Model Validation in Web Environment - Progress in Distributed Modeling

Frank Sellerhoff [1], Peter Milbradt [1], Rainer Lehfeldt [2]

Introduction

This paper reports on progress made in the field of distributed modeling since the 1992 presentation of Lehfeldt and Holz [1] "Model Validation in X-Environment" at the Ninth International Conference on Computational Methods in Water Resources in Denver. Their paper illustrates an efficient user interface for model validation in coastal engineering consisting of a multi-window environment which is entirely based on the standardized X/Window graphics system. New opportunities with networked environments and possible implications for the working processes of coastal engineers were outlined.

We now concentrate on the implications of current developments in information- and communication technology for application software in coastal engineering. The Worldwide Web (WWW) with embedded technologies like cgi, ActiveX, ASP, Java, JavaScript to name just some of the better known, now defines de-facto standards in the internet which can be utilized for efficient distributed project handling.

Using the Java programming language we show a unified software approach with components for graphical user interfaces, presentation graphics, server applications and data base access. Having all of these facilities at hand is of great advantage for model assessment in distributed environments.

We show innovative aspects due to new technologies and also depict techniques which prove rather invariant in retrospect.

Retrospective

The paper on "Model Validation in X-Environment" describes technical aspects of model validation with a standardized graphical user interface in computer networks as of 1992. In principle, the UNIX operating system in combination with the X/Window graphics system are sufficient to set up a standardized graphics oriented application in the internet. Programs are started explicitly on a networked computer which in turn functions as an application server. By specifying another networked computer as display, any graphical output and all interactive events are then redirected to this client station by the application. The work load of number crunching algorithms is carried out on the application server while graphics intensive actions are carried out by an X-server on the computer of the X-client. The communication is based on an X-protocol which handles graphics primitives and all their attributes.

This scenario can be set up between any computers in the internet, provided the linked stations support the X/Window system which relies on the TCP/IP protocol. The overall performance of an application depends on the capacity of each computer involved and the amount of information transmitted through the X-protocol. Software portability is achieved by using the standard C programming language in combination with X/Window libraries.

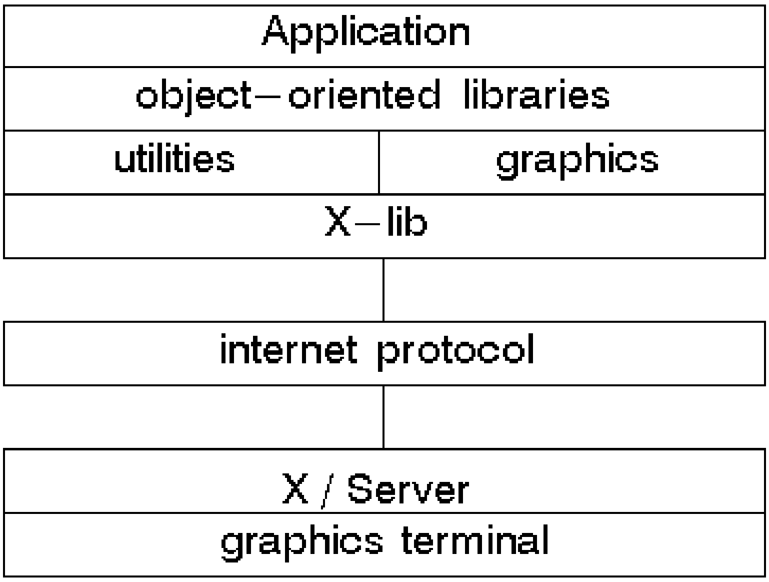

Figure 1: X/Window based internet application.

Modern project handling

Due to the rapid evolution of World Wide Web technologies, computers are used primarily as tools for communication processes with numerical modeling and simulations now being one of many fields of applications. Typical project handling has changed over the last few years in that ever more interdisciplinary teams are working at different locations on common tasks. Through the internet distributed expert knowledge from all areas of numerical modeling and software utilities become readily available and can be focused on new projects.

Natural physical processes ordinarily consist of a network of interactions due to hydrological and meteorological phenomena. Attempts to model these processes introduce simplifications which do not adequately represent the delicate balance in nature.

Simulation models are taking more and more of these interactions into account in order to come close to the natural network of interactions. A fundamental characteristic in the development of numerical simulation models for coastal engineering is the **integrated view on the physical phenomena** under study. If we consider the presence of waves, for example, to be relevant for proper assessment of morphodynamic processes, any numerical simulation must not only include these waves but also the corresponding global hydrological and meteorological circulation patterns for proper cause and effect analysis. There is a trend to no longer use modularized hierarchies of stand alone models simulating the individual physical phenomena and their interactions. Our approach is an **integrated modeling technique** [2] with modularization at the level of the governing equations. In analogy, model hierarchies of extensive domains can be established in order to minimize effects of boundary conditions and initial values on the simulation results.

