Use of Semantic Web Technology for Adding 3D Detail to GIS Landscape Data

Pedro Maroun Eid
Computer Science and Software Engineering
Concordia University, Montreal
pa_eid@cs.concordia.ca

Sudhir Mudur
Computer Science and Software Engineering
Concordia University, Montreal
mudur@cs.concordia.ca

ABSTRACT
GIS content is currently widely available and standardized. However, a great challenge remains in the definition of 3D GIS content. Presently, most 3D content is manually defined and refined by experts in order to result in the fidelity needed by a simulation application such as a virtual training system. In this paper, we present a new approach using semantic web technology that allows experts to formalize their knowledge in the form of an ontology which is then used to provide automated support in the definition of high fidelity 3D GIS content. This approach would allow a low-turn around between acquiring initial data from a certain area and having high fidelity 3D visualization of real world landscapes from GIS data.

Categories and Subject Descriptors
I.3.8 [**Computer Graphics**]: Applications; I.6.5 [**Simulation and Modeling**]: Model Development.

General Terms
Design, Algorithms.

Keywords
3D GIS Detail; Landscape Visualization; Semantic Web; Automated Reasoning.

1. INTRODUCTION
The National Geospatial-Intelligence Agency (NGA) produces most of the digital data (GIS content) used to create simulation representations. The NGA currently has a large collection of internally-produced geospatial data and it continues to focus on substantial data production under contract. It also is part of a large cooperative effort with other nations under the Multinational Geospatial Co-production Program (MGCP) for data collection and sharing. Detailed requirements and quality standards are defined between these organizations [Fillmore, 2006]. This allows easier sharing and reuse of the produced information. Lately, there has been a lot of emphasis on urban operations that demand much higher detail and fidelity of geographical surface features in the 3D visualization when compared to an out-of-the-window (OTW) view high up from an aircraft's perspective. In order to create these individual high detail 3D features, an expert has to study the collection of all the available GIS content and recognize the individual features and estimate their detailed parameter values. This is a very time consuming task and has to be repeated several times and reviewed until an acceptable fidelity is achieved. Some interactive tools exist to assist the expert in defining and refining the geospatial dataset. Some of these include support for feature extraction by pattern recognition in imagery. Others allow the definition of rules that generate the corresponding virtual environment based on the attributes that have been manually preset in the geographical surface feature datasets, a topic addressed to some extent by the field of scene understanding [Gahegan et al., 2002]. No matter what tools are used, there is clearly a gap between obtaining initial GIS data and having this detailed 3D data ready for generating a high fidelity visualization. Expertise appropriately supported by automation could be of great value to make the process more efficient and reliable.

Semantic Web technology [Daconta et al., 2003] is becoming a mature discipline in the definition of formal knowledge representation, reasoning and inference. Its constructs can be applied to a variety of knowledge thirsty systems in order to decrease the dependence on human experts, to consolidate knowledge and make it available in a machine readable form, and to automate otherwise lengthy processes. As a result, a broader range of individuals can help in generating the required data. The data is more easily verified, and the rule set defined can be reused in the form of an ontology. Clearly, the benefits of this technology can be extended to the 3D GIS data domain.

2. AUTOMATION IN DEFINING GIS DATA
Figure 1 shows the different GIS source data types (nodes) and current methods (arcs) used to transform GIS data types in order to generate a 3D representation of various real world objects in the concerned landscape. This GIS data flow diagram was produced after a thorough review of current systems and methodologies in use for acquiring/creating each of the GIS data types. It provides us with the proper context to compare our approach to the techniques used in scene understanding. In particular, while scene understanding techniques are concerned with data transformations denoted by dotted arrows in the graph, our semantic engine will receive the 3 inputs coming into the 3D renderer. The semantic engine will form a significant component of our 3D renderer.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
C³S²E-09 2009, May 19-21, Montreal [QC, CANADA]
Copyright 2004 ACM 978-1-60558-401-0/09/05…$5.00.

Feature extraction by pattern recognition in imagery helps the expert in automating time consuming tasks like identifying road networks, rivers and canals, building contours, and ocean shorelines. These techniques are widely used in the process of creating the feature data set. However, they still require an expert in the loop who has to correct recognition errors, connect broken lines, define the details of the detected road network, and configure a building's details to name a few. The expert uses cartography knowledge in a specific context in order to identify details about each feature element that otherwise are not visible or very hard to detect from a satellite photograph. By consolidating information from different satellite photographs, from the elevation information and from existing surface features of the area, the expert is able to define more details about the feature in question. Examples of systems that do feature extraction by using pattern recognition and scene understanding techniques are BAE SocetSet [SocetSet, 2009] or ERDAS IMAGINE [ERDAS, 2009].

In order to define the digital 3D landscape model for runtime use, the most widely used technique is the pre-generation of the terrain database by using a set of rules that generate a certain 3D element in the environment corresponding to the attributes of a certain feature element. The attributes in this case would have been manually preset in each feature element of the geographical surface feature dataset. There exist a number of commercial tools which make use of GIS data to generate a high fidelity 3D visualization. These tools require considerable amount of user input in the process of building and defining the final 3D environment. They make use of 3D model libraries along with manually input explicit attributions and rules in order to generate the final landscape model to be rendered by a corresponding 3D Renderer system that accepts their output format. Of these, the best known are Creator Terrain Studio (CTS) from Multigen-Paradigm Inc. [Presagis, 2009], TerraVista from Terrex Inc. [Presagis, 2009], and GenesisRT from Diamond Visionics Inc. [DVC, 2009].

