From GISystems to GIServices: Spatial Computing on the Internet Marketplace

Oliver Günther and Rudolf Müller
Institut für Wirtschaftsinformatik
Humboldt-Universität zu Berlin

TR-98-008

Abstract

Many of the functions performed by GIS seem to be amenable to a business model that is fundamentally different from the one we see today. At present, GIS users typically own the hardware and software they use. They pay license and maintenance fees to various vendors. The alternative would be a service-oriented approach where users make their input data available to some GIS service center that performs the necessary computations remotely and sends the results back to the user. Customers pay only for that particular usage of the GIS technology – without having to own a GIS. We discuss this business model and associated problems of privacy and ease-of-use. We also give an overview of our MMM system (http://mmm.wiwi.hu-berlin.de), a distributed computing infrastructure that supports this business model.

1 Introduction

Question: Why do people buy a GIS? *Answer:* Because their neighbor has one. Richard Newell of Smallworld Systems told this joke during his keynote speech at the 1997 *Symposium on Spatial Databases* – and he did not only refer to Smallworld customers. The truth behind his joke is that GIS are often greatly underutilized. Many customers use only a small fraction of the functionalities offered by their GIS. Some of them are aware of that: they simply do not care about the remaining features. Others are not: they may thus miss functionalities that are actually there and use complicated ways to reimplement them with the features they know. Yet other users may not use their GIS at all: they bought it because they thought it may help them with their problems but then found out that it does not. Some customers may not even have bothered to look: they bought the GIS and left it in the package.

To be fair, this can be said not only about GIS but also about many other types of software. Microsoft, for example, estimates that 90% of Excel's functionalities are used by only 10% of its users worldwide. A large part of requests for new functionalities received by Microsoft each day can be answered simply by telling the customer that the requested functionality already exists. What makes the situation somewhat different for GIS, however, is the relatively high price of a GIS license. GIS come in relatively large packages: a single license often costs 3,000 US$ or more, it requires powerful hardware to run on, and it takes considerable training on the customer's part to use the software in a productive way. For most commercial GIS, potential customers face an all-or-nothing choice. Either they invest a relatively large amount to get the license, the required hardware, and some training—or they do not, in which case they get nothing.

We claim that many potential GIS users would become faithful GIS customers if they could do so at a smaller entry cost. Of course, the lower ticket price would not buy them the whole license indefinitely. But rather than putting a time limit on the license, as is typically done, vendors could limit the functionalities they offer. This could mean in particular that the vendor does not sell a classical system license but selected "services" that perform GIS-typical tasks at the customer's request. Services would typically be located at a site run by the vendor or a third party. Customers would make their data available to the service and retrieve the results from the remote site after the computation has been performed. No special hardware or training would be required on the user's part. Payment schemes would follow this service-oriented approach: rather than buying a whole system, users only pay for the particular usage of the vendor's software.

A direct consequence of such a shift from geographic information *systems* to geographic information *services* would be the rise of an electronic market [20] for spatial data and services. Anybody with Internet access could act as both provider and consumer of related goods.

In Section 2 we discuss electronic markets for information goods in more detail. We evaluate the technology that is currently available to support Internet markets, including user interfaces, communication protocols, and server implementations. Section 3 presents MMM, our own middleware approach to support electronic markets for information goods. Section 4 discusses related work, and Section 5 presents conclusions and open problems.

2 Electronic Markets for Information Goods

2.1 Basic Concepts

In an *electronic market for information goods*, providers and consumers trade data and computational services by means of an information network, in particular the Internet (Fig. 1). *Providers* advertise their offerings, make available requested data, and perform requested computations. They may also bundle some of their services with those of other providers. *Consumers* search and select services for a given task, apply them to some data set, and pay providers. Consumers may also integrate services as remote function calls into their local computations and combine services from different vendors. In a GIS context, typical services include data retrieval and conversion, map production, map overlay, spatial access methods, statistical analyses (e.g. multivariate data analysis), and visualization tasks.

