

Intelligent Agent-Based Modeling within Distributed-Spatial Decision Support System

RAMANATHAN SUGUMARAN¹ †, SHRIRAM ILAVAJHALA†† AND VIJAYAN SUGUMARAN †††

† Department of Geography, University of Northern Iowa, Cedar Falls, IA 50614-0125, USA

†† Department of Geography, University of Maryland, College Park, MD 20740, USA

††† School of Business Administration, Oakland University, Rochester, MI 48309, USA

The purpose of this research is to describe the framework for agent-based modeling within distributed Spatial Decision Support System on the Web. A system with three tier client/server architecture model was designed for the development of web-based distributed SDSS based on its function logic. The first level is user interface, offers user interface function; the second level is SDSS level, offers accomplishment of all spatial analysis function; the third level is agent level, offers experts advice. In order to demonstrate this system, a prototype was designed and implemented for planning snow removal operations by integrating real-time weather data, geospatial data, analytical tools, and Web-based technologies.

Keywords: Spatial decision support system; intelligent agent; GIS

1. Introduction

Recently, there is growing interest in using intelligent agent technology to provide decision support capabilities for solving unstructured problems with significant spatial components. Modern intelligent systems contain agents (intelligent computerized assistants) that are capable of acting autonomously, cooperatively, and collaboratively to achieve a collective goal. Intelligent agents can act on behalf of humans and assist them in executing complex tasks. They can be integrated into Knowledge-driven Web-based Spatial Decision Support System (SDSS) environments to shield the complexities of spatial modeling and analyses and help novice users tackle unstructured problems with spatial components. For example, a group of intelligent agents can cooperatively work together to solve the complex spatial optimization problem encountered in the search for the least environmental impact area in a certain region. Recently few authors showed the importance of intelligent SDSS applications (Yeh and Qiao, 1999; Sengupta, et al. 2000; Tsou, 2002; Shahiari and Tao, 2002; Sugumaran et al. 2004; Andrienko and Andrienko, 2006). Most of the available literature showed that Agent-based Spatial Decision Support mainly used for information retrieval and filter, search engine, and knowledge discovery and not many studies demonstrated the agent-based modeling within the SDSS on the Web. This paper describes the framework for agent-based solutions within distributed Spatial Decision Support System and also the implementation of an Intelligent Agent-Based Modeling within Distributed-Spatial Decision Support System on the Web.

2. System Design and Architecture

The architecture of the Intelligent Agent-Based Modeling within Distributed-Spatial Decision Support System is shown in Figure 1. A system with three tier client/server architecture model was

¹ Corresponding Author

selected for development of distributed SDSS. They are divided into three levels from its function logic. The first level is user interface on client side, offers user interface function. The client side is a basic web browser in which a user interacts with the intelligent spatial decision support system. The client side user interface was developed using JavaScript or VBScript, HTML, and applets to dynamically create "user-friendly" HTML pages that present the spatial and non-spatial information to the user. The communication between the client side and the server side is accomplished through HTTP and XML. The second level is SDSS level on server side, offers accomplishment of all spatial analysis function. The server side primarily consists of two components: Web Server and GIS Components. The Web Server facilitates the communication and transfer of data between the client side (Web browser) and the GIS components provides access to the various spatial models stored in the model base as well as the relevant spatial and non-spatial data. The third level is agent level, offers experts advice.

