UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

CHAPTER 2

GIS AND DECISION SUPPORT

A REVIEW OF LITERATURE AND PREVIOUS RESEARCH

2.1 Spatial Decision Making and Computerized Support

Information plays a crucial role in the decision-making process in the planning, site selection and environmental management domain. It is well known that planning starts with the collection of data and information to describe the state and condition of the real word. The purpose of gathering, processing and integrating data and producing information is to understand the environment wherein the planning activities take place. It is accepted that a better understanding of the environment facilitates better planning and decision-making (Harris, 1987). As indicated by O'Hare (1983) poor location decisions in facility placing may result from a variety of factors, the most common of which is an uninformed land use planner.

Some authors (Hopkins and Schaeffer, 1985) see planning, site selection and environmental compatibility assessment as information-producing activities that are performed to reduce the inherent uncertainties in decision-making. Here the major role of a planner and/or analyst is to produce and analyse quality information to aid effective decision-making.

Catanese (1979) and Harris (1987) have clarified the meaning of information by constructing a hierarchy of data (Figure 2.1). They emphasize a progression from observation to data, data to information and information to intelligence. As shown in Figure 2.1 data may be explained as a one-to-one relationship between observation and real word phenomena under concern, while information, on the other hand, is organized data resulting from aggregation, manipulation and other statistical, mathematical, or algorithmic manipulation of data. Information is derived from data to develop specific knowledge necessary to solve a problem or to show patterns or directions.

The whole process of data/information collection and knowledge generation for the problem at hand is a time-consuming activity often requiring more than half of the resources available to the decision-maker. This is also one of the issues that have hampered planning practice.

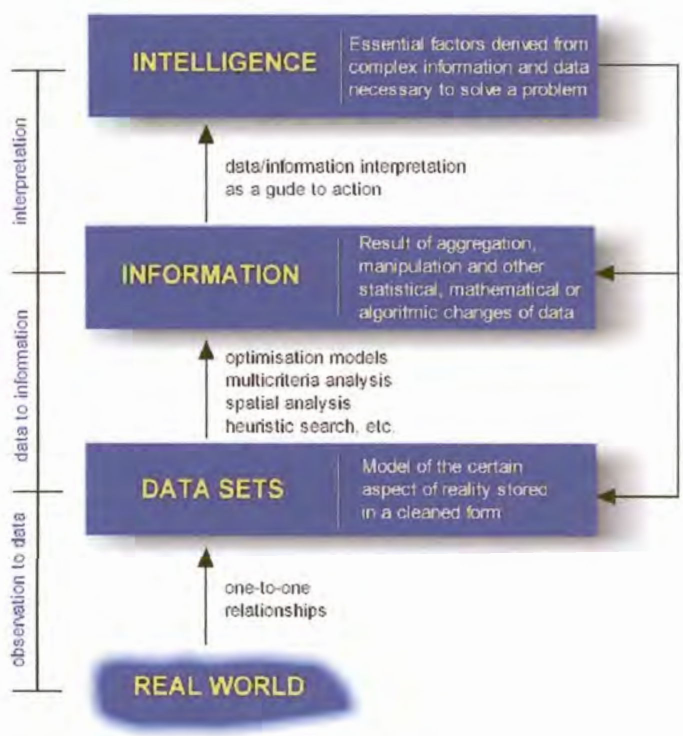

Figure 2.1. Hierarchy of Knowledge, Catenese (1979)

From the viewpoints mentioned above it is apparent that data and information plays an important role in any planning and environmental problem solving task. These tasks are usually information intensive and unstructured, and their effectiveness relies heavily on the availability of efficient tools for data and information manipulation.

General consensus exists in the relevant literature that the advancement of computer and information technology over the last few decades has had a very significant impact on planning, site selection and environmental compatibility assessment. This is mainly due to the extraordinary characteristics of microcomputers, especially in terms of their accessibility, high-speed computational ability and capacity for data/information processing. In addition to this, the development of various computer-based Information Technologies (IT), including Geographical Information Systems, has provided many new ways to work with the spatially related problem solving tasks.

According to *K. Foote and M. Lynch (1997)* many innovations in the application of information technologies in environmental and planning fields began in the late 1950s, 1960s and early 1970s. During that time, methods of sophisticated mathematical and statistical modelling, various environmental models, as well as location-allocation modelling techniques were developed and the first remote sensing data became available. Furthermore, researchers also began to envision the development of Geographic Information Systems. The mid-1970s to early 1990s was a period of far-ranging IT experimentation and development trying to determine how the innovation could be adapted to meet a wide variety of research and commercial needs (K. Foote and M. Lynch, 1997). The same authors emphasized that this was a time in which the development of powerful software coupled with the availability of inexpensive computers permitted many researchers to test new ideas and applications for the first time. In the early 1990s, or perhaps a bit earlier, many of these innovations gradually gained acceptance and were developed collaboratively. The strengths and weaknesses of many information technologies were by then apparent, and researchers began to work together to cultivate the most promising applications on a large scale (K. Foote and M. Lynch, 1997). Two of these IT innovations that attracted considerable attention are GIS and Decision Support Systems (DSS) along with their applications.

2.1.1 Geographic Information System (GIS)

For spatial problem solving tasks in general, and environmental modelling projects in particular, GIS is seen as a convenient and well-structured toolbox. Concepts and techniques of GIS have been extensively discussed in the literature. What follows is a brief discussion of these concepts and techniques.

One of the strengths of a GIS lies in its ability to store, relate and manipulate large volumes of spatial and associated attribute data from diverse sources and formats. GIS is not simply a computer-based system for making and manipulating maps. On the contrary, in respect of the data it deals with, it can be thought of as a special type of database management system (DBMS) distinctly different from the other types of database systems. What distinguishes GIS from other systems is the ability to handle spatial data, to perform spatial operations, and to create new information based on spatial relationships[1] (Cowen, 1988).

Because of GIS's ability to display spatial data in graphic form it is often confused with computer-aided mapping and design systems (CAM/CAD). Although CAD and in particular CAM systems can perform many operations similar to GIS, it is the analytical capabilities of GIS that distinguishes GIS from CAM/CAD.

The abilities of GIS to relate and integrate different data sets and to perform spatial operations on data provide planners and others responsible for location-oriented decision making tasks with a convenient tool for information management, analysis and visualization. Generally, GIS functions can be classified into four categories.

1) Data input
2) Data storage, retrieval, and query
3) Data analysis, and modelling,
4) Data visualisation and reporting

Data input includes functions for capturing, processing and transforming spatial data. The spatial data can be derived from existing maps, aerial photos, satellite images, direct digital inputs, map and image scanning, surveying and other sources. The data input component, and in particular, digitising (converting data from analogue format to one that can be used by GIS) is typically the major bottleneck in the implementation of GIS. It should be pointed out that the development of a large, inventory-related database is a time-consuming and, costly process. Data input, apart from data capturing or format transformation, requires editing operations to verify digital data against the original source.

The second group of GIS functions, that is data storage, retrieval and query aims to organize spatial data into a flexible and topologically structured format which permits it to be shared, updated, and quickly and effectively retrieved on the basis of either spatial or non-spatial queries. The storage and retrieval capabilities of GIS provide users with a superior filing system for a location-based inventory. Thus, large volumes of spatial data from diverse sources and formats can be organized into a single database and incorporated into a common base map. This prevents data redundancy and inconsistency problems often occurring when data are manually maintained and updated.

In addition to its role as a spatial database management and retrieval system, GIS also provides the means for supporting spatial analysis and modelling. This group of functions, unique to GIS, takes advantage of the GIS ability to bring spatial and attribute data together. They perform a number of tasks, such as map overlay, reclassification (changing the format of data through user-defined aggregation rules), proximity analysis, buffer zone generation, etc. These functions and the ability to integrate data justify the use of GIS for location-related tasks.

---

[1] *Because a GIS can perform sophisticated data manipulation and spatial analysis it would be more appropriate to see DBMS is a part of GIS. Accordingly, GIS can be viewed as a collection of specialized tools (routines) linked to a relational database management system.*

