Architecture of an information system in model-based environmental problem solving

[1]**Jolma, A.**, [2]**T. Kokkonen**, [2]**H. Koivusalo, and** [2]**H. Laine**

[1]Laboratory of Cartography and Geoinformatics, Helsinki University of Technology, [2]Laboratory of Water Resources, Helsinki University of Technology, E-Mail: **Ari.Jolma@tkk.fi**

Keywords: *information system; architecture of an information system; modeling; problem solving*

EXTENDED ABSTRACT

Environmental problems are often complex as they typically deal with several disciplines and affect large regions. Tackling such problems requires integration of computational tools. Solving environmental problems requires cooperation between the groups that develop the tools and the groups that use the tools and transfer of information and technology. The process is becoming more and more complex as the problems we try to solve become larger. We are faced with the task of designing an architecture of an information system (IS) for managing the problem solving process.

Attempts to overcome difficulties in solving complex real-world problems, such as environmental problems, include advances in the theory of complex systems, development of information system frameworks, and development of open standards. The theory of dynamic networks gives us tools to understand and possibly design better and more robust and resilient systems (Watts 2003). The idea of an information system framework for environmental modeling is an answer to the general need of linking a model to a database, to another model, to a GIS, or to other tools. In the field of geoinformatics there are several projects that develop open standards, while in the field of environmental modeling there are fewer such projects.

In spite of these efforts there is still a need for new approaches, which link and help to link work carried out in different communities. In this paper we discuss the linkages between simulation models, environmental data and end users, and make an attempt to define the design processes of an IS architecture in model-based environmental problem solving. We present key points of the development of a DSS for the CLIME (Climate and Lake Impacts in Europe) project.

The general IS development process is often divided into five phases: problem analysis, architectural design, development, implementation, and evaluation. In the analysis phase the problem in question is analyzed and an understanding is developed. In the design phase the architecture of the IS is prepared. The development phase deals with the actual construction of the IS. In the implementation phase the constructed IS is transferred to the work environment in which it will be used. The evaluation is the real-world test of the implemented system. From evaluation the process goes back to analysis.

The environmental problem, which the CLIME project studies, is climate change and its effects on lakes and the impacts of the changes in the lakes to the society. Besides the environmental and socio-economic domains, there are also other domains as the future DSS users and the scientists within the project. The CLIME project has a designated, but heterogeneous, end-user community, which is the primary target user group of the DSS.

The architecture of the CLIME DSS is at the highest level a network of four nodes in chain. The offline part of the system produces data, which is stored in a database. The user has access to the information in the database through the online part of the system. The information exchange is two-way in each connection, but, because of the structure, the user does not have a direct access to the database or to the offline system. This architecture is a, very generic, solution to the knowledge transfer problem.

Besides providing a platform for causal Bayesian networks, the design of the CLIME DSS includes visualization capabilities. The visualizations are prepared in the offline part of the DSS, stored in the database, and used in the online part.

Our conclusion is that the problem analysis and architectural design phases are not given enough thought considering how important they are. One reason for this may be that the methodological tools are not yet fully matured, which is understandable as the fields of problem analysis and software architecture are still rather new and developing. New innovations in the above fields could provide a promising avenue for integrating methods used by modelers, software developers, and decision makers in environmental problem solving.

INTRODUCTION

Environmental problems are often complex as they typically deal with several disciplines and affect large regions. Tackling such problems requires integration of computational tools, in particular models and geographic information systems (GIS), into information systems (IS). Different communities develop these tools. There are disciplinary experts, developers of simulation models, communities that develop software, and decision makers who apply information systems. Solving environmental problems requires cooperation between the above groups and transfer of information and technology. The process is becoming more and more complex as the problems we try to solve become larger. Clearly there is a need for design above the level of algorithms and data structures. We are faced with the task of designing an architecture of an IS for managing the problem solving process.

Attempts to overcome difficulties in solving complex real-world and environmental problems include advances in the theory of complex systems (Bar-Yam 2005), development of information system frameworks (Gregersen and Blind 2004, Rahman et al 2001), and development of open standards.

