SYSTEMS FOR INTEGRATED GEOINFORMATION: stages of evolution

Morakot Pilouk

International Institute for Aerospace Survey and Earth Sciences (ITC)
P.O. Box 6, 7500 AA Enschede, The Netherlands
Tel. +31-53-4874358, Fax. +31-54-4874335
E-mail: MORAKOT@ITC.NL

XVIII ISPRS Congress, Commission II, Working Group 2

KEYWORD: GIS, Technology, Design, Systems, Vector.

ABSTRACT

The evolution of geoinformation technology in the last decade has resulted in a rapid change in the way geoinformation can be used and presented. The technological development stimulates the coherent use of information from multi-sources with multiple themes and dimensions to answer questions that demand increasingly complex spatial analysis. This requires to have a powerful geoinformation system (GIS) that can efficiently integrate, manage, process and present different types of geoinformation. The paper points out the importance of an integrated spatial model that well-represents real world objects and relationships among them. An integrated spatial model is considered a very important basis of a GIS. Four variants of system architecture are distinguished corresponding to different levels of integration. The variants are discussed with respect to fourteen criteria. Further development should be directed to structural integration using the integrated spatial model as a basis, as being the most promising approach.

1. INTRODUCTION

The scientific study of the earth to improve human's living conditions, the quality of life and to prevent or mitigate hazards and disasters by exploiting modern digital computing technology requires to represent the real world aspects under interest in a digitally constructed spatial model. Using a tool set provided by a GIS, a spatial model can be constructed and may allow to virtually perform many tasks that otherwise would be carried out in reality, which may be too difficult, too expensive or destructive. A high quality spatial model that represents well all relevant aspects of reality, i.e. real world objects of interest and relationships among them, is needed to make the results obtained from operations on the model comparable to actual operations in reality. The richness of operations on the model, which reflects the functionality of the system, depends on the complexity of the spatial model. As such the spatial model is decisive basis of the system. In present practice, many GISs tend to be developed by combining different spatial models and related functions such as for modelling of terrain relief and 2D terrain characteristics with thematic aspects (e.g. colour, texture). This is achieved by using digital terrain model (DTM) sub-system and typical current GIS software. The approach implies a spatial model whose components are still separately stored and are loosely related. As a consequence the model is not robust and may not well-represent the real world aspects.

In general, a GIS should be aiming at integration of all necessary elements for the spatial model and functions to efficiently create and utilise the spatial model. This paper, therefore, analyses architectures of GISs in order to assess their present status and development trend, which may help to define the direction of the related research and stimulate development of GIS that can easily be used. The first part of this paper reviews general aspects of GIS with a brief outline of its functional components based on the user-perspective. The main concern is system usage, investment, maintenance, productivity and reliability of information. With respect to the robustness of the spatial model as a basis of GIS as well as taking into account present development in information and communication technology, four evolution stages of system architecture for GIS are distinguished ranging from loosely constructed to the well constructed systems.

2. GENERAL ASPECTS OF A GIS

A GIS is to be used to produce a spatial model in a form of database which contains a selective set of aspects of reality. In addition to the production role of a spatial model, a GIS should also play an interface role between human users and the database. This role includes data entry, information query and presentation. Data entry translates human knowledge into a component of the model conforming to the data model and data structure. Information query performs the reverse operation by translating the electronic components into information which should be made understandable to human by the presentation mechanism. Another important role of GIS is maintenance and management of the database. The GIS should provide the possibility to update the database. The management role is to handle various request from the user and activate appropriate operation on the database in response to the user's request.

In the sense of integrated geoinformation two issues must be considered: the first one is the spatial model itself; it must reflect all those aspects of reality that are relevant for the intended use of the GIS. The second one is, whether the system offers all means needed to utilise the spatial model, which is a matter of system functionality. This means various technologies must be adopted to serve different functions; e.g., photogrammetry for acquiring some components of spatial model, computer aided design (CAD) for graphically constructing, editing and visualising the spatial model, database technology to manage the spatial model, virtual reality to naturally explore the spatial model, etc. To effectively exploit the technological development, the spatial model should:

- include three-dimensional (3D) aspects of reality,
- allow both direct and indirect representation of the real world at the desirable level of abstraction,
- be capable to accommodate data from various sources and seamlessly integrate them into one spatial model.

