ATLAS2000 – Atlases of the Future in Internet

M. Friedrich*(mafri@ipg.uni-freiburg.de)
M. Melle†(melle@informatik.uni-leipzig.de)
D. Saupe‡(saupe@informatik.uni-leipzig.de)

July 1998

Abstract

In geography new measuring techniques and imaging modalities have led to a huge amount of distributed data requiring new digital processing techniques. Environmental monitoring programs and analysis of global change in geography, meteorology, and climatology require an interdisciplinary usage of knowledge about the *ecological system earth*. Until recently the traditional atlas has been the primary tool for collection and dissemination of geographical knowledge about the earth. To advance to concepts of the atlas it is necessary to work on a methodological base different from that discussed before under the slogan *digital atlas*. The new tools that are developed in this project will permit an interactive, individual, and problem-related representation, the combination, modeling, and the interchange of multidimensional spatial and temporal data sets. A number of theoretical and practical aspects will be addressed, e.g., the investigation and interpolation of the space-time-continuum, the interpretation of data using different scales, and the use of subject-specific models for representation of measured and simulated data. Within the scope of the project some goals are in the fields of the development of hierarchical methods for compression, visualization and data access, thus enabling efficient handling of distributed resources in worldwide data and computer networks. Another task of the project lies in the development

*Institute for Physical Geography, University of Freiburg i.Br., Germany
†Institute for Computer Science, University of Leipzig, Germany
‡Institute for Computer Science, University of Leipzig, Germany

of didactical concepts for digital use of scientific data and models by a wide class of users. The project is a joint venture of the Institute for Physical Geography at the University of Freiburg and the Institute of Computer Science at the University of Leipzig.

1 The Digital Atlas - A New Approach

An atlas has always been an important aid for the gathering and dissemination of geographical data. The recent advances in geodata processing has furthered a reorientation in the requirements of digital atlases:

- ▷ The basic function of an atlas is the production of maps. The distribution of these maps can occur as either hard copies or in digital form as CD-ROM (active distribution) or in the Internet (passive distribution).
- ▷ The advances in digital image processing provide new possibilities for the digital atlas such as zooming functions and interactive selection of single map layers. Ormeling [3] called an atlas which provides these functions an *interactive* atlas (in contrast to the usual *view only* atlas). If such maps are available on Internet they are called a *map server*. There are many map servers in the Internet, but they always contain one or several drawbacks such as weak cartographic presentation, lack of image processing for the viewer, weak resolution, unknown data sources and so on.
- ▷ A new challenge for a digital atlas is the inclusion of maps resulting from geoscientific models such as maps from weather forecasts or similar less demanding simulations. These models can be derived numerically (e.g. simulation of wind fields) or statistically (e.g. regionalization of precipitation patterns). They provide data for visualization of complicated processes and spatial interpolation of place-depended time series.
- ▷ The integration of these models into digital atlases and the dissemination of maps and atlases in the Internet would improve the use of geoscientific knowledge/data in problem oriented spatial analysis for many ecological problems. Unfortunately, there are no map servers with adequate models in the Internet.

Thus, the main goal of ATLAS2000 is the investigation of various models with regard to their potential for interactive usage and their application for data sets from a provided data base or even for data created by the user. In analogy to the map server a concept integrating interactive maps with various models is called a *Model server*. The model server in ATLAS2000 will contain numerical models, which may need considerable computing power, to simulate specific meteorological phenomena and regionalization models, which require considerable expert knowledge.

ATLAS2000 offers its own climatological database and several spatial data sets of the Upper Rhine Valley. These data sets can be used as an introduction to the basic functions of the digital atlas and provide examples for the user. The user can then apply the various models for a comprehensive climatological analysis of the Upper Rhine Valley.