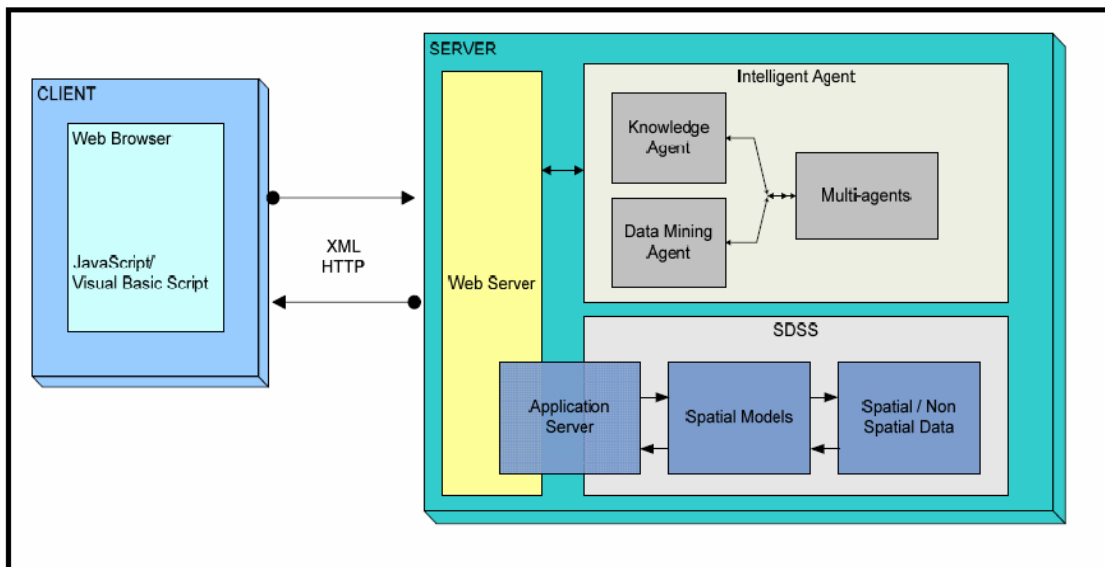


Figure 1. Architecture of Intelligent Agent-Based Modeling within Distributed-Spatial Decision Support System

The current Distributed SDSS comprises of the following intelligent agents: a) interface agent, b) task coordinator agent, c) model agent, d) web services agent, and e) data and knowledge agent. The *interface agent* provides mechanisms for the user to interact with the system. It enables the user to carryout various modeling and analysis tasks and renders the forms and reports created by the system as well as other outputs generated during data analysis and model execution. It also maintains user profiles and facilitates customization and parameterization of the tasks. The *task coordinator agent* serves as the controlling agent that takes care of the communication between the collaborating agents and coordinates the various tasks that need to be executed in order to solve the problem that the user is working on. This agent essentially drives the "what-if" scenario analyses. The *model agent* maintains meta-data about the various models that are part of the SDSS and provides interface to the model base and helps the user in model selection as well as execution. It responds to the coordinator agent by carrying out specific sub-tasks related to problem solving.

The *web services agent* maintains information about the various web services that are available and how to request for specific GIS services. It selects the appropriate web service by searching through the web services registry and takes care of the communication between the system and the web service. The *data and knowledge agent* provides access to relevant data and knowledge stored in various databases and knowledge bases that may be distributed throughout the organization.

This agent maintains meta-data about the various data and knowledge sources and how to retrieve the necessary information based on the models being utilized.

3. Prototype Implementation

In order to demonstrate this system, a prototype was designed and implemented for planning snow removal operations by integrating geo-spatial data, analytical tools, intelligent Agent and Web-based technologies (Figure 2). Most of the existing methods of snowplowing management are based on human experience and trial-and-error methods. Thus the majority of the agencies involved in snowplowing do not use any substantial analytical methods before planning snowplowing in order to reduce the over all costs and maximize the overall effectiveness of snowplowing operations. Also, there is no integrated use of weather information on the web that could alert the snow removal crews and provide a scenario-based decision support. Thus, there is a significant need for developing a system that provides intelligent decision support and a set of analytical tools for snow removal planning operations on the web, while being very user-friendly.

The system has been implemented by creating a website using ArcIMS, the most popular web-based GIS software produced by ESRI Inc. The system utilizes the ArcIMS ActiveX Object Model, XML, ArcXML, and Visual Rule Studio. The GIS software component sits on the server and provides GIS functionality to the user through a web browser. ArcIMS has got several servers such as the application server, query server, image server, and feature server that use the underlying data. These servers, in conjunction with the spatial/non-spatial data, produce and render maps through the browser. The communication between the map display and the spatial server is performed via a standardized and proprietary XML format called ArcXML. The system primarily uses Active Server Pages technology for programming the server side components. Figure 1 shows the various components of the system that has been implemented.

The ActiveX Connector of ArcIMS was chosen over the HTML and Java Connectors for its ease of use and provision for web programming with ASP, an easy-to-use server side programming language. Also, the expert knowledge system created with Visual Rule Studio can be easily integrated into the system. For reading weather data, the RSS feed technology was used to read in the XML data containing live and forecast weather information. The information thus read is automatically used to make decisions regarding resource allocation, generating weather alerts and providing suggestions for resource allocation. For efficient resource allocation, various asset management and assignment algorithms are employed and coded into the system using ASP.