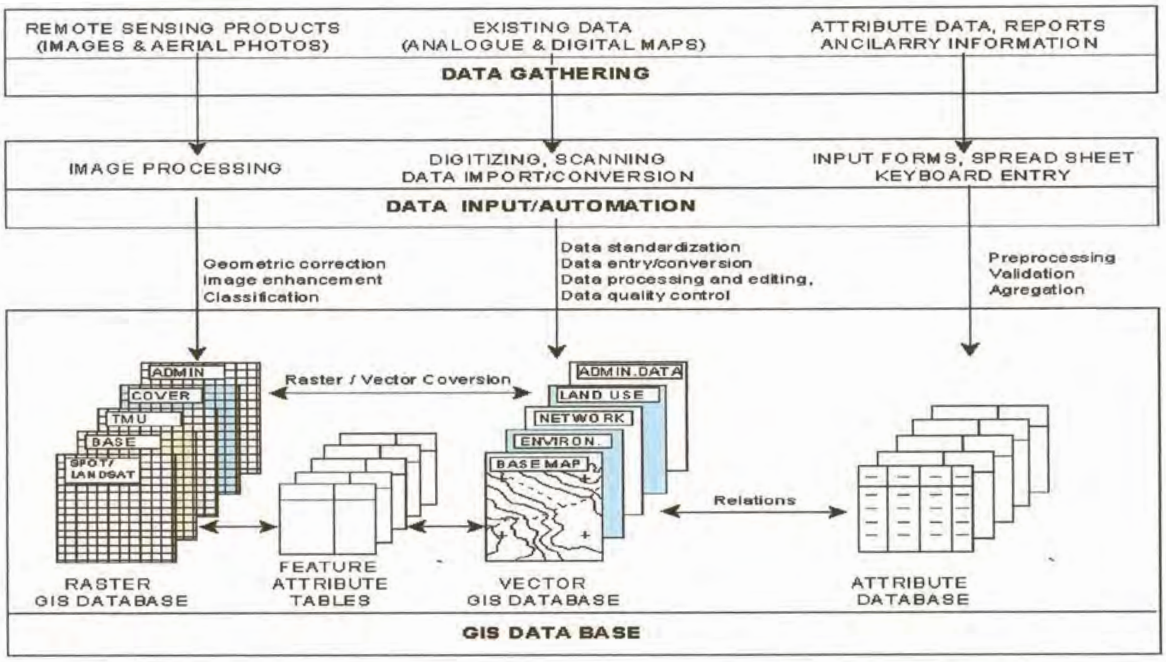

Figure 2.2 GIS data gathering and input procedures (After C. Valenzuela, 1988)

2.1.2 Spatial Decision Support Systems: Context, Concept and Definitions

Decision Support Systems in location and environmental planning fields began to appear during the 1970s and since then they have received a lot of attention in related literature. During this period DSS have been accepted as a specific field of study and practical applications have shown that there are advantages in using them for decision-making. The last decade showed the most rapid growth in DSS development. This development was made possible by the advances in computational capacity, clearer understanding of decision making, design of user-friendly software and operating systems, progress in artificial intelligence coupled with growing familiarity with computers among decision makers.

While there are many definitions of a DSS, there seems to be a general consensus that these systems are usually focused on specific types of decisions and on supporting rather than replacing the user's decision making process (P. Keenan, 1997). They are usually regarded as interactive systems providing the user with easy access to database management and problem specific analytical tools, as well as decision models capable of supporting unstructured decision-making tasks. As the above definition implies, the interaction between user and the system is very important in DSS. The system provides database management tools, application specific modelling capabilities along with a user-friendly interface, while the user (decision-maker) incorporates objectives, criteria, judgement and relevant data for the problem at hand. Their purpose, therefore is not only to automate the decision-making process, but also to aid decision makers in formulating alternatives, analysing their impacts, and interpreting and selecting appropriate options for implementation (Adelman, 1992).

In addition to the increased attention given to DSS there is also a growing interest in the concept of spatially enabled DSS or Spatial Decision Support System (SDSS). The development in SDSS represents an effort to address spatial problem solving and decision making. The term Spatial

Decision Support System implies capabilities for manipulating and analysing spatial data. From the functional point of view, they allow the representation and manipulation of complex spatial data structures, and they include analytical tools for spatial, geographical and other related analysis.

The concept of Spatial Decision Support System was initiated in the late 1980s (Densham and Armstrong, 1986; Densham and Goodchild, 1989). However, the most rapid growth of these systems has occurred in the last six to seven years. During this period many authors addressed the potential of SDSS to support location planning, site selection and environmental compatibility assessment (Densham, 1992; Fedra, 1994; Kim et al, 1993; Mejia-Navarro & Garcia, 1995; Keenan, 1995; Ehler, Cowen & Mackey, 1997). These authors pointed out that the SDSS concept is a feasible solution for improving decision-making processes in the location and environmental planning fields by providing users (decision-makers) with a flexible problem-solving environment. Here, the term "flexible problem-solving environment" refers to easy-to-use and interactive computer based systems that are capable of assisting decision makers to effectively formulate a set of alternatives on the basis of their consequences for the problem at hand.

This definition of SDSS could however refer to almost any computer-based system capable of supporting spatial problem solving. Therefore a further clarification of SDSS is required firstly by emphasizing the fact that that these systems are designed to support specific subsets of spatial related problems. Geoffrion (1983) identifies six distinguishing characteristics of DSS that are also relevant to SDSS:

1) *"They are used to tackle un or semi-structured problems – these occur when the problem, the decision-maker's objective, or both, cannot be fully or coherently specified;*

2) *They are designed to be easy-to-use allowing, sometimes very sophisticated computer technology to be accessed through a user-friendly front end;*

3) *They are designed to enable the user to make full use of all data and models that are available, so interfacing routines and data base management systems are important elements;*

4) *The user develops a solution procedure using models as decision aids to generate a series of alternatives;*

5) *They are designed for flexibility of use and ease of adaptation to the evolving needs of the user;*

6) *They are developed interactively and recursively to provide a multiple-pass approach which is in contrast with the more traditional serial approach – involving clearly defined phases through which the system progresses."*

There is a general consensus that the development of the SDSS concept is an appropriate response to the problems that impede current practice and quality of a decision making processes in location and environmental planning fields. In spite of increased use of information technologies in these planning activities, it has been pointed out that most planners and/or decision-makers have not taken full advantage of the available technologies. The reasons for this are mostly connected to the issue of their complexity, which generally tends to divert the process of decision-making away from decision-makers into the hands of highly trained technology specialist and experts. Although there is a lack of case studies in which the performance of SDSS have been evaluated, it is generally believed that such systems could be a feasible solution for improving the linkage between available IT technology, data/information environment and spatial decision making. As indicated by various authors (Fedra, 1994, Klosterman, 1994, Saenz 1997) the major benefits of SDSS are the ability to extend the boundary of rationality and comprehensiveness as well as quicker and more objective decision-making, cost reductions and improved productivity.[2]

From a design point of view, many definitions of SDSS describe them as a combination (or integration) of different components. Densham (1992) for instance defines the components of a "true" SDSS as an integration of a spatial database management system with analytical modelling capabilities, a visualization component or graphical user interface, and the decision-making knowledge for the problem at hand. He argued that the development and implementation of such systems could be achieved by using a set of linked software modules capable to provide an integrated set of flexible capabilities for solving the specific spatial problem (See Figure 2.3)

The Spatial Decision Support System and Data Visualization Report (CIESIN, 1997) is another constructive effort to summarize the common key components of a SDSS reflecting both its architecture (structure) and its capabilities. These components are summarized in Table 2.1.

Table 2.1 - Common Key Components and Capabilities of a SDSS

Components	Capabilities
Model Management System	Allows for efficient iteration between design development and calculation of impacts, including feedback for the problem at hand; Allows for automatic calculation of attributes for each feature in the scenario design; Allows for automatic checking of compliance with constraints imposed on the design; Support the development of models as well as the use of existing models for the evaluation of scenario design; Allows the integration of spatial objects in the model components
Database Management System	Provides capability for spatial manipulation, and, Provides storage and retrieval capability of entire design scenarios and ability to track scenario development
Display and Report Generator	Provides automatic report generation with graphics and text; Provide links to other programs
User Interface	Provides a graphical user interface; Provides interactive scenario development; Allows user modification of scenario; Provides configurable links to geo-referenced models; Provides selectable user levels

Adopted from: Spatial Decision Support System and Data Visualization Report, 1997, CIESIN

[2] *It is worth to mention that a laboratory experiment undertaken by Mennecke, Crossland and Killingsworth (1998) to investigate the decision maker's performance when using SDSS speaks in favour of the above observations. That research, although unique, examined two independent variables: task complexity (i.e., low, medium, and high complexity, and SDSS use (i.e., no SDSS versus SDSS support) and the results confirmed that the use of a SDSS has an important impact on decision quality and solution time.*

Many authors (Armstrong and Densham (1990), Saenz (1997), Fedra (1994), Kim et al. 1993) feel that expert or knowledge-based systems and other artificial intelligence tools can complement the analysis and modelling capabilities of SDSS. It is also widely accepted that the use of these systems and tools can guide decision-makers in their spatial problem-solving task and also assist them in deriving new facts from existing data and conditions.