During the first phase of the project, various methods for the implementation of stand-alone programs in the Internet were analyzed. The tests were carried out with a model of the nocturnal wind flow patterns in the Upper Rhine Valley and its surroundings. The user can browse through a given simulation of the region of Freiburg i. Br. and interactively change the model variables and the area covering data sets (digital terrain model, land use classification). The model is programmed in C, which requires a client server system in Java, because of the rigid security limitations in common web browsers. On the server side the program manages the configuration files of the model and the model itself as an extern process. Finally, a tool was implemented which allows the import of user specified input data from any web server. Thus, the limitations of the atlas are based entirely on the theory of the individual model.

One of the problems of numerical models in the model server may be the high computing demands. For these cases we are working towards a solution using distributed computing.

2 Calculation Based On A Server Side Model

We have developed an architecture offering geographical model interaction and presentation. This can be done by two different strategies. One is to place the main part of calculation on the client side with the server providing the data. This can be an option only for users with sufficient resources (calculation power, main memory, hard disk memory). Since we do not assume the availability of such resources on the client side we favor the second strategy which places the calculation on the server side. The points from the following checklist should be observed in the design of the corresponding architecture:

1. *User Data Access*: Although the project provides a propriortory data base, the user must also be enabled to attach his own data to the model ([2]). These data may come from different sources: Typically, data are available from databases, but data can also come from distributed resources, from different locations in the Internet and various Database Management Systems (DBMS, figure **??**). A uniform database access has to be developed called *Data Engine* (figure **??**) which is the application server in the three-tier Java to DBMS access hierarchy (JDBC, [5]).

2. *User Classification*: In the Internet various user classes exist and so also ATLAS2000 has to handle these (sorted by knowledge with respect to geographical models): rookies and novices, pupils, students, scientists. Their knowledge differs not only with respect to the geographical model and its associated data but also about geography and computer science in general. The user has to be assisted by a didactical interface depending on his user class. The didactical interface is provided by the *Model Engine*. Methods for user classification have to be developed.

3. *Access Verification*: The user data have to be protected against unauthorized access. This protection depends on both data (database) and user class. User access to data and resources should be filtered by a *firewall* ([4], figure **??**).

4. *Hierarchical Geographical Model*: The execution of the geographical model applied to the raw data may consume high computing times, thus, prohibiting interactive investigation of the model and the data. We propose to build hierarchical models that begin by computing at coarse scales and that progress to finer scales later on. This way interactivity becomes possible again. If necessary the refinement in granularity can be done only for a region of interest, which is selectable from the geographical model time and/or space. The hierarchical model is implemented on the server side and will be handled by the Model Engine (figure **??**).

5. *Progressive Data Transfer*: The output data have to be transmitted to the client for visualization. While data are calculated by hierarchical

geographical models and data often need a long time for transmission to the client the transmission should be done progressively and compressed. Progressive transmission means that coarse scale visualization is transmitted first, while layers of increasing detail are added later.

6. *Model Combination*: In ATLAS2000 the user should be able to combine geographical models to investigate interdependencies between the modeled variables. Such studies should further be supported by multilayer viewing capabilities. One of the problems with this requirement is that in the concurrent hierarchical simulation of all considered models the level-of-detail must be matched or suitably adapted (*Model Engine*).

7. *Calculation Performance*: Calculating a geographical model may need considerable time. But because of the requirements of real-time interaction between user and geographical model the calculation performance has to be scaled-up:

   ▷ *Implicit*: by using special hardware, e.g. using multi processor systems which perform the task in parallel, or by using optimized libraries for the server.

   ▷ *Explicit*: by partitioning the task in subtasks which will be carried out separately on different computers.

   The partitioning of a task has to be supported by the geographical model and can be done explicitly by the *Calculation Engine* or a human operator. The subtasks are distributed to heterogeneous computers via the Internet. It is possible to use different communication standards here: Remote Method Invocation (RMI) for Java, Common Object Request Broker Architecture (CORBA), etc. For an optimal partitioning these communication costs must be accounted for, as always in distributed computing.

8. *Data Compression*: Calculating a geographical model can result in a large amount of data that has to be transfered to the client for further calculation and visualization. Here it is necessary to use suitable data set structures (e.g., [1] specification) and data compression that increase the efficiency of data transmission.