The first requirement results from the fact that only limited spatial analysis of our 3D real world was possible unless we have a 3D model. The direct representation referred to in the second requirement is suitable for real world objects that have well-defined spatial extent. The original observations must be maintained by the model so that the knowledge about the reality will not be degraded. This implies that vector structure may be the more suitable basis of the model. In case a real world object cannot be directly represented, an indirect representation must be provided by the model, e.g. by defining the spatial extent of an object from the neighbours by interpolation of property values. In this case, the relationships among the neighbours given by the direct representation must be used as constraints for the derivation of spatial extent to make the indirect representation accurate. The third requirement points out the necessity of multiple representation in terms of semantics (i.e. an object may have different meanings to different observers) and dimensionality (i.e. ranging from 0D to 3D). The spatial database of a GIS should have the capability to store and maintain multiple representation. Integrating data from various sources implies that redundancy must be minimized whereby human intervention is likely required to decide how uncertainty should be resolved. Moreover, to achieve a spatial model that well-represents relationships among real world objects, the underlying data structure must be unified (see Pilouk and Tempfli 1994, Pilouk et al 1994).

The system should be capable in performing both query-based and computation-based complex spatial analysis across different themes and dimensions and provide the possibility to present the information from arbitrary view points with realistic visualization. This requirement is beyond what 2D GIS can offer. The query-based spatial analysis will allow to obtain information based on direct representation while the computation-based analysis will help to derive information to predict a situation in reality. This information may be directly fed into the model again to avoid repeating time consuming derivation processes.

3. FUNCTIONAL COMPONENTS AND RELATED DISCIPLINES

To support the requirements stated in the preceding section, the five major functional components of a GIS are outlined. This outline may be derived from many existing 2D GISs and other systems that are related to spatial modelling, such as CAD, DTM, etc., as follows (see also Maguire and Dangermond 1991).

(1) Data acquisition
(2) Data structuring
(3) Data storage and management
(4) Data processing
(5) Data output and information presentation

Each of the above components of a system has been separately developed in the past and many of those components have become disciplines of their own, e.g., database management system, photogrammetry, remote sensing, digital image processing, CAD, virtual reality, etc. These disciplines have to be brought together as functional components of future GIS. The next section discusses approaches to system composition by identifying four evolution stages.

4. EVOLUTION STAGES OF SYSTEM ARCHITECTURE

The composition of a system for integrated geoinformation will always depend on the state of the art, policy and economical constraints. Since technology develops, we can consider various system architectures as stages of evolution in handling geoinformation. We use the following fourteen criteria to analyse for different evolution stages.

1) Compactness of the system
2) Common operating system (OS) or hardware platform
3) Functional access
4) Data access
5) Relationships among components of spatial model
6) Commonness of user-interface
7) Investment cost
8) Maintenance
9) Data redundancy
10) Handling of uncertainty
11) Productivity of geoinformation
12) Potential toward automation
13) Supporting personnel
14) Size of user organisation

4.1 Evolution stage 1: Independent sub-systems

A GIS can be composed from a set of sub-systems as shown in **Figure 1**. Although this seems to be a rather easy (but expensive) approach to construct a GIS, it can only be regarded as a low level of integration with a low degree of unification. Each sub-system offers only a subset of all the functions of a GIS to carry out some specific tasks along the geoinformation production line. With respect to the above defined criteria, this kind of system has the following characteristics.