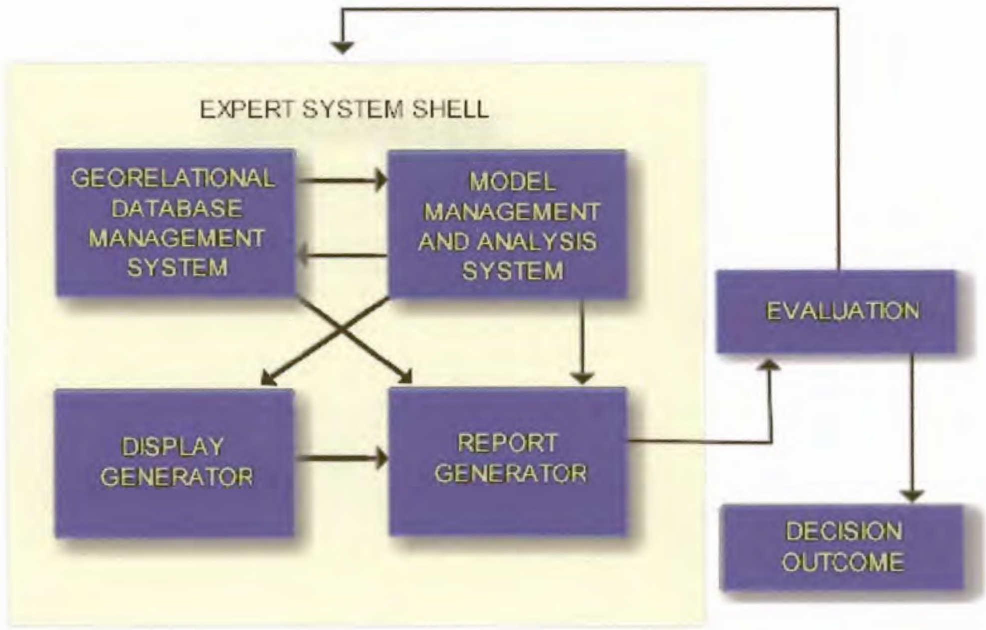

Figure 2.3 Components (software modules) for Spatial Decision Support System (Adopted from Armstrong, Densham, 1990)

Bearing in mind the multidisciplinary nature and the diversity of applications in the field of location and environmental planning, it is very unlikely that any SDSS will have all of the characteristics mentioned above. Some of them, depending on the goals of the system and the complexity of the spatial problem, might be simple and include only a few capabilities. The others might be more demanding, requiring the integration of various models and the use of expert systems and/or other artificial intelligence tools in order to provide effective decision support.

2.1.3 Geographic Information System as a Spatially Enabled DSS

The potential for growth of SDSS both in the field of research and practical application has been facilitated by the development of spatial technologies of which Geographic Information System (GIS) is the most important. Since the late 1980s, GIS has been recognized as the appropriate technology, or foundation stone for SDSS development. There is a general consensus that this technology provides users not only with an array of tools for managing and linking attribute and spatial data, but also for analysis and the visualization of model outputs. While many of these capabilities also exist in other systems, such as visualization and virtual reality systems, as well as CAM/CAD systems, GIS is unique. It is still the main spatial technology to date because of its emphasis on supporting analysis and providing users with a representation of objects in a common and cartographically accurate spatial system.

According to Armstrong and Densham (1990) SDSSs are evolving from GIS in the same way that DSS evolved from management information systems. Saenz (1997) also pointed out that of all computer-based technologies being integrated into DSS, GIS is perhaps the most popular. This is reflected in the increasing number of research and development papers referring to SDSS at various GIS conferences (ESRI International User Conferences 1996-1999; International Symposia on Spatial Data Handling, 1995; International Conference/Workshop on Integrating GIS and Environmental Modelling, 1997; Joint European GI Conference, 1997, etc.). Muller (1993), in his review of GIS, also identified SDSS as a growing area in the application of GIS. Fedra (1994) commented on the importance of GIS for SDSS development as follows: "in a hefty volume on Computerized Decision Support Systems for Water Managers (Libido et al., 1989) a conference proceedings, of close to 1000 pages, GIS is not mentioned once (at least according to the subject index). In contrast, and three years later, at a session of the 1991 General Assembly of the European Geophysical Society, dedicated to Decision Support Systems in Hydrology and Water Resources Management, more than half the papers discuss GIS as a component of the research method (EGS, 1991)". While this literature search was neither systematic nor exhaustive, it certainly indicates that GIS has become an emerging field with a lot of potential for SDSS development.

Many authors, relying on GIS for a variety of routine decision support and analysis applications, tend to go further and define GIS as a SDSS. Cowen (1988) for instance has characterized GIS as "a decision support system involving the integration of spatially referenced data in a problem-solving environment". Mennecke (1998) also identified GIS as a spatially enabled decision support technology. These definitions however seem to suffer from a lack of agreement on what a SDSS is and what it actually constitutes. Keenan (1997), Densham (1992), Fedra (1994), and many others argued that defining GIS as a type of SDSS is not supported by the DSS literature and that the capabilities of many of these systems are insufficient to assist decision makers in their deliberations. They also pointed out that GIS applications are often described as being SDSS because they where used for the collection or organization of data used by decision-makers. This is a reflection of the trend identified by Keen (1986) and many others that any computer-based system that somehow supports decision-making is (or could be) considered as being a decision support system.

The view of GIS as a SDSS is not however entirely without support and justification. It is considered important to stress the fact that decisions, as indicated by Simon (1977), fall in a continuum that ranges from highly structured (programmed) to highly unstructured (un-programmed) decisions. Structured decision processes as indicated by Simon refer to routine and repetitive problems for which standard solutions exist. For example, land development, land use control and similar activities regarding monitoring the state and changes in an area could be seen as routine and repetitive problems. The objectives of these activities are to keep control over the space and to register phenomena and trends of interest for planning and management. These activities are regarded as structured, data driven decision processes that do not require substantial modelling components as provided by the majority of GIS systems. Consequently, the technology of GIS with its facilities for storing, retrieving, manipulating, displaying and analysing spatial data and related attributes could be considered as a SDSS for structured decision making activities. In addition, Saenz (1998) argued that a GIS by itself could indeed function as a SDSS but only in the situation where the spatial analysis and modelling functions it provides are sufficient or adequate to support a particular spatial problem solving task and related decision making process.

As pointed out in the above definitions, a common requirement for SDSS is the availability of explicit models and capabilities to support a particular type of spatial problem solving which in practice tends to be a semi-structured or ill-defined. Complexity, uncertainty and even conflicting objectives usually characterize these problems and, as indicated by Spargue (1982) they cannot be solved by structured computerized decision support. Decisions for this class of problem can be understood as revolving around a choice between different options (Fedra, 1997; Armstrong and Densham, 1990). It is well known that site selection and similar problems often involve a number of possible solutions requiring a decision-making process to decide on the final solution. In principle, each decision means acceptance of one solution and rejection of a number of other solutions that are also feasible. It would normally be more effective if the selection of one solution out of a number of potential solutions could be based on exact criteria. For this type of decision there are often neither generally accepted criteria, nor the possibility to test all the possible solutions before making a final choice. In practice the usual approach would be to select one solution from a set of options that appear workable (Armstrong and Densham, 1990). Furthermore, the decision-making process for this class of problem is usually judgmental, iterative and integrative. As such, it requires more analytical competence as well as the availability of multi-criteria and other application specific models and modelling techniques capable of supporting the respective tasks.

From this point of view, there is widespread agreement that GIS systems, in spite of their significant contribution in assisting decision makers, could not be regarded as a fully developed SDSS, since they obviously lack the modelling tools required to adequately explore the solution space of semi-structured problems. This is specifically applicable to various fields of human activities, including location planning, site selection and environmental compatibility assessment (Openshaw, 1997).

Another widely cited criticism concerning GIS as being a fully developed SDSS is based on the issue of the complexity of the technology, specifically the framework and language for dialogue between decision-maker and computer system (man-computer interface). Albrecht (1998) in his overview of universal GIS operations for environmental modelling argued that current GIS systems are so difficult to use that it requires some expertise to handle them and that it could take up to a year to master a GIS. According to Albrecht (1998) and many others, these systems are cumbersome for occasional users (decision-makers) who require decision-support environments (man-computer interface) that are interactive, flexible and easy to use. In other words, various actors in the decision making process do not wish to be immersed in the technicalities of a full-blown GIS. What they usually desire is a fairly simple command structure with an understandable language and graphical user interface along with the ability to answer complex spatial questions.

2.1.4 The Role of GIS as a SDSS Generator

The abovementioned deficiencies that are preventing GIS to be used as a decision support system for spatial problems has attracted increased attention in related literature. Keenan (1997) defines of-the-shelf GIS as a GIS system that can be used as a generator to build a SDSS for a specific problem domain. There is strong evidence in the SDSS literature that GIS technology can be used as a generator for a SDSS. This is mainly due to the continuous development of GIS abilities to integrate diverse spatial and non-spatial data and information from various sources. This is one of the major requirements for the development of a SDSS. GIS also provides an appropriate, usually called "geo-relational" database management system characterized by comprehensive facilities for storing, retrieving and manipulating spatial and non-spatial data. Furthermore, it provides an interactive user interface, as well as a link between the interface and database which can be customized to allow actors in a particular decision making process to easily query spatial and related attribute data. Spatial visualisation is another fundamental capability of GIS that is also of particular importance in supporting spatial problem solving. However, the main issue that is preventing the acceptance of GIS as a complete SDSS generator for a full range of spatial problem solving tasks, is the almost complete absence of problem-specific models and modelling techniques developed to support decision making even before the emergence of GIS (Carver, 1991; Keenan, 1997). Some of these models (location-allocation techniques, uncertainty analysis, multi-criteria evaluation) are however of interest to potential users of spatial decision support system.

One possible example where modelling is required, is site selection and similar problem solving tasks. As pointed out above, these spatial decision problems have to do with choices between various options (possible locations) that are usually analysed, compared and ultimately ranked according to a number of user defined objectives and relevant criteria. The selection process is then based on a comparative analysis of the ranked options (possible locations) and on the selection of those that appear to be the most workable. For this type of problem, the majority of GIS systems only allow the decision-maker to identify (through a map overlay process) a list of sites meeting a predefined set of criteria. What is missing is a mechanism for representing choice and priority in the context of evaluating conflicting criteria and objectives. This is usually accomplished by implementing multi-criteria evaluation techniques such as the analytical hierarchy process (AHP). This restriction in terms of choice or preference makes GIS a very static modelling environment and thus reduces its scope as a decision support tool (Heywood et al., 1994).