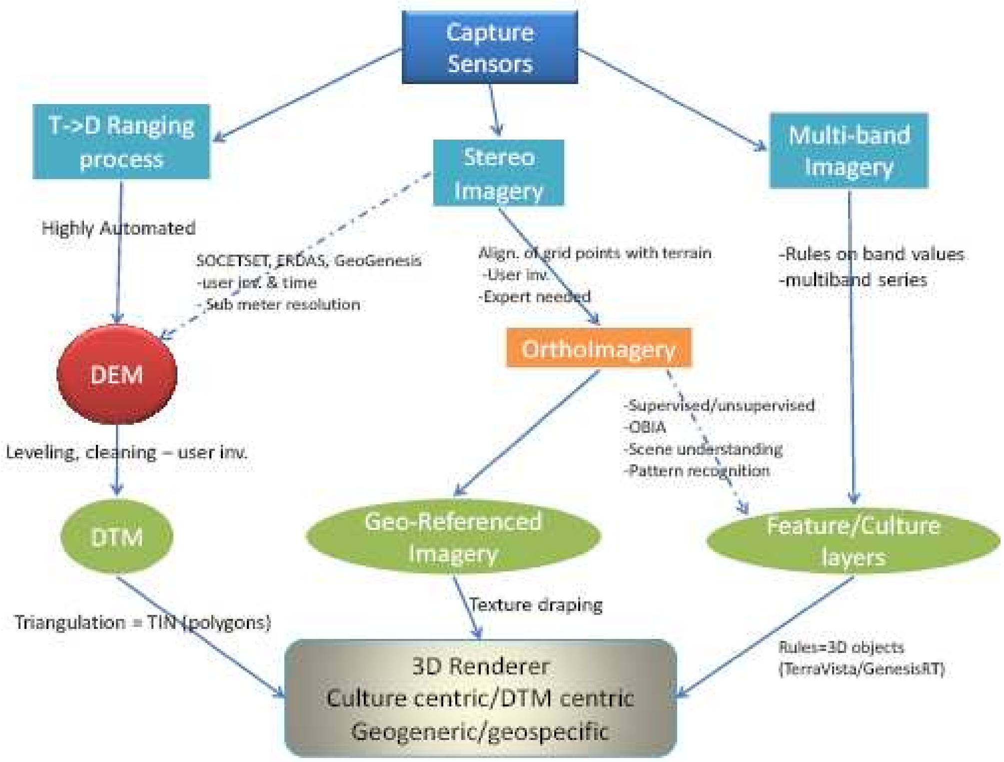

Figure 1. GIS Content Definition and Usage

TerraVista[1], for instance, provides a user interface to define procedural rules and edit feature attributes. After building the environment, the output is a model in an Open Scene Graph (OSG) format according to [Presagis, 2009], suitably modified from the standard OSG format. This is in contrast to GenesisRT's on-the-fly 3D generation where no preprocessing of the geometry needs to happen before the visualization is started. The rules are meant to describe what to do with a certain feature with specific attributes. For example, if a linear feature record has an attribute type with value "Secondary Road", the rule defines the texture and width to use along the linear object and generates the model of the road. The user has to define all rules based on feature types and attributes; rules are parsed procedurally as listed. The first rule matched will be executed in the build process and will generate the 3D model of the road. For each feature object in the source data, the rule list is parsed and if a matching rule exists, it is executed. While defining the rules, the user may edit feature record attributes to make the task easier. Users normally do so in order to reduce the number of rules to define, usually at the cost of reduced fidelity. Also, it is common for most feature source

[1] All product names used in this document are the registered trademarks of the respective owner companies.

data to be delivered without attributes, unless numerous hours are spent by experts in attributing the records properly (by feature extraction software).

On the other hand, CTS is a 3D model editor based tool that allows users to create, add and modify feature models from a model library in order to construct the 3D model of the environment [Brockway, 2002]. The output is a large OpenFlight database with referenced models and textures to create a large area environment. The user cannot define rules in CTS to automate the process of building the 3D environment; rather she/he uses common 3D model editing techniques to build the final 3D environment.

Defining rules to automate the construction of the 3D digital landscape is currently the most widely adopted technique due to the possibility of reuse of the rule sets defined and due to the correlated output of the final 3D environment elements. However, the user has to manually create the input data and explicitly attribute it in order to output the required detail. If the rule sets need to be reused, the user needs to match the attributes in the input to a standardization that would be enforced where the rule set is used.

We need a methodology that can simultaneously achieve accuracy/detail in the digital 3D landscape model and low turn-around between acquiring new GIS data and having a high fidelity 3D visualization of the landscape. We believe that this can only be achieved by automating some of the manual tasks involved in the preprocessing of the data. We shall describe next our proposed approach of adding semantics to the data and having a semantic engine define identities of GIS data objects. This would also allow the integration of different GIS data coming from distinct sources and sensors.

Semantic Web allows the definition of formal knowledge in the form of an ontology. A terminology box or TBox defines formal knowledge while an assertion box or ABox defines instances of concepts in the TBox. Both the TBox and the ABox define the domain knowledge. This knowledge can then be queried using a Semantic Web Reasoner to return results that systems can understand and interpret. Since the knowledge is formal, the information returned as a response to a query about a certain instance in the domain can be used to infer further information about that instance and that information can be used by computer programs. The TBox is normally static for any domain, while the ABox can change. The structure of the Semantic Web allows the separation of formal knowledge and instances. While formal knowledge can be reused, instances can be added, removed or modified. The ABox can be constructed procedurally based on available knowledge in the TBox provided the dataset being analyzed is tagged properly for a computer process to map data to concepts in the TBox. The Semantic Web Reasoner provides services to query, analyze and modify the TBox or the ABox.

