Chapter 14

An Intelligent Software Agent Architecture for Distributed Cartographic Knowledge Bases and Internet Mapping Services

Ming-Hsiang Tsou, Department of Geography, San Diego State University, CA USA

Abstract

The Internet can provide interactive display and multimedia functions for digital maps and remotely sensed imagery. One major problem for the development of Internet mapping facilities is information overload. It is a challenge for the cartographic community to make the power of Internet mapping accessible to users, but at the same time to help users adapt cartographic concepts and rules to their web mapping applications. This chapter will introduce a possible solution by adopting software agents in the architecture of Internet mapping facilities. In contrast to a traditional expert systems approach, the use of software agents emphasizes that their knowledge bases are located in hundreds of distributed small agent programs instead of a single huge omnipotent computer machine. The design of an intelligent software agent could facilitate the establishment of distributed cartography knowledge bases (CKB), which could help map users to access/distribute/exchange different cartographic rules, map symbols, color schemes, design layouts, via the Internet.

1 Introduction

Internet mapping is one of the newest Internet applications in support of the access and dissemination of geospatial information across computer networks. Since the Internet can provide interactive display and multimedia functions for digital maps and remotely sensed imagery, Internet-based mapping facilities will become the major components for the next generation of geographic information systems (GIS), including web-based geographic information services (GIServices) and wireless-based mobile GIS. In the future, the popular use of Internet mapping facilities will allow users to access real-time geospatial data from anywhere at anytime. Geospatial information will be available at users' fingertips by utilizing wireless communications and mobile devices, such as

Personal Digital Assistants (PDA), Pocket PCs, and smart cellular phones. Internet mapping facilities will provide fundamental mapping services for these advanced GIS applications.

Currently, thousands of web-based map servers are available on the Internet and hundreds of different GIS commercial packages and technologies are being adopted for various Internet mapping applications, including real estate, natural habitation management, urban planning, traffic monitoring and controls, and emergency rescue. One of the major challenges for the development of Internet mapping is the problem of information overload. This chapter will introduce an intelligent framework for Internet mapping applications by using software agents to help users in information searching, automatic data conversion, and on-line mapping. Hopefully, the design of an intelligent agent-based architecture will facilitate the development of Internet mapping applications from ad-hoc, technology-centered projects to sustainable, cartography-oriented developments.

Many recent research projects and studies focus on the development of Internet mapping and distributed GIServices (OGC, 1998; Goodchild et. al., 1999; ISO/TC211, 2000). The development of modern cartography includes three different stages: paper maps, GIS applications, and Internet mapping. The first stage of modern cartography focused on the design of static paper maps and communication functions. Paper maps are generated by predefined rules of information representation, such as symbolization, generalization, projection, and color schemes. Along with the progress of GIS, the second stage of modern cartography focused on the use of GIS packages and visualization tools. Electronic maps and GIS applications are generated by computers in digital format, which can provide interactive, multimedia map display and dynamic spatial query functions.

The third stage of modern cartography is Internet mapping. Internet mapping services can provide ubiquitous, flexible and real-time access to geospatial information and dynamic map representation. The targeted users of Internet mapping are highly diverse and different from GIS users. Most Internet mapping users may lack sufficient cartographic training to manage or interpret the dynamic representation of geospatial

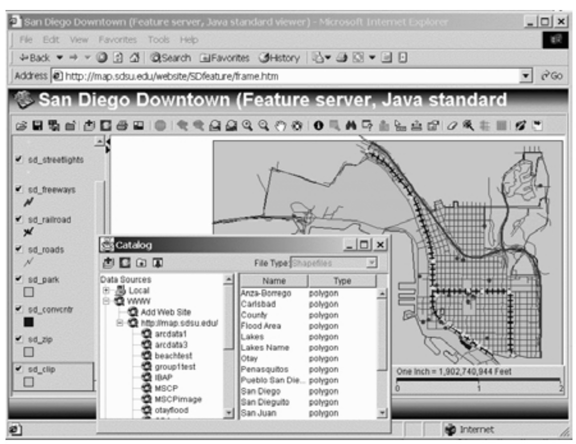


Figure 1. The multiple access and powerful tools of Internet mapping applications.

