Tools, and DJV's ability to answer queries related to threats, anomalies, and justifications related to land transportation problems.

Figure 5.3 shows the TDS interface where an end user can select option from the main menu. Menu options for adding facts to the KB are used to model a transportation scenario. For instance, facts about a trip vehicle V, preassigned route R_p, actual route R_a, expected time T_e and actual time T_a are initially created. Furthermore, information computed by

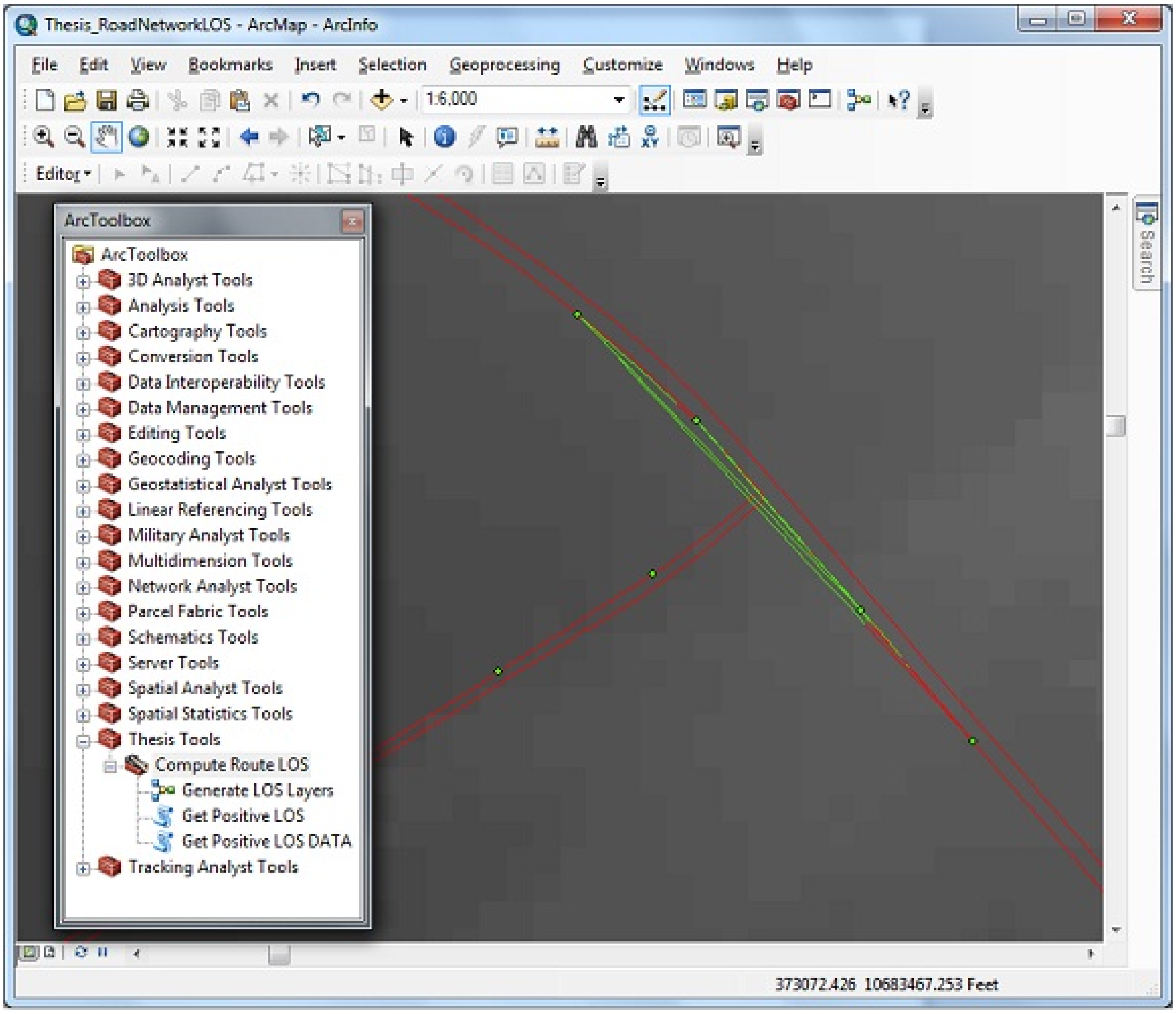

Figure 5.2: ArcMap shows point to point visibility using ESRI's line of sight tool.

