ON GEOSPATIAL AGENTS

Merik Meriste, Tõnis Kelder, Jüri Helekivi, Andres Marandi
Tartu University Institute of Technology , 51014 Tartu, Estonia

Leo Motus
Tallinn Technical University, 19086 Tallinn, Estonia

Keywords: geospatial technology, geospatial agent, multi-agent system, agent-based computing

Abstract: As access to spatial and real-time data improves, the need for appropriate software tools has become prevalent. To develop geospatial applications in this context requires an approach to software architecture that helps developers evolve their solutions in flexible ways. Two concepts are today considered reasonable here – web services and agents. This paper presents generic geospatial agents in a prototype of agent development environment KRATT. Pilot applications are described and experience discussed.

1 INTRODUCTION

As access to spatial and real-time data improves, the need for appropriate software tools has become more and more prevalent. To meet the demands of such systems one needs tools for developing systems of autonomous, interactive software entities. The Open Geospatial Consortium (McKee 2003a, 2003b) tackles the challenge of developing tools to support the integration of spatial capabilities into applications than don't depend on a full-featured Geographic Information Systems (GIS). The main problem is - embedding of location-aware and time-aware components into geospatial applications.

The current practice of agent-based systems (Bigus, 2002) is mostly focused on agents' intelligence related issues. The problem of multi-agents' organisation and engineering has received less attention (Odell et alii, 2003). In principle, *an active object with full control over its state* forms a pragmatic basis for agent's implementation. Two concepts are today considered reasonable to how to address this – web services and agents. This paper presents implementation of generic geospatial agents in an agent development environment KRATT (Motus, 2004). Basics of KRATT agents as well as conceptual line of development are described in section 2. Basic collection of geospatial agents is introduced in section 3. Pilot applications are superficially considered in section 4.

2 AGENT'S STRUCTURE

Agents exist and interact in a computer system that is distributed across a set of, not necessarily homogeneous, networks. The agents can exist completely in a virtual world – interacting only with the other agents, or also interacting with non-agent components of the system. The prototype development is carried out in C# and .NET since this platform is suitable for controlling multiple threads needed in agents. An application as an agent system consists of administrative agents, and application agents. All the agents are generated from pre-specified classes. Classes form the namespace *AgentComponents* written in C#.

An agent is implemented as (see Figure 1) an instance of a class Agent in the namespace AgentComponents. This base class determines functionalities and lists related components that are to be applied to generate an instance of a location- and time-aware agent. As a rule, a multi-agent system comprises several agents; each agent is implemented as Windows application (WinA).

Basic components of class Agent are:

(1) Communicator, exchanging (time-stamped) messages with the other agents;
(2) Manager, managing control of agent's ;
(3) Actor, performing the functional tasks;
(4) Monitor, monitoring specified aspects and time-stamped events in the agent's behaviour.

Meriste M., Kelder T., Helekivi J., Marandi A. and Motus L. (2005).
ON GEOSPATIAL AGENTS.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 210-213
DOI: 10.5220/0001230402100213

Each component is described by a class in the namespace AgentComponents. In the most cases the Communicator is standardised, whereas the other components are different in different applications.

Majority of agents, i.e. application agents carry out their autonomous (maybe) proactive tasks. When specialising an application agents it is necessary to specialise its components Communicator, Manager, Actor, and Monitor. In the same WinA are constructed all components of the agent.

Administrative agents are specific in a sense that they store and execute the rules, and provide common services, required for expected normal operation of application agents: Basic types of administrative agents are:

(1) *AMS Agent* - management of application agents.
(2) *Proxy Agent*s - transferring messages to and from agents that reside behind a firewall.
(3) *Monitor Agent* - collecting and processing activity reports of application agents.

Proxy and *Monitor* agents have also to register themselves with the corresponding agents *AMS* type.