1) The system is composed of several sub-systems either in form of hardware or software. Therefore, it is not compact.
2) Different sub-systems may need different hardware and OSs, e.g., VMS, Unix, MsDOS, MacOS.
3) The system cannot provide a central control panel. Functions of a sub-system are only reachable from the respective local control panel.
4) The data cannot be accessed from a single entry point. The data transfer from one sub-system to the other may need to be done manually, e.g., using floppy disks, if the sub-systems used for consecutive operations are not connected on-line. Data conversion is likely to be required because typically each sub-system will used its own data structure.
5) Components of information are usually stored separately in the local database of each sub-system. For example, data representing man-made objects may be stored in the CAD sub-system, data of terrain relief in the DTM sub-system, data of other terrain objects in the 2D geoinformation sub-system. This implies that metric computation must be used to integrate data from different sub-systems before topological relationships can be created.
6) The system does not provide a common user-interface. The user-interface is locally provided and dependent on each sub-system. This implies elaborate user training and operation liable to mistakes.
7) The investment costs are high because each sub-system has to be purchased separately. A number of sub-systems is required to achieve the required functionality.
8) Maintenance is difficult and also expensive. Different vendors may be responsible for different sub-systems.
9) Data redundancy is most likely very high because of separate and independent storage.
10) Dealing with uncertainty is necessary in operations that involve datasets from different sub-systems.
11) Users have to cope with many problems, so the productivity is not likely to be very high.
12) Difficult to turn the production line into automation due to several limitations mentioned earlier.
13) This approach requires various supporting personnel, e.g. OS specialist, application specialist, application programmer, etc., to ensure operation.
14) The size of user organisation is quite large in terms of number of personnel and space required for placing the sub-systems.

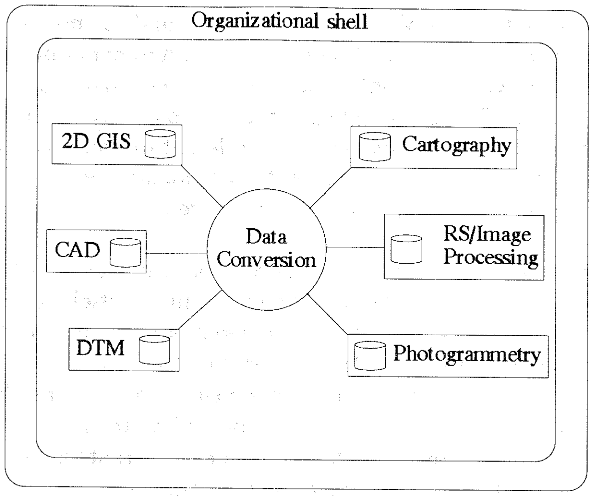

Figure 1 *System composition by independent sub-systems*

Figure 1 is a graphical illustration of this approach. The data conversion plays a central role to integrate components of the spatial model which is stored separately and independently as databases in various sub-systems.

4.2 Evolution stage 2: Functional integration

A system based on this architecture combines all necessary functions into one software package. **Figure 2** illustrates this approach. The system has the followings characteristics.

1) The system is compact because all sub-systems are shrunk down into functions or software modules that are implemented within the system.
2) The system is based on one OS and hardware platform.
3) The system provides a central control panel.
4) The data can be accessed from a single entry point. The data transfer between software modules can be done as background process.
5) Each module may have its own data structure to store data. For example, coverage data and TIN data in Arc/Info are stored in separate datasets with different data structures. Topological relationships among data elements across different datasets do not exist.
6) The system can provide common user-interface.
7) This approach is less expensive than the independent sub-systems and the client/server (see section 4.3) because it is based on only one software package.
8) Maintenance is easy, because of fewer pieces hardware and software are to be maintained and only one vendor to be dealt with.
9) Data redundancy still exists among different datasets.
10) Problems in handling of uncertainty are similar to evolution stage 1.
11) The productivity is likely better than the client/server approach because all processes are locally performed under one system shell, thus, requires less time for data transfer and message translation.
12) Many operations can be automated which makes it more feasible to automate the whole production line. The user has a possibility to optimize and streamline the operation after

getting familiar with the system, e.g. by using script or macro language, usually provided by this kind of system, to combine basic functions together, thus many inter-processes that require manual operation can be reduced.
13) Less supporting personnel is needed than in evolution stage 1.
14) The size of user organisation becomes small in terms of number of personnel and space for accommodating the system. The vendor organisation, however, becomes larger because the design and implementation of the system requires personnel from many disciplines.