4. Sample Applications

4.1 Data Visualization and User Interface

One of the main objectives of the system is to provide an easy-to-use interface that can be used even by a non-technical user to visualize the spatial and non-spatial data on the web. With this objective in mind, the system provides an uncluttered interface, divided neatly into various “areas” (Figure 3). For example the menu and the tool bar appear on top, and the map and map layers area are beneath the menu, that displays both spatial and non-spatial data to the users. The interface is mainly menu driven, making it very easy to use. All the menu options are given appropriate, non-technical names so that a naïve computer user can also utilize the system. The system generates verbose and detailed alerts, warnings, tips and other messages that help a non-technical user better understand the system. A detailed help system is available, which avoids technical jargon and provides the user an easy guide to use the system.

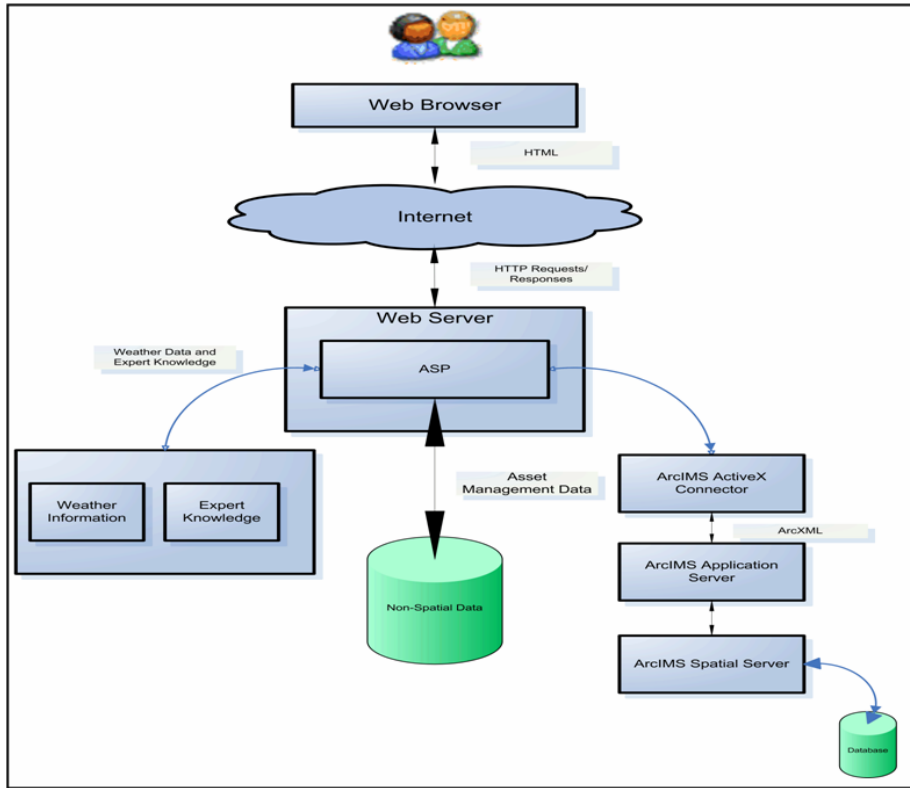


Figure 2 shows the various components of the system that has been implemented

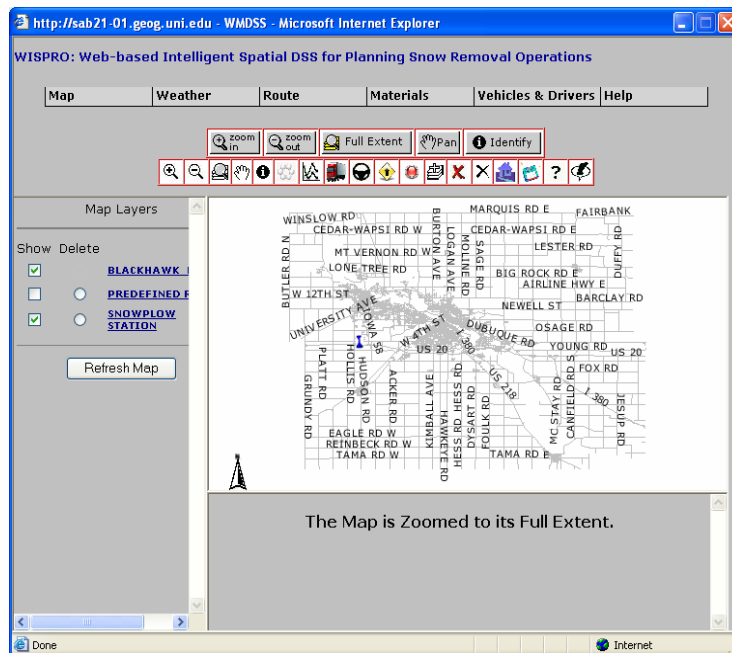


Figure 3. The Main Interface and Data Visualization Window for the System

4.2 Routing Application

The routing module essentially deals with generating routes for snowplowing. The system provides two options for routing, namely manual and automatic (Figure 4). Manual routing is the process where the user selects the route to be plowed, starting from the snowplow station, whereas automatic routing produces the