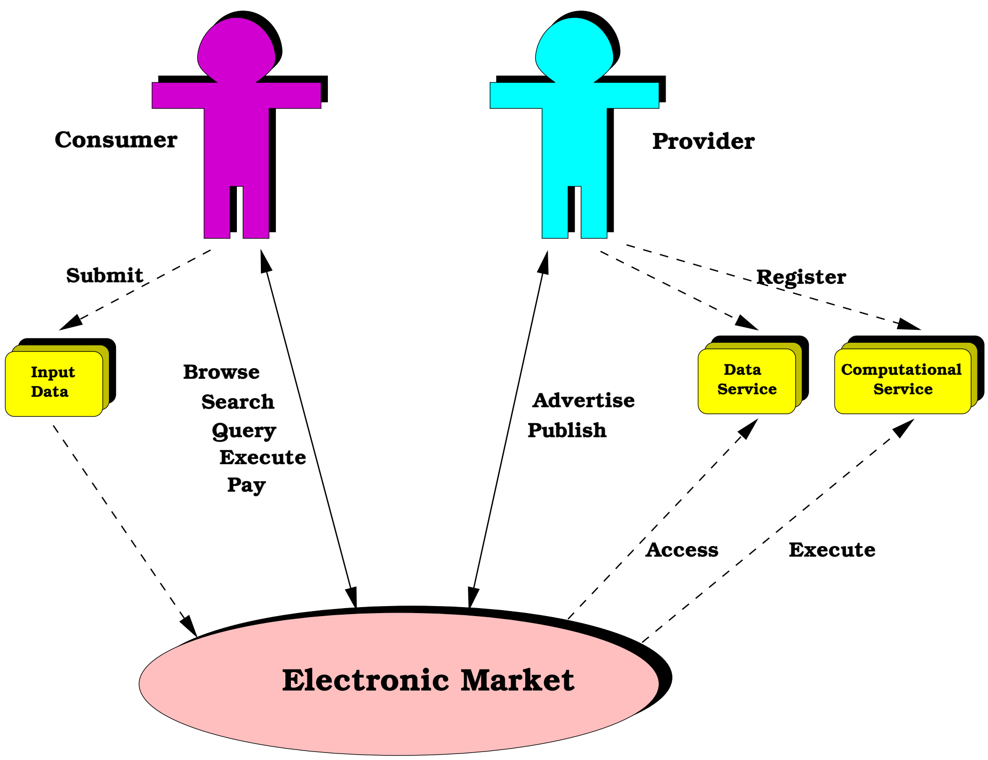

Figure 1: Interactions on an Electronic Market

Our vision is an Internet-based market for information goods, where anybody with Internet access can participate as consumer or provider. It should be as easy to *register* a service as it is to write a Web page today, and it should be as easy to *use* a service as it is to read a Web page today. With regard to GIS, this vision would most likely translate into an electronic market for GI services, where the number of GI service providers would be considerably greater than the number of GIS vendors today. Moreover, the number of consumers would be much greater than the number of GIS users today. Although the average revenue *per consumer* would be lower than the average revenue per GIS user today, *overall* revenue could well increase considerably. We also

Format/Technology	Consumer	Provider
HTML	++	0
Native document formats (other than HTML)	0	++
HTML forms + CGI + database access	+	–
HTML forms + CGI + general services	0	–
Java applets (as user interfaces) + general services	0	–
Java applets (to implement services)	+	– –

++ = very easy, – – = very difficult

Table 1: Consuming and Providing Web Content Today

expect customer satisfaction to improve because users pay only for those functionalities they really take advantage of.

Which technologies are available today to implement such electronic marketplaces for GI services? Table 1 rates technological alternatives in terms of ease of use for consumers and providers; it shows that the situation for content providers is still rather unsatisfactory. For simple documents, providers can either offer them in their native format, which requires additional effort on the consumer's side, or they have to convert their holdings into HTML. The Common Gateway Interface (CGI) – a *de facto* standard to connect dynamic information sources – requires even more advanced skills on the provider's part, especially when used for services that are more complex than simple database access. Non-technical content providers will rarely be willing to go through the required learning effort. A similar argument applies to Java-based solutions: many potential service providers, especially in the research community, would not be willing to spend much time on converting their software to Java. Even the effort required to provide an existing service with an applet-based user interface is often prohibitively high.

The evolution of GI service markets thus requires more than application programming with HTML, CGI, and Java. We need appropriate conceptual foundations for the required technologies and their economics, and we need the transfer of these concepts into organized *electronic marketplaces*. A marketplace should free providers from implementing a fully integrated service, thus significantly reducing their market entry cost. Such marketplaces were first proposed for decision support applications [6, 10].

The Internet technologies underlying such a marketplace can be grouped into four categories: *user interfaces*, *communication*, *connectivity*, and *market infrastructure*. Any combination of choices from each category defines a middleware [5] for an information market. For example, if one takes a Web browser as user interface, HTTP as communication protocol, Web sites for connectivity, and Web search engines as market infrastructure, then one obtains the existing Web.

The remainder of this section briefly reviews technologies in each category, concentrating on their roles in an electronic marketplace. Bhargava and Krishnan [7] give more implementational details in the application context of operations research and decision support.

2.2 User Interfaces

User interfaces for Internet services include simple Web browsers, programs embedded in a Web browser, or stand-alone programs that embed communication protocols for remote services. Among the most important technologies for browser-embedded GUIs are markup document languages (HTML, HTML forms, XML), JavaScript, plug-ins, Active-X components, and Java applets.

These technologies differ in terms of complexity for both consumer and provider. Writing applets with the Java Application Windowing Toolkit, for example, is far more complex than writing HTML forms. Of course, the higher complexity also buys more functionality. Most significant is that all solutions but plain HTML allow client-side computing, i.e., the downloading of a software module to a client with subsequent execution. One may use this option, for example, to validate user input before it is sent to the remote service, or one may download a whole service and execute it locally. Other differences exist with respect to the ways downloaded programs can access client resources (such as files), and with regard to communication protocols. Plug-ins, Active-X, and Java applets can use any combination of protocols discussed in the next section, while the simpler approaches are more limited.

Note that user interfaces to GI services would not be restricted to browser-embedded solutions. One could also envision new lightweight GI systems that embed transparent access to an Internet-based market of spatial data and computational services.