In the theory of complex systems we have seen many new ideas in the last twenty or so years. The theory of chaotic systems gives us tools to understand complexity and emergence (Holland 1998). The theory of dynamic networks gives us tools to understand and possibly design better and more robust and resilient systems (Watts 2003). The idea of a network is very simple to understand but it is sometimes hard to see what are or what should be the nodes, what do the links between them mean in our problems, and where to go after these have been identified. A general principle is to divide the whole into only a few connected domains (Miller 1956; Jackson 1995, 2001). Complex domains have an internal structure, which can be depicted also as a network. Some methods, for example those based on object orientation, are more rigorous and geared towards specific tasks like data modeling.

The idea of an information system framework for environmental modeling is an answer to, and outgrowth from, the need of linking a model to a database, to another model, or to a GIS. It is related to the idea of a decision support system (DSS), which is an analytical tool, designed to support a decision maker with the help of a database, models, and a human-computer interface. The technology enabling the linkage of models and databases along with the intricacies of linking, i.e., tightly vs. loosely coupled systems and specific techniques such as introspection, is often the focus of attention of research in information system frameworks. The problem of analysis and problem solving is lost into the interestingness of technology or science.

Projects that develop open standards are laudable community efforts. In the field of geoinformatics there are several such projects, many of them coordinated under the auspices of Open Geospatial Consortium, Inc. (OGC), while in the field of environmental modeling there are fewer such projects. Open standards may be descriptions of data formats, often based on the XML meta language, such as the geographic markup language (GML), or they may be service descriptions, such as the web map service (WMS).

In spite of these efforts there is still a need for new approaches, which link and help to link work carried out in different communities (see also van Deursen et al 2000). In this paper we discuss the linkages between simulation models, environmental data and end users, and make an attempt to define the design processes of an IS architecture in model-based environmental problem solving.

The paper is based on the authors' previous experience in environmental problem solving and information system development, and on the development of a DSS for the CLIME (Climate and Lake Impacts in Europe) project. CLIME is a cooperative, international research project aiming at developing a suite of methods and models that can be used to manage lakes and catchments under future as well as current climatic conditions.

1. INFORMATION SYSTEM DEVELOPMENT PROCESS

1.1. Five phases of the information system development process

The general IS development process is often divided into five phases: problem analysis, architectural design, development, implementation, and evaluation. In this paper we concentrate in the first three phases. In the analysis phase the problem in question is analyzed and an understanding is developed. In the design phase, which relies on creativity, architecture of the IS is planned. The development phase deals with the actual construction of the IS. In the implementation phase the constructed IS is transferred to the work environment in which it will be used. The evaluation is the real-world test of the implemented system. From evaluation the process goes back to analysis.

In the development phase the architectural design is mapped into a technical design, which uses specific programming languages, database designs, APIs, and other existing and specified technologies or products.

1.2. Problem analysis

The problem analysis is divided into two parts from the point of view of IS development: domain analysis and requirements of the user (Jackson 2001).

Typically in domain analysis the analyst considers issues like whether the domain in subject is static or dynamic, and if it is dynamic, what is the temporal scale of modeling. The table 1. spells out some questions the analyst asks in this process. Some of the questions may seem obvious but in reality are often forgotten. Jackson (1995) discusses these and other domain distinctions.

Table 1. Questions related to domain analysis for developing environmental DSS.

Characteristic	*Questions for the analyst*
Temporal	Are we mainly dealing with operational, tactical, or strategic issues? Can we consider something static? How to deal with mismatches between domains?
Spatial	Do the location, size, or topological nature of objects matter?
Organizational	What is the decision making process? What roles do people/institutions have? What are the information needs?
Tangible	What kind of existence is suitable for this domain? Many things related to modeling are very intangible, even subjective.
Active	How to deal with things happening by themselves in the domain? How to keep the system up-to-date?