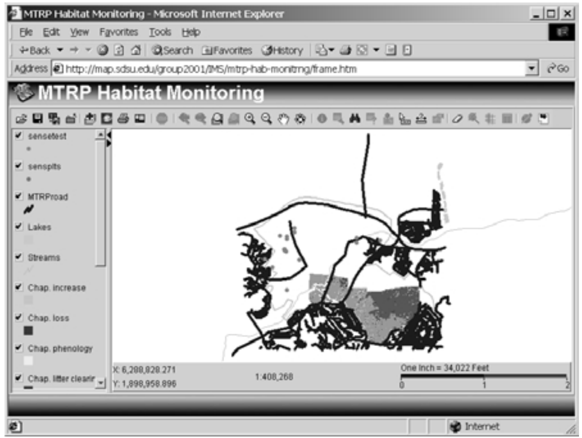


Figure 2. Poorly designed and misrepresented Internet mapping application.

information. For example, Figure 1 illustrates how an Internet map user can access hundreds of map layers from different Internet map servers at the same time. However, users may not know essential procedures for combining different layers, changing map symbols, applying color schemes, setting display thresholds, or modifying map projections. Without sufficient cartographic knowledge and training, the Internet map may be poorly designed and misrepresented (Figure 2).

It is a challenge for the cartographic community to make the power of Internet mapping accessible to users, but at the same time to help users adapt cartographic concepts to their web mapping applications. Many cartographers have already noticed this problem. Kraak indicated in his book, *Web Cartography*, that "The WWW will certainly have a great positive impact on the cartographic discipline. However, the quality of the maps produced will be, at first, relatively poor, but will improve in the near future." (Kraak, 2001, p. 11).

This chapter will introduce a possible solution by adopting software agents in the architecture of Internet mapping facilities. The design of an intelligent software agent has three main goals:

1. Software agents have the ability to search, carry, and apply cartographic rules for web mapping applications;
2. The agent-based architecture will provide a dynamic framework to combine different cartographic rules for different mapping tasks; and
3. The collaborations between software agents will facilitate the establishment of distributed cartography knowledge bases (CKB), which can help map users to access/distribute/exchange different cartographic rules, map symbols, color schemes, design layouts, via the Internet.

2 Artificial Intelligence Research in Cartography

Software agents developed from the research of artificial intelligence and distributed computing (Bigus and Bigus, 1998). This section will introduce the development of

Artificial Intelligence (A. I.) particularly in the domain of Cartography in the past few decades. The earliest AI approach in cartography can be traced back to the development of expert systems in the 1970s (Openshaw and Openshaw, 1997). Expert systems are computer program that manipulate symbolic knowledge and heuristics to simulate human experts in solving real-world problems (Weiss and Kulikowski, 1984; Shea, 1991). In the 1980s, many cartographers tried to develop expert systems for various mapping tasks, including automated point label placement (Christensen, et. al, 1995; Doddi et. al., 1997), automatic generalization (Buttenfield and McMaster, 1991), and map label conflict detection (Freeman and Ahn, 1984).

There are four major elements in a traditional cartographic expert system: a knowledge base (rules), an inference engine, an object database (facts), and an input/output user interface (Shea, 1991). The knowledge base includes cartographic rules and logics for different mapping tasks and guidelines. For example, "the color of contours in a topographic map is brown," is a rule that can be encoded inside an expert system. The inference engine provides the control mechanism to direct the reasoning processes. The object database stores the actual entities for mapping tasks, including map layers, label names, and geometry features. The user interface provides communication channels between computers and users. Figure 3 (left) illustrated the four basic components of a traditional expert system.

More recently, cartographers have explored other types of AI research, such as fuzzy logics, machine learning, and software agents. Many research projects focus on the abstraction and representation of cartographic knowledge by using AI or object-oriented modeling techniques (Zhan and Buttenfield, 1995). For example, by adopting a machine learning approach, computers can automatically build some cartographic rules from a set of map examples given by an expert rather than the rules defined directly by a cartographer (Zucker, 2000). The development of software agents is another research direction in the AI community. The AGENT (Automated Generalization New Technology) project, developed by Lamy and Weibel (Lamy et al. 1999) in the late 90s demonstrates the great potentials of agent based methodologies in providing solutions in autonomous map generalization. However, only a few research projects focus on the actual implementation of cartographic software agents currently. The following section will introduce software agents and focus on the comparison between expert systems and software agents.

3 Software Agents

The goal of software agents is to reduce user work and information overload (Maes, 1994). A software agent is a software entity which functions continuously and autonomously in a particular environment (Shoham, 1997, Brenner et. al., 1998). Each software agent has specific functions and responds to specific events, based on pre-defined knowledge rules, the collaborations of other agents, or users' instructions. Software agents can help users search information, interpret or translate different data formats and cartographic rules, and make logical decisions or suggestions. For example, Web search engines like Excite, Lycos, and AltaVista are using software agents to help users search the information on the Web. Current research suggests that intelligent software agents will be widely used and implemented in the future, especially in open, distributed systems (Bradshaw, 1997).