2.3 Communication

We distinguish four categories of communication protocols: stream socket protocols (e.g. libraries that implement TCP/IP-based sockets between client and server), Internet application protocols (HTTP, FTP, SMTP, or proprietary telnet solutions), remote function calls (RPC, DCE, etc.), and distributed object environments (RMI, CORBA, COM/OLE, or proprietary solutions).

These categories are not mutually exclusive. In fact, the latter three categories are all based on socket protocols. From the programmer's perspective, however, they provide quite different functionalities. The first two are particularly appropriate if byte strings (e.g. images or formatted ASCII documents) are exchanged between client and server. The latter two are more appropriate if communication relies on the exchange of instances of data types.

In the last category, RMI by JavaSoft only works between Java clients and servers. It may use stream sockets *or* HTTP for communication, which allows it to work through firewalls. Microsoft's COM/OLE is supposed to work across different programming languages (but only on Windows platforms). The Common Object Request Broker Architecture (CORBA) provides a language-independent communication layer between distributed objects across operating systems and programming languages (currently supporting C, C++, Java, Smalltalk, and ADA).

GI services may use any of these communication protocols in various combinations, depending on the particular application requirements. Remote function calls or method invocation mechanisms may be less suitable for performance-sensitive applications since data has to be passed through several protocol layers. Data confidentiality and integrity could likely be implemented by secure sockets under any of these protocols. An authentication service, on the other hand, may require higher-level communication protocols, like a CORBA object service.

2.4 Connectivity

The term *connectivity* refers to the different ways to wrap existing software in order to make it accessible via one of the communication protocols listed above. Choices include:

- CGI scripts, which process a request (typically encoded as an HTML form) by invoking a program, generating an HTML form to present the result, and forwarding it to the requesting client;
- server-side scripts, which are embedded in pre-compiled HTML documents and interpreted at the server before the document is forwarded to the client;
- callable libraries, which are invoked via an RPC or DCE daemon;
- object wrappers, which give the software system an object-oriented interface.

An essential design decision concerns the question whether the response of the system should depend on previous requests, i.e., whether communication should be *stateful*. The first three solutions do not maintain state *per se*, although one may mimic state on top of their stateless protocols. In CGI, for example, this could be done by depositing *cookies* at the client's site. A cookie is a string that contains some state information for the server. The server may later retrieve it and use it to restore the state reached at the end of a previous session.

Another important characteristic of an information market is topology. The classical Web solution is to contact one service at a time and get the result back to the browser. In a market, however, it may be desirable to advise services to get their input from other services. This requires wrappers for services that can communicate with each other.

Finally, wrappers should be adaptive with respect to the service they encapsulate. Simple wrapping techniques, such as CGI scripts, may have to be changed with every service modification (e.g. in order to change the HTML form used to present the result). More advanced techniques, such as server-side scripts, can adapt automatically.

2.5 Market Infrastructure

The term *market infrastructure* refers to services that are shared between providers and consumers. They can be divided into four categories: repositories, serving as yellow pages for market content; data transformation services; support for service registration; and execution planning.

The need for *repositories* is motivated by the complexity of describing a service. Indexing techniques as used by Web search engines may fail, since they are based on full-text analysis. Relationships between different services, such as *belong to the same domain* or *solve an equivalent problem*, need to be made explicit. The repository may also serve for certification purposes by implementing ratings, annotations, and reviews of services. Repositories may simply be the result of an integration of services. As an example, consider the TSIMMIS architecture [23, 16] of wrappers and mediators (Fig. 2). The mediator between the application and the wrapper is able to decompose a query and spawn subqueries to the wrappers. The wrapper then translates the incoming queries into requests to the encapsulated legacy system. While there is actually considerably more functionality provided in TSIMMIS, this simple description already shows a simple market infrastructure where the mediator realizes a repository implicitly.

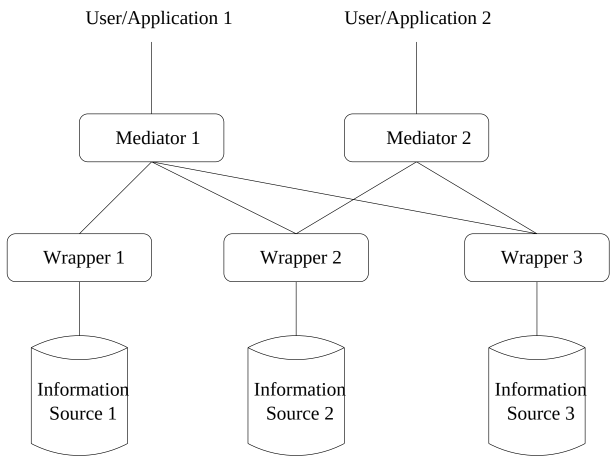

Figure 2: Mediators and Wrappers in the TSIMMIS architecture

Data transformation services are required as long as there is no single data interchange format in the chosen application domain. In this case, data transformation at market level lowers the entry barriers for providers because they can register existing services in their original form, without adapting them to other data formats. If a market has already converged towards a common language to define formats (e.g., XML) or interfaces (e.g., CORBA's Interface Definition Language), data transformation may still be necessary at the semantic level. Consider, for example, two tables of geographic distances that are both represented in XML but whose semantics are defined differently. Interoperation between these two tables requires some mapping between them.

