Automating Extraction of 3D Detail from GIS Data using Semantic Web Technology

Pedro Maroun Eid[1]
Department of Computer Science and Software Engineering
Concordia University, Montreal
pa_eid@cs.concordia.ca

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

ABSTRACT

In many domains, low turnaround time is highly desirable between obtaining new geographical data of an area and having the information suitable for simulation systems and training. In modern warfare for example, proper tactical planning and training needed to prepare effectively for a certain mission, mandate familiarity with details of the area of operation. Most existing techniques would not achieve a high level of fidelity when rendering the area in 3D unless the GIS data is further augmented and refined by humans. For instance, given initial geographical source data layers consisting of elevations, road surface features and imagery, many techniques would only render road texture over steep terrain. Whereas, a human would immediately distinguish this as improbable by collectively looking at the data layers and note a missing element, an overpass or tunnel.

This paper describes a system which uses deductive reasoning in conjunction with specialized per-element spatial tests and applies it to the GIS data to extract, identify and classify individual spatial elements along with values for their properties. An expert cartographer's knowledge is formalized by means of an ontology. Description Logic reasoners are then used to infer information about instances revealing their true identities and to provide associated property values. In our specific example above, from the analysis of the road feature, the elevation pattern under this road, and image analysis of the specific sub-region in the imagery, the reasoner draws the correct conclusion of the existence of an overpass or tunnel and provides quantitative information needed for 3D rendering, such as location and other parameters for a procedural model. Previous semantics based research in this domain has concentrated more on improving the fidelity through the addition of artifacts like lights, signage, crosswalks, etc. Our work differs in that separating formal knowledge from data processing allows fusion of different data sources which share the same context.

ABOUT THE AUTHORS

Pedro Maroun Eid is a Ph.D. Candidate at the Department of Computer Science and Software Engineering of Concordia University, Montreal, Canada. He is investigating methodologies and processes that help users automate the process of using GIS source data to define 3D detail for use in high fidelity visualizations. He also occupies a Technical Consulting position at Presagis Canada Inc., a worldwide COTS solutions provider for the training and simulation markets.

Sudhir Mudur, Ph.D., is a professor in the Department of Computer Science and Software Engineering of Concordia University, Montreal, Canada. His areas of research include 3D graphics, virtual environments, visualization and data mining.

[1] Author for correspondence

Automating Extraction of 3D Detail from GIS Data using Semantic Web Technology

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

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

INTRODUCTION

Several organizations including the National Geospatial-Intelligence Agency (NGA) cooperate under the Multinational Geospatial Co-production Program (MGCP) to collect, produce and share digital geographic information. Despite the availability of information, and established quality standards and requirements (Fillmore, 2006), there is a need to reduce the turnaround time between obtaining new Geographical Information System (GIS) data of an area and having the information suitable for simulation systems and training applications which depend on high detail 3D model definitions for visualization and semantic attribution for shared data.

Urban operations for example, have lately been the focus of several research efforts due to the high requirement in detail and fidelity of the geographical surface features. As compared to an Out-The-Window (OTW) view of a flight simulation where the viewpoint is often high above the ground level and terrain surface and features are only represented as a texture map, urban simulation requires detailed 3D modeling of close-by objects. To produce geospecific 3D models for a certain area, an expert cartographer would need to study the collection of data available and define each feature individually by suitably estimating parameter values for 3D objects based on their appearance and other data available in the different layers within the GIS dataset. Such a task can be effort intensive requiring many reviews and repetitions so that acceptable fidelity in 3D representation is achieved. The main challenges in automating this task are (1) the need for expert knowledge in cartography to extract and estimate parameters from the various actual GIS sources and (2) the time required to process all the available information in the available dataset for the area of interest.

A few interactive tools exist which can assist the expert in defining the data and parameters according to the fidelity required. But it is very difficult for any particular tool to consolidate knowledge independent of the actual data being studied, when this data is in different forms and standards for elevation, imagery and features. In a paper by Eid and Mudur (2009) a process was briefly described which shows how an expert's knowledge can be defined by means of an ontology independent from actual data and layers. This formal knowledge can then be used by a computer program to extract facts from the different data layers and add asserted information to the knowledge base. A description logic based semantic engine would realize the knowledge base automatically resulting in instances of generic 3D entities being reclassified to their actual specialized subclasses according to the semantics they represent. The properties of instances are also inferred automatically and the collective knowledge used for visualization.