Today, large scale engineering projects are usually no longer carried out by small groups concentrated at one particular location. A distributed project team co-operates on common objectives and each member develops sub models of his own, which are immediately available for all participants in a networked environment. Modern communication technologies thus help to bridge the gap between model makers and model users.

Prerequisites for distributed project handling

A number of basic services must be established in the network which cover essential strategies and methods in the field of coastal engineering. This is an essential prerequisite to guaranty efficiency within the distributed project team. Compared with 1992, these components have been ported to and optimized for the Web, thus eliminating dependencies on particular hardware or operating systems. The diversity of applications and the gamut of offers from the internet must be accompanied and supported by appropriate documentation. You cannot expect training for applets, say. Intuitive user interfaces with comprehensive help functionality are being developed [3].

Application server

Separating the activities of model makers and model users favors the encapsulation of repetitive task which rely on expert knowledge. Realization of such tasks are called services. The resulting applications proper need maintenance, documentation and revision handling. Efforts for installation and getting used to new functionality must be insignificant. Typical examples of services are data selection and analysis, scientific visualization, grid generation, fast overview simulations etc.

By using Java technology, there is no need for software adaptation to different platforms. Further more, Java applets will always be downloaded from the server in their latest version and Java servlets will be carried out in one single version on the server. Quality control of software through the model maker gains considerable reliability by "centralized maintenance" and standardization. The pertaining documentation is also available in platform independent HTML format.

Transparency and integrity of data

The distribution of project engineers and vast amounts of model data require network transparent availability of the complete project data base. Most currently available database management systems handle crucial requirements like concurrent data access, security and integrity.

Over and above this functionality, there is a need for concepts related to acceptable query performance and minimum transmission duration for mass data. The amount of data and the character of information held requires sophisticated visualization far apart from representation in ASCII - format.

Documentation and analysis

Distributed project handling brings about a division of the common task into modules which usually are based on each other. In order to report progress or availability of partial models, a standardized and central documentation must be provided which can be accessed at any time by any team member. Publishing on such documentation server is possible from each working place. In particular, experience and insight can be stored as additional meta information to simulation results and documented for interested persons.

Form and scale of many tasks in the engineering practice demand concepts and resources for efficient use of the information available. Examples for mechanisms to navigate HTML documents are presented by Brüggemann [4].

Software development with Java

Using the object oriented programming language Java in combination with other Web technologies implicitly guaranties independence of the applications from underlying platforms with respect to hardware or operating systems which has inhibited efficient co-operation in the sense described so far.

Programs written in Java are translated with a Java compiler into platform independent byte code [5]. A so called 'virtual machine' which is a software product written in ANSI-C and individually optimized for different platforms then executes the standardized byte code. The virtual machine is available as runtime environment for the shell, Web browser and Web server. Each runtime environment requires certain profiles of the code for optimal execution and fulfills different security levels. Java applications can be run in 3 distinct versions:
as stand alone application,

as applet in a web browser and
as servlet in the virtual machine of the web server

Java beans
Java beans promote the reuse of classes. The definition of standardized interfaces and naming conventions for methods lead to encapsulated components. Due to their standard form of documentation [6], Java beans can easily be handled with graphical software builders. They can be combined to new applications without detailed knowledge of their inner structure. The bean technology strongly supports the engineer with implementation and documentation of tools (programmed know how) for further use.

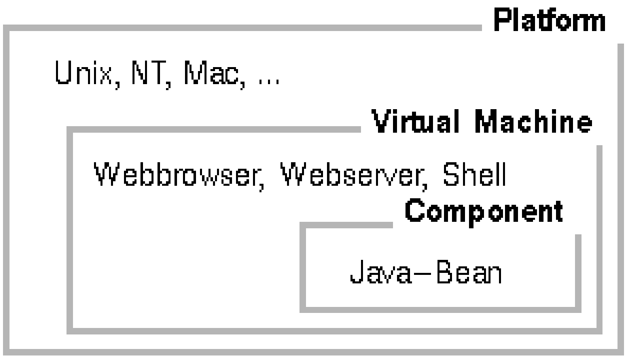

Figure 2: Reusing of software components in different runtime environments and operating systems.

Java packages
The Java development kit provides a number of useful class libraries (packages) for graphical user interfaces (awt), data bank access (jdbc) and container classes (collections) among others. Two classes essentially handle the communication between Java components installed on remote computers.