Support for *service registration* can come in different flavors. One way consists of wrapper frameworks that interoperate with the repository as well as with data transformation services. The latter aspect is particularly important since it enables interoperation with registered services, thus creating *positive externalities* (i.e., the benefit of participation increases with the number of participating consumers and providers). Another kind of support consists of component frameworks (such as CORBA's vertical common facilities or SunSoft's Java beans) that may serve as starting points for the implementation of a service. Interoperation with Web browsers could then be based on systems like CorbaWeb [21], Web∗ [3], and W3Objects [19]. A third kind of support applies to the positioning of a service in a given classification scheme. Finally, support could come in the form of *execution agents* to be created from service descriptions [8].

Execution planning in GIS marketplaces has been proposed by Abel et al. [2]. They define a plan as an acyclic digraph. Its source nodes are data services or data provided by the user, interior nodes represent computational services, and target nodes are results from services. Arcs relate to data flow between data sources, computational services, and data targets. Execution planning may take place manually by means of scripting languages [4] or graph editors, or it could be done automatically. Automatic execution planning would generate a plan from a declarative request [15]. A typical example of plan generation comes up in the context of interactive map making (e.g. on a WWW map server). Depending on the user's specifications, the system has to determine appropriate data sources, retrieve the required data sets, possibly perform some translation between formats, align the data sets, and perform a map overlay. In addition, one may have to perform some complex aggregation on the associated non-spatial data.

2.6 Summary

Figure 3 summarizes the different solutions for user interfaces, communication and connectivity. Increasing distance from origin represents higher complexity and functionality. A region spawned by intervals on the three axes describes a middleware for which a market infrastructure could provide value-added services.

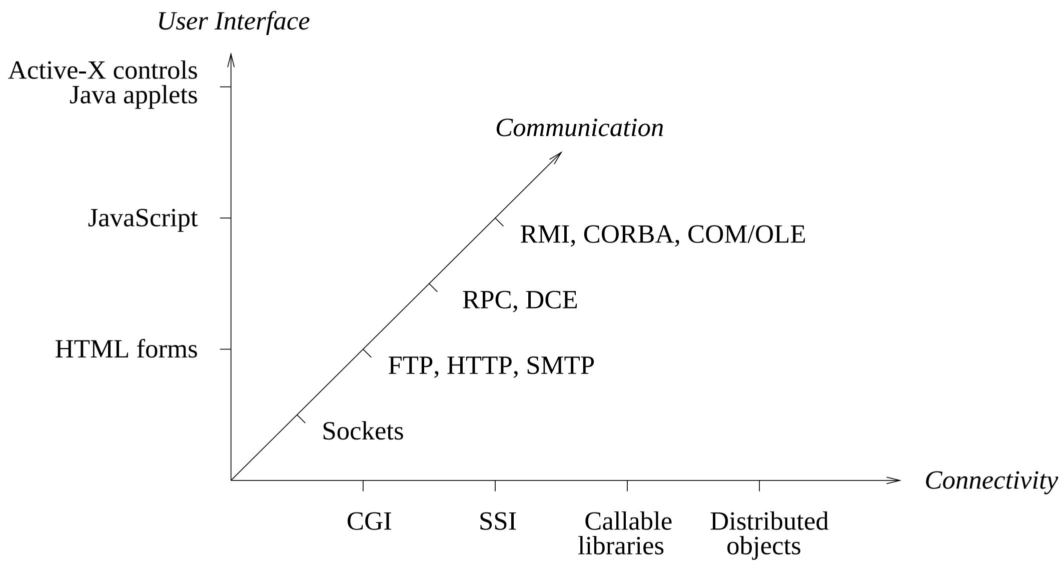

Figure 3: Technologies for electronic marketplaces

While one could argue for maximum functionality in each dimension, it is not clear whether this would stimulate market participation because it may have a negative impact on interoperability with other technologies used in the market. Take, for example, an infrastructure that heavily relies on CORBA. The fact that CORBA environments implement binary data interchange formats excludes the corresponding data from being processed with other tools. A better approach might be to agree on the exchange of documents written in a markup language defined in XML. Such documents can be processed by many tools and are also readable by humans. NAOMI, the North American Openmath Initiative (http://naomi.math.ca/) and a follow-up to the European Openmath project (http://www.openmath.org/), is currently defining such languages for mathematical data. As part of these projects, the *Java OpenMath Library* (http://pdg.cecm.sfu.ca/openmath/lib/) implements Java classes that are instantiated by XML documents. Applications using these classes may communicate directly with each other via RMI or CORBA, while communication with other applications could be based on the exchange of XML documents via TCP/IP sockets.

Overall it can be said that most of the technology to create GI services is available. Other than pure implementation issues, however, we identify five major problem areas that need to be addressed before our vision of an electronic market for GI services could become reality:

- user interfaces that are simple enough even for the occasional user;
- formats for metadata to describe data and computational services, and to match them with consumers' problems;

- data transport mechanisms that are reliable, efficient and secure;
- data interchange formats that interoperate with customers' desktop data representation;
- business models that address related legal, marketing, and pricing issues [10, 11].