In respect of the aforementioned examples in the field of location planning and environmental modelling, many researchers have advocated the perspective that some sort of extensions to proprietary GIS are needed in order to achieve the desired level of problem solving (Goodchild at al, 1992; Openshaw, 1991; Fedra, 1995; Burrough, 1992; Anselin, 1993, Armstrong and Denham, 1990; Keenan, 1995; etc.). In other words, in order to be used as a SDSS generator for a full range of spatial problem solving, GIS software must provide a mechanism for incorporating appropriate models and modelling techniques drawn from other disciplines (Keenan, 1997). From this point of view, the key to useful SDSS is basically an integration of proprietary GIS with other application specific models (software) through direct and indirect links (see Figure 2.4).

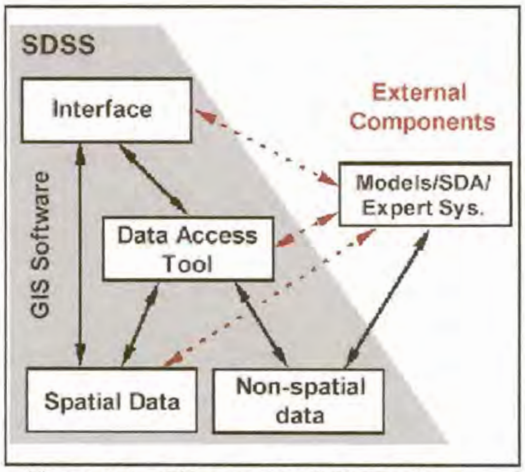

Figure 2.4 GIS as a SDSS Generator (Keenan, 1987)

From the literature review, it appears that such integration essentially seeks to fuse capabilities available in the individual systems and to provide some desired level of usability. Therefore, as argued by many authors (Grimshaw et al. 1996, Fedra, 1994, Goodchild at al. 1992) integration of specialist software supporting spatial and non-spatial data analysis and modelling with GIS through direct and indirect links is a key to useful SDSS. Such integration seems to have the necessary power and flexibility to assist decision-makers "in sorting out their perceptions and preferences after the information gathering stage" (Grimshaw, 1996).

2.1.5 GIS Integration (Classification of Systems Integration)

Integrating GIS with data analysis and modelling software drawn from other disciplines provides a method for communication between these systems so that they can share resources. It usually deals with issues related to data/information exchange. For instance, how can data be shared or exchanged effectively and precisely between different systems? The purpose of integration as indicated in related literature is to develop an environment in which users (decision-makers) are able, in a user-friendly manner, to access all functions from the systems being integrated in order to implement their analytical and problem solving deliberations.

The problem of integrating (or linking) analytical and modelling tools to proprietary GIS has over the past decade begun to emerge as an important research area (Goodchild et al., 1992; Anselin and Getis, 1992). Various logical ways of coupling spatial data analysis and models with GIS have been identified and there is still work underway to explore them. The most frequently cited classification is the architectural basis for integration, where the integration is expressed in terms of the closeness or the extent to which two separate systems are interfaced (Goodchild et al, 1992; Fedra, 1994). According to Goodchild (1992) three major approaches can be distinguished, namely:

1) Loosely coupled integration between proprietary GIS and spatial/non spatial analysis and modelling software.
2) Close coupling between spatial data analysis and modelling software and GIS;
3) Full integration of spatial analytic procedures and modelling techniques with the GIS;

(1) Loosely Coupled Integration

Loosely coupled integration is the simplest and by far the most frequently adopted approach for using GIS in many applications. In this approach problem specific models (specialist software) and GIS are used as two separate (independent) applications that just exchange files (see Figure 2.5). The data resulting from one system are fed into another through direct-link transmission or using other, indirect ways. When using this approach, GIS is very often employed as a pre-processor (preparation of model-input data), and as a post-processor (display and possibly further analysis of model results). Each system therefore complements the other - the model reads some of its input data from GIS files and produces some of its output in a format that allows further processing and display with GIS.

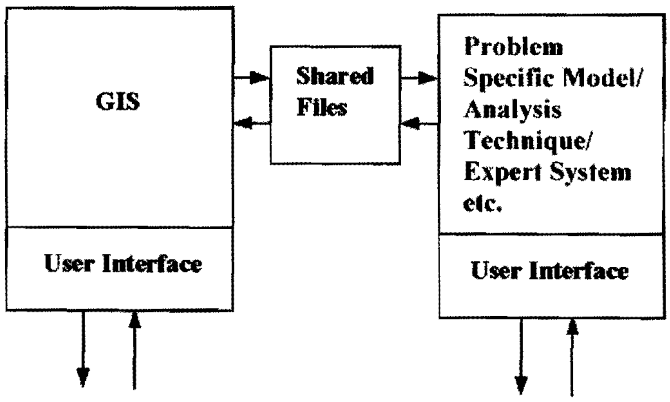

Figure 2.5
Loosely Coupled Integration
(After Chulmin,J. 1999)

From the software engineering point of view, loosely coupled integration seems to be a rather common and by far, perhaps the most straightforward approach of linking different application (software) components. It requires little if any software modification and customisation. The disadvantage of this integration method is usually related to issues of exchange of input and output data between applications. Many authors (Fedra, 1994; Singh and Treleaven, 1998,) argued that a solution based on files shared between two separate applications could be sometimes lengthy and cumbersome, especially in performing iterative modelling over a large number of problem (spatial and non-spatial) variables with sufficient speed. They indicated that although modelling may be fast, the process of data transfer can be slow and even error prone. It has also been pointed out that importing GIS data into other applications and vice versa is not always straightforward requiring either use of special products or development of software routines to convert (pre-process) the data into proper formats.

Recently however these shortcomings, particularly in applications running under the same operating system, are becoming less prominent mostly due to the IT improvements in the field of inter-application connectivity. Examples are:

- *DDE (Dynamic Data Exchange) - an object oriented technology which is an MS Windows supported method of exchanging data fairly rapidly amongst applications on the same computer;*
- *OLE (Object Linking and Embedding)- MS Windows supported technology which permits an object of one application to be either linked or embedded within another, from where it may be edited directly;*
- ODBC (Open Database Connectivity standard) - Microsoft's open interface for accessing data in a heterogeneous environment of relational and non-relational database management systems.
- RPC (Remote Procedure Calls) – an inter-application communication protocol most commonly found on UNIX platforms.

(2) Close Coupling Integration

A close coupling approach involves deeper integration of a problem specific model(s) with GIS. With this approach different applications (software systems) share not only the communication files but also a common graphical user interface (GUI). The GUI provides the veneer to assist and guide the user through the whole modelling process. Apart from the common user interface, closely coupled systems provide transparent file and information sharing and, therefore easy and error-free data/information transfer between the respective SDSS components (Figure 2.6).

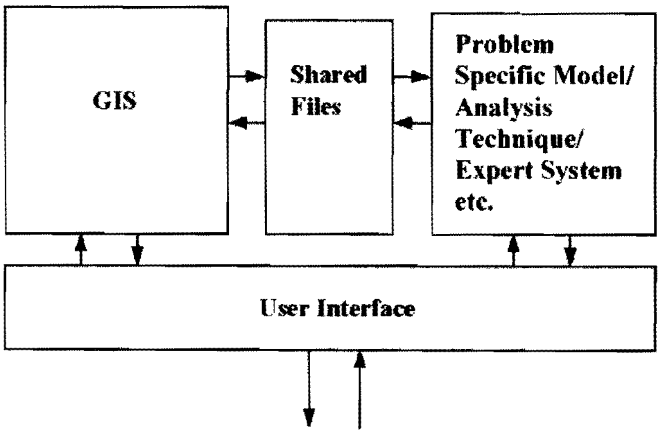

Figure 2.6 Close Coupling Approach (After Chulmin, J.1999)

An example of close coupling that draws together GIS and problem specific models is the Integrated Planning Decision Support System (IPDSS) (Mejia-Navarro and Garcia, 1995). IPDSS, as described by the authors, represents an Unix based SDSS designed to assist communities in the evaluation of geological hazards, vulnerability and risk, as well as to assist urban planners in analysing, modifying and re-evaluating spatial information within land-use planning activities. As such, it can be viewed as a computerized framework that is used to support complex decisions based on spatially distributed information. IPDSS incorporates the GIS named GRASS (Geographic Resource Analysis Support System), various problem specific numerical models and multi-criteria analysis techniques within the common graphical user interface (GUI). While this architecture is in fact the collection of diverse, independent software tools, the IPDSS GUI is assembled in such way that the analyst always has an impression that he/she is interacting with a single coherent system.

Another example of closely coupled integration is the Land-Use Change and Analysis System (LUCAS), developed during the "U.S. Man and Biosphere project" in the Computer Science Department of the University of Tennessee (Berry et al. 1996). LUCAS, as defined by its authors, is a prototype computer based SDSS specifically designed to integrate the multidisciplinary data stored in GIS (GRASS) and to simulate the land-use policies prescribed by the incorporated analytical models. It was implemented as an "object-oriented" C++ application to promote modularity and to allow different or additional software modules to be added to existing code easily, as the needs of investigators changed. The central component of LUCAS is a common, user-friendly graphical user interface capable of extracting different types of data for addressing research questions concerning land use and its impacts. The types of data include spatial and tabular data, results of mathematical models, spatial models, maps and/or visualization of land-use simulation exercises, etc.