The knowledge can be defined specific to a given context, say, city rules and urban laws for a specific area or region in our case. The defined knowledge can then be used to further automate extraction of facts in the form of instances of TBox concepts that would constitute the ABox. This in turn would identify objects and their feature attributes resulting in further automation of object and detail identification and correspondingly lower dependence on an expert cartographer.

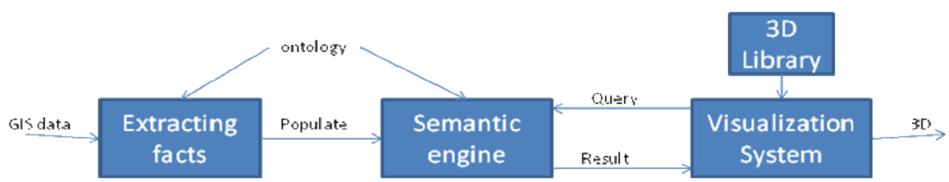

Figure 2. 3D GIS Geometry Generation Process

3. SEMANTIC ENGINE-BASED APPROACH

This section presents an overview of our approach in the format of a process (shown in Figure 2) which generates detailed 3D landscape data from GIS source data.

The first step is the extraction of facts from GIS source data (elevation data, landscape imagery and feature data sets). The following procedure can be followed:

Procedure A*: Extracting Facts*

1. *Retrieve the list of all concepts defined in the ontology (TBox)[Daconta et al., 2003]*
2. *Retrieve the list of all defined features in the features layers*
3. *Decompose the known feature object types into spatial elements using the highest level of detail available*
4. *For every available spatial element, use basic spatial tests to determine the best ontology concept that represents it*

This process is feature centric where the primary elements of the visualization are the features and the terrain is constructed around. The list of basic spatial tests possible should be available as input and region dependent. It could also be integrated in the ontology where each ontology concept class could be associated to a spatial test. As an example of step 3, if we are processing elements in a road shape file, then each segment between 2 vertices is considered a spatial element and is added as an individual road segment. If a parallel road segment exists close by, then the parallel property involving the 2 road segments is added as a fact to the ABox. The semantic engine could try to determine if the roads are part of a highway. With this process, the semantic engine has the needed information to make some facts about the spatial elements explicit using inference on the knowledge that exists in the system. The process does not execute spatial tests in the feature layers alone, if the elevation values under a specific segment/set of segments suggest a recognizable pattern, then a corresponding property can be added to the semantic engine facts. The same is applicable to imagery data, although it is less important as we assume that understanding the imagery was already done extensively in the process of creating the feature layers. We will use it however later to determine specifics about the object being drawn. For example, a bridge in real world could be built to connect 2 or more roads as the result of multiple situations: the slope of underlying terrain is too steep, a road is more expensive to build, an access to another bridge is needed, or over an inaccessible area such as water, valley, or a residential/commercial area. Each of these can be associated a spatial/imagery test and if this test succeeds, the associated property will hold in the semantic class.

After extracting the facts from the GIS feature layers, an ABox would be constructed [Daconta et al., 2003]. Before giving this ABox to the semantic engine, the semantic system needs to be initialized and verified. The semantic engine would have (1) a TBox already loaded, (2) a coherence check on the TBox would have been done, (3) the ABox is input to the semantic engine, and (4) the ABox consistency is then verified using an inference service. Finally, in the visualization step, the visualization system would use the GIS data layers to render the terrain and its texture. However, it will use the information in the ABox with all the inferred data (explicit) to render the feature objects. The following process would be used to visualize the feature objects in 3D:

***Procedure B**: Visualization*

1. *Query the semantic engine for the ABox*
2. *For each instance in the ABox*
3. *Construct a reasoner query to get the specialized class and properties.*
4. *Use the response to select the best available model that matches the concept class*
5. *Graphical details of the model are populated from the imagery, elevation and feature attributes using the object's properties, when available.*
6. *Draw the object with the available information*

When constructing the query, the visualization system needs to build it based on its graphical capabilities. For example, if the visualization system does not have a procedure to draw a bridge, but knows how to draw a road, then the query would not involve the concept bridge. Therefore, the query construction should be based on available 3D models and 3D construction procedures. An example query would be: is object at P0(W157.890, N21.334) a bridge? Using the response, the visualization system would be able to select the 3D model or rendering procedure to use. This is not only based on the concept class, but also involves the explicit properties. The properties can also be used to populate needed parameters for 3D construction procedures. The associations between concept classes and 3D models can be one-to-many. The default model is used unless further key information is available for this specific concept from the imagery or elevation values.