Different from the traditional expert systems approach, the use of software agents emphasizes that their knowledge bases are located in hundreds of distributed small agent programs instead of a single huge omnipotent computer machine. For a long time, cartographic expert systems were the Holy Grail for the cartographic community. The following statement is from early research in cartographic expert systems, and illustrates a possible direction of cartographic expert systems and functions:

> A full cartographic expert system would be capable of producing, without human intervention, maps of all types, ranging from graphics to rapid on-line GIS display to high-quality printed wall and atlas maps. … The cartographic expert system would allow the user to specify scales, projections, colors, symbols, and other map elements, but would make good decisions about defaults for any of these if the user chose not to specify them." (Buttenfield and Mark, 1991, pp. 139)

However, the development of cartographic expert systems has not taken off in the real world. There are many reasons for the slow progress in cartographic expert systems. One of the major concerns is that cartographers have not been able to agree on the rules of symbolization and generalization (Wang and Ormeling, 1996). In fact, the fundamental problem with a cartographic expert system is that the traditional expert system is a closed, ad-hoc system where the rules, the facts, and the inference engine are all confined to a single box (Figure 3). The centralized expert systems cannot meet the different types of mapping requirements, or provide dynamic rules for different applications. It is impossible to design an expert system that performs all of the mapping tasks.

Software agents will be an alternative solution for automatic map making. Software agents search, combine, and dynamically apply different cartographic rules according to specific mapping tasks. For example, to display a digital elevation model, software agents collect the color scheme for elevation and interpolation algorithms for a 3D data model. If the next mapping task is to create a road map, software agents will be able to dynamically change these rules and find an appropriate color scheme for the road map display. The dynamic architecture of intelligent software agents has more flexible rules and helps in various Internet mapping tasks. Figure 3 illustrates the mechanisms of software agents that can combine user-defined cartographic rules, knowledge bases, and GIS metadata (facts) for different types of tasks. The following section will introduce the design of cartographic software agents and specify their roles and functions.

3.1 The Design of Cartographic Software Agents

Software agents help users to access distributed data objects and GIS components on heterogeneous GIS platforms by interpreting, filtering, and converting information automatically (Tsou and Buttenfiled, 1998a). Three fundamental roles of software agents are essential to Internet mapping: information finder/filter, information interpreter, and decision maker.

3.1.1 Information finder/filter role

The first role of a software agent is as an information finder/filter, which helps users find the requested information and filter out unnecessary elements to a reasonable level according to a specified user task. The interactive process between software agents and users is similar to the relationship between car dealers and car buyers. When a buyer

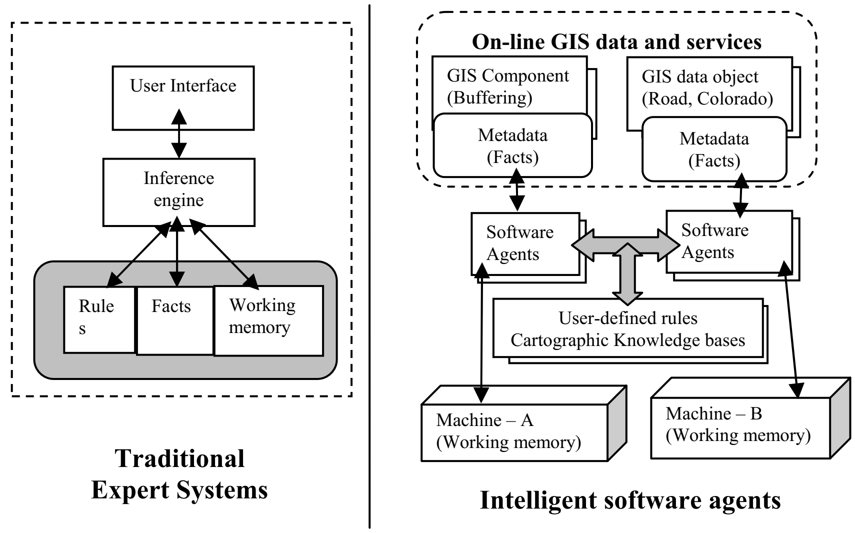


Figure 3. The Comparison of cartographic expert systems and intelligent software agents..

tells the car dealer what kind of car they prefer and what price they can afford (task specification), the car dealer will filter and simplify the information for all available cars. The car dealer will in theory only provide a reasonable range of choices to the buyer. Similarly, the intelligent agent will filter out unnecessary information and provide a reasonable number of choices. Moreover, software agents can play a more active role than a simple information filter. If the task cannot be executed or completed, the software agent may suggest modifying requests and tasks, or provide an alternative choice to users. Figure 4 shows the role of software agents as an information filter/finder. In this example, a user submitted a search request ("DEM color scheme") to the software agent. The agent will use different search methods to search appropriate color display schemes from multiple cartographic knowledge bases located in web servers A, B, and C. Then the software agent will apply these rules to the actual display of the digital elevation model.