The main focus of this work is on development of a system that uses the capabilities of a semantic engine to serve a visualization system and to help in defining higher fidelity 3D models. Through queries on the realized knowledge base and with the help of an ontology of parameterized 3D models, this system can provide the data needed to define high fidelity 3D models. The system can extract quantitative parameters such as bridge width, span and cover texture from the knowledge base using standardized knowledge base querying languages. The extracted information is then used in conjunction with a parametric 3D model to procedurally construct a higher detail geospecific equivalent of the real entity defined in the collection of GIS information available.

PROBLEM DESCRIPTION

Low-turn around is desired between acquiring new GIS source data and it being ready for use by a simulation system, requiring high-fidelity visualization of this data. We first briefly understand the process of creating visualization ready data from GIS source data. Then we define a new process that could potentially lower time consuming efforts and risk in current state-of-the-art processes.

Figure 1 summarizes 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 systems and methodologies currently in use for acquiring/creating each of the GIS data types.

The current process used to create visualization ready data depends very much on a cartography expert. The human expert has to analyze every object in the area of interest using the various sensed data available and then define, attribute and model this object for 3D visualization. Many systems that create the virtual world based on given GIS input do perform certain amount of automated reasoning. For example, currently Presagis Terra Vista (Presagis, 2009) allows the user to predefine a ruleset to model exactly how the final 3D digital environment will be generated. There is a certain overhead for the user to define additional rules, if he determines it is necessary, but the resulting output is highly customized with the ruleset determining the final fidelity achievable. DVC GenesisRT (DVC, 2009), on the other hand, uses a limited set of fixed built-in rules which the user can invoke using input data. There is clearly a challenge in having both high fidelity and high automation in a single process.

In many cases, the process of creating high-fidelity 3D visualization has to be done iteratively from the analysis of sensed data to the final visual rendering step until the required fidelity is achieved. Clearly, doing this task, typically on several thousand objects in a region of interest would be highly demanding in skill, time and effort.

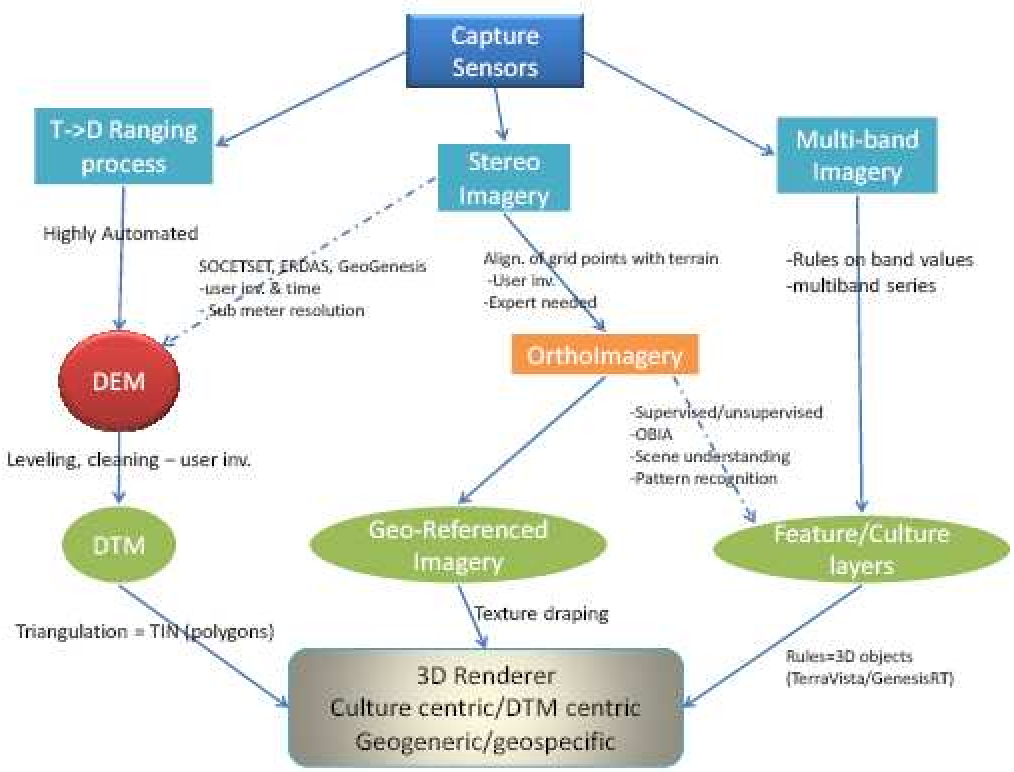

Figure 1. GIS Content Definition and Usage (after a paper by Eid and Mudur (2009))