It is worth mentioning that the macro languages of GIS software such as MapInfo's MapBasic, Arc/Info's Arc Macro Language (AML), ArcView's Avenue, makes it possible to employ GIS as generators by providing a common interface capable not only of invoking external programs (models) from the GIS environment, but also to secure transparent file and information sharing. One of the most recent examples is the utilization of ESRI's Arc Macro language (AML) in development of a graphical user interface for the incorporation (close coupling) of Soil and Water Assessment Tools (SWAT) with ARC/INFO. As describe by Zhou and Fulcher (1997), the menu interface provides a tool to identify the relative contribution of different watershed areas to agricultural non-point source pollution and evaluate the effects of alternative land use management practices on surface and ground water quality at the watershed scale.

(3) Full Integration

Full integration implies the coupling of problem specific models and GIS within one single application with shared memory and communication routines (Figure 2.7). The focus of this approach is on the system consistency (common data structure and data model, data handling and visualization) that obviously guarantees optimal system performance, particularly in comparison with the loosely coupled integration approach. However Fedra (1994) argues that the most elegant form of integration is also the most costly one in terms of development efforts since it requires appropriate programming knowledge as well as a good understanding of proprietary GIS and other application development environments.

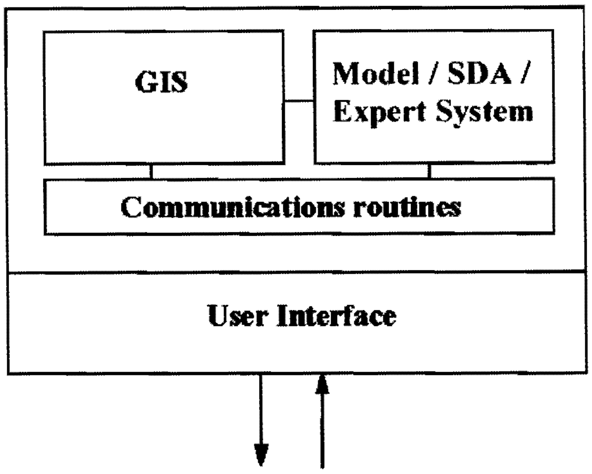

Figure 2.7 Full Integration (After Chulmin, J. 1999)

From the available research and practical applications it is apparent that full integration can be achieved by different methods. One method is to use a proprietary GIS-based programming language to create and implement a problem specific model that consequently becomes one of the analytical functions of the GIS. Examples are desktop GIS packages such as ArcView and MapInfo. Both of them provide macro-programming languages that enable their functionality to be extended by writing new programs or customisation of the user interface. There is evidence that third party developers are creating powerful extensions that can be added to these GIS desktop systems. The majority of them are designed to support many types of either data-driven or the model-oriented spatial problem solving tasks.

Full integration can also be achieved by creating user-specified analysis and modelling routines through high level programming languages such as Fortran, C, C++ and adding them to the existing tool box of the proprietary GIS. Examples include integration of multi criteria evaluation techniques into GIS such as the Simple Weighted Linear Combination Procedure embedded into the SPANS GIS software and Saaty's Analytical Hierarchy Process (AHP) added to IDRISI.

Another frequently cited and recently the most prominent method for achieving full integration is the method of incorporating GIS functionality into models through the use of a variety of development frameworks including popular programming environments such as Visual Basic, Delphi, Visual C++, and others. This method of integration, however, requires a sufficiently open GIS architecture capable of being accessed by other software applications. To facilitate this type of development, GIS software developers have recently adopted application development environments based on trends and technologies from various other fields such as computer sciences and data engineering. More specifically, GIS systems are moving towards a true distributed, object-oriented geo-processing environment, sufficiently modular to permit their integration with other software components within one single application. An example is ARC/INFO's Open Data Environment (ODE) on both UNIX and PC platforms. ODE allows developers to access ARC/INFO (GIS) functionality from different non-geographic information system applications or through a custom created interface. This approach means that the applications incorporating GIS functionality could be developed in more modular fashion and within programming environments other than Arc Macro Language (AML – Arc/Info's platform scripting language and interface toolkit).

At the PC level, GIS software developers (ESRI, MapInfo, for instance) are rapidly adopting and making available so-called ActiveX controls and a collection of programmable ActiveX Automation objects. As reusable and programmable software components, these controls and automation objects allow application developers to add required elements (or subsets) of mapping and GIS functionality to applications developed in another programming environment outside a GIS. A good example of this type of GIS tool is the software "What If?" developed as a cooperative effort by LDR International Inc, Data Chromatic and Prof. R.E. Klosterman. (1997). It represents an interactive, easy to use GIS-based SDSS that, as described by the authors, supports all aspects of the land use planning process (land use suitability analysis, projection of future land use demand, evaluation of the likely impact of alternative policy choice and preparation of a land use plan). *"What if?"* incorporates various site selection and planning models and modelling techniques into a fully integrated and portable MS Window application. It was developed in Microsoft's Visual Basic programming environment. Required mapping and GIS functionality were integrated into the application by using ESRI's MapObjects GIS component software.[3]

All the approaches to integration described above have certain advantages and disadvantages. It is therefore difficult to draw a conclusion as to which approach is superior. Bailey (1994) for instance is somewhat pessimistic about the prospects of tight integration of statistical and other models with GIS. He advocates a form of loose coupling based on open-systems computing environments wherein a GIS, statistical and other analysis package would be accessed simultaneously but independently on the same GUI. Fedra (1994) furthermore argued that fully embedded models into GIS appear to be rather simple and restrictive. Batty (1998) also pointed out the limitations of available GIS scripting languages, notably the size of problem that they can effectively handle. He argued that complex spatial problem solving tasks can only be handled by combining (linking) GIS with independent software tools (models) through a common interface written in some high level language outside the GIS environment (close coupled approach). Likewise, Djokic (1993) made a strong case for the use of available software tools within a SDSS shell. He argued that the one-off effort of developing an interface between software components would require much less effort that customizing existing or writing new software. On the other hand, Walsh (1993) emphasized the need for an open architecture and interdisciplinary collaboration in the development of SDSS with fully integrated GIS functionality.

As can be seen from the above, any decision concerning an appropriate integration approach is obviously case-driven. It depends on many factors such as contents and complexity of the problem to be supported by SDSS, availability of software components required, system characteristics and performance, available resources, data requirements etc.

---

[3] ESRI's MapObjects is an ActiveX Control bundling a large number of programmable ActiveX automation objects. They provide application developers with powerful mapping and GIS capabilities which can be used in a wide variety of development frameworks including popular programming environments such as Visual Basic, Delphi, Visual C and others.

2.2 GIS and Knowledge Based Systems

2.2.1 What is a Knowledge Based System?

Expert systems (ES) or, interchangeably Knowledge based systems (KBS) have evolved as a branch of Artificial Intelligence (AI)[4] and from a broader perspective they are apparently the principal area of AI applications in various fields of human activities.[5] They have been successfully introduced for decision support in many areas, notably medicine, chemistry, engineering, military, finance etc. Recently, however KBS techniques have also been seen as a useful complement to SDSS analysis and modelling tools.

KBS technology was conceived during the 1960s and up to the late 1970s it was limited to the academic scene as a field of AI enquiry and research. By the 1980s it began to appear as a commercial application. As indicated by Turban and Anderson (1998), this was the result of substantial efforts made to develop approaches and techniques that embodied languages or tools allowing the construction of programs capable of closely resembling human reasoning.

In the literature one can find a broad spectrum of definitions and/or functional and structural descriptions of expert systems. As observed by Fedra (1991), they range "from rather narrow automata selecting pre-defined expert answers to better-than-human reasoning performance in the complex problem domains". In general, however, KBS can be regarded as "a class of interactive computer software that uses human expertise in a narrow, problem specific area (referred to as a domain) in order to perform functions similar to those normally performed by a human expert(s) in that domain (Goodall, 1985). They are fashioned along the line of how an expert would go about solving a problem and are designed to provide expert advice (Fedra, 1991). Like any other model, KBS can vary from an extreme simplification to a knowledge intensive encapsulation of expert knowledge for the particular problem domain.

As can be seen from the above definition the essence of KB systems is in that they attempt to incorporate human expertise and imitate the expert's reasoning process. What actually makes KBS feasible is an appropriate use of task-specific, empirical knowledge usually in the form of rules or heuristics and the availability of inference mechanisms for utilizing this form of information in order to derive either workable solution or expert advice for the problem at hand (Ignizio, 1991; Fedra,1991).

In the related literature one can find two principal approaches to developing a KBS. The first approach includes the use of a programming language and writing original code for the particular KBS. When this approach is selected, nearly any higher level programming language can be used, although some have been more popular than others. Generally, Prolog, SmalTalk, Lisp, and C were often called AI languages due to their characteristics. The new generations of object oriented programming

---

[4] *The term Artificial Intelligence is used to collectively group differing sets of techniques, which as their main common goal, strive to build computer software capable to mimic human knowledge.*

[5] *It should be also pointed out that many other AI techniques, besides ES, have also been utilized successfully for a wide class of problems, namely Neural Networks, Genetic Algorithms, Fuzzy Logic etc. However, those intelligent systems techniques are out of the scope of this research.*

languages are, however, considered even more useful for a KBS development as they allow new routines to be added to a program without modifying existing codes.