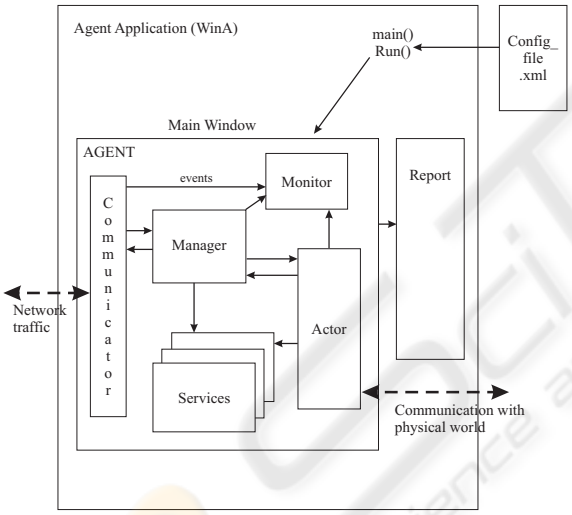

Figure 1: Generic description of an agent.

3 GEOSPATIAL AGENTS

Agent based digital map enables a customer to use in his/her computer a variety of digital map based applications. The customer's computer must have a web browser, and sufficient memory space for automatically downloaded active map object together with some other active objects, necessary to solve the particular task stated by the customer. An application is configured dynamically, depending on which agents are active and accessible at this particular moment. Map agents can display different areas from a variety of digital maps (raster or vector based maps, maps with different colour schemes, etc). Reasonably short response time of the agent system is achieved due to carefully designed network traffic, combined with caching and parallel processing of source maps and databases in the servers. Agents, and the applications that use services provided by agents are not in one-to-one relation; one agent can simultaneously work with many applications. Also, an application may use services from different agents in different situations.

The map-specific part of the web browser's page in a customer's computer is *map_object* (presented as ActiveX component). The other parts of the page depend on the specific requirements of the application and cater for dynamic support of the page, and for interactions between the *map_object*, the customer, and web-servers. To display the requested area of a map the *map_object* requests the necessary parts of the map from appropriate agents, the request may also contain additional details and/or constraints. Typically the whole page or parts of it are generated by a web-server based application (see Figure 2). The main operational response and support to the queries from the *map_object* and the web-server application comes from a dedicated multi-agent system.

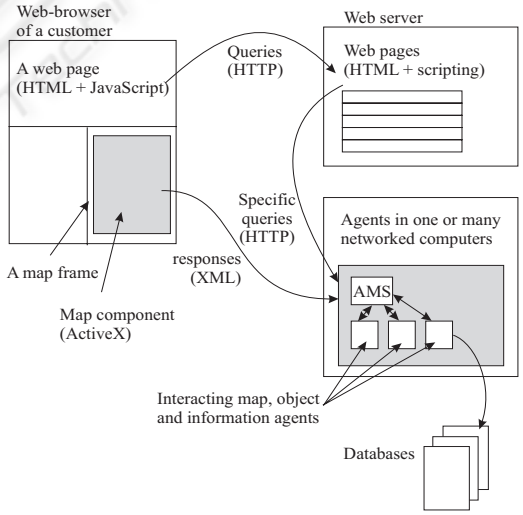

Figure 2

The community of basic geospatial agents forms the kernel of any particular agent-based digital map. The collection of agents is easily extendable depending on tasks required by the customer, types of digital base-maps and additional databases to be used, etc. Agents have generic features that facilitate their application in a variety of applications with minimal modifications.

For pilot application there are implemented following classes of generic agents:

(1) Map agents transmit fragments of the map, an agent can offer one or more raster base-maps.
(2) Vector agents transmit vector layers or fragments of layers, if possible.
(3) Agent that transmit additional information in one of the two possible formats – XML- record or a picture, sound- or video file

The requirements on additional information are obviously application-specific. Sample cases are listed in the following in order to illustrate the approximate character of the additional information:

(a) information regarding a graphical primitive,
(b) information on a certain location in a map
(c) combination of the two previous cases – response to the queries for additional information may contain several large volume objects (pictures, sound- or video files); the customer can later choose suitable information.