We see this task as composed of two distinct modular steps, the facts extraction step and the spatial knowledge visualization step. Facts extraction, or feature extraction (GIS to knowledge), mechanisms are used to add knowledge to a knowledge base. This knowledge is formal and the collection of this knowledge allows for inference of new information about the existing knowledge; new information becomes explicit. Spatial knowledge visualization (knowledge to 3D) uses the available information in the knowledge base, representing an area of interest, to construct a corresponding 3D scene. Its focus is to model, in its best ability, what is implicit but described formally in the knowledge base. The task is analogous to creating the corresponding 3D world from different descriptions of the same. Two individuals might describe a "house" differently in words, but as more information becomes available, all depictions converge to the knowledge provider's version.

We use Semantic Web technology for formally modeling the knowledge base. Facts are extracted from the available GIS source data and added as assertions to the knowledge base. The semantic engine then analyzes all available facts and infers information about the available instances in the knowledge base. For example, a segment that is identified as a transport network element can be identified as a covered bridge if the elevation pattern under the segment and the texture information from the imagery along this segment are available for corroboration. The type, width, thickness, angles, and cover texture of the bridge are extracted using the inferred attributes and the scaled imagery. Using Semantic Web technology therefore allows data independence; the same process can be used on different sensor data sources (various features, elevations, and imagery sources) to extract needed data for defining high-fidelity visualizations.

A Geometry Definition Engine is used to automate the creation of 3D features using the inferred information and the values of the extracted attributes available in the knowledge base. There is no requirement for this engine to perform any analysis on its input data as that would have been already done during the knowledge base realization process. Using the SPARQL querying language, described in the following section, the semantic engine can service the Geometry Definition Engine to procedurally construct high fidelity models. This is achieved using an ontology of parametric models. The Geometry Definition Engine has needed parameters for a given model to automatically construct queries for every instance in the knowledge base and define parameter values for modeling the 3D equivalent of this instance.

The rest of this paper is organized as follows. The next section briefly reviews previous work related to our research. The process used to define the 3D features using available details in the knowledge base by query construction and execution using Semantic Web technology forms the focus of the remaining sections. A real world example is used for illustration purposes.

PREVIOUS WORK

The problem of automating the extraction of well-defined data from GIS sources has been addressed by several fields including computer simulation, earth studies, government systems and visualization. In all these fields the emphasis is on standardization of the results of the extraction process for sharing data between different systems and disciplines. Standards such as Shapefiles (ESRI, 1998) have thrived in the definition of surface feature information. However, they lack formal semantics and users can interpret a Shapefile record differently. In most cases, there is no formal semantics behind the classes and attributes used in the Shapefile record. Very few have attempted to convert available GIS information to equivalent formal knowledge that can be used as a single data collection.

Background on Semantic Web Technology

The Semantic Web (Daconta et al., 2003), or the web of data with meaning, allows the definition of formal knowledge in the form of ontologies. The knowledge can be decentralized, but still forms a satisfiable and consistent knowledge base. 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 has a corresponding formal ontology and 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.

When the ABox is realized through a reasoner service, all the instances are processed and reclassified, by entailment, to their actual specialized subclasses according to the semantics they represent. A description logic reasoner can deduce information based on formal semantics defined in the TBox such as axioms representing equivalence classes, property and role domains and characteristics and rule definitions. Description logic is a subset of first-order predicate logic which allows formal logic-based semantics, through the definition of axioms, while being more expressive than propositional logic. It is a subclass of first-order predicate logic since it defines some restrictions on binary relations, also referred to as roles, to allow further decidability.

SPARQL (McCarthy, 2005), defined as SPARQL Protocol And RDF Query Language, is a querying language which allows the query of a connected Resource Description Framework graph representing the knowledge base. It became an official W3C recommendation on 15 January 2008. The SPARQL query language is similar to the SQL query language in its syntax and allows formatting and retrieving of knowledge from the knowledge base by pattern matching. It also supports quantitative value testing as part of the query. The result of a SPARQL query is returned in the form of a result set.