One of the design issues for information finders/filters is the search mechanism. Three types of agent search mechanisms are commonly used in current software agent models: message broadcasting, agent roaming, and the metadata repository (Knapik and Johnson, 1998). The message broadcasting mechanism is a traditional network communication approach, where the sender broadcasts messages to local networks and waits for other machines to respond (Peterson and Davie, 1996). The second type of search is the roaming agents, in which a software agent will move or copy itself to different locations or servers and collect requested information directly, then return back to the original location. The third type of search mechanism is to build a connection between agents and database servers directly and generate a metadata repository of connected databases in advance. When software agents receive a request from users, the agents can search the metadata repository immediately and then redirect the linkage to the target database server.

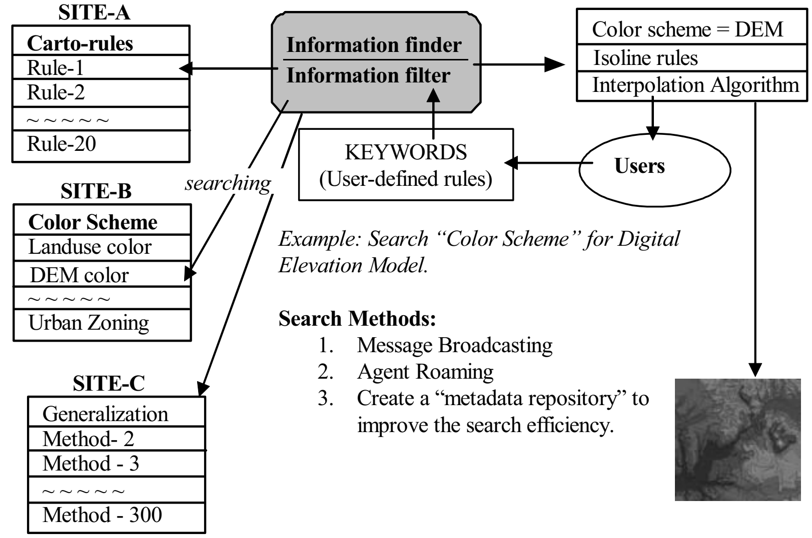


Figure 4. The information finder/filter.

3.1.2 Information interpreter role

The second role of the software agent is as an information interpreter that accesses and conveys information from one side to the other. In distributed network environments, heterogeneous data models and systems can not communicate directly. A software agent can bridge heterogeneous information islands and translate different data types and models for different systems. In order to translate the information correctly, this agent has to acquire certain knowledge and methods during the process. The knowledge and methods are defined and encapsulated in the metadata of geodata objects or components (Orfali et. al., 1996; Tsou and buttenfield, 1998b), and help an agent to interpret the information correctly (Figure 5).

The design of map display metadata is essential for information interpreters, which specify the representation methods of geodata objects on electronic media or computer screens. The contents of the map display metadata may include the following items:

- Feature type (raster/vector, point, line, polygon, or volume)
- Spatial references (coordinate systems, projections, map units)
- Attribute type (nominal, ordinal, interval/ratio, or multiple attributes)
- Map symbols (attribute lookup table, symbol size, symbol icons/shapes)
- Color scheme (2 bits-B/W, 8bits-256 color, 32 bits-true color)
- Scale threshold

These metadata contents can be interpreted automatically by software agents to dynamically apply the color scheme and map symbols. With the help of map display metadata, Web-based geodata objects become self-describable and self-manageable map layers.