Jackson's (2001) problem frames is a method of analyzing and describing problems, and matching them with well-known classes of problems. Jackson calls this task domain analysis. Problem frames are small networks of domains and required conditions (Figure 1. depicts the essential elements of problem frames). A required condition, which is linked to two or more domains, creates a named relationship between the domains. Jackson (2001) presents, describes, and analyses a number of problem classes. For example in his "workpieces" frame the workpiece is a domain, which is operated with the help of a machine using a finite set of operation requests, which is another domain.

From the point of view of modeling there are issues of: understanding the modeled system and cooperation between modelers, observational design and data collection, and parameter estimation, calibration, and validation. The result of modeling is usually some kind of formal description of the system. There are actually two steps in modeling: in the first the description of the system is the workpiece, and in the second (calibration) the set of the parameter values is the workpiece.

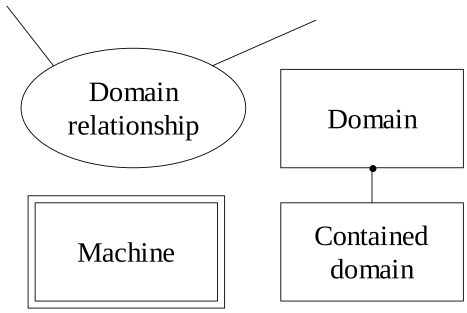

Figure 1. Types of elements in Jackson's (2001) frame diagrams. A named domain relationship is a condition that is required to hold. A contained domain is separate because it is a separate part in the problem, but all phenomena in it are also phenomena in the containing domain.

From the point of view of model-based environmental problem-solving there are issues like conducting a what-if analysis, cooperation between different model users, and understanding the modeled system with the help of the model. Getting from modeling to model use also presents the problem of knowledge transfer if the modeler and the model user are not the same person.

1.3. Architectural design

Garlan and Shaw (1993) introduced architectural design of an information systems as a new field of study. Their thesis was that design of software at this level is different from the design of algorithms and data structures. The research is progressing rapidly and the current state-of-the-art is, for example, to consider architecture of information systems, which use the Internet in many ways and whose boundaries are thus very difficult to define.

The architectural design of an IS is based on an increasing number of more or less readily available elements or components. The corresponding domain in the modeling world contains the different types of models. The architecture of an IS can also be examined as a network. A single node in an architectural design may be for example a processor, a user, a database, a client, or a server. The link represents a connection, typically information exchange, between nodes. Figure 2 illustrates how an IS may be depicted as network of elements, which easily become large as requirements and possibilities increase.

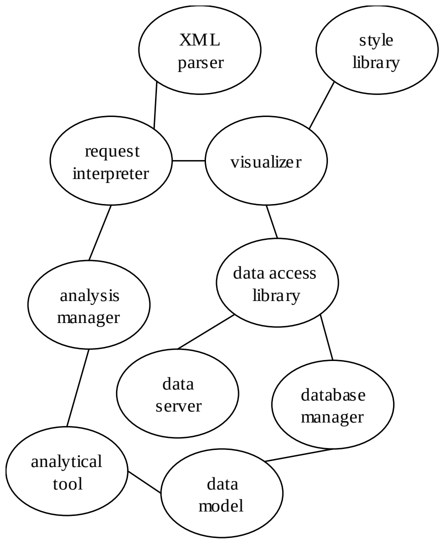

Figure 2. A collection of architectural elements of IS represented as nodes and connected for a hypothetical system.

In an IS an element often has a general role or a set of elements follow a general pattern. For example a processor node is typically seen as an element, which gets data and possibly processing instructions as an input, and yields as an output data, e.g., a number, a GIS layer, or a HTML document. Processors can be joined together in a chain or a tree to produce more complex processors or pipelines. An example of a pipeline is the chain of processors, which produces a viewable image. For example a map is an image, which is produced from spatial data in a couple of processors with the help of, e.g. scale and projection information and formatting information. The "visualizer" in Figure 2. could be a processor implementing a pipeline, where raw geospatial data is retrieved, overlaid, and rendered according to a style. These processing steps are done in separate, sequential actions. In A set of processors is commonly referred to as an "engine".