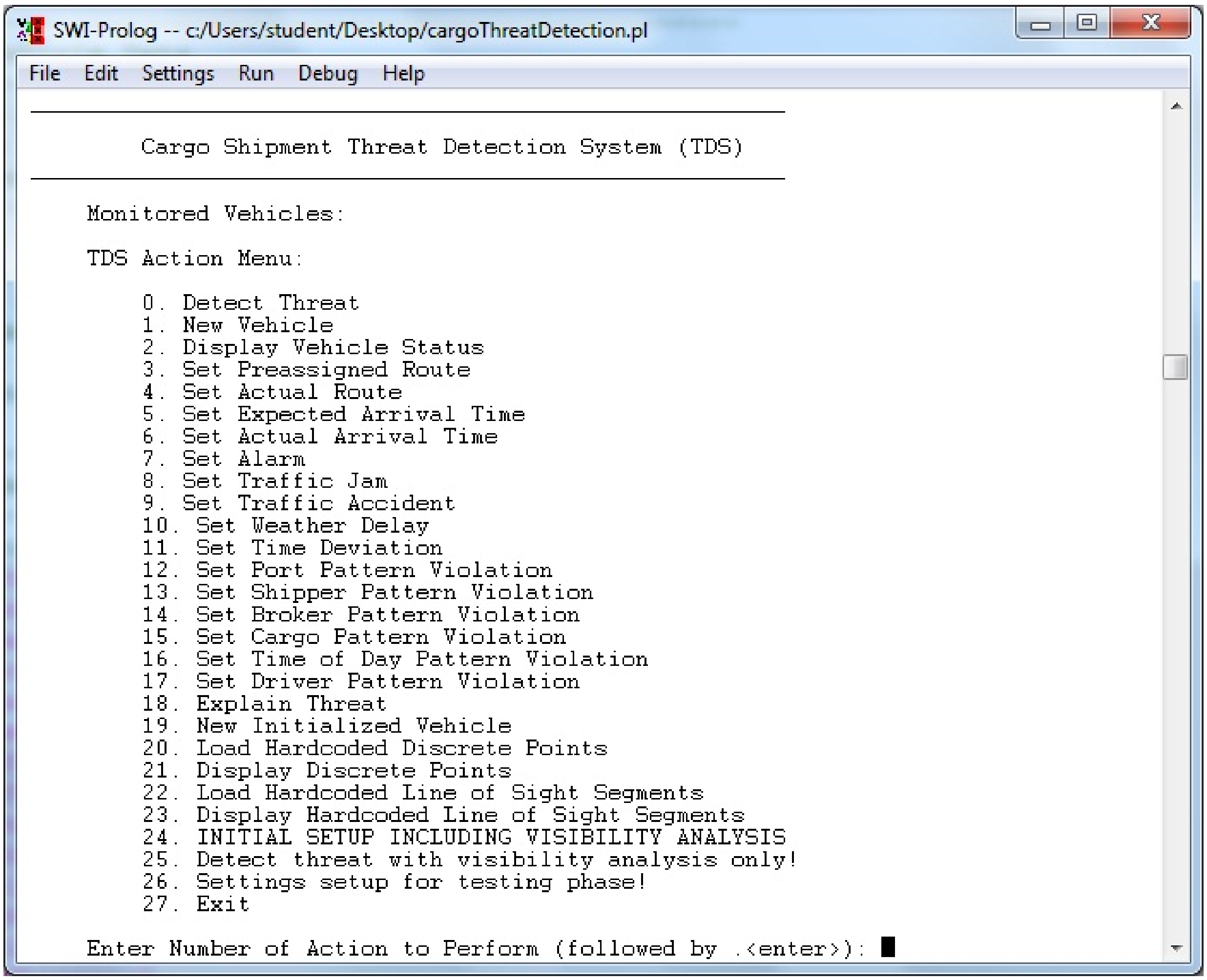

Figure 5.3: Main menu of the Thread Detection System application used by DJV.

the Visibility Modeling Toolbox implemented for this thesis, i.e. discretized route points and point-to-point visibility, can also be added with options in this menu.

Scenario information to be processed is added to the KB as facts which are described in the Prolog section of chapter 2. Once the KB is populated with the information, the KB can be queried to obtain new information. For an inspection officer who needs to verify a route deviation and whether it has a justification, using GPS provided location information for an accident, which is also identified by the driver, and the location of the driver where the accident was spotted, which must have been before reaching an intersection between the preassigned route and a taken secondary route,the user can verify visibility using the system. For input data which does not contain any threat, the result would be a "No threat detected in vehicle!" message indicating that there is a positive line of sight between those points which can be used as part of the validation for a driver justification for a route deviation.

For the case above with a modification of the KB facts about the segment of the road network, the same vehicle would be evaluated with a new observer point $p3$ and a new target point $p4$. In case the resulting message were "Points are not visible: $p4$, observed point is not in the preassigned route. RED LIGHT!", this would indicate that the target point is not part of the preassigned route. Hence, there should be no need for the driver to mind an accident at that location of the road network. This scenario would give good reason for an inspection officer to doubt and further question a driver's route deviation justification. Modeling geographic awareness in this manner, the DJV logic application was created to process queries for evaluating the validity of a driver's trip anomaly justifications.