The second approach relies on the utilization of one of the tools developed specifically to aid the construction of a KBS. These tools are called Expert System Shells (or frameworks). They are composed of editing facilities for the construction of the knowledge base for a particular domain, general control mechanism for knowledge processing, as well as a facility for building a man-computer interface. Various shells of this kind are currently available. Examples are CLIPS (C language Integrated Production System) developed by NASA at the AI Section of the Johnson Space Centre, JESS (Java Expert System Shell), recently developed by Friedman-Hill at Sandia National Laboratories in Livermore and many others with different levels of sophistication in supporting KBS construction and implementation.

2.2.1.1 Components of the Knowledge Based System

As shown in Figure 2.8, a KBS can be described as a programming environment that contains all of the necessary utilities for developing and running the system. From the structural point of view it usually consists of the following components:

- **Knowledge base** – collection of domain specific information,
- **Inference engine** - the knowledge processor, that works with available information on a given problem in order to draw conclusions or recommendations,
- **Blackboard (Working memory)** – contains data (facts) entered by the user or inferred by the expert system during a consultation,
- **User interface** - a user friendly system front-end that controls and guides communication between user and system, allowing him/her to provide necessary input data to the system,
- **Explanation facility** - provides explanations on the reasoning of the system.
- **Knowledge acquisition** - usually seen as a subsystem for transformation and accumulation of problem specific expertise from experts and other documented knowledge sources to a computer program in order to initialise or expand the system's knowledge base.

The first three components, i.e. knowledge base, inference engine and working memory, along with the user interface are usually indicated as the generic components of KBS.

The **Knowledge base** is one of the essential parts of a KBS. It can be understood as a collection of facts representing knowledge on various known aspect of the KBS's subject area. It can otherwise be thought of as a collection of generic rules (facts) that direct the use of knowledge to solve, or provide advice for a specific instance of problem in a particular domain. The information in the knowledge base is incorporated into a KBS by a process usually called **knowledge presentation**, which will be discussed later.

If the knowledge base could be viewed as the heart of a KBS, then **the inference engine**, also known, as control mechanism is the brain of the system. This component is essentially a computer program composed of a set of procedures and algorithms for the manipulation of information contained within the system's knowledge base and its working memory in order to infer or draw conclusions. The most common strategy for drawing conclusions is based on logical deduction of conclusions from a set of facts and rules. They are very often provided in the form of "IF (premise) THEN (consequences)". This component also includes procedures and directions on how to use the system's knowledge base, as well as which facts to obtain by querying the user.

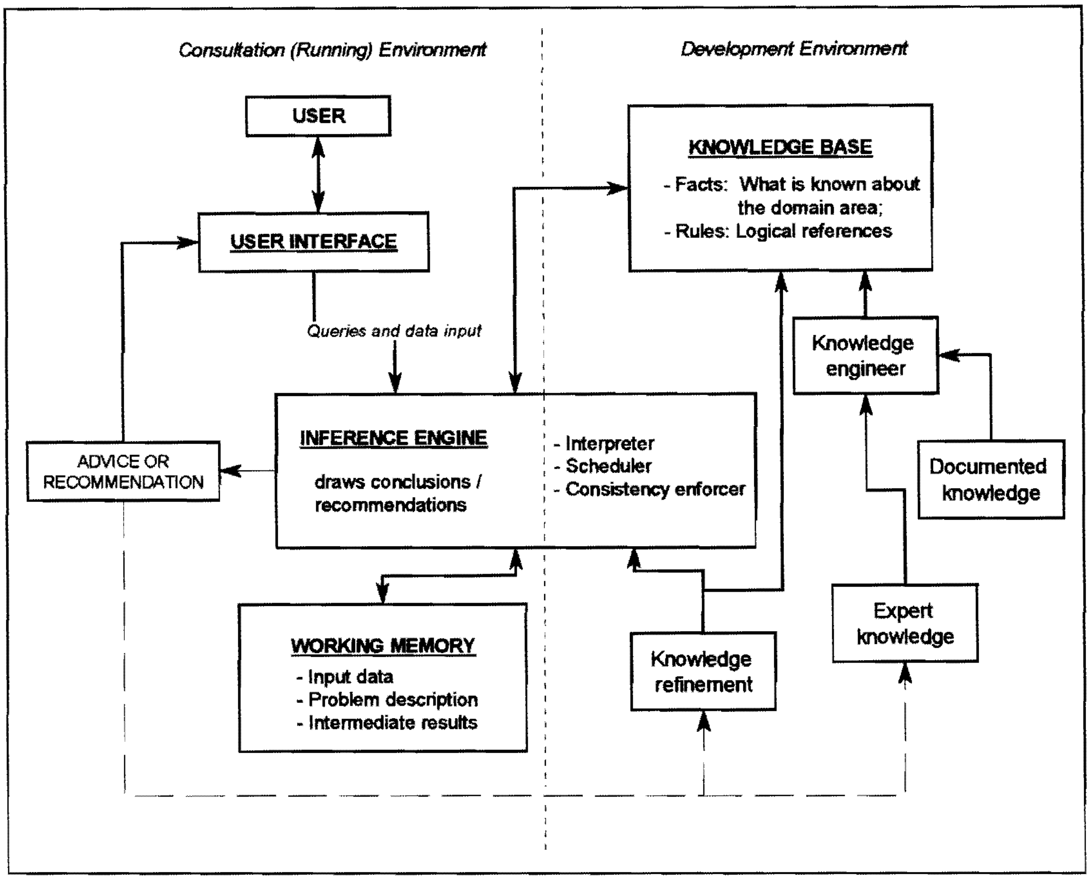

Figure 2.8 Structure of a Knowledge Based System (Turban & Aronson, 1998)

Besides the knowledge base and the inference engine a KBS system typically has a so-called "Blackboard". It is usually perceived as an area of the system's working memory set aside for both, namely the description of the current problem-solving task, as specified by input data, and for recording the system's intermediate results.

Separation of the knowledge base and the inference engine is yet another key feature that distinguishes KBS from conventional programs. This separation, usually referred to as a "plug-in" KBS architecture, allows the existing knowledge base to be detached from the system and a new one containing different sets of rules and facts to be inserted into a system. This characteristic is a basis for generic KBS software known as an expert system shell. As already indicated above the expert system shell usually consists of a general control mechanism (inference engine) along with editing facilities for entering the knowledge base for a particular subject area.

2.2.1.2 Knowledge Acquisition, Representation and Implementation

Knowledge acquisition, representation and implementation (inferencing) are the essential steps in developing a KBS. In the problem related literature they are usually defined as "knowledge engineering".

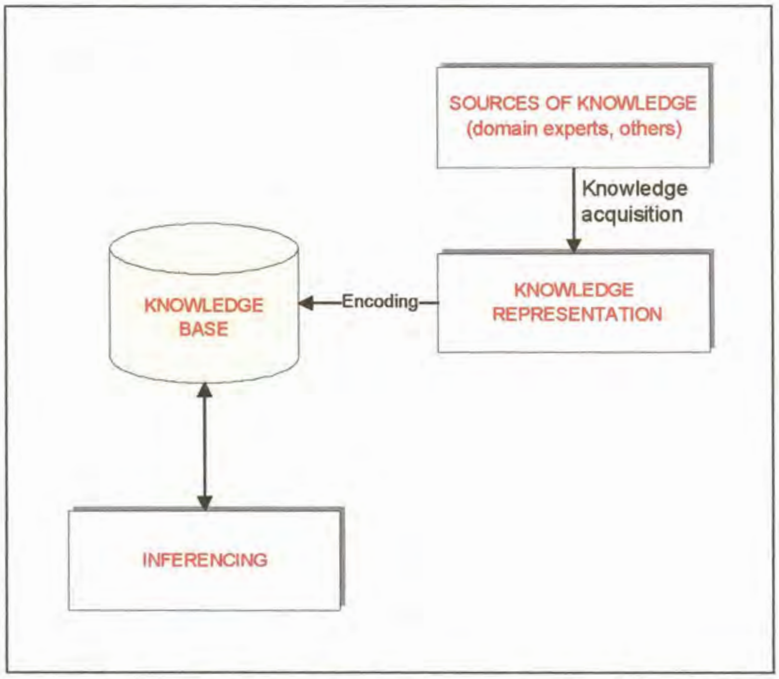

Figure 2.9 Process of Knowledge Engineering (Turban & Aronson,1998)

Knowledge acquisition involves the process of knowledge extraction from "documented" and "undocumented" sources of expertise and its transfer to the knowledge base. Usually, however, this process is accomplished through meetings between the developer (knowledge engineer) of the system and domain experts, during which the experts' knowledge is elicited, refined and encoded in the knowledge base. The elicitation of knowledge from experts is done with the aid of different methods. They can generally be classified in three categories, namely, manual, automatic, and semi-automatic.