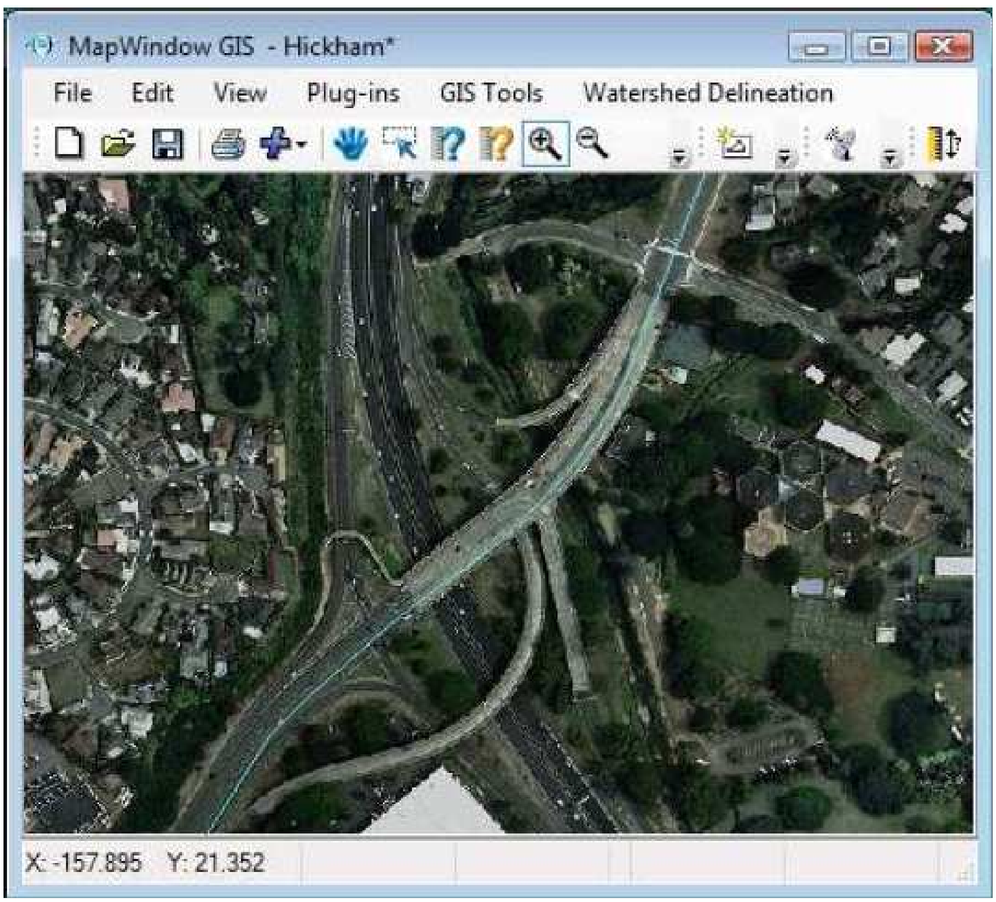

Figure 3. Hidden Overpass satellite view

4. AN EXAMPLE

This example depicts a situation where the GIS source data is not very well defined and the information a human can gather by studying the elevation data and satellite imagery can generate further details for a high fidelity visualization. Our objective is to generate the 3D object (in this case a transportation feature element) with the highest detail possible using the available GIS source data for this area. The GIS source data sets will be used to generate and visualize details that are otherwise implicit in the given data and where in a normal case, an expert cartographer must edit the GIS sources in order to include such information explicitly. We emulate the use of our approach with the goal of generating 3D detail and illustrate the various steps involved. While some of the components of this process have been implemented, we must mention here that a complete integrated semantic web based system using this approach is under development.

4.1 Problem

Figure 3 shows the 1-meter satellite imagery of an area in Hickham, Hawaii along with the correlated line feature obtained from the NGA data repository. We can clearly identify the overpass stretching from the bottom-left to the top-right of the satellite image. Figure 4 shows the same area without the satellite imagery revealing the elevation data available for this area and the linear and areal features available.

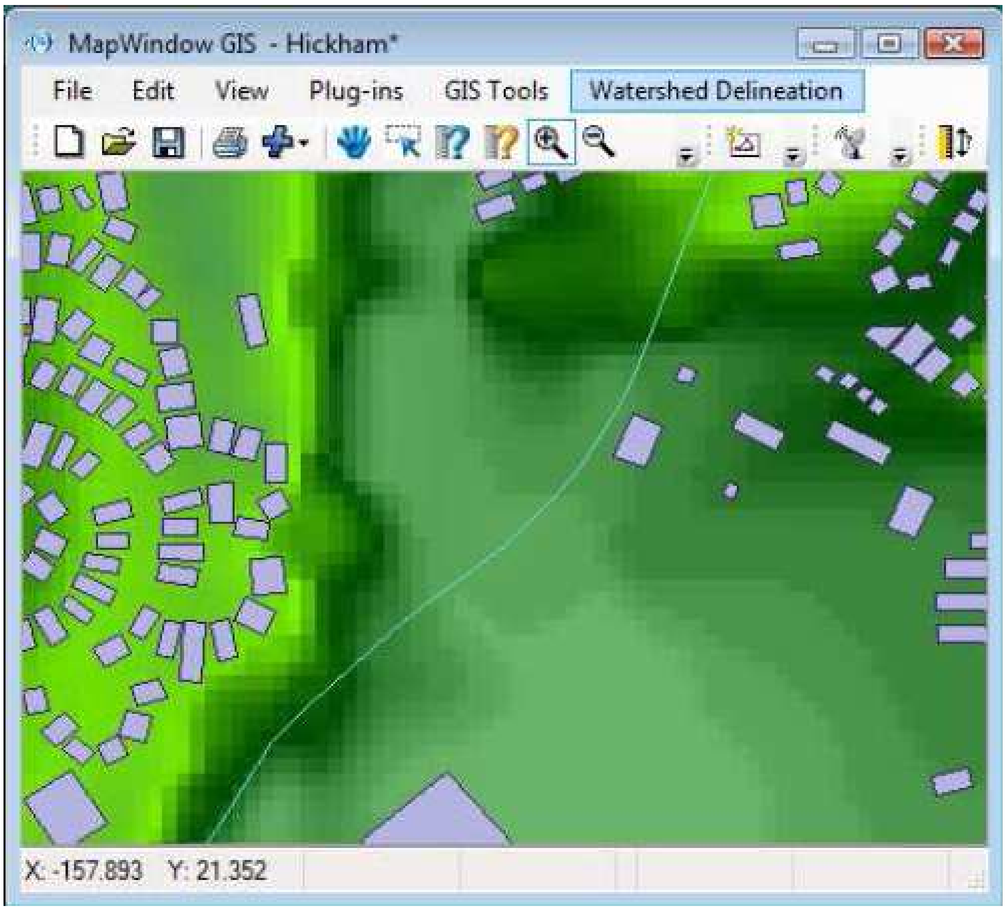

Figure 4. GIS data of Hidden Overpass

The elevation source can be viewed as the background height image and is a DTM (Digital Terrain Map) of type GridFloat. The line feature, which can be identified in Figure 4 drawn from the bottom-left crossing to the top-right of the area under consideration, represents the overpass road and is defined in the ESRI Shapefile file format [ESRI, 1998] with the filename containing the word “roads”. This line feature is poorly defined and does not contain information about its elevation over ground which is typical with feature data that has not been manually altered after a feature extraction process from the imagery. Hence, the fact that this is an overpass is not known, unless it is explicitly attributed to that line feature segment by a human.