The last component in figure 5.1 to discuss is Probe-It!, which can be mapped as the sixth stratified component mentioned in the Solution Approach section of chapter 4. Probe-It! is an information visualization tool for rendering proofs encoded in the Proof Markup Language (PML) [30]. For instance, such visualizations provide advanced users and developers with details about how logic rules are applied by a Prolog application. For the purpose of this thesis, Probe-It! allows auditing of the processing implemented by the

Prolog application, i.e., it helps users to understand all the inference steps taken to arrive at certain conclusion. Probe-It! takes as input a PML file which has information about the rules and facts used for a logical inference, builds a graphical view of the inference process, and displays the graphical representation to allow exploration of the proof.

Figure 5.4 depicts DJV application of the rule $isPositiveLOS(Po, Pf) : -positive\text{-}LOS(Po, Pf).$, which receives two points as input, an observer point and a target point, and evaluates whether there is positive LOS visibility between them. If the points are present in the KB, the visualization generated by Probe-It! will confirm the result of applying the rule. The graphical view of the inference process taken to arrive at this conclusion can help the user to understand the rule application and the result better.

This chapter presents the implementation and sample execution results of the three main software components designed, developed, and/or used for this thesis: the GAM GIS application, the DJV Prolog application, and Probe-It! The overall goal of the GAM system is to show one way to model geographic awareness in order to provide a reliable verification of a driver route deviation justification. The following chapter presents conclusions and ideas for future work.