9. *Data Visualization*: The resulting data can exist in multi dimensional space and time and, moreover, are given at a certain scale. Different views on this data are possible and therefore this data should be transfered in a *raw* form. After transmission it can be used for visualization using a library offering different views, scales and layers (maps, 3D views, VRML, tables). The user interface for the visualization should be adapted to the user class (*Visualization Engine*).

Figure 1 shows how a user's SQL statement is interpreted by the ODBC system. The ODBC system is called by the JDBC driver from Java. The result of the action is caught by the JDBC driver and can be handled by Java. This simple two-tier model can be improved by a three-tier model which encapsulates the database access. This makes the database access virtual, allows distributed access and maps SQL statements in an object oriented manner to Java. This layer is then called *Data Engine*. The access to user data will be implemented here.

The data access granted by the *Data Engine* give the *Model Engine* the possibility to work on this data and answer the user's geographical model changes. The *Calculation Engine* spawns the calculation of the geographical model by sending pre-partitioned subtasks to different computers. The *Visualization Engine* handles requests from the user and visualizes the results. Based on the *Visualization Engine* the application or applet is the interface (front end) to the user depending on his user class.

References

[1] *Spatial Data Transfer Standard*. http://mcmcweb.er.usgs.gov/sdts/.

[2] Wolfgang Benn and Ingo Gringer. Zugriff auf Datenbanken "uber das World Wide Web. *Informatik-Spektrum*, 21:1–8, 1998.

[3] Ferjan Ormeling. Konzeptionelle Konsequenz f"ur die Bearbeitung elektronischer Atlanten. In *Wiener Schriften zur Geographie und Kartographie*. 1996.

[4] Joachim Posegga. Die Sicherheitsaspekte von Java. *Informatik-Spektrum*, 21:16–22, 1998.

[5] Sun Microsystems, Inc. JDBC Guide: Getting Started. Technical report, Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A., 1997.

A Pictures

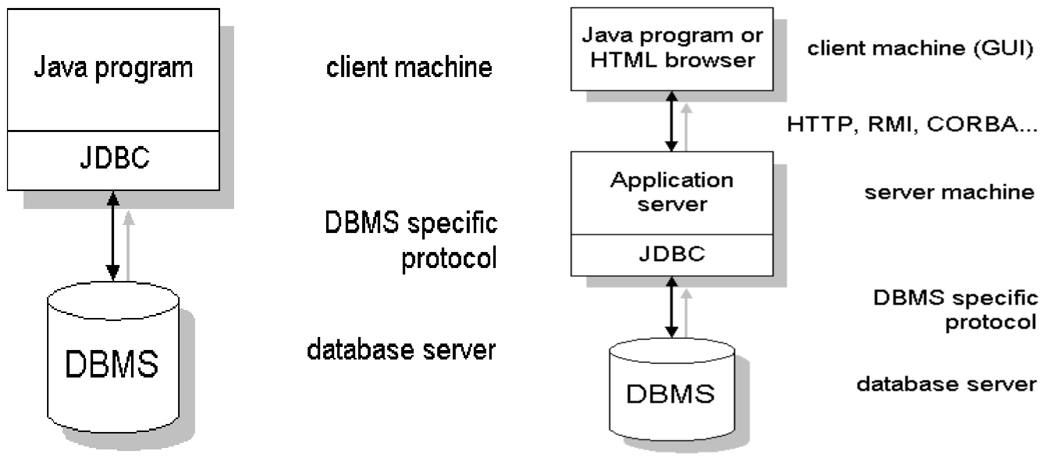

Figure 1: Two-tier and three-tier model for database access using JDBC connection in Java. This required a JDBC driver that can communicate with the particular database management system being accessed.