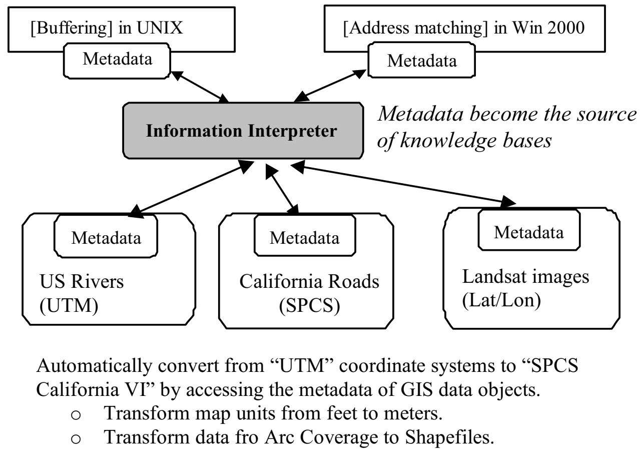


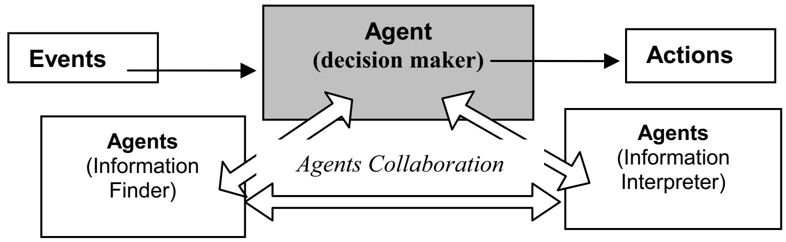
Figure 5. The information interpreter.

As an information interpreter, software agents help users perform GIS operations more accurately and efficiently. For example, an Internet map user wants to perform a GIS overlay by combining a (Colorado roads) in Web-site-A and a (US rivers) in Web-site-B, but the two different GIS data sets may have different coordinate systems (UTM and SPCS) and different map units (meters and feet). Software agents would help users to convert their coordinate systems and map units to the same system for the map overlay task. With the help of an information interpreter, Internet map users can easily work with different types of mapping data and GIS programs.

3.1.3. Decision maker role

The third role of a software agent is that of a decision maker, which can make decisions autonomously based on rational rules defined by its own knowledge base, user-defined rules, or the collaboration of other agents (Ferber, 1999). A software agent can collect and analyze information according to specific events, make an optimal decision through the collaborations of users and other agents, and execute the final decision (Figure 6). For example, a GIS user wants to add a new polygon layer to his existing point data layers. Software agents (such as an information finder) will begin to search for related cartographic rules, and will find one that indicates that "all point data should be above polygon data for the effective display". Then the information interpreter can convert this cartographic rule to an executable command, and users will confirm the agent's actions. With the help of other agents, the decision maker agent will move all point data above the new added polygon data layer.

There are several issues in the design of decision makers. First, the setting of decision-making rules will require the design of appropriate user interfaces and procedure verification during the period of collaboration between users and software agents. The second issue is the collaborations among software agents, which need to define an appropriate communication protocol and formalize agent interaction mechanisms. The third issue is how to choose the participants in the decision-making



Event: If a new [polygon] layer is added into point data layers.
Agents/Users Collaboration:
Info. Finder: search for cartographic rules
Interpreter: convert cartographic rules to executable procedures
User: confirm agent's actions.
Action: Move [point] data layers above the polygon layer

Figure 6. The decision maker.

process and the setting of the decision-making procedure. Since a decision maker will have more power than other types of software agents, the design of software agents will need to define the hierarchy of software agents (Weiss, 1999).

4 The Establishment of Distributed Cartographic Knowledge Bases

The previous section illustrates how a software agent model can be applied in Internet mapping tasks, and specifies of their roles and behaviors. This section will focus on the establishment of distributed cartographic knowledge bases from which software can access and retrieve cartographic rules. In recent software agent research, the A.I. community uses a different term, "ontology", to replace "knowledge base." "An ontology is a particular conceptualization of a set of objects, concepts, and other entities about which knowledge is expressed, and of the relationships among them." (Labrou, et al. 1999, p. 46). In the cartographic software agent framework, cartographic ontology is the knowledge base for guiding cartographic software agents' behaviors and decisions. This chapter will use both ontology and knowledge base interchangeably. The following statements are a few examples of cartographic ontology:

- All maps have coordinate systems.
- The map units used in the State Plane Coordinate System are feet.
- The map units used in the UTM coordinate system are meters.
- 1 foot = 0.3048 meters.
- Point layers should be above polygon layers in map displays.

These statements indicate the concepts, rules, and relationships among cartographic objects and entities, such as map units, map types, and coordinate systems. However, these rules and concepts alone will not be understood by software agents and need to be processed and converted into actual computer algorithms for software agents. The next section will illustrate the procedures for converting ontology statements into agent-readable computer programs.

4.1 The establishment of a single cartographic rule

The first step in converting the ontology statement is to express these rules more specifically in the first logic format. The computer science and A.I community use the first logic expression to specifically formalize the rules of ontology (Russell and Norvig, 1994). The first logic expression is the mathematical language developed specifically for the predicate logic operations. The following example is the first logic expression of "*all point layers should be above all polygon layers*".

$$\forall x, y \quad \textbf{\textit{Above}}\textit{(PointLayer(x), PolyLayer(y))}$$

The first part defines ALL x and y variables. The second part indicates the relationship (Above) between x and y. The advantage of the first logic expression is to provide a clear definition of the relationships among different objects and events.