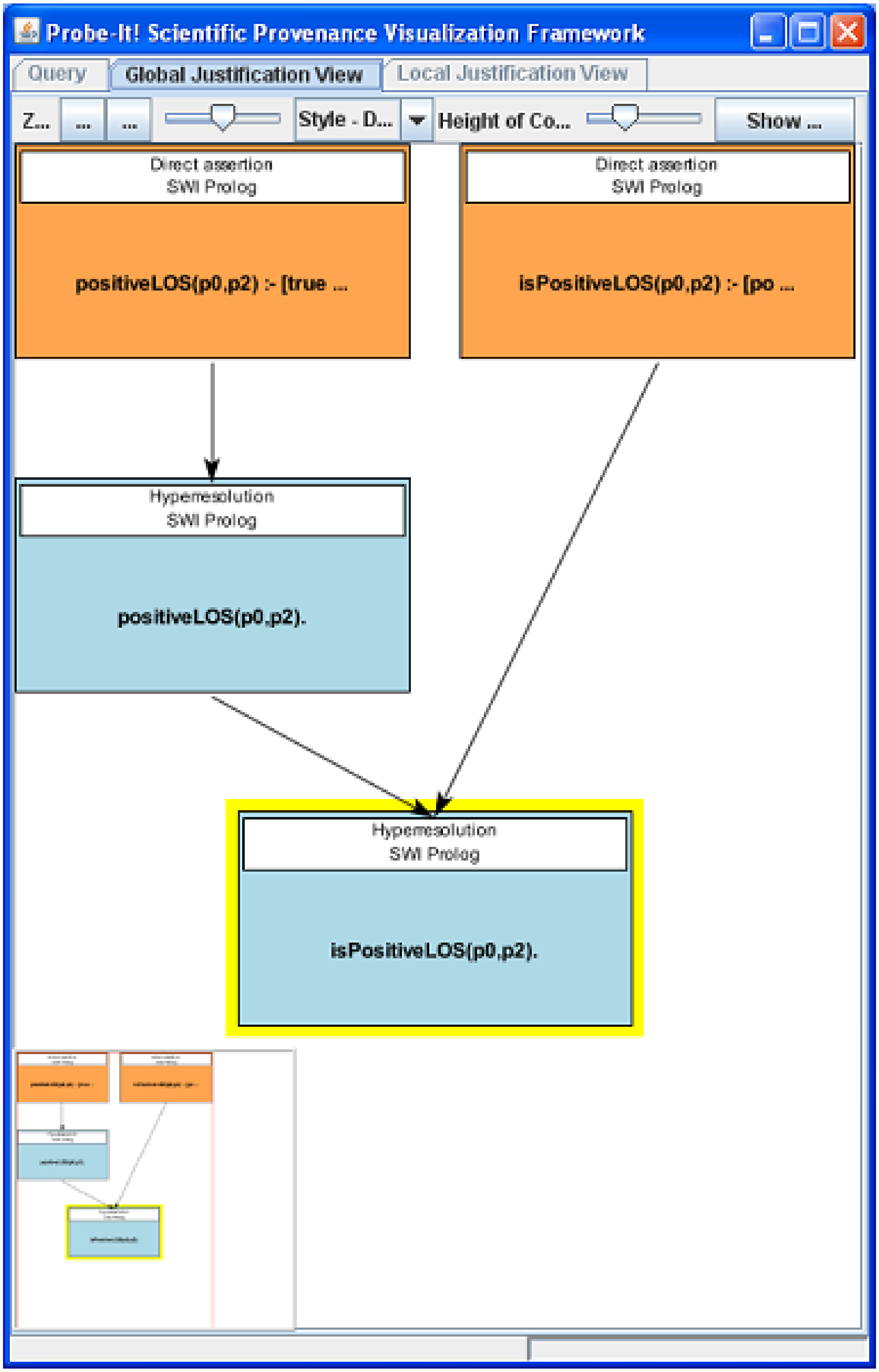

Figure 5.4: Probe-It! Visualization of isPositiveLOS rule application

Chapter 6

Concluding Remarks

6.1 Significance of the Result

This thesis presents the analysis and a solution for problems related to the ground transportation of valuable or sensitive assets such as commercial products and materials, cultural assets and works of art, and hazardous materials. By using pre-established, authorized routes between shipper and inspection locations, vehicle trips can be monitored for route following compliance and deviations. This thesis presented a solution approach in which the GAM system can detect route deviations and validate driver-provided justifications for such anomalies. By combining the results of GAM GIS visibility analysis with DJV's logic rules and facts, this thesis models driver geographic awareness which can be used in an inspection support tool to detect route deviations and validate driver justifications.

The presented solution approach implemented in the GAM system can enable enhanced monitoring of land transportation operations through validation of driver provided information about route deviations. By modeling and gathering various facts about trip instances such as road network visibility, vehicle locations, predefined trip routes, and actual trip routes, the presented work can be used to assist vehicle and payload inspectors to make better informed decisions about the need for detailed inspections of a trip elements. The GAM system aims to complement the expertise and the experience of inspectors by providing extra information about route deviations and justifications which can increase the effectiveness of inspection targeting policies and sampling rates. As a complement for advanced users and developers modifying the DJV logic application, the presented solution also uses Probe-It!, a PML proof visualization tool, to enable users' analysis of the inference

derivation process supporting a route deviation detection or deviation justification result.

To speed up application performance and leverage the slow rate of change of actual road networks, the implemented solution precomputes visibility analysis along the predefined routes once. The computed results are added as KB facts at the beginning of an application session.

6.2 Future Work

The following are recommended starting points for continuing and future research based on the work presented in this thesis.

- It would be interesting to use a complex LOS algorithm, such as the one presented in [22], which uses diverse criteria to analyze visibility. Having complex LOS computation capabilities would enable a monitoring application to deal with visibility constraints which are present in various road networks.

- The discretization of predefined routes used in this thesis to perform visibility analysis could be replaced by a model based on constraints and interval computations to reduce the potential impact of errors caused by location approximations based on route sample points.

- The Threat Detection System used in this thesis utilizes elapsed time to measure expected and actual arrival times to indicate a potential trip anomaly when arrival time values are not within an authorized tolerance. This and other time-related criteria and processing could be considered to support processing of dynamic transportation scenarios.

- The GIS implementation utilized in this Thesis is based on 2D visualizations of the results of visibility analyses. Extending the implementation to utilize three-dimensional visualizations may enhance the usability of the displayed results.

References

[1] P. Blackburn, J. Bos, and K. Striegnitz, "Learn Prolog Now!," http://www.coli.uni-sb.de/kris/learn-prolog-now, 2001.

[2] J. Brainard, A. Lovett and J. Parfitt, "Assessing hazardous waste transport risks using a GIS," *International Journal Geographical Information Systems*, vol. 10, pp. 831-849, 1996.

[3] I. Bratko, "PROLOG Programming for artificial intelligence 2nd Ed.," Addison-Wesley, 1990.

[4] City of El Paso GIS, "Paso del Norte Mapa," http://www.pdnmapa.org/pdnmapa/HTML/epshps/trkrts.zip.

[5] S. Costantini and A. Tocchio, "About declarative semantics of logic-based agent languages," *Post-Proc. of DALT 2005, LNAI 3229*, Springer-Verlag, Berlin, 2006.

[6] S. Costantini and A. Tocchio, "The DALI logic programming agent-oriented language," *Logics in Artificial Intelligence, Proc. of the 9th European Conference, LNAI 3229*, Springer-Verlag, Berlin, 2004.

[7] S. Costantini, A. Tocchio, P. Tsintza and L. Mostarda, "Agents and Security in a Cultural Assets Transport Scenario," *WOA* pp. 78-86, 2007.