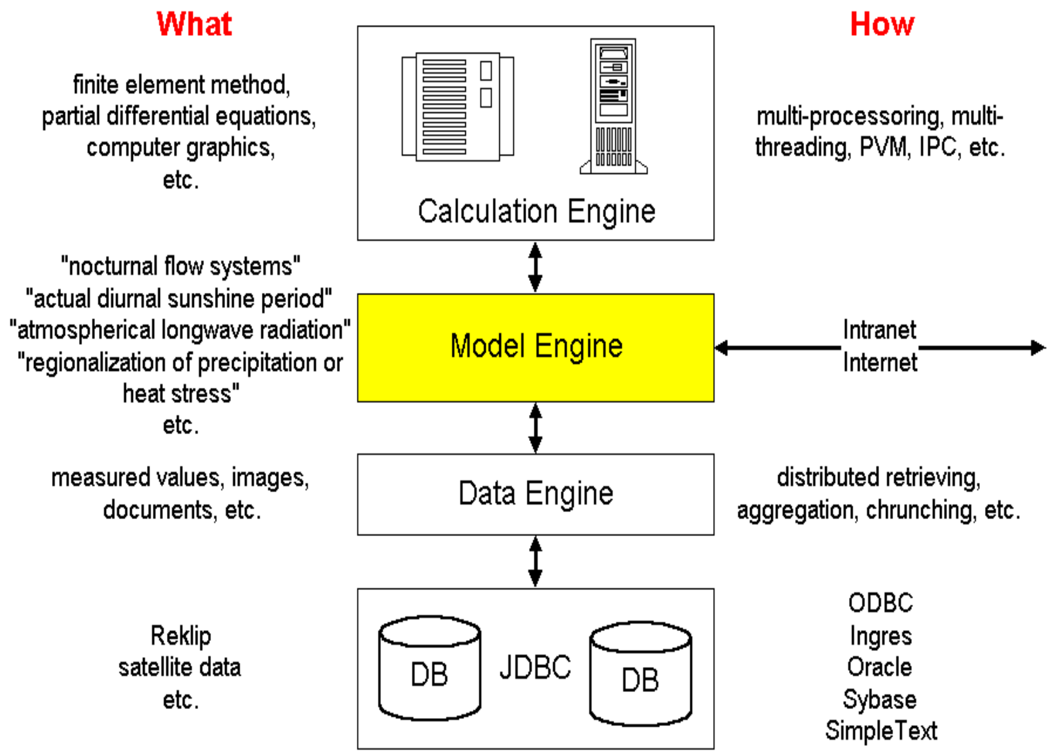

Figure 2: The server side of the ATLAS2000 architecture.

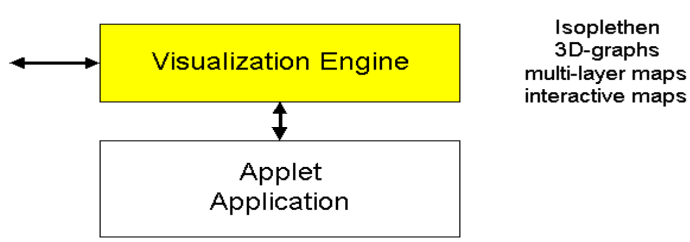

Figure 3: The compute power on the client side is assumed to be low. After transferring data from server only the visualization is done locally using appropriate plug-ins. Also the model interaction from the client to the server side is handled by this client plug-in.