Manual methods are structured around some kind of interview with domain experts, as well as through tracking their reasoning process and observing specific problem-solving procedures. What is typical for these methods is that they are usually very slow, expensive and sometime even inaccurate due to the fact that experts are typically asked to perform tasks that they do not ordinarily do. In that regard there is a trend in the research and practice to automate the knowledge acquisition process through development and implementation of various computer-aided methods (i.e. rule induction, case based reasoning, neural net, and intelligent agents).

Among the different techniques implemented so far, an *interactive, expert driven knowledge acquisition method* is of particular interest for this research. As illustrated in Figure 2.10 it supports experts by allowing them to build a knowledge base for a particular problem domain without the involvement of a knowledge engineer. As such it can be understood as a smart computer-based tool for capturing the expert's knowledge, distilling it, and then automatically generating a knowledge base. In addition, its purpose is to help experts in bypassing their cognitive defences and biases, as well as to identify relevant criteria and level of knowledge needed in supporting the subject area of a KBS.

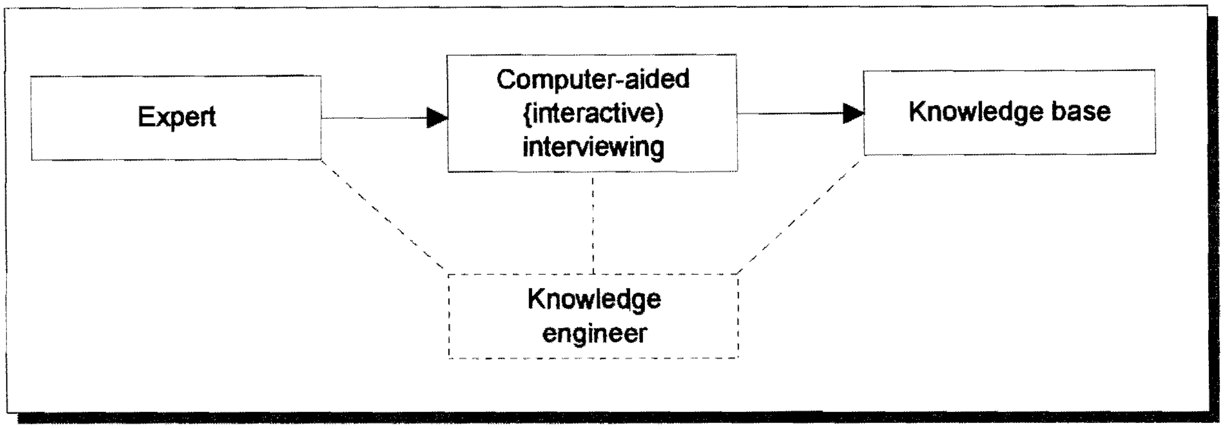

Figure 2.10 Schematic representation of an interactive, expert-driven knowledge acquisition method implemented in this research (Turban & Aronson, 1998)

Knowledge presentation is a process of defining the approach (form and format) that will be used in a KBS program to represent domain knowledge collected during a knowledge acquisition session.

As indicated in the related literature, knowledge is represented in various forms and formats including semantic networks, frames, attribute value lists, decision tables, conventional programs, etc. These knowledge representation schemes usually follow diverse algorithms and software construction in the process of building a knowledge base. Among them, *production rules (rule-based programming)* seem to be by far the most commonly used and the most directly understandable form of knowledge presentation. In this programming paradigm, rules are used to represent heuristics, or "rules of thumb" which specify a set of conclusions/advice for a given situation and/or condition. The basic idea of knowledge representation is simple. Knowledge is namely represented as IF_THEN and/or IF_THEN_ELSE rules. These are essentially association pairs; i.e. IF is a particular fact (premise/condition), THEN (ELSE) is the conclusion or action to be taken or expert advice for the problem at hand. An example is given below:

Rule1:	*Rule2:*
IF soil = type A	*IF potential = high AND flood potential = high*
THEN erosion potential = high	*THEN environmental suitability = low*

As can be seen from the example above, rules are basically a formal way of specifying how an expert reviews a condition, considers various possibilities and recommends conclusions and/or advice.

In this research however the form of knowledge presentation in supporting the proposed KBGIS model will be based on a so-called domain decision (or the truth) table approach, rather then on production rules. More detailed discussion concerning this issue is provided in the chapter 4.

Knowledge implementation can generally be understood as the process of translating the organized knowledge into an operational KBS system. It basically relies on the existence of the computer algorithm (program) that controls and carries out the reasoning process. As mentioned before, that program is called the inference engine or the problem processor. Its major task is to direct the search through the knowledge base in order to:

- Draw (infer) conclusions from a set of expertise (facts) about the problem area stored in the knowledge base;
- Reports solution, provide expert advice for the problem at hand and explain the line of reasoning;
- Interact with the user for additional information.

The process usually involves the application of various inferencing techniques (inference rules, case-based reasoning, model-based reasoning etc.) that attempt to mimic reasoning similar to those of experts. Among them, the application of inference rules is of particular interest for this research.

Inferencing with rules normally involves implementation of a so-called **modus ponens** reasoning procedure for drawing conclusions/actions. In this procedure, given a rule ***"IF A and B, then C"***, and the fact that ***"A"*** and ***"B"*** are true, then it is valid to conclude that ***"C"*** is also true. In the terminology of logic, one can express this as:

A		B
PREMISE	→	ACTION

IF premise is true THEN action is valid

(Premise could be one or more conditions)

To understand this reasoning procedure, especially from the point of view of this research one can consider the following example of an inference rule in its generic form:

```
IF   < Premise 1> AND <Premise 2>
THEN  <Inference 1> or <Inference 2>   .........
```

As can be seen from the example above when both premises (conditions) are found true the conclusion, using modus ponens, is considered valid. Under such circumstances one can say that the rule fires. In other words, firing a rule occurs only when the rule's entire hypothesis (the IF parts) are satisfied. The interface engine then acts appropriately searching the knowledge base and infers conclusions that can be either reported or stored in the system's working memory as a premise for other inference rules.

From the above example, it is clear that the inference engine contains rules that differ from those of the knowledge base. Knowledge base rules (sometimes called declarative or production rules), when implemented relate to a specific domain or expertise stating the facts and relationships about the problem. On the other hand, inference rules pertain to a more general control and search strategy for deriving actions and/or conclusions. These rules are usually called procedural rules referring to an algorithm (mechanism) on how to search the knowledge base and infer conclusions /actions, given that certain facts are known.

There are two fundamental modes of search strategy used by an inference engine, namely: *a backward chaining strategy and* a *forward chaining strategy*. The basic mechanism of the *forward chaining strategy* is reasoning from a given set of premises or rules hypothesis (the IF parts) to derive conclusion or action that follows them. On the other hand, the *backward chaining* is reasoning form conclusion/action (the THEN side of the rule) to the premises that caused them. Accordingly, a backward chaining strategy is termed a goal-driven approach, while forward chaining is usually called a data-driven approach. The latter is of particular interest for this research.

2.2.2 Knowledge Based Systems in Spatial Problem Solving

Application of KBS in location planning, environmental compatibility assessment and other spatial problem solving tasks began to appear during mid 1980s and have since than been discussed in the related literature (Robinson, 1987; Frank, 1987; Karimi, 1987; Kim, 1990; Wiggins, 1990; Wright, 1990; Fedra, 1997; Openshaw, 1997). This was the result of an increasing demand for such systems especially in problem solving situations where formal mathematical models appear to be less effective or impractical for deriving workable solutions (Ignizio, 1991; Fedra, 1991; Han and Kim, 1989). Another reason for this growth is related to the increased availability of a number of software tools (expert system shells) for building and speeding up the construction of KB systems that are not software and hardware specific and can be run on standard desktop computers.

It should, however be pointed out that the application of KBS in supporting location planning has not yet reached maturity. The most fundamental reasons, as argued by Kim and Han (1989), are disparities between the type of problems that decision-makers in spatial planning deal with and the type of problems for which the problem-solving approach of KBS is suited. Another reason can be an absence of information on successful practical application of KBS. In this regard Fedra (1991, 1994) argued that most of the KBS being described in the related literature were in a so-called research and development stage and that the number of operational ones in spatial problem solving seems to be rather small.

An overview of the literature has generally pointed to two basic types of KBS applications in the spatial problem-solving domain, namely:

1) Purely knowledge driven systems, and
2) KBS coupled with other systems either as intelligent front ends or fully embedded knowledge based models for a specific problem domain.

Purely knowledge driven systems could be seen as standalone KBS based on an empirical "model" or "qualitative understanding of how things work". They rely on sizable domain knowledge usually represented in the form of rules or heuristics, and on inference mechanisms for utilizing this form of information in order to derive either workable solution or expert advice for the problem at hand. One of the most popular areas in applying these types of KBS refers to land use control and management. This, typically well structured, spatial problem solving area appeared to be appropriate for implementation of the KBS's problem solving approach (Fedra, 1994; Turban and Aronson, 1998; Han and Kim, 1989). A typical example of such a KBS is the Decision Aid Planning Tool (ADAPT), described by Davis and Grant (1990) as a knowledge based DSS system specifically designed to assist land-use planners in producing a local government zoning scheme. Other cases utilizing standalone KBS for solving location-based problems include applications in the site selection and suitability analysis. Examples found in the related literature are: SISES (Site Selection Expert System, Findikaki, 1990), ESTMAN (Expert System for Manufacturing Site Selection, Suh et al., 1990), ESSAS (Expert System for Site Analysis and Selection, Han and Kim, 1988), ETCON (Expert System for Conservation Land Use Planning, Ahma et al., 1994).