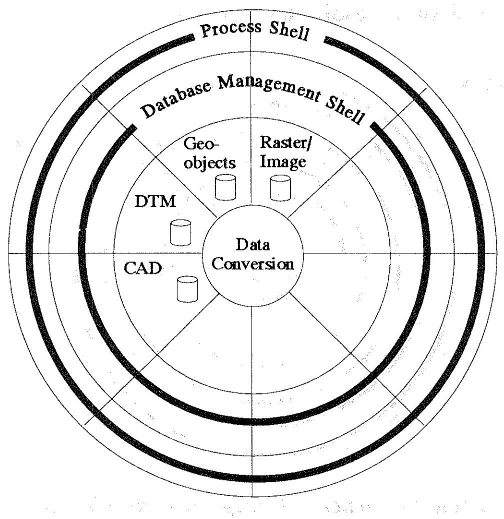

Figure 2 Functional integration

Although this approach represents most of the present attempts of development of GISs, it turns out that no single system can offer all required functions (yet). An example is that most of the GISs still use 2D spatial models as bases thus cannot offer 3D modelling capability. This implies that the users still have to adopt the architecture of evolution stage 1 to achieve required functionality.

4.3 Evolution stage 3: Client/Server architecture

This approach is based on the communication between a "client" and a "server". The client is a module provided to interface with the end-user. The server provides operations to process requests from and gives feedback to the client. The server and the user can only communicate via the client module. The system has the following aspects.
1) The client/server approach does not attempt to improve the compactness of the system. The system is still composed of several independent sub-systems as in stage 1.
2) The client/server approach is able to carry out task on different hardware platforms and OSs by using a standard communication protocol. The role of the OSs on different platforms are suppressed thus the user only needs to deal with the OS and the hardware platform of the client module.
3) The client provides a central control panel. Necessary functions are reachable from the client module. Each function of the client turns a user action into an appropriate request message which will be sent to the server. Upon receiving the request message, the server evaluates the request and triggers a process if the request is valid. The client can also attach some data, which should be processed by the server, to the request message. The server processes data, if necessary, and then sends back the result to the client. The user may not have the freedom to explore the functionality of each server unless better access to the functions of the server through a mechanism called "object link and embedding" (OLE) are provided (see Microsoft 1993, Brockschmidt 1993). The link between each dataset and its specific server application is maintained through this mechanism. More than one application server can be attached to a dataset and a choice of servers may be provided. Upon the selection of dataset and server, OLE activates and transfers all control and the user-interface to the server which allows the user to access all functions provided by the server. The user returns to the client by quitting from the server. The OLE approach seems to work well on a single OS platform. The client/server approach does not require the user to move around and entering several OS shells to reach functions that are available on different sub-systems, thus, offers an improvement in this respect.
4) The data can be accessed through the client whose database is a container that embeds and encompasses different datasets, which may have different data structures native to specific servers. This means that each dataset embedded in the client database may be recognisable only by a specific server (see **Figure 3**). The data transfer from one server to another server is done on-line. The client application may have to locally provide a data conversion function, if the destination server cannot recognise a non-native data structure.
5) Components of information are stored separately in different embedded datasets. The client database can be regarded as a collection of different datasets. There are no topological relationships across different datasets.
6) Common user interface with intuitive graphical user interface (GUI) can be provided by the client, however, this is limited by the ability of the client in knowing all functions of different servers.
7) The investment cost can be as high as the independent sub-systems approach with the additional cost of the client application and the network connection to all server applications.
8) The client application must be upgraded in accordance to the upgrading of one of the servers or communication protocol.
9) Data redundancy is not minimized by this approach caused by the independency of the embedded datasets.
10) Uncertainty has to be dealt with every time relationships among data elements across different datasets have to be created.
11) Users deal with less problems in functional access and data access and transfer, so the productivity can be improved.
12) The production line can be more automated because some manual processes can be eliminated.
13) Less supporting personnel is needed because the user needs not deal with many low level operations.
14) With respect to personnel, the size of organisation is may be smaller than in the independent sub-systems approach

but not so as far as space concerned for accommodating the sub-systems.