maps for the user automatically. Automatic Routes are shown on the map as the predefined routes. They are the standard routes used by the Iowa DOT to snowplow, based on a priority. The manual route creation tool helps add new routes as necessary. For creating a route, expert knowledge is used to determine the estimated snowplowing time, number of runs to snowplow and the total snowplowing time. The program utilizes the expert knowledge encoded in the form of business-rules and embedded as a DLL to obtain the results. This routing application assists the novice user to make better decisions.

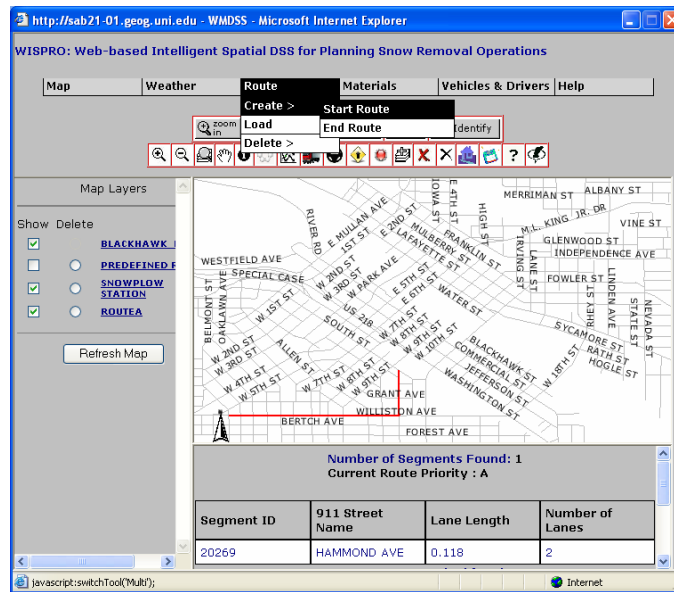


Figure 4. Process of Creating a Route Manually

4.3 Resource Allocation

Upon the creation of routes, vehicles, drivers and materials can be assigned to snowplow these routes (Figure 5). In order to assign the vehicles, the Hungarian algorithm was employed. The allocation of drivers to the vehicles is based upon a heuristic-based algorithm. Assignment of materials is based on the suggestions generated by the system based real-time weather conditions. The system provides for the allocation of material by the type of material like liquid salt, sand, etc. Material for each route is assigned based on the snowplowing time, the priority of the route, etc. Data such as the unit cost are retrieved from the materials' database and used for material assignment. For material assignments, the system uses the same assignment guidelines used by the US Department of Transportation in the form of encoded expert knowledge in the system.