[8] CUSPIS Project Website, http://www.cuspis-project.info/demonstrations.htm

[9] N. Del Rio and P. Pinheiro da Silva, "Probe-it! Visualization support for provenance," *Proceedings of the Second International Symposium on Visual Computing (ISVC 2)*, pp. 732-741, Lake Tahoe, NV, 2007.

[10] K. J. Dueker, "Land resource information systems: A review of fifteen years experience," *Geo-Processing*, Vol. 1, pp. 105-128, 1979.

[11] M. Egenhofer. "Toward the semantic geospatial web," emphIn Proc. of the 10th ACM Int. Symp. on Advances in Geographic Information Systems (GIS), pp. 14, 2002.

[12] M. J. Egenhofer and D. M. Mark, "Naive Geography," *Spatial Information Theory-A Theoretical Basis for GIS, International Conference COSIT'95, Semmering Austria, A. Frank and W. Kuhn, eds., Lecture Notes in Computer Science*, vol. 988, pp. 1-15, Spring-Verlag, Berlin, 1995.

[13] ArcGIS 10 Website, http://www.esri.com/software/arcgis/index.html

[14] ESRI's ArcGIS 9.2 Desktop Help, Creating a line of sight, http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=2788&pid=2783&topicname=Creating_a_line_of_sight

[15] G. L. Foresti and B. Pani, "Monitoring motorway infrastructure for detection of dangerous events," *Proc. IEEE Int. Conf. Image Analysis and Processing*, pp. 1144-1147, 1999.

[16] M. F. Goodchild, "A spatial analytical perspective on geographical information systems," *International Journal of Geographical Information Systems*, Vol. 1, No. 4, pp. 327-334, 1987.

[17] M. F. Goodchild and J. Lee, "Coverage problems and visibility regions on topographic surfaces," *Annals of Operations Research*, No. 18, pp. 175–186.

[18] P. Govindan, Y. Tian and R. Howlett, "A GIS framework for improving the harbor security," *Proceedings of the 26th Annual ESRI International User Conference*, San Diego, USA, 2003.

[19] K. Hiramatsu and F. Reitsma, "GeoReferencing the Semantic Web: ontology based markup of geographically referenced information," emphJoint EuroSDR EuroGeographics workshop on Ontologies and Schema Translation Services, Paris, France, 2004.

[20] B. Huang, R. L. Cheu and Y. S. Liew, "GIS and Genetic Algorithms for HazMat route planning with security considerations," *International Journal of Geographical Information Science*, vol. 18, pp. 769787, 2004.

[21] J. Lee, "Analysis of visibility sites on topographic surfaces," *International Journal of Geographical Information Systems*, vol. 5, no. 4, pp. 413–429, 1991.

[22] A. Kassin, T. Magoc, R. Romero, "A line of sight algorithm using fuzzy measures," *North America Fuzzy Information Processing Society (NAFIPS),* pp. 1-6, 2010.

[23] M. Lepofsky, M. Abkowitz and P. Cheng, "Transportation hazard analysis in an integrated GIS environment," *G.E.G. Beroggi, W.A. Wallace (Eds.), Computer Supported Risk Management, Kluwer Academic Publishers, Dordrecht*, pp. 115132, The Netherlands.

[24] L. Liu, Z. Liqiang, C. Chen, and C. Hong, "An improved LOS method for implementing visibility analysis of 3D complex landscapes," *International Conference on Computer Science and Software Engineering*, no. 2, pp. 874–877, 2008.

[25] Los Alamos National Laboratory, "Los Alamos tracking system helps flag route errors for WIPP trucks," *News and Communications Office*, http://www.lanl.gov/news/index.php/fuseaction/home.story/story_id/1123

[26] K. McCarty and M. Manic, "Line-of-sight tracking based upon modern heuristics approach," *Industrial Electronics and Applications*, pp. 40–45, 2008.

[27] L. Mostarda, A. Tocchio, P. Inverardi and S. Costantini, "A geo time authentication

system," *Trust Management, IFIP International Federation for Information Processing*, vol. 238, pp. 123-138, 2007.

[28] A. Nerode and R. A. Shore, "Logic for applications 2nd Ed.", Springer, 1997.

[29] P. Pinheiro da Silva, D. L. McGuinness, N. Del Rio, and L. Ding, "Inference web in action: Lightweight use of the proof markup language," *International Semantic Web Conference*, 2008

[30] P. Pinheiro da Silva, D. L. McGuinness, and R. Fikes, "A Proof Markup Language for the semantic web," *Information Systems*, vol. 31, issues 4-5, pp. 381-395, 2006.