Applications of Formal Knowledge

Goodwin (2005) introduces the use of semantic web knowledge bases in Ordnance Survey, Britain's national mapping agency. The ontologies at Ordnance Survey are manually developed and produced into two types: domain ontologies and data ontologies. Domain ontologies describe domain knowledge, e.g. a house is a building and has a footprint. Data ontologies formalize the structure of the data representing the elements in the domain ontology, e.g. the instance of a house would have a file type, a record format and other database entries representing it. These ontologies work together with the actual data in order to provide search queries and mapping systems with needed query results; for example, where is the closest mall next to Montreal's Town Hall? However, this does not address the need for specific parameters needed to produce high-fidelity 3D representations of feature objects.

Hummel et al. (2008) introduce the use of Semantic Web to the problem of scene understanding of urban road intersections in image sequences. The use of description logic in this context allowed a more generic approach in detecting types of complex road intersections and to infer information about the involved lanes in the intersection. This information was then used to define and predict movements and restrictions of cars. The paper approaches the problem by TBox definition and ABox dynamic construction as data becomes available from the image sensor sequences and the land surveying office map.

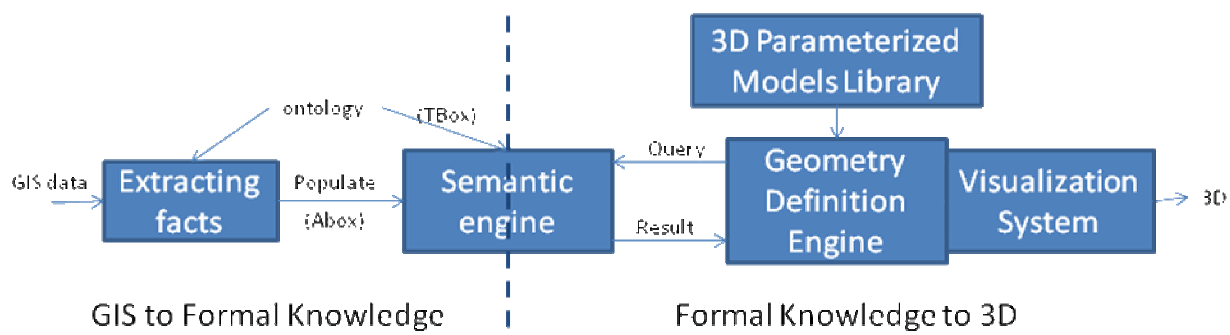

Figure 2. 3D GIS Geometry Generation Process

Our Process Using Semantic Web Technology

In a previous paper by Eid and Mudur (2009), we have formulated a process to extract facts from available GIS source data and add it as knowledge to the knowledge base system. Inference was used to make further information explicit; more than what was available in the original GIS source data. The process was demonstrated using an example and we were able to emulate the modeling of elements with higher detail when compared to the original information in the used dataset.

In this paper, this process is elaborated to include further details on how we use the knowledge base information, by query construction, to define high-fidelity 3D models. Figure 2 shows the diagram of this process extended to include the Geometry Definition Engine which defines 3D models of needed features.

INFORMATION IN THE KB

We now briefly describe the process used to populate the knowledge base. The first phase is extraction of facts as given in the procedure below.

Procedure Extracting Facts

1. *Retrieve the list of all concepts defined in the ontology (TBox)*
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*

It is important to note that Semantic Web allows decentralization and knowledge can be added by various entities to the knowledge base.

Our extraction of facts procedure is customized to identify available facts about the road network using the collection of GIS layers given. A TBox is already constructed and given to the semantic engine. The process considers that primitive pattern recognition of the road network on the imagery has already been done by tools such as SocetSet (SocetSet, 2009) and stored as Shapefile linear records. It may be noted that the pattern recognition process used in SocetSet can happen within our process itself, and is a topic extensively researched. We do not consider image based feature recognition to be within the scope of our research. Records resulting from the pattern recognition process might neither be attributed nor precise. However, if attributes are available, they are inserted as facts in the knowledge base.

Our procedure first uses the available basic spatial elements (for example, segments of the linear record) and decomposes them further into finer segments by detecting pattern changes on the underlying elevation data and imagery texture changes along the linear segment. New segments are therefore identified and inserted in the knowledge base with properties such as *connected_to(segment1, segment2)* and *over(segment2, high_slope_area)* if the elevation pattern under the segment determines high slope values. In this case, s*egment1, segment2 and high_slope_area define instances in the knowledge base.* Next, since the imagery is scaled, the corresponding imagery subset along the linear segment (the road texture) is analyzed for texture and signage and further information such as *covered* or *uncovered*, *road_width*, *lane_numbers*, *lane_type*s and *roadside_type* is added. These are pattern recognition problems and they fall under the extraction of facts process. The visualization system does not have to integrate with these procedures to be able to generate the required scene.