Current real-time visualization systems would render this information without modeling a 3D object for the overpass, but by simply wrapping the generated TIN (from the DTM) using a the satellite image as a texture, and if necessary drape a road texture along the linear feature as shown in Figures 5 and 6.

Figure 5. Ground View using GenesisRT

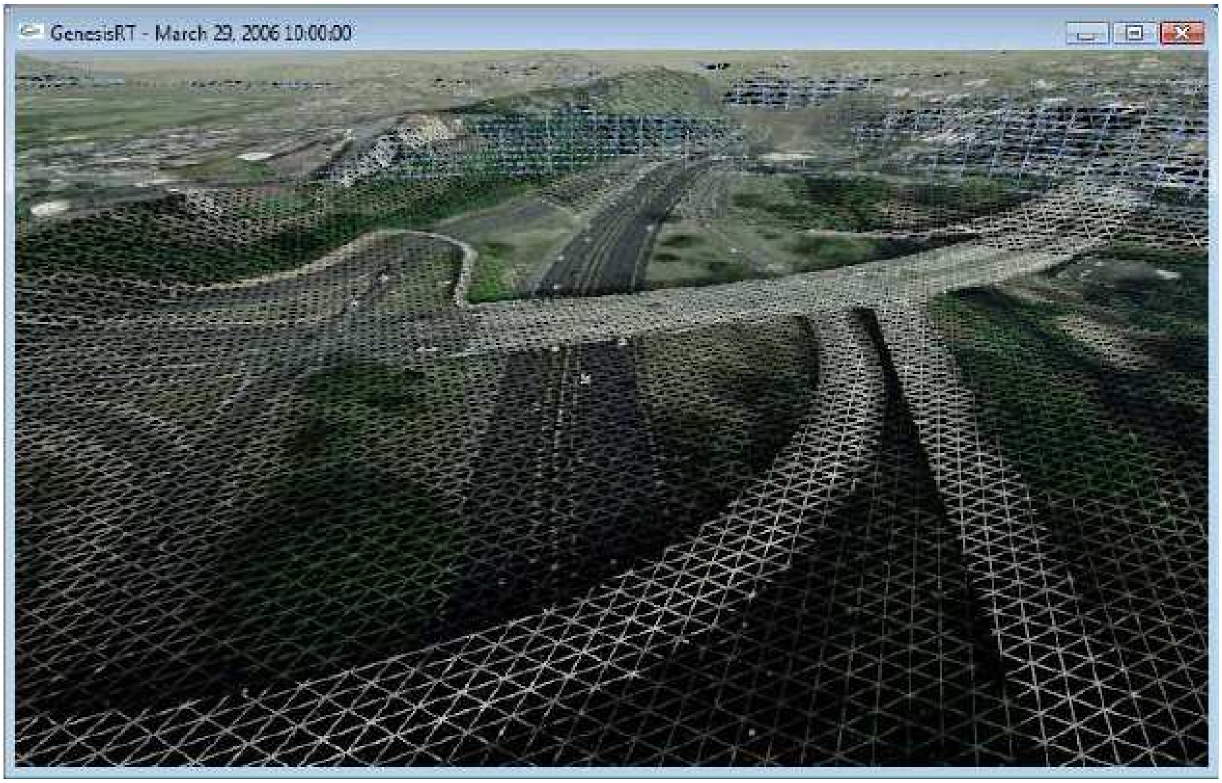

Figure 6. Aerial View using GenesisRT in wireframe

Figure 7. Ground View using GenesisRT without imagery

GenesisRT, in this case, was used to generate a 3D model of the terrain on-the-fly using the GIS data sources and some configuration files that we implemented. When the system has completely finished loading with satellite imagery as texture, the fidelity presented is the one shown in Figure 5. Without satellite imagery as texture, Figure 7 shows a generic visualization of the same GIS data from the top-right point-of-view looking towards the overpass. As can be seen, there is a simple road texture and no 3D model representing the overpass. GenesisRT has a very low turn-around with almost no delay between the time GIS data sources are initially available and the time before having a usable 3D visualization of the data.

Other systems could be used to generate the 3D model of the overpass with higher fidelity but through an offline exercise. They do offer more extensive user interfaces to manipulate how the final 3D model is generated. These systems include TerraSIM and TerraVista which are based on defining procedural rules and an extensive framework to allow the user to manually edit the feature records and their attributes. For example, a pre-built 3D model would be referenced during graphics modeling. By properly providing the input to these systems, a comparable fidelity to what we are proposing is possible at a cost: expertise and time. None of the known systems (both commercial and published) are capable of taking the source data in this example as input and outputting a high fidelity real-time 3D rendering automatically, even though a human could easily recognize this linear feature as an overpass by looking collectively at the elevation data and satellite photograph. Image understanding is not at a point where systems can successfully identify objects of this sort automatically for other systems (such as visualization systems) to use.

4.2 Solution Using Our Approach

Given the collection of GIS source data available, we make use of the elevation and imagery data in order to extract further facts about the attributes of the crossing road. The elevation data analysis points out that the area where the road crosses is impossible to pass based on the slope of the terrain thus pointing to a possibility of a bridge. On the other hand, the imagery analysis using bridge patterns along the line, helps determine the type of bridge in this area using closest image match. The resulting data is then used to define a higher detail object than simply a road texture in this area.