[31] P. Pinheiro da Silva, N. Del Rio and H. Porras, Probe-It! Website, http://trust.utep.edu/probeit/

[32] Python Programming Language Website, http://www.python.org/

[33] SWI-Prolog Website, http://www.swi-prolog.org/

[34] R. B. Seixas, M. R. Mediano, and M. Gatass, "Efficient line-of-sight algorithms for real terrain data," *III Simpsio de Pesquisa Operacional e IV Simpsio de Logstica da Marinha*, Rio de Janeiro, 1999.

[35] T. Tezuka, R. Lee, Y. Kambayashi and H. Takakura. "Web-based inference rules for processing conceptual geographical relationships," *Proceedings of Web Information Systems Engineering*, pp. 14-21, 2001.

[36] D. Tuft, B. Salomon, S. Hanlon, and D. Manocha, "Fast line-of-sight computations in complex envorinment", Department of Computer Science, University of North Carolina at Chapel Hill, Technical report TR05-025, 2005.

[37] P. Wonka, M. Wimmer, and F. X. Sillion, "Instant visibility," *Computer Graphics Forum*, vol. 20, no. 3, pp. 411–421, 2001.

[38] J. Zhang, J. Hodgson and E. Erkut, "Using GIS to assess the risks of hazardous materials transport in networks," *European Journal of Operational Research*, Vol. 121, pp. 316-329, 2000.

Appendix A

DJV/TDS Prolog Source Code

The Prolog source code included in this appendix is the implementation of the TDS/DJV logic rules introduced in chapter 4. The rules were written with SWI-Prolog version 5.6.64[33].

```
hasThreat(V, A) :- hasAnomaly(V,A),!;
    hasUnjustifiableAnomaly(V, A).

hasUnjustifiableAnomaly(V, A) :- hasRouteTimeAnomaly(V, A),
        write('Enter observer point: '),
        read(Po),
        write('Enter target point: '),
        read(Pf),
        not(hasJustification(V,A, Po, Pf)).
```

Listing A.1: Threat and unjustifiable anomaly entry rules

```
hasRouteTimeAnomaly(V, A) :- hasTimeDeviation(V,A);
                             hasRouteDeviation(V,A).

hasTimeDeviation(V,D) :- not(timeOK(V)),
    D='There is a problem with the expected arrival time
    and actual arrival time.'.
timeOK(V) :- hasExpectedArrivalTime(V,Te),
```

```
    hasActualArrivalTime(V,Ta), onTime(Te,Ta).
timeTolerance(T) :- T = 5.
onTime(Te,Ta) :- timeTolerance(E), Ta > Te - E, Ta < Te + E.

hasRouteDeviation(V,D) :- not(routeOK(V)),
    D='There is a problem with the preassigned route
    and actual route.'.
routeOK(V) :- hasPreassignedRoute(V,Ra),
    hasActualRoute(V,Rr), Ra = Rr.
```

Listing A.2: Route and time anomaly rules

```
hasJustification(V, A, Po, Pf) :-
    not(invalidPoints(Po,Pf,[p0,p1,p2,p3,p5],[p0,p1,p2,p4,p6]));
            jamJustify(A),trafficJam(V,_,_);
        accidentJustify(A),trafficAccident(V,_,_);
            weatherJustify(A),weatherDelay(V,_,_).
```

Listing A.3: Has justification rule

```
invalidPoints(Po, Pf, Rp, Ra) :-
    processPointsVisibility(Po, Pf, Rp, Ra, Expl),
        write('Points are not visible: '), write(Expl),
    write(' RED LIGHT!'),nl.

processPointsVisibility(Po, Pf, Rp, Ra, Expl) :-
    not(isValidPoint(Po)),
    concat(Po, ', point is not valid.', Expl);
        not(isValidPoint(Pf)),
```

```
    concat(Pf, ', point is not valid.', Expl);
        not(isFinalPointInPreassignedRouteOnly(Pf, Rp)),
    concat(Pf, ', observed point is not in
        the preassigned route.',Expl);
        not(isInitialPointInBothRoutes(Po, Rp, Ra)),
    concat(Po, ', initial point is not before
        the intersection.',Expl);
        not(isPositiveLOS(Po, Pf)), concat(Po, '-', Str1),
    concat(Str1, Pf, Str2),
    concat(Str2, ', points do not have LOS
        between them.', Expl).
```

Listing A.4: Point processing rules

```
isValidPoint(P) :- discretePoint(P,_,_).
isPositiveLOS(Po, Pf) :- positiveLOS(Po, Pf).

pointInRoute(X,[X|_]).
pointInRoute(X,[_|Ys]):- pointInRoute(X,Ys).

isPointInRoute(P, Route) :- pointInRoute(P, Route).

isFinalPointInPreassignedRouteOnly(Pf, Rp) :-
    isPointInRoute(Pf, Rp).

isInitialPointInBothRoutes(Point, Rp, Ra) :-
    isPointInRoute(Point, Rp), isPointInRoute(Point, Ra).
```