On completion of the extraction of facts procedure above, the ABox is realized and the determined type of *segment2* is say, *Bridge_part*. A new instance, *bridge1*, is then added to the knowledge base defining all the *connected_segments* belonging to the bridge as *part_of* the bridge entity such as *part_of(segment2, bridge1).* The extracted information is added to the knowledge base by modifying the ABox. Using reasoning, inference services available through the reasoner allows for the ABox realization where all instances are processed and classified according to their most specific subclasses (specialization). Pointers to data objects can also be added as data properties referring to pieces of data that will be used by the 3D rendering procedure. For example, the extracted bridge texture can be stored in a specific format and referred to, through a Uniform Resource Identifier (URI). The visualization system would use the URI to locate and load the texture for the bridge.

ONTOLOGY OF PARAMETERIZED MODELS

An ontology of parameterized models is needed for the system to be highly automated. The Geometry Definition Engine will use this ontology to automatically define and construct needed queries to extract information from the knowledge base. Bitters (2005) has addressed the problem of organizing generic feature models representing visual objects into a taxonomy called the Visual Objects Taxonomy and Thesaurus, VOTT. His work could be extended to create an ontology of parameterized models and the corresponding needed parameters. In this paper, we show this through an example of a simple parameter taxonomy which we constructed manually. We used Presagis Creator, a Visual Database Modeling System, to identify the generic transport, common bridge and covered bridge concept attributes based on the Bridge Wizard tool and a taxonomy was created as shown in Figure 3. Using an ontology with formal meanings for the parameters allows a system to automatically match the graphical parameter values of the parameterized models to equivalent concepts from the semantic engine's GIS sources knowledge base. Goodwin (2005) states that Britain's National Mapping Agency, Ordinance Survey, uses Semantic Web technology primarily to translate or map concepts between different organizations or domains.

Selecting a specific model type from the taxonomy is done through a generic query that returns the specialized type of instance being queried. For example, given the example in the previous section where the *bridge1* instance is inserted to the ABox as an anonymous instance along with *part_of* properties for the *Bridge_part* instances and *covered* as a *type* attribute value, the *bridge1* instance query returns *covered_bridge* as its type (made explicit by ABox realization). The Geometry Definition Engine, based on this information, selects the corresponding parameterized model from the taxonomy and constructs SPARQL queries for each of the listed parameters to retrieve this model's parameter values, whenever they are available from the knowledge base. Although most reasoners work with the knowledge as an RDF graph natively, all Semantic Web reasoners can translate the available knowledge in the knowledge base to an RDF graph. SPARQL is used to match graph patterns and return a formatted result set.

As can be seen in Figure 3, *covered_bridge* is a subclass of *bridge* and, therefore, inherits all the available properties for *bridge* and adds some properties specific to *covered_bridge.*

Generic Transport Attributes	**Bridge Attributes**	**Covered Bridge Attributes**
Start vertex position	SubClassOf: Generic Transport	SubClassOf: Bridge
Start Angle	Span Dividers	Width dividers
Start width	Deck Thickness	Cover Height
End vertex position	Starting vertical angle	Wall angle
End Angle	Ending vertical angle	Entrance angle
End width	Support width	Covered Bridge Textures
Number of segments	Support depth	
Left overhang size	Bridge Textures	
Right overhang size		
Overhang height		
Transport Textures		

Figure 3. Generic Transport, Bridge and Covered Bridge Example Taxonomy

GEOMETRY DEFINITION ENGINE

After the ABox instances are retrieved, the Geometry Definition Engine analyses every instance and forms a definition for each using the available extracted parameters. The procedure used for this is given below:

Procedure Instance Geometry Definition

1. *Retrieve the instance's most specific class type from the knowledge base*
2. *Select the corresponding type from the parameterized models ontology*
3. *Retrieve the list of all listed parameters and super-parameters needed by the selected type from the ontology*
4. *Construct a query for every parameter found to retrieve the corresponding value from the knowledge base.*
5. *Store type, parameters, and values as a Shapefile record*

First, to retrieve the instance's most specific class type from the knowledge base, a generic SPARQL query for all instances is used (Figure 4). Second, the inferred class or instance type by ABox realization is used to select a specific concept in the parameterized models ontology. The returned type using the query in Figure 4 matches a concept name in the parameterized models ontology.

```
Select ?type
Where { InstanceX rdf:type ?type }
```