The same results can also be obtained through standard analysis procedures and a set of simple rules. However, semantics in this context would allow formal knowledge to be defined and edited independently from how the system works and brings all the advantages of the semantic system into the process. Semantics can be used to determine the following: since the line crosses steep elevations with bell shape type elevations under the set of line segments (refer to Figure 8), and depending on depth and leveling of the underneath elevation values of the terrain, most likely this segment is an overpass or bridge.

Semantic Web has all the necessary constructs to define formal knowledge independently of the system's methods to derive knowledge. It allows various tools for creating and editing ontologies as well as provides inference engines that provide a wide range of inference services. Based on first-order predicate logic, inference engines can derive implicit information by equation solving making it faster than conventional semantic systems such as neural networks which depends on several cycles and are based on heuristics and rules. To satisfy the Semantic Web system's needs, a TBox and an ABox need to be constructed.

A TBox is pre-constructed (generic or per-area/context) and given as input:

Classes

bridge
transport
segment
impassable_area
connected_segments (has *connected_to* property)

Properties

is_part_of(segment, transport)
is_a(bridge, over(transport, impassable_area))
over(passable_way, impassable_area)
connected_to(segment, segment)
is_a(transport, connected_segments)

We use the semantic system to help in deciding which procedure to use for a certain situation. For example, the *bridge* class has an associated elevation pattern finding algorithm (bell shape finder) that is not the case for the overpass or tunnel classes.

The Shapefile GIS source layer given as input in this example defines the line feature record as *ShapeID = 4*, *ShapeType = SHP_POLYLINE*, and *numPoints = 26*. Here is a subset of the points that we are concerned with. They are listed in the file using WGS84 WCS projection:

(-157.896959, 21.348452)	*(-157.896704, 21.348709)*
(-157.896440, 21.348956)	*(-157.896166, 21.349193)*
(-157.895860, 21.349429)	*(-157.895595, 21.349641)*
(-157.895406, 21.349792)	*(-157.895228, 21.349943)*
(-157.895063, 21.350116)	*(-157.894912, 21.350301)*
(-157.894775, 21.350497)	*(-157.894652, 21.350702)*
(-157.894546, 21.350916)	*(-157.894474, 21.351122)*
(-157.893998, 21.352351)	

The points and shape information presented have been extracted using a plugin that we implemented on the MapWindowGIS® system, an open source project initiated by Idaho State University and a group of renowned GIS researchers.

First, we need to initialize the semantic system, (1) we start the semantic engine and load the TBox defined in this example and (2) we run the ontology coherence inference service to verify the coherence of the knowledge defined in the TBox. We then load the roads Shapefile (US63840_roads.shp), the DEM 1foot resolution and the 0.5 foot resolution imagery into a defined memory structure.

Second, we extract facts, (1) we retrieve the list of all concepts defined in the ontology (using an inference service, *Taxonomy retrieval*), (2) we retrieve all defined features from the features layers and the list of all shape records in the US63840_roads Shapefile, (3) we decompose the shape record *ShapeID 4* by extracting all the point coordinates defined in this shape and assigning segments to each 2 connected coordinates and (4), for every available shape, we use basic spatial tests to determine the best concept that represents it. Our example *ShapeID 4*, using spatial test "*DEM sampling*", will be sampled at defined intervals and the output dataset will be compared with known patterns from the pattern library (associated with the ontology). The procedure returns the start and end points in the shape that have the best match for a certain spatial test with a certainty value. For example, in this case, the data returned defines *pts 2* to *13*, as *impassable_area*.

Figure 8 shows the curve returned by sampling the elevation data. It was generated using another plugin to MapWindowGIS® that extracts elevation values from the GridFloat data layer at the points listed in the shape file record. The graph is generated using a Microsoft Excel® datasheet that plots the points with elevations in meters on the y-axis and relative distance between the latitude/longitude points in meters on the x-axis. The total shape length is 2km. The solid line section of the graph represents the plot of the latitude/longitude points used in this example. It is clear that the steep slopes and bell-shaped curve over the short distance would suggest an impassable area where it is unlikely that the road is continuous over this region of 400m (between x=1400 and x=1950) and elevation differences of almost 25m.