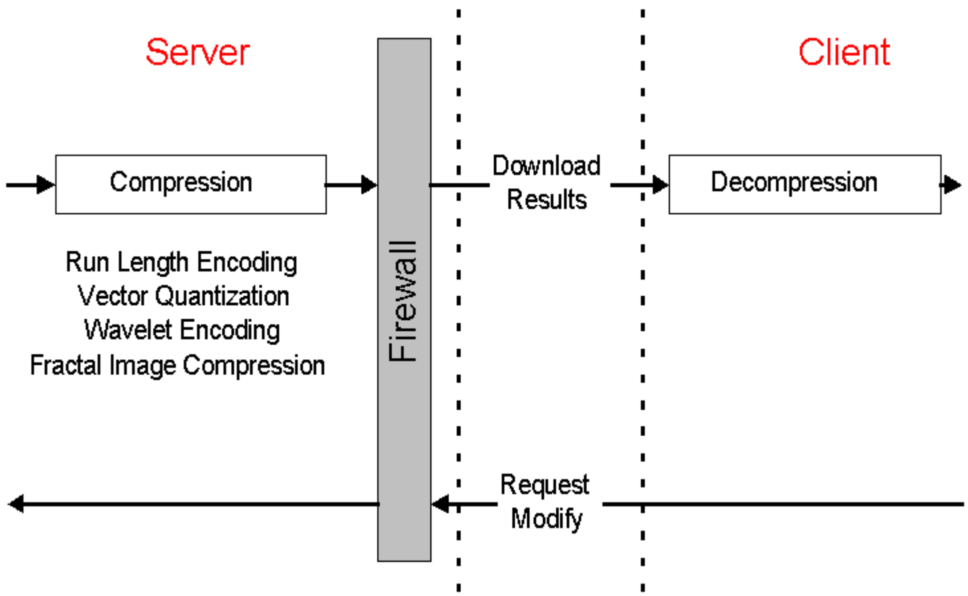

Figure 4: Network connection needs compression of data and secure information transfer.

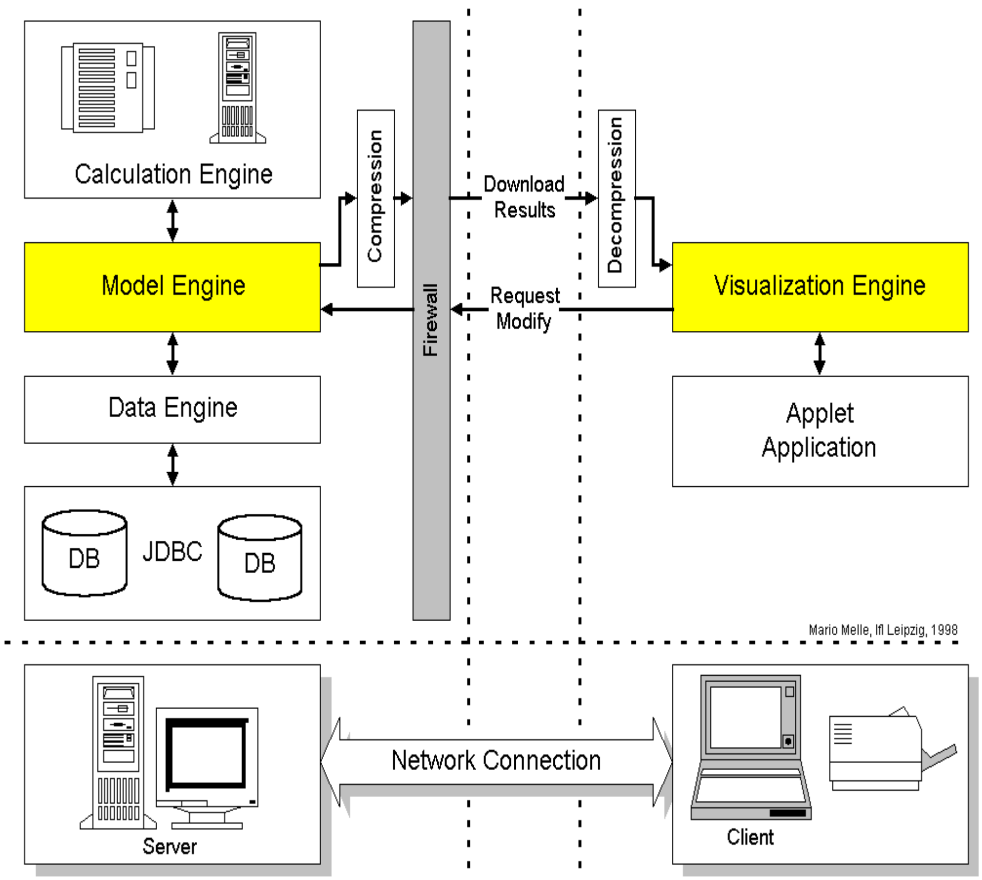

Figure 5: The overall structure of the model for ATLAS2000.