Listing A.5: Visibility and route processing rules

Appendix B

GAM Visibility Analysis Tools

B.1 Generate GAM LOS Layers Model

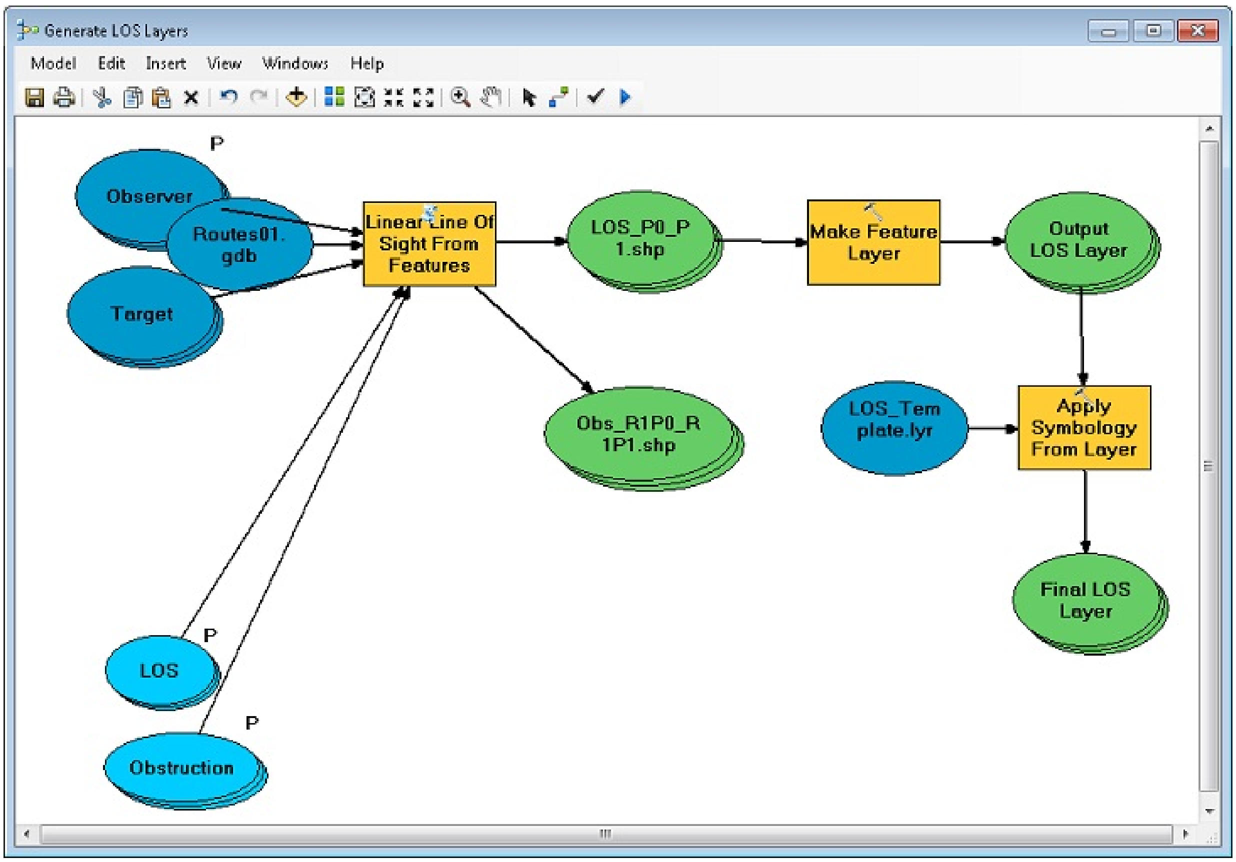

Figure B.1: This model triggers the line of sight tool along the road network.

Figure B.1 shows the **Generate LOS Layers** model created in ESRI's ArcMap 10. It has three processes, the first one is the *Linear Line Of Sight From Feature* tool represented

as the first yellow rectangle. This tools can be found under the Military Analyst Tools → Analysis → Terrain Analysis → Linear Line of Sight from Features. Its inputs are the observer point, target point and the workspace, it also needs two variable names to use as the output features name. The next process is the *Make Feature Layer* tool and it can be found under Data Management Tools → Layers and Table Views → Make Feature Layer, its objective is to convert the features created in the previous process into layers because the third process needs layers as input. The third process, *Apply Symbology From Layer* receives a layer and a template in order to format the resulting layers in such a way that they can be shown in the screen. This process can be found under Data Management Tools → Layers and Table Views → Apply Symbology From Layer.