3 MMM: A Method Management System

We recently presented our MMM (Method ManageMent) system, a distributed computing infrastructure that supports a business model and electronic marketplace as described above [18]. MMM is a collection of middleware services that facilitate Web-based access to software modules. In particular, MMM supports providers and consumers in

- browsing, searching, and querying for relevant services;
- executing services in (possibly proprietary) execution environments;
- transferring and converting input and output data;
- state maintenance to support exploratory data analysis;
- registering new services (publication, check-in, interface definition).

Early prototypes of MMM were based on traditional CGI technology. Consumers entered their requests by means of HTML forms, which were then translated into CGI and forwarded to MMM. In more recent versions, we have replaced the CGI component by a Java applet. A prototype is available at http://mmm.wiwi.hu-berlin.de. A CORBA-based reimplementation is currently under consideration [24].

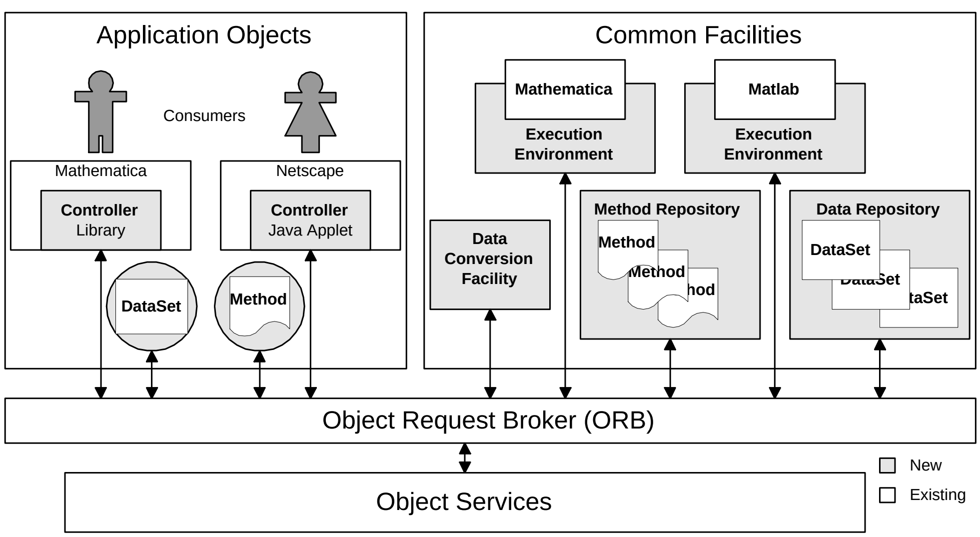

Figure 4: MMM Architecture in OMG Terminology

As illustrated in Fig. 4, MMM maintains a registry of services in the form of data and method repositories. They contain two kinds of meta objects [17]. *Data service objects (DSOs)* serve to wrap and describe *data* services. *Method session objects (MSOs)* wrap and describe *computational* services. Data and method repositories are organized along subject-specific taxonomies. In MMM's initial application domains – statistical computing and operations research – these are based on the *Guide to Available Mathematical Software (GAMS)* taxonomy [13].

Meta objects may be stored in different formats, such as XML documents or relational database tables. Figure 6 gives an XML example of an MSO for a Matlab routine, which resides on a Princeton WWW server. The metadata includes all the information that is necessary for the MMM infrastructure to place the method in the given taxonomy, to query a consumer for appropriate input data, to retrieve the routine via the Internet, and to execute it on the consumer's data by a Matlab execution service. The provider of such a routine only has to create an MSO in order to register it as a service in the MMM marketplace. Registration can either be done by editing an XML document and submitting it to MMM, or by going through a registration session with a Java applet.

When a consumer decides to run the Matlab script, the MMM infrastructure uses distributed object technology to create a temporary session object, which resides remotely. In CORBA terms, one creates an *application object* that uses middleware services and common facilities to support its operations (e.g., billing and authorization facilities). In order to generate a user interface to the session object, the Java applet creates a dynamic stub. On method invocation, the session object uses in turn stubs of MMM common facilities to negotiate the execution (Fig. 5). A session object for the Matlab routine described in Fig. 6 would first invoke an operation on an object that implements a Web client to download the implementation from Princeton. Then it would use a stub of the execution engine to install the implementation temporarily and transmit the user's input data. Finally, it would trigger execution and return the result to the applet.