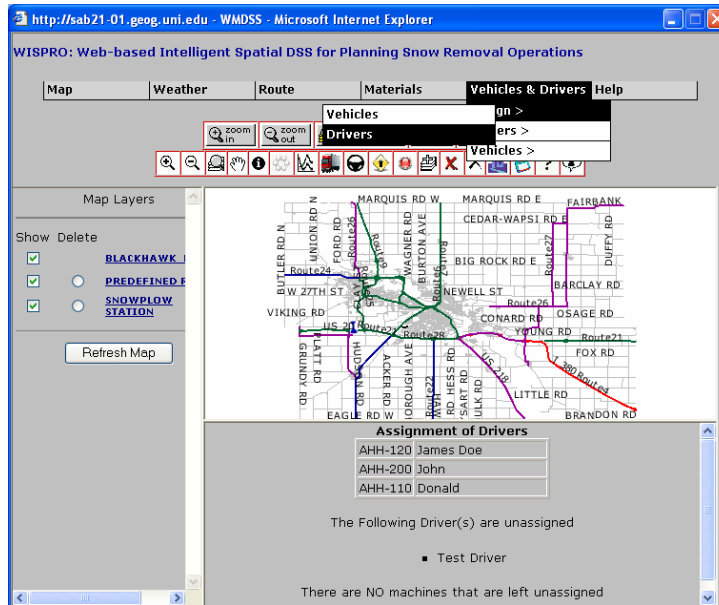


Figure 5. An example Driver Assignment using a heuristic-based algorithm

5. Lessons Learned

Developing this system has been a challenging and an interesting experience. This project provided us with many important lessons in implementing the prototype, particularly in areas of ArcIMS customization, knowledge engineering, asset management algorithms' application, real-time weather data usage, and expert systems integration with a Web-based GIS. The technology choice for the current system – ArcIMS ActiveX Connector – was made based on its availability on Windows operating platform and the wide availability and usage of related technologies like Active Server Pages and IIS. For the knowledge base, Visual Rule Studio was chosen for its tight integration with our chosen technology of Windows, IIS and ASP. Visual Rule Studio provides the ability to integrate a knowledge-base as a Windows DLL within the ASP pages on the server. The feedback mechanism chosen for this project was a modified rapid prototyping approach from software engineering, where we obtained feedback from end users each time we added new features to the prototype, and thus improved the prototype over a period of time.

There were several technical issues that we dealt with during the development of this prototype. On the server side, managing various ArcIMS virtual servers and services was challenging and a good learning experience. An important lesson learned while administering ArcIMS Servers was that an optimal balance of virtual servers can handle constant traffic to the website without hogging memory resources. Another issue was the slow response of the ArcIMS servers. This can also be solved by adding or deleting virtual servers as necessary. Further, there were some issues dealing with obtaining live weather data in the form of XML feeds. Since most of the weather feeds are provided for free, the updates are less frequent, which inhibits our ability to provide actual “real-time” weather data at all times. Finally, for extending this prototype, ArcGIS Server 9.x can be used in conjunction with ArcObjects to provide advanced functionality like adding new features (e.g. new roads) and advanced geo-analytical tools.

6. Conclusion

We have presented an architecture for agent-based solutions within distributed Spatial Decision Support System on the Web. This architecture integrates geo-spatial analytical techniques,

intelligent agent technology, and spatial decision support systems. We also demonstrated this system with a prototype for planning snow removal operations. The system prioritizes the snowplowing routes using intelligent agent technology and allows decision makers to manage the allocation of snow removal assets and resources. The use of intelligent agent in our system reduces the cognitive load on the user and provides a user-friendly interface even for a naïve user.

REFERENCES

Andrienko, N., Andrienko, G. (2006). Intelligent visualization and information presentation for civil crisis management. *9th AGILE conference on geographic information science*, Visegrad, Hungary.

Ming-Hsiang Tsou. (2002). Adopting Software Agents for Internet Mapping and Distributed GIServices. ESRI International User Conference 2002, paper session, 234.

Shahiari, N. and C.V. Tao. 2002. GIS Applications using Agent Technology. *Symposium on Geospatial Theory, Processing and Applications*. Ottawa, Canada.

Sengupta, R., Bennett, A. D., & Armstrong, P.M. (2000). Agent-Oriented Modeling environment for Spatial Decision Support. First International Conference on Geographic Information Science.

Sugumaran, R., Sugumaran, V., Salim, M.D., & Villavicencio, A. (2004). An Agent-Based Spatial Decision Support System for Snow Removal Asset Management: A Web-Based Approach. Americas Conference on Information Systems.

Anthony Gar-On Yeh, and Jiming Qiao, 1999. Intelligent Solution Support System for Spatial Modelling and Decision Support. Thirty-second Annual Hawaii International Conference on System Sciences-Volume 6 p. 6016.