B.2 GetPositiveLOS and GetPositiveLOSDATA

The **GetPositiveLOS** and **GetPositiveLOSDATA** scripts work together to evaluate each LOS element produced in the **Generate LOS Layers** model with the purpose to identify elements with positive line of sight. If such an element is found, it is written in a table placed under the default GeoDatabase and into a Prolog file as Facts to be added to the DJV KDB. A batch job performs all the operations. While the GetPositiveLOSDATA script has input data, the GetPositiveLOS script performs the LOS computations. Below is the Python source code of each script.

```
# ------------------------------------------------------------
# GetPositiveLOS.py
# Description: This script evaluates visibility in the
# destination point and writes the result in a table and pl file.
# Created by: Ari Kassin
# ------------------------------------------------------------
import sys, string, os, arcgisscripting, GetPositiveLOSDATA
# Create the Geoprocessor object
```

```
gp = arcgisscripting.create()
# Load required toolboxes
gp.AddToolbox("C:/Program Files/ArcGIS/Desktop10.0/ArcToolbox/
    Toolboxes/Data Management Tools.tbx")
gp.AddToolbox("C:/Program Files/ArcGIS/Desktop10.0/ArcToolbox/
    Toolboxes/Analysis Tools.tbx")
VisiblePointsTable = "C:\\Users\\student\\Desktop\\GIS Data\\
    Routes01.gdb\\VisiblePoints"
myFile = open('C:/Users/student/Desktop/GIS Data/
    visivilityAnalysisFacts.pl', 'w')
for i in range(11):
    # Open a search cursor
    rows = gp.SearchCursor(GetPositiveLOSDATA.LOS[i])
    row = rows.Next()
    # Check if the point has a positive LOS visibility.
    # TarIsVis = 1 makes it true, TarIsVis = 0 makes it false.
    if row.GetValue("TarIsVis") == 1:
        myFile.write('asserta(positiveLOS(' +
        GetPositiveLOSDATA.ObserverPoint[i] + ',' +
        GetPositiveLOSDATA.TargetPoint[i]+ ')),')
        OutputRows = gp.InsertCursor(VisiblePointsTable)
        OutputRow = OutputRows.NewRow()
        OutputRow.LOSLayer = "Visible " +
            GetPositiveLOSDATA.LOS[i]
        OutputRows.InsertRow(OutputRow)
myFile.close()
```

Listing B.1: Get Positive LOS script

```
# ——————————————————————————————
# GetPositiveLOSDATA.py
# Description: This script has the variable definitions used in
# the GetPositiveLOS.py script
# Created by: Ari Kassin
# ——————————————————————————————
# Local variables
ObserverPoint = ["P0", "P0", "P0", "P0", "P0", "P1", "P1", "P1",
    "P1", "P2", "P4"]

TargetPoint = ["P1", "P2", "P3", "P4", "P5", "P2", "P3", "P4",
    "P5", "P3", "P5"]

LOS = ["LOS_P0_P1_Layer", "LOS_P0_P2_Layer", "LOS_P0_P3_Layer",
    "LOS_P0_P4_Layer", "LOS_P0_P5_Layer", "LOS_P1_P2_Layer",
    "LOS_P1_P3_Layer", "LOS_P1_P4_Layer", "LOS_P1_P5_Layer",
    "LOS_P2_P3_Layer", "LOS_P4_P5_Layer"]
```

Listing B.2: Get Positive LOS DATA script

Curriculum Vitae

Ari Kassin Fuentes was born on August 5, 1979 in México City, México. The first son of Jacobo Kassin Zaga and María Elena Fuentes Peralta, and older brother of Jana Aida Kassin Fuentes, he was raised in the Montessori Method throughout his childhood and completed Elementary School based on that philosophy. He graduated from Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) Campus Ciudad Juárez's High School in 1997. He entered Instituto Tecnológico de Ciudad Juárez (ITCJ) in the Fall of 1997 and graduated in the Fall of 2002 with a Bachelor of Science degree in Computer Science.

After graduating from ITCJ, he worked at Infolink Aplicaciones, a company in Ciudad Juárez, México, as a Web Applications Developer from late 2003 to the summer of 2005. Then he joined EDS (now HP) in the same city as a Systems Analyst and worked there from the fall of 2005 to the summer of 2006. After that, he worked for Genpact Americas in Ciudad Juárez, México as a Software Engineer from the summer of 2006 until the summer of 2007.

In the spring of 2007, he entered the Graduate School of The University of Texas at El Paso (UTEP). While pursuing a master's degree in Computer Science, he worked as a teaching assistant for one semester in the Department of Computer Science. Then he worked as a research assistant for Dr. Raed K. Aldouri in the Regional Geospatial Service Center (RGSC) for four semesters.

Permanent address: 5720 Simona Barba, Col. San Miguel Allende
Ciudad Juárez, México 32310