The Remote Method Invocation (rmi) package starts a service which is based on the TCP/IP protocol and calls methods of remote objects. It establishes references to objects which lie outside the limits of its own memory.

Communication between distributed Java applications is organized as low-level byte stream. At run time, the serialization package (serialization) transforms Java objects for transmission and deserializes received data in turn. This important mechanism is integral part of each virtual machine.

Data and methods
Object oriented software development reflects the work packages within an engineering task as a number of classes similar to the abstraction necessary for handling the project. An application program consists of objects which are instances of relevant classes. Information related to the application can be stored in object oriented databases simply by declaring the respective classes as persistent. Persistent storage in relational databases requires an additional level of abstraction for mapping the objects into tables and references.

For the time being, the schema of persistent classes is kept in a separate dictionary where information on the class attributes is stored. The next stage will be to store class methods as well in the data base. The aim is to provide a database including platform independent methods.

Field data for model validation
We discuss two examples of distributed work processes in coastal engineering with regard to model validation.
The first application deals with providing software for recurrent tasks and illustrates an on-line approach to grid generation. The second example describes compilation and management of a central data base which is an important corner-stone for proper model setup and assessment of simulation results.

Digital terrain modeling
Local agencies and coastal data centers maintain separate archives of bathymetry data, beach profiles, soundings of navigation channels and topographical information. Often these data sets are stored in incompatible data formats and different coordinate systems as well. Combining it all to create a digital terrain model for further use in numerical modeling can become a rather difficult job. For the time being it is hardly possible to produce a complete synoptic overview of all data that may be obtained without much effort.

If, on the contrary, all data sets are held in an identical form in the net with their particular geographic locations indicated in a map, the desirable and necessary overview can automatically be presented. Apart from possible classifications, all terrain data are online open to the public.

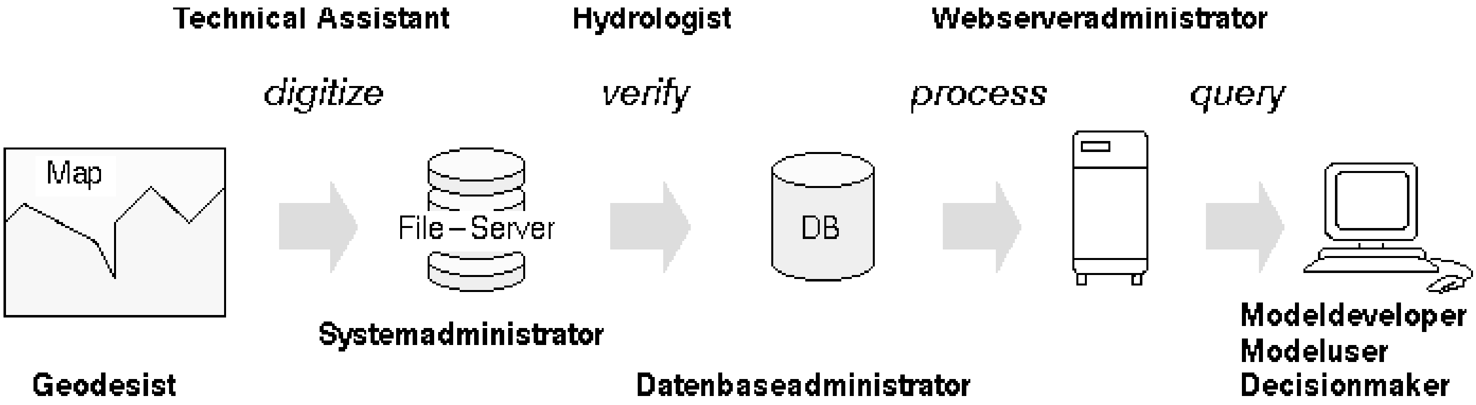

Figure 3: Tasks and fields of activity related with setting up digital terrain models in distributed working environments.

Information from surveys taken at different times in a particular domain can also be chronologically documented. Using simple visualization techniques in online comparison of historical system states, insight can now easily be gained concerning morphological changes along the coast. All interested institutes of research and administration can answer e.g. questions with regard to morphodynamics according to most recent data or download data necessary for numerical modeling.

Hydrological systems analysis
Time dependent data like water level recordings, wind fields or current measurements are also collected and archived by a number of different institutions. Just like in the case of terrain data, it is hardly possible to provide an easy access or overview of the distributed information.

Interactive maps indicate the locations of measurements, the parameters recorded and the periods of registration.
Again using simple visualization of time series and related statistical analysis provide an orientation within the data collection and may introduce specific extreme events. Online access to the underlying data is realized via these graphs by starting downloading for further analysis with the ftp services.