Figure 4. Retrieving The Most Specific Type of InstanceX Using SPARQL

Third, using this concept name, all the attributes listed as defined in Figure 3 are retrieved from the ontology. Fourth, a SPARQL query as shown in Figure 5 is constructed for every attribute to retrieve the corresponding value from the knowledge base. %data_property% represents an attribute being queried for InstanceX.

```
Select ?value
Where { InstanceX %data_property% ?value }
```

Figure 5. Retrieving An Attribute Value Using SPARQL

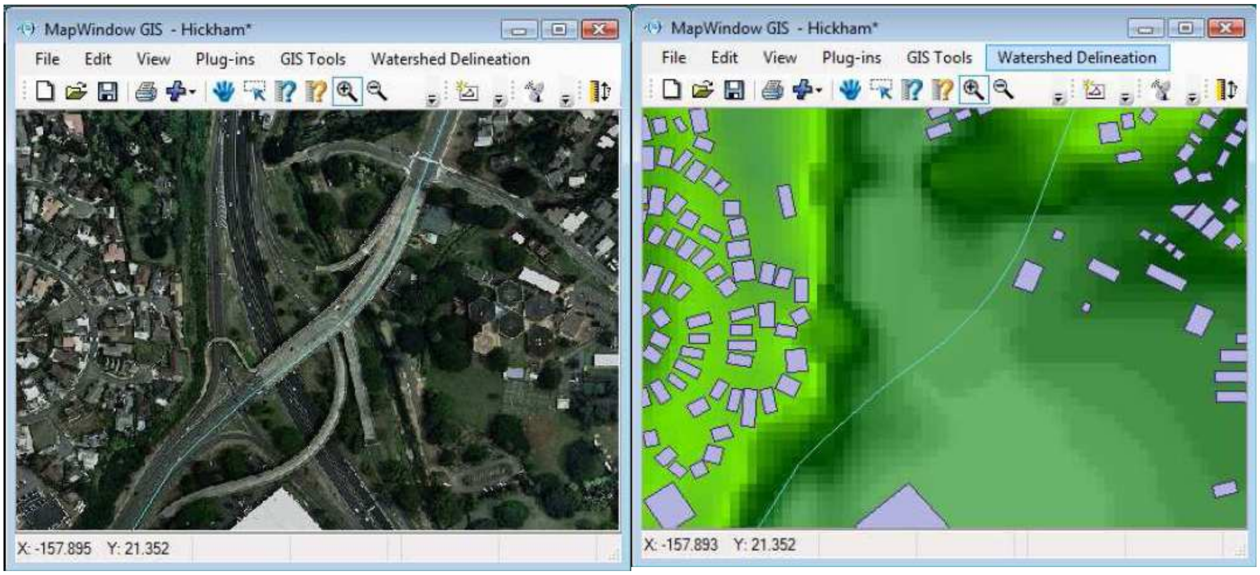

Figure 6. Satellite Imagery, Gridfloat Elevations, and Features GIS Source Data

Start vertex position	(-157.896959, 21.348452, 10m)
Start Angle	43.04 degrees
Start width	28m
End vertex position	(-157.894474, 21.351122, 23m)
End Angle	67.31 degrees
End width	28m
Number of segments	14
Left overhang size	1m
Right overhang size	1m
Overhang height	1m
Transport Textures	&localmachine;void.rgb
Span Dividers	5
Deck Thickness	2m
Starting vertical angle	3 degrees
Ending vertical angle	0 degrees
Support width	12m
Support depth	5m
Bridge Textures	&localmachine;void.rgb

Figure 7. Retrieved Values for Bridge Concept

We shall now illustrate the above process with a real world example. We take as an example the GIS source data for an area in Hickham, Hawaii. Figure 6 shows the output of the MapWindowGIS® system, an open source project initiated by Idaho State University and a group of renowned GIS researchers. The presented linear feature, crossing bottom-left to top-right, defines a set of segments representing the road (including implicitly the bridge). There is no formal explicit information that a bridge exists in this area, though looking at the satellite imagery, we can easily see that an overpass actually exists. A knowledge base is first created and realizing the ABox after application of *extracting facts* procedure yields the entailment and inference that a certain set of segments from this linear record actually define a bridge. Using the SPARQL queries described earlier in this section, values are extracted for the inferred *bridge* instance. Figure 7 shows the results returned by the queries.