A user node is a human user of a system, who observes or examines the output of the system, provides input o the system, or interacts with the system. A database is a storage of data, whose structure can be designed with aid of a data model. A client node and a server node form an interacting, connected pair. The client asks the server to perform a task, or to produce some data or a document. A client may be connected to more than one servers and one server may serve one or more clients. The "request interpreter" could be a server, listening to user's requests, but it also is a client, which utilizes the services "visualizer" and "analysis engine".

Gamma et al (1995) introduced a formalism for describing common design patterns, which can be applied in development. Jackson (1995) emphasizes the distinction between design, which is more related to analysis, i.e., logical model of the real world, and development, which is more related to implementation, i.e., programming. The problem is that the distinction between "real world" and the programmed world is becoming harder and harder to make.

Types of models include simulation models, optimization models, causal Bayesian models, and data models. A simulation model contains a description of the simulated system, its state and how the state changes in time when something affects or does not affect the system. In the case of an agent-based simulation model the simulated system is divided into two parts: the agent or agents and the environment. The description contains also parameter names and parameter values. The internal structure of an optimization model contains the objective function, constraints, decision variables, parameters, and parameter values. The parameters may be the size of the system or other kind of a parameter. A causal Bayesian model is a directed graph, where the nodes are random variables, each of which represents a state of a domain, and the links are causal effects between these domains. In mathematical sense the links are joint probability distributions. A data model is a network, which describes the internal structure of data.

Architectural design is a mediator between the problem description and the technical design of real world tools. It should present a solution to the requirement of the problem the user has. It seems to be the concern of only a few people since decision makers are interested in practical problem solving, modelers in the internals of a model, and tool developers and especially software companies are interested in the real world tools of the development phase. The corresponding domain in modeling does have all the different model types but modelers are typically interested in only one type of a model at a time.

1.4. Development

The development of an IS used to be very much a programming task. Programming is still required but often the basic data types and algorithms are pre-defined and programming is needed for linking different tools together or it is done at a higher conceptual level (Cahegan and Macgill 2005). The development environments have become much more powerful but also complex, and the software development process requires much more attention.

The same basic functionality can be achieved with different technologies, but the technical or institutional requirements dictate the selection. The technical design has to consider issues like performance, reliability, usability, compatibility of different architectural elements, integration, and interoperability with other systems. The technical domain is notoriously ridden with commercial interests, protectionism, and fixed opinions.

Many development platforms are becoming more component-based. The benefit seems to be that the technical design may now be closer to the architectural design. On the other hand, skipping the architectural design as such is never a good idea since then the main point of architectural design, presenting a solution to a clear problem, is easily overlooked if focus is on technology. Many component platforms have also serious performance problems since the data exchange interface is implemented at high level and may become a bottleneck.

2. DEVELOPMENT OF THE CLIME DSS

2.1. The climate-catchment-lake-society problem studied in CLIME

The environmental problem, which the CLIME project studies and for which it aims to develop tools, is climate change and its effects on lakes and the impacts of the changes in the lakes to the society. The domains, which can be separated from the whole, are depicted in Figure 3. Besides these environmental and socio-economic domains, there are also other domains worth analysis in the CLIME project. The most important other domains are the future DSS users and the scientists within the project.

The CLIME project has a designated, but heterogeneous, end-user community, which is the primary target user group of the DSS. The requirement of the DSS user depends on the user. The user may be a lake manager, a water utility using the lake, a regional environmental manager responsible for many lakes, a national environmental policy maker, or a international policy maker or an authority. The problem is often to predict the effect of a climate change, e.g., on the ice regime of a lake, on the leaching of dissolved organic matter from the catchment, or on the eutrophication process of the lake and the estimation of the associated cost.