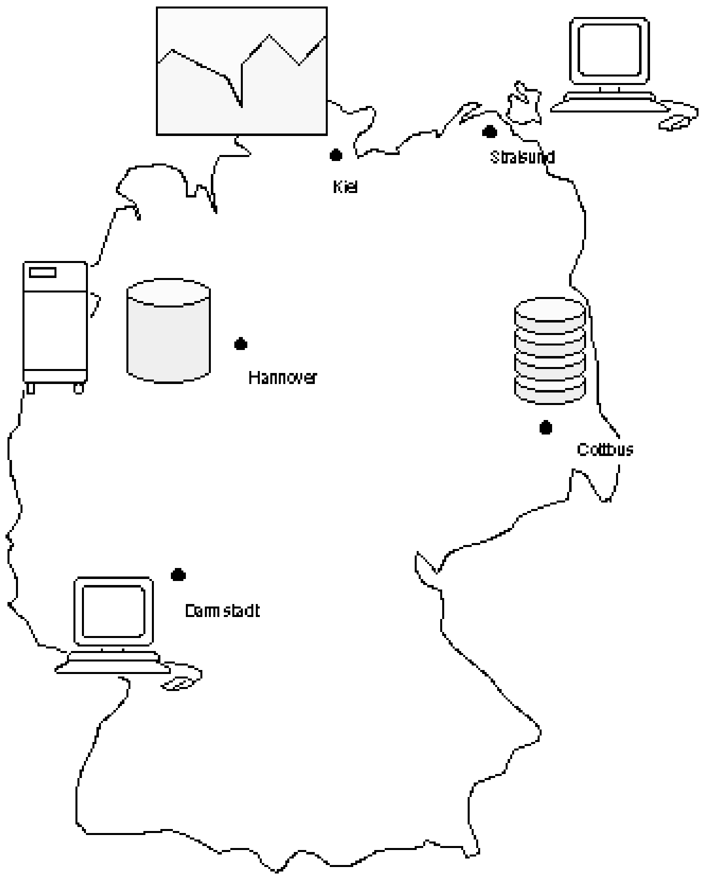

Figure 4: Distributed work packages in the MorWin Project [7] with regard to digital terrain models.

Distributed grid generation
Production of digital terrain models relies on efficient methods for grid generation. The tools for this recurrent service are provided and maintained on one particular server. Since grid generation needs a lot of computing resources, it is obvious to carry it out on the server via a servlet.

Possible constraints for grids to be used in numerical modeling may be

a bounding polygon limiting the computational domain,
criteria related to characteristics of the intended application like simulations of currents, waves or sediment transport or
a reference digital terrain model with information on bathymetry and topography.

Polygons and criteria can be expressed in compact form and are easy to transmit through networks. The service for grid generation in a distributed working environment is therefore designed as an applet-servlet combination explained below.

A web browser contacts the web server providing the desired services and documents. The documentation contains a link to another HTML page referencing an applet which is downloaded in its latest version. This interaction turns the web server into an application server. The applet then provides a user interface and contacts a particular servlet A on the web server which in turn sends a query to the data base (OODBMS - Poet). The requested information on available data sets regarding the digital terrain model are returned and appropriately displayed as lists or sensitive maps. The applet based user interface relates a graphical representation to the corresponding data set in a data base.

Furthermore, this applet can be used to specify further constraints for the grid to be created. It accepts a number of bounding polygons which can be manipulated or newly created with a polygon editor and the above mentioned intrinsic criteria for grid generation. Transmitting this set of information to another servlet B on the web server terminates the online session.

When servlet B is invoked in this manner all information necessary for the grid generation tools are received and the automated procedure based on established methods is carried out. The constraints according to specified criteria avoid later manual modifications and the resulting grid can be identically reproduced at any time. Once the triangulation has been completed, the requested mesh is returned via email in a preselected format. Alternatively, it may be provided on an ftp server for later downloading. Any request and action is registered in a log file.