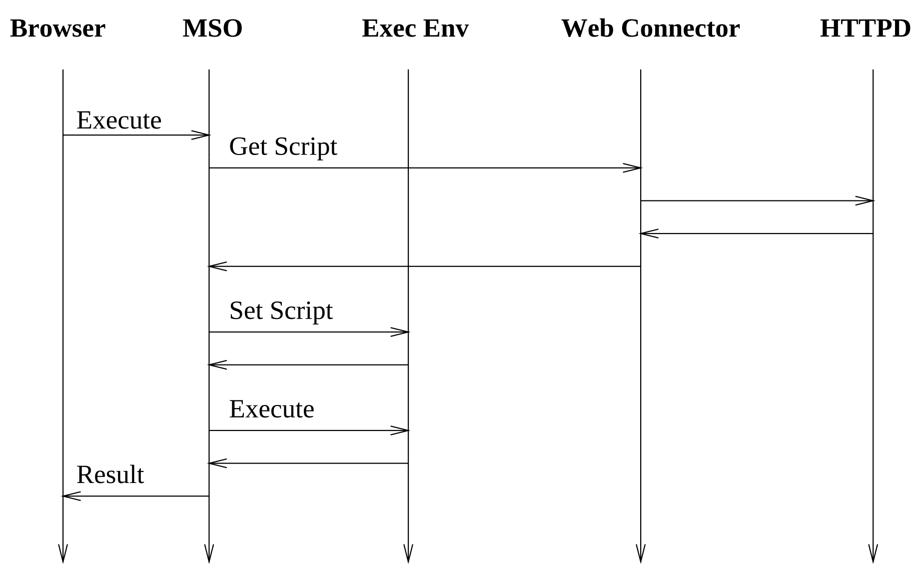

Figure 5: A vertical time sequence diagram illustrating method invocation in MMM

In order to locate relevant services, one can browse or search the MMM registry. This includes simple queries in the style of a WWW search engine (e.g., `time series AND estimation`), as well as more complex SQL queries (e.g., `SELECT plan FROM MSOclass m WHERE category =`

``estimation'').

With regard to the taxonomy presented in Section 2, MMM combines different choices for its user interface, communication, and connectivity. MMM's *user interface* is a Java applet that implements generic layout functions for a set of data types (basic types like string or integer, and abstract data types like records or tables). The applet uses type information and dynamic stubs to create input masks on the fly for the operations offered. For *communication*, MMM gives objects the choice between stream sockets and method invocation via stubs. For example, when an MSO transmits the script for execution, it can either pass it as parameter value in a method call, or it can create a stream socket from which the execution service reads the script via plain TCP/IP. *Connectivity* is realized by a framework of object-oriented wrappers, which may in turn encapsulate other protocols to contact services on the Internet. An example is provided by the Web Connector object in Fig. 5 that encapsulates the HTTP protocol. Other wrappers encapsulate connectivity to engines that can interpret routines in specific languages, like the Matlab routine in the example.

4 Related Work

MMM is cooperating closely with two other projects that have related objectives: DecisionNet and SMART. *DecisionNet* [6, 9] is an organized electronic market for decision support technologies. The market infrastructure consists of agents that support consumers and providers in transactions. The decision technologies themselves reside on provider machines distributed across the Internet. DecisionNet is accessible at http://dnet.sm.nps.navy.mil/.

SMART [1, 2] is another Internet marketplace model with an emphasis on spatial data and related algorithms. Like MMM and DecisionNet, SMART is based on the asynchronous communication between service providers and consumers. It offers *query services* to obtain data from a provider, *function services* to model computational tasks (such as conversion between representations), *planning services* to combine and coordinate different tasks, and *execution services* to execute a plan on behalf of a customer.

In 1994, an international group of GIS users and vendors founded the *Open GIS Consortium (OGC)*, which has quickly become a powerful interest group to promote open systems approaches to geoprocessing [22]. The OGC defines itself as a "membership organization dedicated to open systems approaches to geoprocessing." It promotes an *Open Geodata Interoperability Specification (OGIS)*, which is a computing framework and software specification to support interoperability in the distributed management of geographic data. OGC seeks to make geographic data and geoprocessing an integral part of enterprise information systems.

An interesting application of some of the technologies described in this paper has recently been presented by Bhargava and Tettelbach [12]. They implemented a waste disposal planner for the Monterey area by combining geographic data about waste disposal points with data about which types of waste are taken at what price. Providers of waste disposal services can register their location and prices online. Users may employ the service for simple look-up or for the computation of an optimal disposal strategy. The system combines a form-based user interface that operates via CGI with a relational database.