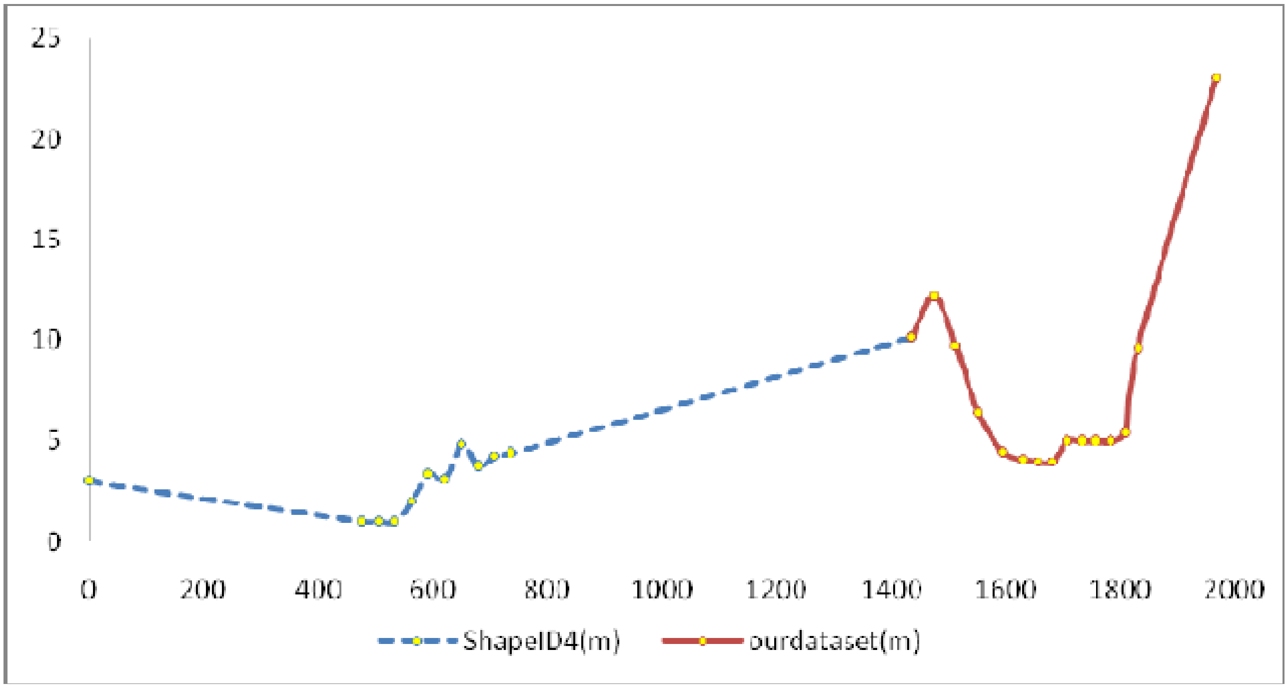

Figure 8. Our dataset elevations over distance from start point

Note that the system did not figure out that this is a bridge yet; we might discover this later due to other objects and facts in the system: e.g. overpass over another segment in another shape. If the certainty value passes a set threshold, the system would return the segments/points involved as "segments" with properties such as:

over(segment1, impassable1),
over(segment2, impassable2),
…
connected_to(segment1, segment2)

The *connected_to* property is established when there is a unique GIS vertex shared between 2 segments. Also, we only defined the line parts as *segments* over an *impassable_area* (we didn't define them as a *transport* concept class). All the above information determined through the facts extraction step constitutes the ABox of the system.

After extracting all the facts and to complete the initialization of the semantic system, (3) we dynamically construct the ABox and insert it into the semantic engine and (4) we verify the ABox consistency using an available inference service.

Third, at this stage, the 3D geometry creation system uses the semantic system for querying purposes. After (1) retrieving the ABox from the semantic engine, we (2) process every instance and (3) formulate a list of queries using the available instances, their properties and the available 3D models with semantic definitions. As an example of such query, the 3D system would construct: *segment1 is_part_of bridge*? Where *segment1* is an ABox instance, *is_part_of* is an available property, and *bridge* is an available concept linked to an available 3D model or procedure. The semantic engine could then determine that *segment1* has:

connected_to->connected_segments->transport
->over(impassable_area)->bridge

Although, in this case, it is really an overpass with no information available about the underneath visible road from the satellite photograph. The semantic system finds that segment1 actually is part of a bridge with some other segments involved. This information would be returned as a true result on our query. The 3D system would (4) use a bridge 3D model that is associated with the concept class discovered about our instance. We can have multiple 3D bridge models available for a single concept, although, for efficiency, a single concept for each possible 3D model should be defined. We use the default model unless further key information is available for this specific concept from the imagery, elevation values, or other feature records. In our case, our bridge class would have some associated pattern recognition techniques to determine covered/uncovered bridges along the segments constituting the bridge. This allows us a choice of the 3D model based on certainty which we can configure based on image texture values. Bases for the bridge can also be configured based on the underlying terrain (contours map, other feature records, etc…). An overpass has its base legs normally before and after the overpassed feature which is different from a bridge. Furthermore, (5) graphical details for the model are populated from the GIS imagery, elevation and feature attributes, when available (model texture, size, components and additional details). We can finally (6) model and draw the object with greater detail based on the information and properties collected (Figure 9), as compared to the results from current systems (Figure 7).

We believe that the above approach would result in better fidelity than the state-of-the-art tools/techniques known for dynamic 3D scene generation from GIS data sources, a specific example of which was shown earlier using GenesisRT visualization snapshot.