The next step in building cartographic knowledge is to convert the first logic expression into the actual computer programs that can be embedded inside software agents. The following example is the converted computer program for the previous first logic expression.

```
#Define int x, y
#Define int Number
# the "Number" variable indicates the layout sequences (1:top,
2:second..) #

polylayer(x).Number = i
pointlayer(y).Number = j
If (i < j) then {
       polylayer(x).Number = j
       pointlayer(y).Number = i
       }
```

This computer program will perform a sorting function to make sure that all of the point-layer sequence numbers are lower than the polygon-layer sequence numbers. The lower sequence number will be displayed above the higher sequence number. Then, software agents can access and apply this algorithm to specific mapping tasks.

Besides the conversion of the ontology statement, another important procedure for automatic mapping tasks is the integration of metadata facts and cartographic rules. The following example illustrates how to apply the cartographic rules to actual mapping tasks by bring the metadata and the rules together.

A software agent receives the following cartographic rules: *If the color of the new polygon layer is the same as one of the existing layers, carto-agents will change the color of the new layer to a unique color.* The first step is to convert the rule into the first logic expression:

$$\exists x_{\text{E (existing layers)}} \quad \textbf{\textit{Color}}\ (X_E) \equiv \textbf{\textit{Color}}\ (X_{new}) \ \Rightarrow \ \textbf{\textit{Change}}\ (\textit{CartoAgent, Color}(X_{new}))$$

Then the software agent will need to know what colors are used in the existing layers and find a unique color for the new layer. The layer color (facts) information can be retrieved from the metadata associated with each map layer. Software agents can use a random number to pick a unique color for the new layer.

```
Color[AllPolyLayers] = [blue, red, green]
Color(NewLayer) = NewLayer.metadata.color

While ( Color[AllLayers] contain Color(NewLayer) )
  {  Color(NewLayer) = Color(Randam)
  }
```

These examples illustrate the procedures for converting cartographic knowledge to actual computer programs and the implementation of cartographic principles by combining the metadata facts and rules. The next section will introduce another important feature of software agents, the capability to access multiple knowledge bases from different locations and dynamically combine cartographic rules.

4.2 Inference from multiple cartographic knowledge bases

The advantages of cartographic software agents are mainly the capabilities to access distributed cartographic knowledge bases (CKB) and conduct the inferences by combining multiple rules from different servers. The following example illustrates how an inference procedure is accomplished by accessing two different CKBs, one at San Diego State University and another at UC-Santa Barbara (This is a hypothetical example of distributed cartographic knowledge bases).

Multiple Cartographic Knowledge Bases (CKB):

- Rule#1: **"Landuse" maps are qualitative.**
  (located at San Diego State University) http://map.sdsu.edu/001.ckb
- Rule#2: **Color-hue is the best visual variable for displaying qualitative area maps.**
  (located at UC-Santa Barbara) http://geog.ucsb.edu/hydro.ckb

Inference:

Combine Rule#1 **AND** Rule#2

→ **Landuse maps should use "Color-hue" for area symbol display.**

Computer Program (Software Agent Actions):

```
Landuse.Symbols = ColorScheme(Hue).Attribute(LU)
```

This example illustrates the potential power of the distributed cartographic knowledge bases. After combining two cartographic knowledge bases, a new cartographic rule was generated through the inference procedures. Different research institutes and universities can develop their own specialized cartographic knowledge bases. For example, UC-Santa Barbara can focus on the color scheme rules for thematic mapping and San Diego State University can focus on the land use and land cover map display. By combining multiple cartographic knowledge bases, the software agent framework will be able to provide more flexible, intelligent mapping services for different tasks.

Currently, the detailed specifications for cartographic knowledge bases are still under development. This research suggests that the implementation of cartographic knowledge bases should adopt XML as the software development platform because XML is used by the current Internet mapping software packages and many software agent models, such as ESRI's ArcIMS and AutoDesk's MapGuide, Java's Jini, and

OGC's GML. XML is a metadata-oriented markup language that can provide a mechanism to impose constraints on the storage layout and logical structure (Bray, et. al., 1998). Many software companies have adopted XML as a major tool for the development of content-based services. Several software agent projects in academia also focus on the adoption of XML in agent communication languages (Nolan, et. al. 2000).

5 Conclusion

The design goal of cartographic software agents is to create distributed cartographic knowledge bases via the Internet. Each computer and web server connected to the Internet can contribute to the cartographic rules, in effect becoming one cartographic brain. Figure 7 illustrated the dynamic construction of Internet maps by combining multiple cartographic knowledge bases via the Internet.