4 EXPERIENCE

4.1 Participatory GIS

Participatory GIS for regional planning and management was tested as an application example. The aim of the application created for Tartu City is to view and update (as a dynamic collection of spatial and information frames integrated into one application) the information distributed in databases of different owners.

The task came from the city government, who needs the information of different infrastructures (gas-, water-, sewage-, heating pipelines, power lines and phone lines) for planning activities every day. In reality, the infrastructure is maintained by different organizations and the information, therefore, is also kept in different GIS and databases.

The prototype of the participatory GIS for the tasks described above is implemented as a federation of agents. This agent-based system provides the updated information from different GIS and DB as well as represents and manages the information supplied by agents involved.

User can select one main theme and compare it with different subthemes. The number of base maps (provided by map agents) and different data categories (provided by vector agents) used in application depend on maps and data which are available for users. As a result, users can have overview of a set of infrastructure of the area they are interested, e.g. the location of pipelines, availability of gas line or central heating system etc. First experience shows that dynamic compilation of frames intermediated by agents, offers reasonable processing speed, better than that of a typical fixed menu of web services of a traditional GIS.

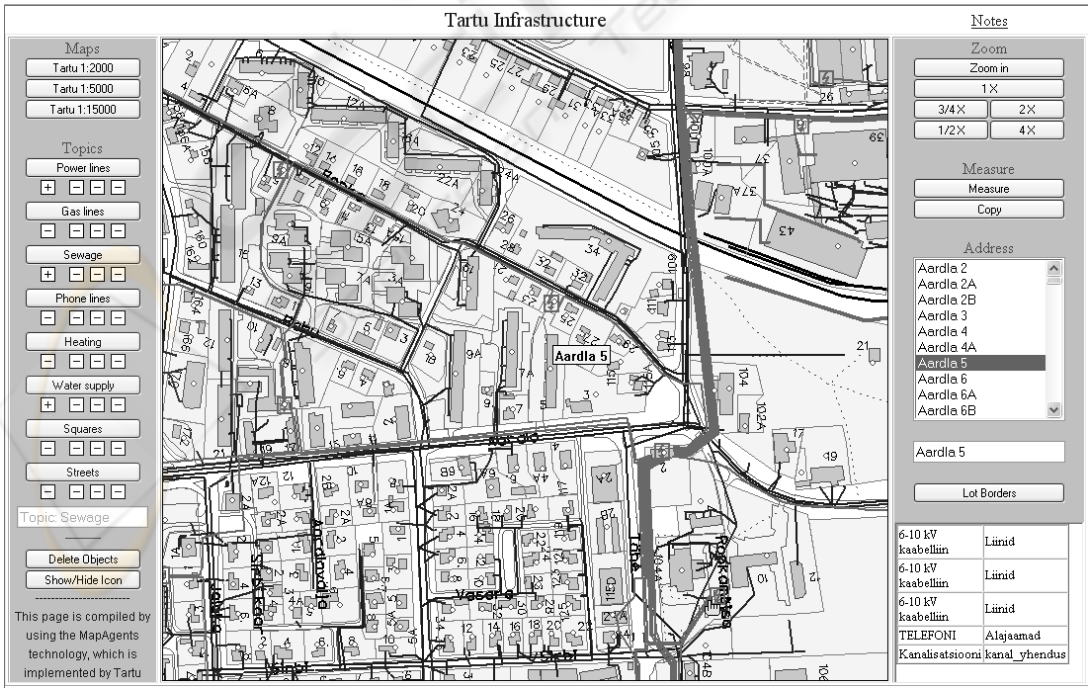

Figure 3: Participatory GIS.

4.2 Tracking active objects

To include active or mobile objects in the framework of geospatial agents, a prototype for tracking vehicles was developed. As usually, a GPS module, which is attached to moving object, receives the signal from satellites and transmits the coordinates with the unique ID of object to operator agent. The agents provide the coordinate info of particular object to map application. The map application is built up as any other agent-based application and active object acts as additional agent.