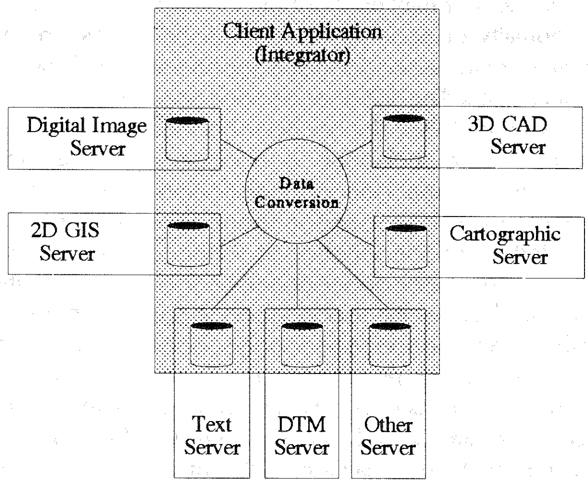

Figure 3 Client/server architecture

This approach has been developed after the evolution stage 2 in response to demands from the user community. It can be regarded as an intermediate solution because of the fact that complete functionality is not yet available on a single system at this time. The client/server approach can be a good solution for providing access to all required functions existing on different independent sub-systems. The client may be provided by a third party who has experience in offering service on interfacing users with systems from various vendors (e.g. training, design of process flow) or through cooperation among the vendors (see Intergraph 1995).

4.4 Evolution stage 4: Structural integration

To create a spatial model that better represents reality, the components of geoinformation must be closer related. This requires well-defined relationships among data elements by means of topology which is likely to be difficult without an appropriate unified data structure (UNS). The propose architecture is based on UNS as described in Pilouk and Tempfli (1994), Pilouk et al (1994).

4.4.1 General consideration:

This approach is aiming at a better representation of real world objects and relationships among them while trying to maintain good aspects of the evolution stages 1 to 3. The major considerations for the structural integration are:

- All components of the 3D spatial model must be stored in one database so that topology can be applied to represent the spatial relationships of the real world objects. This will help to avoid metric computation (if not necessary) thus speed up many operations.
- Both direct and indirect representation must be possible within the system. The direct and indirect representation need different processes to present the information to the user. To allow both representations in one database the system must use an appropriate data structure.
- Data redundancy must be minimized. Redundancy is normally introduced by the lack of awareness of existing data. The storage of redundant data does not provide additional information but only consumes extra storage space and rather result in conflict and, therefore, requires additional operations to resolve conflictions.
- The frequency of handling of uncertainty must be minimized and be taken away from the end-user as much as possible by converting it into a data quality attribute which is beneficial to the end-user. The database creator has more chance to access the original sources and, is therefore in a better position to resolve the uncertainty.

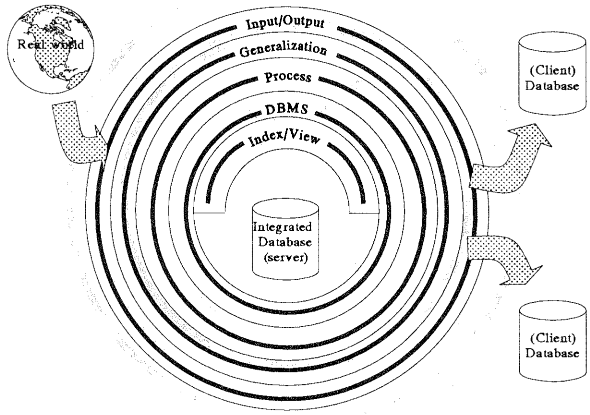

Figure 4 Structural integration architecture

4.4.2 A propose system architecture:

The proposed architecture for a GIS is illustrated by **Figure 4**. There are various layers of the system that encompass the integrated database that is based on UNS. The integrated database contains a spatial model accommodating both direct and indirect representation of real world objects. The database management shell embraces this database on which various indices or database views exist. A database index helps to speed up spatial access to some specific data elements. It can be regarded as a specific view of the earth that is closer to the application domain or operation requirement. The existence and status of each index and view are depend on the integrated database. All indices and views are updated according to the changes in the integrated database and any changes must be directed first to the integrated database and subsequently to the indices and views. The database management shell provides functions and rules to access and update the integrated database or views. The process shell is the next outer level to the database management shell. It contains various functions to process the integrated database or views by using database management functions provided by the database management shell. The next outer shell is the generalization shell which helps to simplify the input from the real world to the GIS and provides an appropriate representation for the output from the GIS.