Software agent-based communication mechanisms can facilitate the dynamic integration of geospatial data, GIS programs, and cartographic rules and knowledge bases in distributed network environments. The design of a cartographic software agent model can be applied to many Internet mapping applications, such as natural resource management, environmental monitoring, and transportation. Current Internet mapping facilities have very powerful cartographic tools but very few cartographic knowledge rules. Many web map users encounter serious problems when dealing with multiple map layer presentation and the map display controls, such as projection and coordinate systems.

This paper introduced the initial development of a cartographic software agent model for Internet mapping. The preliminary design identified the three dynamic roles of software agents: information finder/filter, information interpreter, and decision maker. Hopefully, with the help of software agents, the quality of web maps will improve and users will be able to use more effective color schemes and appropriate map symbols to display geospatial information on their computer screens or hand-held

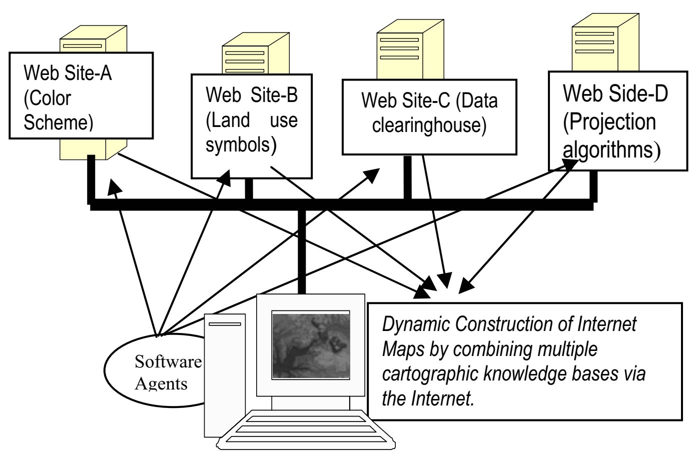


Figure 7. Dynamic Construction of Internet Maps by Software Agents.

devices.

The detailed specifications for cartographic ontology and inference engines are still under development and are the major tasks for the future. Internet mapping provides many potential directions for the development of cartographic knowledge bases and rules, such as 3D display, multiple layer transparency, animation, etc. It will be a great opportunity for cartographers to focus on these new topics and create a new paradigm for modern cartography. Dynamic, machine readable cartographic rules will become the essential components for future GIS applications and geospatial web services. Moreover, the development of cartographic software agents can combine multiple A.I. approaches, such as fuzzy logics for setting a scale threshold, probabilistic theory for the representation of data uncertainty, and neural networks for setting classification numbers. In the near future, a fully cartographic expert system may be feasible in the form of distributed cartographic software agents and dynamic Internet mapping frameworks. With the help of cartographic software agents, GIS professionals, Geoscientists, and the public can access and generate their own Internet maps more easily and effectively.

Acknowledgements

This paper forms a portion of the "Software Agent Communication Mechanisms" project (http://map.sdsu.edu/geoagent), sponsored by San Diego State University Foundation and the "Web-based Geospatial Information Services and Analytic Tools for Natural Habitat Conservation and Management" project supported by NASA's Affiliated Research Center at San Diego State University. Funding by the NASA ARC program and matching funds from the San Diego State University Foundation are acknowledged and greatly appreciated.

References

Bigus, J. P. & Bigus, J. (1998). *Constructing Intelligent Agents with Java: A Programmer's Guide to Smarter Applications.* New York: John Wiley & Sons, Inc.

Bradshaw, J. M, editor (1997). *Software Agent*. Menlo Park, California: AAAI Press.

Bray, T., Paoli, J., & Sperberg-McQueen, C. M. (1998). *Extensible Markup Language (XML) 1.0 Specification.* W3C. URL: http://www.w3.org/TR/1998/REC-xml-19980210

Brenner, W., Zarnekow, R. & Wittig, H. (1998*). Intelligent Software Agents: Foundation and Applications*. Berlin: Springer.

Buttenfield, B. P. & Mark, D. M. (1991). Expert systems in cartographic design. In *Geographic Information Systems: The Microcomputer and Modern Cartography*, ed. D. R. F. Taylor, Oxford: Pergamon Press. pp. 129-150.

Buttenfield, B. P. & McMaster R. B., editors. (1991). *Map generalization: Making rules for knowledge representation.* London: Longman Scientific & Technical.

Christensen, J., Marks, J., & Shieber, S., (1995). An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics* 14(3), pp. 203-232.