Finally, in order to share the definition of the geometry generated by this process with the visualization system, we have chosen to use the Shapefile format with standardized attributes corresponding to the parameterized models ontology definition. The process creates or modifies a Shapefile with a record for every instance processed. The class type of the instance, the list of the corresponding attributes defined by the ontology and their values are stored in the Shapefile record for this instance. The attributes used will have a formal definition defined by the ontology; mapping or translating these attributes to different applications will be possible.

VISUALIZATION

As discussed before, some systems integrate knowledge extraction and logic within the 3D construction phase. Our process isolates the knowledge extraction and inference procedures into a separate coherent and semantically sound framework. It allows the visualization system to focus on visualizing formal knowledge available within the knowledge base. Using the Geometry Definition Engine, it will be easy to integrate our process with most of the visualization systems currently available. To even make the output more generic, Shapefiles are used to publish the parametric information of objects as standardized record attributes for visual objects.

For creating a high fidelity visualization in our example, we used the Bridge Wizard from Presagis Creator to construct the 3D model of the bridge after we have extracted and defined the needed parameters shown in Figure 7. This emulates how a visualization system would dynamically generate the 3D model of our bridge instance by procedural construction using the Shapefile record and the standardized attributes. The inferred type is used to select a specific procedural model algorithm (in this case the Cantilever Beam bridge parameter model algorithm). Other record attributes (refer to Figure 7) are used as parameters for the selected procedural model to generate the needed representation of the feature instance. Figure 8 shows the final result of the Bridge Wizard tool from Presagis Creator using the parameters in Figure 7 as input.

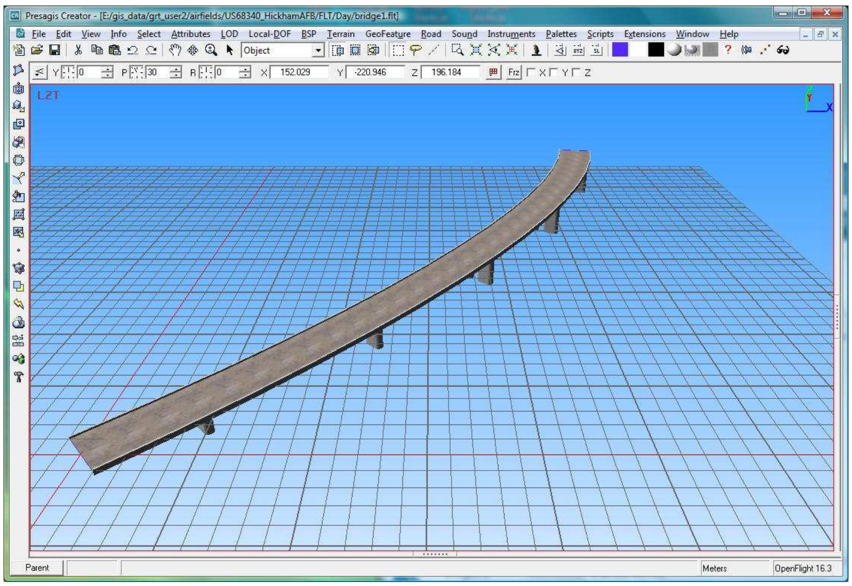

Figure 8. Generated Cantilever Beam (East View)

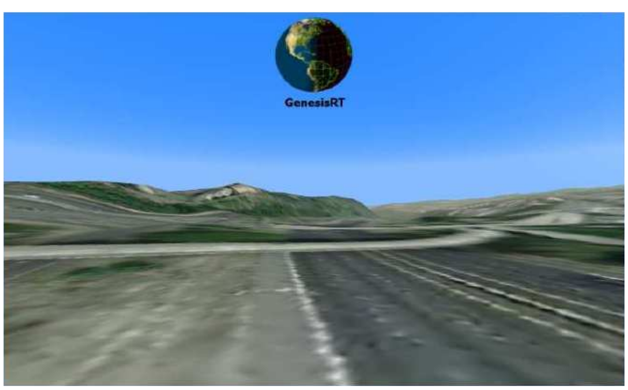

Figure 9. Ground View Visualization using DVC GenesisRT

Figure 10. Ground View Visualization using Presagis Terra Vista

Figure 11. Ground View with Generated Cantilever Beam Model (West View)

Using commercially available state-of-the-art visualization systems would ordinarily generate an environment with only a simple road texture over steep terrain as is explicit in the original GIS source data. The results of using two popular state-of-the-art systems on our example GIS dataset are shown in Figures 9 and 10, and the result of our process is shown in Figure 11. It is clear that we are able to obtain a higher fidelity 3D model for the object in question with much higher level of automation support.