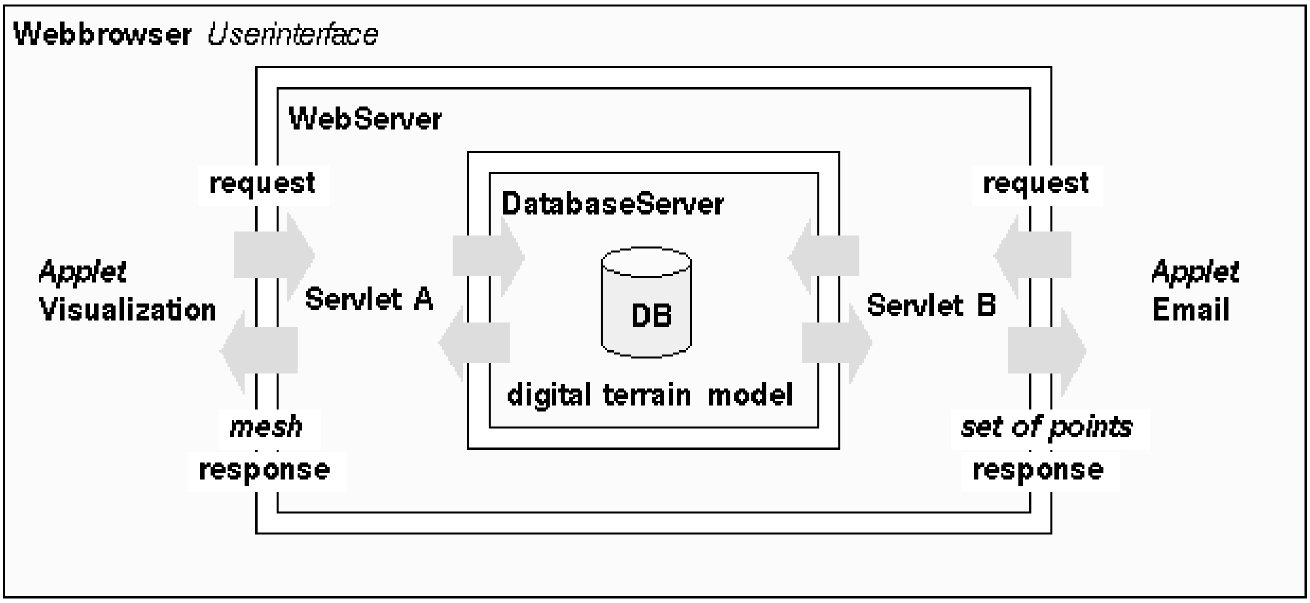

Figure 5: Application of Java applet-servlet technology for distributed grid generation.

Distributed modeling and simulation assessment
Apart from network transparent handling of geographical data and the creation of related grids, there is another type of data which is essential in assessment of coastal hydrodynamics. Time series of water levels, wind and wave data as well as current meter recordings are investigated for system analysis and identification of significant events. They are also used as boundary conditions in simulation models for hind- and forecasting purposes.

Within a distributed working environment numerous methods of analysis and tools for visualization are available as either client or server applications. Statistical procedures for instance constitute typical applications run on the server side with few statistical parameters being returned instead of transmitting bulk measurements. The client side then carries out visualization procedures appropriate for these results.

Simulation models in coastal engineering are usually set up as model hierarchies embedded in large scale circulation models which supply initial values and boundary conditions necessary to drive small scale models. Mapping hind cast data to the coastal simulation domain is another prominent example of client/server services in distributed modeling.

Future developments with Java beans for hydro-engineering purposes
Operation of simulation models within distributed working environments is still in its infancy. Efficient monitoring methods including visualization and trend analysis are being developed. In terms of client server technology using Java, the numerical model proper is operated as servlet application on a certain web server and interested clients peer into and/or manipulate the simulation progress via applet applications.

References

[1] Lehfeldt R, Holz P, 1992. Model Validation in X-Environment. Proc. 9th Int. Conf. Computational Methods in Water Resources, Denver, pp 733-740.

[2] Milbradt P, 1995. Zur Mathematischen Modellierung großräumiger Wellen- und Strömungsvorgänge. Dissertation. Institut für Bauinformatik, Universität Hannover.

[3] Sellerhoff F, Lehfeldt R, Milbradt P, 1997. Ein Konzept zur multimedialen Informationsaufbereitung wasserbaulicher Daten. http://morwin.bauinf.uni-hannover.de/modelle/infobasis/multimedia/konzept.html

[4] Brüggemann M, 1998. Dynamic Interactive Documents in Hydroengineering. Proc. 3rd International Conference on Hydro-Science and Engineering

[5] What is the java Platform? http://www.javasoft.com/nav/whatis/index.html

[6] Java-Beans Specification http://www.sun.com/beans/glasgow/index.html

[7] BMBF-research projekt "Morwin" http://morwin.bauinf.uni-hannover.de

---

1 Dipl.-Ing. Frank Sellerhoff, Dr.-Ing. Peter Milbradt
Institute of Computer Science in Civil Engineering, University of Hannover, Germany

2 Dr.-Ing. Rainer Lehfeldt
Federal Agency for Waterways, Kiel, Germany