Olsen and Associates, a Swiss company founded in 1985, operates a commercial decision support service on the World Wide Web (http://www.olsen.ch). The site features a complete set of models, library resources, and customer support for financial market traders. The underlying decision technology consists of mathematical models to compute directional forecasts and timing indicators

```
<OBJECT><MSO><ID>ADJPH</ID>

<GENERAL>
<AUTHOR><PERSON> ... </PERSON></AUTHOR>
<PUBLISHER><PERSON> ... </PERSON></PUBLISHER>
<CREATION> ... </CREATION>
</GENERAL>

<ACCESS location="extern" decompression="none" tarfile="">
http://www.aos.princeton.edu/WWWPUBLIC/tapio/arfit/adjph.m</ACCESS>

<SIGNATURE>
<CALL>adjph(x)</CALL>
<INPUT> <PARAMETER> <ID>x</ID>
...
<DESCRIPTION>complex matrix</DESCRIPTION>
</PARAMETER> </INPUT>
<OUTPUT> <PARAMETER> <ID>ox</ID>
...
<DESCRIPTION>complex matrix</DESCRIPTION>
</PARAMETER> </OUTPUT>
</SIGNATURE>

<ENVIRONMENT> <ENGINE><MATLAB></MATLAB></ENGINE> </ENVIRONMENT>

<CONTENT>
<TYPE><MSO_TYPE> <SCRIPT format="ascii" language="matlab"></SCRIPT> </TYPE></MSO_TYPE>
<DESCRIPTION>
<URL>http://www.aos.princeton.edu/WWWPUBLIC/tapio/arfit/</URL>
% ox=ADJPH(x) produces a complex matrix ox
% obtained by multiplying column vectors in x
% with phase factors exp(i*phi) such that the
% real and imaginary parts of column vectors in ox
% are orthogonal and the norm of the real part is
% at least the norm of the imaginary part.
</DESCRIPTION>
<KEYWORDS>
<KEYWORD>estimation</KEYWORD>
<KEYWORD>decomposition</KEYWORD>
<KEYWORD>autoregressive model</KEYWORD>
</KEYWORDS>
<ABSTRACT>
<URL>http://www.aos.princeton.edu/WWWPUBLIC/tapio/papers/arfit.html</URL>
<URL>http://www.aos.princeton.edu/WWWPUBLIC/tapio/papers/arfit_alg.html</URL>
</ABSTRACT>
</CONTENT>
</MSO></OBJECT>
```

Figure 6: A method session object (MSO) for a Matlab routine

and to provide trading support for applications in investment and risk management.

The non-commercial *Network-Enabled Optimization System (NEOS)* [14] of Argonne National Laboratory can also be regarded as an electronic marketplace for computational services. NEOS allows users to solve complex optimization problems via a Web interface (http://www-c.mcs.anl.gov/home/otc/Server/neos.html) and a proprietary client that users may install locally. The CGI-based system gives access to a large number of related services provided by research institutions in the US and Canada.

5 Conclusions and Open Problems

Many of the functions performed by GIS seem to be amenable to a business model that is fundamentally different from the one we see today. At present, GIS users typically own the hardware and software they use. They pay license and maintenance fees to various vendors and they have to train their staff in using the system. The alternative would be a service-oriented approach where users make their input data available to some GI service center that performs the necessary computations remotely and sends the results back to the user. Customers pay only for that particular usage of the GIS technology – without having to own a GIS. The system is open for anybody to participate as consumer or provider of services. We believe that with such an approach the number of consumers will be much greater than the number of GIS users today. We also expect customer satisfaction to increase.

Our MMM system is one example of a middleware to implement such an electronic marketplace and associated business model. MMM's architecture is based on object request broker technology. Its support for consumers includes features for browsing, searching, and querying. Support for providers concentrates on the registration of new methods and on the related management of metadata.

Before electronic marketplaces for spatial services will become commonplace, however, several critical issues will have to be resolved.

First, the development of appropriate licensing and (micro-)payment schemes is still work in progress. It is crucial for the success of Internet marketplaces that service providers can be certain to collect fees from all customers that use their services (directly or indirectly). Systems like MMM are independent of any particular payment scheme. They can in principle be combined with any of the major systems. The main problem is economical, not technical: what prices are consumers willing to pay, such that it is still attractive for potential providers to enter the market?

Second, one needs ways to guarantee confidentiality for the input and output data transmitted. If the service in question requires the input data in its original form, only organizational measures are possible to identify trustworthy service providers that maintain confidentiality. In many cases, however, a service can be performed on some transformed version of the data (see Fig. 7). The traveling salesman problem is a typical example of this case: in order to solve a given problem P, one can first apply a series T of geometric transformations (such as rotations or scalings) to P, then ask the service provider to solve the resulting problem $T(P)$. Once the service provider has found a solution S of $T(P)$, one obtains the solution of the original problem P by applying T^{-1}, the inverse of T, to S. This way the service provider never has access to the input and output data in its original form. It is an open research problem to define problem classes for which this kind of transformation is possible, i.e., where $S(P) = T^{-1}(S(T(P)))$ or, more general, for which there is *some* transformation (not necessarily T^{-1}) to generate $S(P)$ from $S(T(P))$.

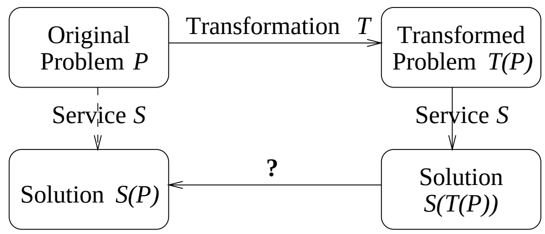

Figure 7: Possible transformation of a problem P to maintain data confidentiality.

Third, there is the related problem of *certification* of services. How can we make sure that a service does what it promises to do, in the desired quality? Some providers may already have acquired a reputation in their respective field. Others, however, may have to submit proofs of their competence and trustworthiness before customers decide to use their services.

Fourth, the usage and configuration of services should not be too complicated. Many GIS vendors pride themselves on the *turnkey* nature of their systems: setup efforts are minimal, and one can start using the system shortly after purchase. This may not always be the case in a digital marketplace type of situation, where users have to select and combine the services they need before they can use them.