Doddi, S., Marathe, M. V., Mirzaian, A., Moret, B. M. & Zhu, B. (1995). Map labeling and its generalizations. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*, pp. 148-157. URL: http://www.cs.unm.edu/~moret/map.ps

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence,* English Edition. Harlow, England: Addison-Wesley.

Freeman, H. R. & Ahn, J. (1984). AUTONAP – an expert system for automatic name placement. *Proceedings of the First International Symposium on Spatial Data Handling* , Zurich, Switzerland, August 1984. pp. 544-569.

Goodchild, M. F., Egenhofer, M., & Fegeas, R., editors. (1999). *Interoperating Geographic Information Systems.* Dordrecht: Kluwer Academic Publisher.

ISO/TC 211 Secretariat (2000). *Program of Work,* Version 8. ISO/TC 211 N 854. URL: http://www.statkart.no/isotc211/dokreg09.htm

Knapik, M., & Johnson, J. (1998). *Developing Intelligent Agents for Distributed Systems: Exploring Architecture, Technologies and Applications*. New York: McGraw-Hill.

Kraak M & Brown A. editors (2001). *Web Cartography*. London: Taylor & Francis.

Labrou, Y. Finin, T., & Peng, Y. (1999). Agent Communication Languages: The Current Landscape. *IEEE Intelligent Systems,* March/April 1999, pp. 45-52.

Lamy S, Ruas A, Demazeu Y, Jackson M, Mackaness W, and Weibel R 1999 The Application of Agents in Automated Map Generalisation. Paper presented in the 19th International Cartographic Conference in Ottawa, August 20, 1999. URL: http://agent.ign.fr/public/ica/index.html

Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM,* 37(7), pp.31-40.

Nolan, J. J., Sood, A. K, & Simon, R. (2000). An Agent-based Architecture for Distributed Imagery and Geospatial Computing. *Proceeding of the 29th Applied Imagery Pattern Recognition Workshop*. Washington, DC. October 16-18, 2000. pp. 252-257. URL: http://citeseer.nj.nec.com/439621.html

Open GIS Consortium, Inc. (OGC) (1998). *The OpenGIS Abstract Specification,* Version 3. Wayland, Massachusetts: Open GIS Consortium, Inc., URL: http://www.opengis.org/techno/specs.htm

Openshaw S. & Openshaw C. (1997). *Artificial Intelligence in Geography.* New York: John Wiley & Sons, Inc.

Orfali, R., Harkey, D., & Edwards, J. (1996). *The Essential Distributed Objects Survival Guide*. New York: John Willey & Son, Inc.

Peterson, L. L. & Davie, B. S. (1996). *Computer Networks, a Systems Approach*. San Francisco, California:.Morgan Kaufmann Publishers, Inc.

Russell S. J. & Norvig P. (1994). *Artificial Intelligence: A Modern Approach*. Englewood Cliff, New Jersey: Prentice Hall.

Shea, K. S. (1991). Design considerations for an artificially intelligent system. In Buttenfield, B. P. and McMaster R. B. edited *Map generalization: Making rules for knowledge representation.* London: Longman Scientific & Technical, pp. 3-20.

Shoham, Y, (1997). An Overview of Agent-oriented Programming. In *Software Agents*. edited by J. M. Bradshaw. Menlo Park, California: AAAI Press, pp. 271-290.

Tsou, M. H., & Buttenfield, B. P. (1998a). An Agent-based, Global User Interface for Distributed Geographic Information Services. In *Proceedings of the 8th International Symposium on Spatial Data Handling,* Vancouver, Canada, pp. 603-612.

Tsou, M. H., & Buttenfield, B. P. (1998b). Client/Server Components and Metadata Objects for Distributed Geographic Information Services. In *Proceedings of the GIS/LIS' 98,* Fort Worth, Texas, pp. 590-599.

Wang, Z. & Ormeling, F. (1996). The representation of quantitative and ordinal information. *Cartographic Journal* 33(2), pp. 87-91.

Weiss, G., editor. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial* Intelligence. Cambridge, Massachusetts: The MIT Press.

Weiss, S. M. & Kulikowski C. A. (1984). *A Practical Guide to Designing Expert Systems*. Totowa, New Jersey: Rowman and Allenheld.

Zhan, F. B. & Buttenfield, B. P. (1995). Object-oriented knowledge-based symbol selection for visualizing statistical information. *International Journal of Geographic Information Systems.* 9(3), pp. 293-315.

Zucker, J.-D., Mustiere, S. & Saitta, L. (2000). Learning Abstraction and Representation Knowledge: An Application to Cartographic Generalisation. The fifth International Workshop on Multistrategy Learning (MSL'2000), Guimarães, Portugal.