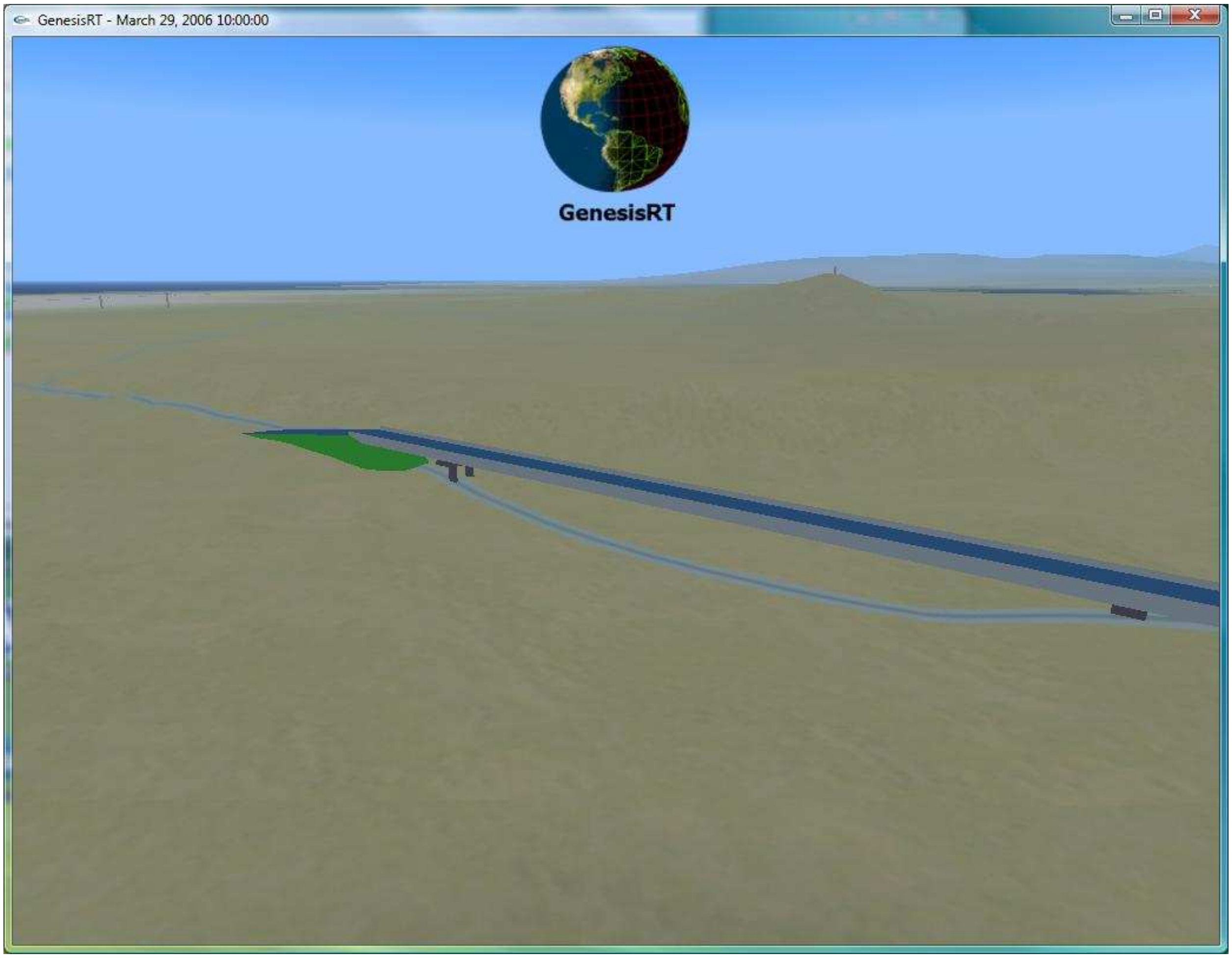

Figure 9. Ground View using GenesisRT without imagery but with 3D Detail of Bridge

5. CONCLUSION AND FUTURE WORK

In conclusion, this approach promises to automate much of the manual effort which the cartography expert spends to study a specific data set of the area in context and to modify the GIS content in order to define details of objects that would be used for visualizing a 3D landscape in high fidelity. It offers an alternative method for defining knowledge formally using existing well-established techniques and favors the reuse of this knowledge. There are many established systems that allow the editing and verification of ontologies as well as many semantic engines with various inference services. However, having a modularity between the knowledge and the semantic reasoner creates a powerful mechanism to define and automate information classification. Details and extraction of feature information is one example.

This technique extends [Bitters, 2008] in that it maintains a realistic classification of features based on other available information and does not use a probabilistic mechanism in determining non-existing details. It also extends the inferred information by providing other GIS source data collection information to the equation. It is not restricted to Shapefile features analysis. It allows the extraction of further accurate details by specific pattern matching and attribute extraction techniques. For example, techniques that can query elevation patterns, texture size, visual pattern classes, and attributes for procedural models which can be used to configure and show a surface feature in higher detail.

It is worth exploring further semantic engine services such as the ABox realization service. This service could remove the need for querying each individual surface element by the 3D engine and would instead provide a list of elements with their most specific attributes. Some references list this technique as exhaustive, but it might still be faster than querying every element by cross-referencing to the available 3D model template classes.

6. REFERENCES

Bitters, B. (December 2008). *Spatial Relationship Networks: Network Theory Applied to High-Detail Virtual Environments.* Proceedings of I/ITSEC 2008, Orlando, Florida.

Brockway, D. (2002). *Architecture for Managing Vector, Raster, and 3D Geometry in GIS*. ESRI International User Conference Paper 1068. MultiGen-Paradigm, Inc.

Daconta, M. C., Obrst, L. J., & Smith, K. T., (2003). *The Semantic Web: A Guide to the Future of XML.* Web Services and Knowledge Management. Wiley Publishing, Inc., Indianapolis.

DVC (2009). In *Diamond Visionics Corporation.* Diamond Visionics, LLC. Available from http://www.diamondvisionics.com/. Internet. Retrieved on 14 February 2009.

ERDAS (2009). In *Erdas, The Earth to Business Company*. ERDAS Inc. Available from http://www.erdas.com. Internet. Retrieved on 14 February 2009.

ESRI (1998). *ESRI Shapefile Technical Description – An ESRI White Paper*. Environmental Systems and Research Institute, Inc. ESRI, U.S.

Fillmore, R. (2006). *The MGCP is making big strides towards getting global high resolution data common across the board*, Military Geospatial Technology, 23 March 2006, V 4:1.

Gahegan, M. & Flack, J., (2002). *The Integration of Scene Understanding within a Geographic Information System: A Prototype Approach for Agricultural Applications,* Transactions in GIS 3(1) pp 31-49.

Presagis (2009). In *Products | Content Creation | Presagis.* Presagis Canada Inc. or Presagis USA Inc. Available from http://www.presagis.com/products/content_creation. Internet. Retrieved 14 February 2009.

SocetSet (2009). In *BAE Systems Geospatial eXploitation Products.* British Aerospace Ltd. Available from http://www.socetset.com/. Internet. Retrieved on 14 February 2009