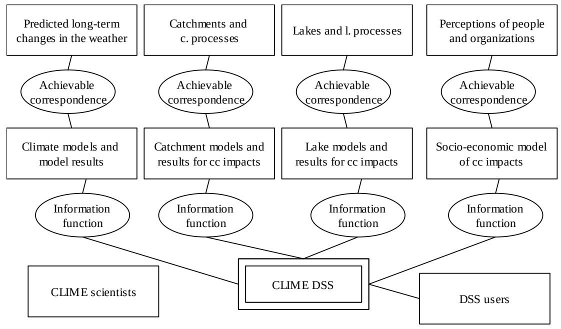

Figure 3. A frame diagram of a simulation model based DSS for climate change impacts on lakes implemented in CLIME. The links from the "scientists" domain are not showed. Scientists are connected to the real world through measurements and to the models through use. The connection from the scientists to the DSS is also via the information functions. An information function does not necessarily indicate an online connection.

In the CLIME project the above mentioned domains are studied separately and together and in the context of general understanding and in the context of specific problems or lakes. There are several domains of study in CLIME: describing and predicting the climate of Europe, describing some processes in catchments and lakes and predicting the direct effect of climate change on them, describing the regional coherence of lakes, describing and predicting the socio-economic consequences of changes in the lakes, and developing decision support tools (Anonymous 2001).

2.2. Information processing in CLIME

The approach to problem solving in CLIME relies on the application of environmental simulation models. There is the local computing conducted by each researcher and then there is the coordinated project scale computing. A major challenge is the chained execution of several simulation models, i.e., the regional climate models, the weather generator, the catchment models, and the lake models. In CLIME this is carried out by file-based, i.e., very loose, coupling of the models. The parallel calibration and execution of the same model at different sites is supported by a shared database of calibrated parameter values identified at different sites.

The shared CLIME database, depicted in Figure 4., is an attempt to collect and organize the knowledge obtained in the project. The database is part of the database of the DSS.

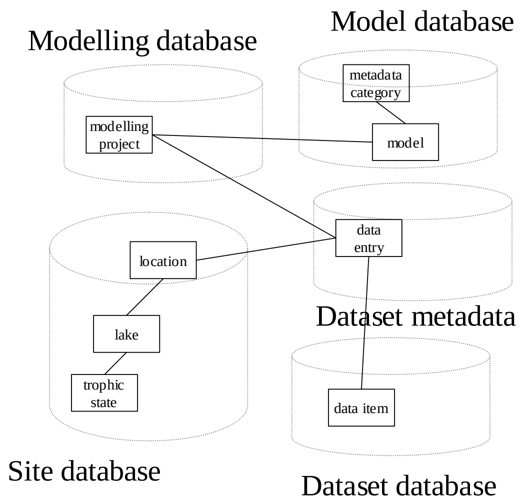

Figure 4. A simplified diagram representing the design of a part the CLIME database as a set of connected databases, which are composed of connected classes of objects.

2.3. Architecture of the CLIME DSS

The architecture of the CLIME DSS is at the highest level a network of four nodes in chain (Figure 5.). The offline part of the system produces data, which is stored in a database. The user has an access to the information in the database through the online part of the system. The information exchange is two-way in each connection, but, because of the structure, the user does not have a direct access to the database or to the offline system. This architecture is a – very generic – solution to the knowledge transfer problem.

A solution to the general prediction problem is attempted in CLIME with simulation models. The simulation models are run completely in the offline part of the IS. An attempt to present, regionalize and extrapolate the results will be made using a simpler probabilistic, causal models (Koivusalo et al. 2005). The causal models and their input data are prepared with the offline tools but the results of this work will be stored into the database for subsequent use in the online system. The prediction of the socio-economic response is attempted through sociological fieldwork, i.e., interviews and research, whose results can be incorporated also into the causal networks.

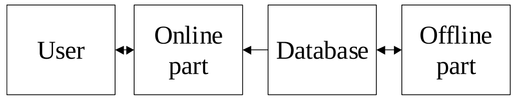

Figure 5. The general architecture of the CLIME DSS.

Besides providing a platform for the causal Bayesian networks, the design of the CLIME DSS includes visualization capabilities. The visualizations are prepared in the offline part of the DSS, stored in the database, and used in the online part (Jolma et al 2005).

The online system is the immediate solution to the decision support problem. It provides information of the CLIME project and its results. The results of the simulation models are processed with several tools in the offline system, and stored into the database for the online system. This information will give the user a possibility to comprehend the climate change phenomenon. The causal models are stored in the database and can be loaded into the online system. Use of the causal models give the user a chance to comprehend the effects of climate change and risks associated with them.