The characteristics of a system adopting the structural integration approach are:

1) The system is as compact as in the functional integration approach.
2) A single OS and hardware platform are used.
3) The system provides a central control panel.
4) The data can be accessed from a single entry point. Each data element, i.e. a component of the spatial model, can be access directly or by navigating via other data elements.
5) Relationships among data elements can be established to represent spatial relationships in reality.
6) Common user-interface can be provided.
7) The investment cost for the system may be as low as for the functional integration. The investment on data storage capacity is expected to be lower than for the other architectures due to less redundancy.
8) The system is easy to maintain with respect to hardware, software and database.
9) Data redundancy is eliminated.
10) Uncertainty has to be handle only at the database creation and updating stages. Other operations only exploit the data quality information.
11) High productivity is expected.
12) Production can to be automated because manual operations such as data transfer, conversion and resolution of uncertainty have been eliminated.
13) Few supporting staff is needed.
14) The size of organization is comparable or smaller than for the functional integration architecture.

When comparing the four evolution stages we can say, performance of structural integration is better or at least equal to any other form of system architecture with respect to all fourteen criteria.

5. CONCLUDING REMARKS

The propose structural integration implies great effort in designing and constructing the system. This effort, however, is transferred to the vendor rather than leaving it to the user as was done in evolution stage 1. The expected superiority is not only in handling uncertainty but also on benefits from explicit relationships and minimized redundancy--with all consequences for the user.

Present systems are in the evolution stage 1 with increasing attempts concerning stage 2. There is a trend toward evolution stage 3 due to commercial driving forces that prefer to keep proprietary development from public while at the same time offer improved user-friendliness to partially fulfil user demands. This is evidenced by the adoption of specification for OLE extensions for computer-aided-design (CAD), computer-aided-manufacturing (CAM), computer-aided-engineering (CAE) and GIS by a group of vendors related to spatial information industries, namely the Design & Modelling Application Council (DMAC) which includes ANSYS Inc., Autodesk, Bentley System Inc., Cadence Design Systems Inc., Crisis in Perspective Inc., Intergraph, Microsoft, Ray Dream, SDRC, Shapeware Corp. and Spatial Technology, since the beginning of 1995 (see Intergraph 1995). The evolution stage 4 promises an ideal system, but can only be achieved if there is a consensus on a formal design of the spatial data model. This consensus may also result in the adoption of the client/server architecture on top of the structural integration.

ACKNOWLEDGEMENT

The author wishes to thank Dr. Klaus Tempfli for his valuable discussion, suggestion and correction of this writing.

REFERENCES

Brockschmidt, K., 1993. Programming for Windows with Object Linking and Embedding 2.0, Microsoft press.

Intergraph, 1995. New OLE Extensions for CAD/CAM/CAE and GIS Adopted, Press releases, Intergraph Corp. Huntsville, http://www.intergraph.com/press95/dmpr.html.

Marguire, D.J., and Dangermond, J., 1991. Functionality of GISs, In: Geographical Information Systems: vol. 1, Principles, Marguire, D.J., Goodchild, M.F., and Rhind, D.W. (eds.), Harlow: Longman Scientific & Technical, pp. 319-335.

Microsoft, 1993. Object Linking and Embedding: OLE 2.0 Design Specification, Microsoft Corporation.

Pilouk, M., and Tempfli, K., 1994. Integrating DTM and GIS Using a Relational Data Structure, In: Proc. Eight Annual Symposium on Geographic Information Systems (GIS'94), Vol. 1, Vancouver, Canada, pp. 163-169.

Pilouk, M., Tempfli, K., Molenaar, M., 1994. A tetrahedron-based 3D vector data model for geoinformations, In: Advanced Geographic Data Modelling: Spatial data modelling and query languages for 2D and 3D applications, Molenaar, M., and De Hoop, S., (eds.), Netherlands Geodetic Commission, No. 40, Delft, The Netherlands, pp. 129-140.