Acknowledgments

Support from the German Research Society (DFG grant nos. SFB 373/A3 and GRK 316) is gratefully acknowledged. We would also like to thank the members of the MMM team for their contributions to this paper: Matthias Asche, Brian Fox, Kathrin Gayer, Louis Perrochon, and Gerrit Riessen.

References

[1] D. J. Abel. Spatial Internet marketplaces: A grand challenge. In M. Scholl and A. Voisard, editors, *Advances in Spatial Databases*, volume 1262 of *LNCS*, Berlin/Heidelberg/New York, 1997. Springer-Verlag.

[2] D. J. Abel, V. Gaede, K. L. Taylor, and X. Zhou. SMART: Towards spatial Internet marketplaces. Technical Report SIS-1997-01, CSIRO, Canberra, Australia, 1997.

[3] G. Almási and V. Jagannathan. Integrating the WWW and CORBA-based environments. In *Proceedings of the Fourth World Wide Web Conference*, 1995. See also http://webstar.cerc.wvu.edu/www4/www4.html.

[4] P. Becker. An embeddable and extendable language for large-scale programming on the Internet. In *Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS'96)*, 1996.

[5] P. Bernstein. Middleware: A model for distributed system services. *Communications of the ACM*, 39(2):86–98, 1996.

[6] H. K. Bhargava, A. S. King, and D. S. McQuay. DecisionNet: An architecture for modeling and decision support over the World Wide Web. In T. X. Bui, editor, *Proceedings of the Third International Society for Decision Support Systems Conference, Vol. II*, pages 541–550, Hong Kong, 1995. International Society for DSS.

[7] H. K. Bhargava and R. Krishnan. The World Wide Web: Opportunities for OR/MS. *INFORMS Journal on Computing*, 1998.

[8] H. K. Bhargava, R. Krishnan, M. Casey, D. Kaplan, S. Roehrig, and R. Müller. Model management in electronic markets for decision technologies: A software agent approach. In R. Sprague, editor, *Proceedings of the 30th Hawaii International Conference on System Sciences*, 1997.

[9] H. K. Bhargava, R. Krishnan, and D. Kaplan. On customized access to a WWW-based distributed network of decision support services. In T. X. Bui, editor, *Proceedings of the Third International Society for Decision Support Systems Conference, Vol. II*, pages 551–560, Hong Kong, 1995. International Society for DSS.

[10] H. K. Bhargava, R. Krishnan, and R. Müller. Decision support on demand: Emerging electronic markets for decision technologies. *Decision Support Systems*, 19(3):193–214, 1997.

[11] H. K. Bhargava, R. Krishnan, and R. Müller. Electronic markets for decision technologies: A business cycle analysis. *Journal on Electronic Commerce*, 1(4):109–127, 1997.

[12] H. K. Bhargava and C. G. Tettelbach. A Web-based decision support system for waste disposal and recycling. *Computers, Environment, and Urban Systems*, 1998.

[13] R. F. Boisvert. The guide to available mathematical software advisory system. *Math. and Comp. in Simul*, 31:453–464, 1989.

[14] J. Czyzyk, M. P. Mesnier, and J. J. Moré. The networked-enabled optimization system (NEOS) server. Preprint MCS-P615-1096, Mathematics and Computer Science Division, Argonne National Laboratory, March 1997.

[15] V. Gaede. Execution planning in Internet marketplaces. Discussion paper, Sonderforschungsbereich 373, Humboldt-Universität zu Berlin, January 1998.

[16] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. Integrating and accessing heterogeneous information sources in TSIMMIS. In *Proceedings of the AAAI Symposium on Information Gathering*, pages 61–64, 1995.

[17] K. Gayer. Metaobject repositories for managing services in electronic marketplaces. Discussion paper, Sonderforschungsbereich 373, Humboldt-Universität zu Berlin, January 1998.

[18] O. Günther, R. Müller, P. Schmidt, H. Bhargava, and R. Krishnan. MMM: A WWW-based approach for sharing statistical software modules. *IEEE Internet Computing*, 1(3), 1997.

[19] D. Ingham, M. Little, S. Caughey, and S. Shrivastava. W3Objects: Bringing object-oriented technology to the Web. *World Wide Web Journal*, 1995. http://www.w3.org/pub/WWW/Journal/1/ingham.141/paper/141.html.

[20] R. Kalakota and A. B. Whinston. *Frontiers of Electronic Commerce.* Addison-Wesley, Reading, Mass., 1996.

[21] P. Merle, C. Gransart, and J.-M. Geib. CorbaWeb: A generic object navigator. In *Proceedings of the Fifth International World Wide Web Conference*, Paris, 1996. INRIA. http://www5conf.inria.fr/fich_html/papers/P33/Overview.html.

[22] The Open GIS Consortium and the OGIS Project, 1997. URL http://www.opengis.org.

[23] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogenous information sources. In *Proceedings of the 11th International Conference on Data Engineering*, pages 251–260, 1995.

[24] L. Perrochon and R. Müller. Integrated scientific computing in global networks. In H. El-Rewini, editor, *Proc. 31st Hawaii Int. Conf. on System Sciences*, volume VII, pages 628 – 636, 1998.