2.4. Development of the CLIME DSS

The platform of the CLIME DSS is, for the online part, the web. The database is only partly centralized and only a part of the central database is built on a relational database while the rest is a loose collection of files. The core of the online part of the DSS is a Java application, which contains custom code for handling the interface and a solver for Bayesian networks, and visualization tools. The DSS application relies for all its meta data and data on a custom server.

3. CONCLUSIONS

This paper discusses the common staged process for developing IS, in this case for model based environmental problem solving. The challenge seems to be the separation of the three stages but in the same time managing their connections. The IS development must also be coordinated with the modeling.

Our conclusion is that the problem analysis and architectural design phases are not given enough thought considering how important they are. One reason for this may be that the methodological tools are not yet fully matured, which is understandable as the fields of problem analysis and software architecture are still rather new and developing. New innovations in the above fields could provide a promising avenue for integrating methods used by modelers, software developers, and decision makers in environmental problem solving.

It is important to see that computational platforms are not solutions, and they must not be regarded as such, although it is equally important to see that good solutions usually need to be implemented as computational tools.

4. ACKNOWLEDGMENTS

This study was partly funded by the European Commission's Environment and Sustainable Development Programme under contract EVK1-CT-2002-00121 (CLIME).

5. REFERENCES

Anonymous. 2001. CLIME: Climate and Lake Impacts in Europe. The project website: http://clime.tkk.fi (accessed 23.9.2005).

Bar-Yam, Y. 2005 Making Things Work: Solving Complex Problems In A Complex World. Necsi Knowledge Pr. 306 pp.

Cahegan, M. and Macgill, J. 2005. Fitting the pieces together: a geocomputational problem-solving environment to integrate disparate analysis and visualization tools. GeoComputation 2005: The Art and Science of Solving Complex Spatial Problems with Computers. Ann Arbor, Michigan, USA, August 1 – 3, 2005.

van Deursen, W., Wesseling, C., and Karssenberg, D. (2000). How do we gain control over GIS technology? 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs. Banff, Alberta, Canada, September 2 - 8, 2000.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995) Design Patterns. Addison-Wesley.

Garlan, D. and Shaw, M. (1993). An introduction to software architecture. In V. Ambriola and G. Tortora (ed.), Advances in Software Engineering and Knowledge Engineering, Series on Software Engineering and Knowledge Engineering, Vol 2, World Scientific Publishing Company, Singapore, pp. 1-39, 1993.

Gregersen, J.B. and Blind, M. 2004. OpenMI: The essential concepts and their implications for legacy software. IEMSS 2004: Complexity and Integrated Resources Management. 6 pp.

Holland, J. (1998). Emergence: From Chaos to Order. Addison-Wesley. 258pp.

Jackson, M. (1995). Software requirements & specifications. Addison Wesley. 228 p.

Jackson, M. (2001). Problem frames. Addison Wesley. 390 p.

Jolma, A., Kokkonen, T., Koivusalo, H., and Laine, H. 2005. Computation of Information for Environmental Management from the Results of Regional Climate Model Simulations. GeoComputation 2005: The Art and Science of Solving Complex Spatial Problems with Computers. Ann Arbor, Michigan, USA, August 1 – 3, 2005.

Koivusalo, H., Kokkonen, T., Laine, H., Jolma, A. and Varis, O. 2005. Exploiting simulation model results in parameterising a Bayesian network – A case study of dissolved organic carbon in catchment runoff. ModSim 2005.

Miller, G.A. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, 1956, vol. 63, pp. 81-97.

Rahman, J.M., Cuddy, S.M., and Watson, F.G.R. 2001. Tarsier and ICMS: Two Approaches to Framework Development. MODSIM 2001 International Congress on Modelling and Simulation. December 10-13, 2001. Canberra, ACT, Australia. 6 pp.

Watts, D. J. (2003). Six degrees. W.W. Norton & Company. 374 p.