CONCLUSION

In this paper, we have described the key components of a system which uses Semantic Web technology to define high fidelity features for use in 3D visualizations. Our process uses a knowledge base system which is populated by extracting facts and adding them as assertions to the knowledge base. A Semantic Web reasoner realizes the knowledge base to infer on available instances; information about instances is made explicit. Using the SPARQL querying language and an ontology of parameterized models, we have defined how a Geometry Definition Engine constructs queries and retrieves visual attribute values. Using the extracted values, a procedural construction mechanism models a high fidelity 3D representation of the feature instance for subsequent rendering. The process is automated and needs very little user involvement. It is modular where the extraction of facts step happens separately from the visualization system by adding knowledge to a knowledge base, independent of the type of the GIS data source. The knowledge can be shared among several applications and is available within a single coherent and semantically sound framework.

The results are very promising and several open topics are being considered such as reducing the facts collection phase requirements using nRQL and metrics comparing this process to the state-of-the-art processes in GIS to 3D conversion. Currently, the process assumes that the facts extraction step does all the additions to the knowledge base prior to starting with the Geometry Definition process. However, users generally define attributes on a per-need basis. Using nRQL (Haarslev et al., 2004) allows speeding up processes by triggering inference services only when needed. It only works with RacerPro, a professional scale semantic web reasoner, and avoids the need for ABox realization which could be very costly with large knowledge bases as all instances are processed with this service. Second, actual use case comparison studies are needed to compare our new system with available state-of-the-art systems that acquire GIS source data, attribute it, and transform it in order for a visualization system to model the intended information with fidelity comparable to what we are proposing. The comparison studies will allow us to gather metrics that will show how well the techniques used in this system works and would highlight the low-turnaround for creating higher fidelity visualizations intended by this system.

ACKNOWLEDGEMENTS

We would like to thank the continuous help and guidance of Dr. Volker Haarslev, Concordia University. His experiences in Machine Intelligence, and specifically the Semantic Web, add great value to this research. We would also like to thank Kevin Smith, Technical Marketing Manager at Presagis Canada Inc., for numerous experience-driven comments. We thank Bill Gerber, our birddog and paper champion for this conference, for his reviews and help. A final thanks to Diamond Visionics and especially Presagis who has provided authorization and support to use their software for experimentation and analysis in the context of this research.

REFERENCES

Bitters, B. (Spring 2005). *A Geographical Ontology of Objects in the Visible Domain*. Ph. D. Dissertation submitted to Department of Geography. University of West Florida, Florida.

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.

Eid, P., & Mudur, S.P. (2009). *Use of Semantic Web Technology for Adding 3D Detail to GIS Landscape Data.* In proceedings of the Canadian Conference on Computer Science and Software Engineering 2009 (C3S2E'09), 205-215. Montreal, Canada.

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.

Goodwin, J. (2005). *What Have Ontologies Ever Done For Us – Potential Applications at a National Mapping Agency*. OWL: Experiences and Directions workshop. Galway, Ireland.

Haarslev, V., Möller, R., & Wessel, M. (2004) *Querying the Semantic Web with RACER + nRQL*. Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL'04). Ulm, Germany.

Hummel, B., Thiemann, W., & Lulcheva, I. (2008) *Scene Understanding of Urban Road Intersections with Description Logic*. Proceedings of Dagstuhl Seminar 08091, paper 1616. Schloss Dagstuhl, Saarland University, Deutschland.

McCarthy, P. (2005) *Search RDF Data with SPARQL*. IBM DeveloperWorks white paper. Available from http://www.ibm.com/developerworks/xml/library/j-sparql/. Internet. Retrieved 09 June 2009

Mizen, H., Dolbear, C., & Hart, G. (29 November 2005) *Ontology Ontogeny: understanding how an Ontology is created and developed*. In First International Conference on GeoSpatial Semantics (GeoS 2005), 15-29. Mexico City, Mexico.

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

Wiegand, N., & Garcia, C. (17 June 2007). *A Task-Based Ontology Approach to Automate Geospatial Data Retrieval*. Transactions in GIS Vol. 11. Blackwell Publishing Ltd. 355–37