The tracking multi-agent is a federation of agents, consists agents of following types:

- agents processing fragments of the map;
- agents for searching, processing, and sending the vector information for the image or situation that has been activated in the *map_object*
- GPS agents, that in pair with periodically transmit to other agents the position of active objects equipped with standard GPS device.
- agents that manage, search, and forward additional information related to active objects.

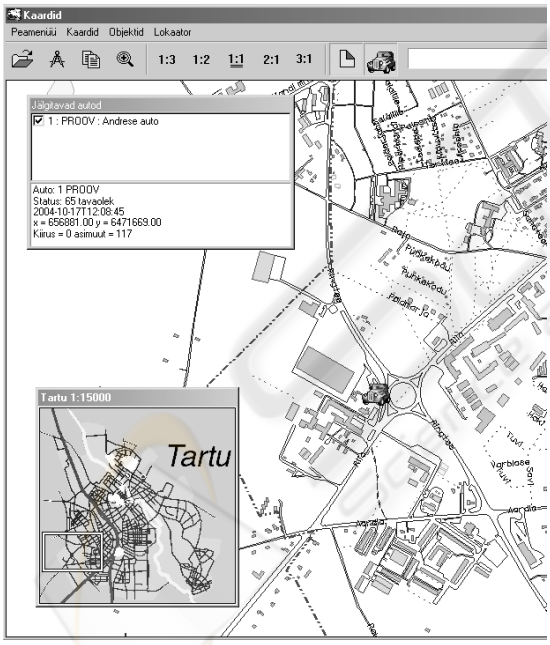

Figure 4: Tracking active objects.

5 CONCLUSIONS

The developed collection of agent classes forms the basis of agent-based digital maps. The collection is extendable and depends on tasks required by the customer, types of digital base-maps, additional databases to be used, and on other factors. Each agent is to be programmed as required by the specific application. Agents have generic features that facilitate their usage in a variety of applications with minimal modifications. The experience shows that dynamic compilation of agents' intermediated map frames offers a reasonable processing speed, better than that of a typical fixed menu of web services of a traditional GIS.

In contrast to the traditional approach in GIS, the geospatial agents empower developers and end-users to build their custom maps independently and post useful information for their clients. Advantages include the shift of ownership from digital map and GIS suppliers to the users and the opportunity for the users to build their own dynamic and interactive maps. This opens a venue for developing many more and tailor-made applications.

The described agent-based solution provides the basic ideology that can be extended for application on mobile & thin platforms. Further experiments and the development of the instrumental software and classes of geospatial agents are planned, in pair with carefully selected geospatial application areas.

ACKNOWLEDGMENTS

This research has been partially financed by Estonian Science Foundation grant no. 4860, and by grants no. 014 2509s03 and no. 018 2565s03 from the Estonian Ministry of Education and Research. This support is gratefully acknowledged.

REFERENCES

Bigus, J.P., Schlosnagle, D.A., Pilgrim, J.R., Mills, W.N., Diao, Y., 2002. ABLE: A toolkit for building multi-agent autonomic systems. *IBM Systems Journal*, vol. 41, no. 3, 350-371.

McKee, L., 2003a. The Importance of Going "Open". *Open Geospatial Consortium*, White paper, 8 p. www.opengis.org

McKee, L., 2003b. The Spatial Web. *Open Geospatial Consortium*, White paper, 8 p. www.opengis.org

Motus, L., Meriste, M., Kelder, T, Helekivi, J., 2004. Agent-based Templates for Implementing Proactive Real-time Systems. Proc. Int. Conf. on Computing, Communications and Control Technologies, Austin, Texas, ISBN: 980-6560-17-5, 199-204.

Odell, J., Parunak, H., Fleischer, M., 2003. Modeling agent organizations using roles. *Software and Systems Modeling.* Springer, vol. 2, 76-81.