Although these and other similar examples demonstrate that KBS could be a useful tool or approach in supporting spatial problem solving, various references in related literature, revealed specific limitations. One limitation of a standalone system for spatial problem solving is that they were not able to represent relationships between non-spatial data and spatial locations. These relationships are crucial particularly when decision rules built into these systems depend strongly on geographical location (Chulmin, 1999). Another, even more important limitation is that the typical multidimensional problem solving methods in location planning are difficult to articulate and encapsulate in the existing forms of knowledge presentations within a KBS. Kim and Han (1990), Fedra (1991) and many others argued that the nature of location planning problems, including their complexity and spatial orientation, makes purely knowledge driven systems unsuitable for a wide range of applications mainly due to their current technical limitations. They furthermore pointed out that only by integrating KBS with other information systems could one hope to effectively support a wide range of location tasks. The idea to combine the unique capabilities of KBS with other systems and vice versa has recently gained widespread attention.

2.2.3 Coupling KBS with GIS – Knowledge Based GIS

One example of functional integration that is of particular interest for this research is the linkage between GIS and expert systems, also referred as knowledge based GIS (KBGIS). The goal of this type of integration is to produce more useful computer tools that can assist in spatial problem solving, not only by conventional computing, but also by some sort of reasoning similar to those of human experts (Han and Kim, 1990).

Research efforts to couple KBS with GIS, and in the process overcome the deficiencies of GIS as a spatially enabled decision support technology, have rapidly increased since the late 1980s. (Borrough, 1986; Robinson at al, 1987; Wright at al., 1990; Kim and Han, 1990; Fedra, 1997; Densham and Armstrong, 1990; Coulson, 1992; Cowen et al., 1994; Miller, 1994; Openshaw, 1997; Matthews and Sibbald 1998; Lam, 1998, etc.).

In an effort to develop and intelligent GIS for natural resource management, Coulson (1992) noted that the usefulness of a proprietary GIS can be notably enhanced by incorporating the elements of AI techniques, especially the rule based reasoning and the expert system concept. They pointed out that for the purpose of natural resource management a GIS is an exceptionally useful tool for representation and analysis of landscape elements in the form of geographically referenced and related attribute data. However, they found a GIS an inferior tool for representing and analysing relationships among landscape elements since it does not provide any decision making and/or pattern matching modules that can reason about these relationships. Therefore individuals should have their decision rules in place before GIS can be utilized. In other words, the relations between landscape elements cannot be interpreted without the intervention of an expert. To resolve these limitations they developed a so-called Intelligent GIS (IGIS). This was accomplished by preparing a KBS containing rules or heuristic knowledge of a domain expert and, then linking it with GIS database developed with the aid of GRASS GIS software.

Leung (1993) demonstrated that the KBS concept could be an appropriate approach for approximating human reasoning and consequently enhancing the level of intelligence of current GIS systems. Through their work they argued that current GIS systems suffer from certain conceptual shortcomings that prevent their successful development as a spatially enabled decision support technology. Among these shortcomings inappropriate logical foundations and the low level of intelligence are the most important and require immediate attention. In respect of the logical foundation, they indicated that current GIS systems are predominantly based on Boolean logic which gives no room for imprecision in information, human cognition, perception and thought process. Regarding the level of intelligence, they claimed that human knowledge and expertise have not been effectively integrated into current GIS systems. To overcome these conceptual shortcomings of the present day GIS systems they developed a flexible, general purpose and fuzzy-logic based Expert System shell (FLESS) as a tool for construction of a GIS with a higher level of intelligence. The prototype of the shell has been tested on two simple knowledge-based GIS systems prepared as didactic examples. The first dealt with remote-sensed data and land-type classifications, while the second was focused on climatic classifications with regular GIS data layers. The two examples have clearly illustrated the possibilities and usefulness of the KBS approach in providing an "intelligent GIS system's front end" that could be effectively used to build a knowledge base model for a domain specific spatial problem-solving task.

Another example where a KBS is used as an intelligent front end for a GIS is the SDSS for Rural Land Use Planning developed by Matthews and Sibbald (1998). As described by the authors, the system was developed to assist rural land managers in the examination of land use/ allocation options and the potential impacts of land use change. It includes Smallworld GIS software, a land use and impact assessment model management system as well as an intelligent interface overlaying the GIS database. The interface contains a control mechanism capable of passing data from GIS to the model management system and also to capture essential information from both databases and to derive or deduce conclusions regarding land-allocation that meet the preferences of the land manager. In contrast with the earlier example, this one illustrates the role of the KBS approach in providing a descriptive dialogue between the user and the system.

Rosenblit and Jankowski (1991), Fedra (1995), Saenz (1997) and many other authors argued that the KBS concept can also be useful in designing intelligent front ends in the form of advisors and in the process minimizing or even avoiding misuse of complex models running under the GIS environment.

The incorporation of knowledge and heuristics into the GIS environment has resulted in the development of a so-called "fully embedded expert system" utilising a KBS to support spatial problem solving. In contrast with intelligent front ends designed to enhance human-system communication and the use of models and data in the decision-making process, these fully embedded knowledge based systems tend to enhance models and decision-making results. Therefore they are typically problem oriented rather than method oriented. What makes these systems useful is the appropriate use of domain specific knowledge or heuristics embedded into a GIS as a set of tools. These tools, in combination with other conventional modelling techniques available within the GIS environment, add a considerable amount of flexibility to problem solving and representation (Fedra, 1991; Ignizio, 1991).

Yialouris et al (1997) gave an example of such a flexible system. They designed an Integrated Expert Geographical Information System (EXGIS) for the assessment of land suitability for agricultural uses. The EXGIS, as described by the authors, is a modular designed knowledge based GIS that combines the capabilities of a commercial GIS - pcARC/INFO (used for spatial data storage and processing) with the rule-based knowledge system specifically developed for this project. The KBS was implemented in CLIPPER to allow transparent data transfer with pcARC/INFO. Its knowledge base contains more than 600 rules. Both the FAO system for soil evaluation and the local experience and knowledge of soil and climatic conditions were combined for the formulation of production rules of the EXGIS knowledge base. Integration of the system's components (KBS and pcARC/INFO) under the common operating environment was done through the interface developed with the aid of SML - the macro language provided by pcARC/INFO. EXGIS has been applied to study soil suitability and climatic conditions for five crops within an area of about 30,000 ha. Its evaluation, as claimed by the authors, showed satisfactory results since the conclusions drawn by the system match those of an expert.

Incorporating a knowledge-based approach to enhance GIS and spatial decision-making has been found particularly interesting in the environmental domain. Tasks in this spatial problem solving area are very often unstructured allowing heuristics, and therefore knowledge based techniques to be applied. The idea is to use GIS as a proper tool for visualisation, manipulation and analysis of spatially oriented data, while the KBS should provide a basis for catching the essential information from the database and converting it into practical advice. An example found in the literature is MEXSES - an expert system for environmental impact assessment (Fedra et al., 1991). It combines a GIS with the rule-based KBS in order to provide support for a screening level assessment at the early stage of projects planning. The KBS, as described by the authors, is composed of hierarchical impact assessment (EIA) checklists designed to guide the analyst through a reasonably complete set of expected impacts for a given project type. The checklists are combined with the inference mechanism that also includes an explanation function and a knowledge based browser connected to a hypertext system. The inference mechanism can, when necessary obtain the required data from the GIS and ask the user to choose or set values for a project type. The knowledge and explanation browser displays rules in a form transparent to the user, while hypertext links them to a handbook style definition and explanation of the term and concepts used by the system.

Another example is the Ecosystem Management Decision Support System (EMDS) recently developed by USDA Forest Service (Reynolds, 1998). The EDMS, as noted by the authors, integrates ArcView GIS and knowledge-based reasoning technologies in the Microsoft Windows environment. To conduct an assessment with the EMDS, the user is requested to: (1) prepare and/or design a template view that includes all required GIS themes; (2) construct knowledge bases that describe relations among ecosystem states and processes of interest to the assessment. To support these activities the EMDS basically integrates two key applications: (1) the NetViewer that provides a knowledge base development environment, and (2) the EMDS extension to ArcView that includes system objects and methods for processing knowledge bases in a GIS application.

Considering the above examples, it seems that KBS for spatial problem solving tends to become more sophisticated and useful when they are combined with GIS and other conventional models. Many authors argued that KBS should not be seen as a substitute for methods and models already applied within the GIS environment but rather as complementary techniques that can improve the performance of GIS in supporting spatial problem solving.

2.3 Conclusion

The purpose of this chapter was to examine how modern information technologies can provide better support for solving spatial problems. Various approaches to the integration of GIS with other decision support tools were reviewed. The intention was to examine different ways to logically integrate these systems and identify trends in system integration and thus providing the theoretical background for this research.

As indicated in the previous sections of this chapter, GIS technology is recognized as a very useful technology for most spatial problem solving tasks. However, in spite of their significant contribution, current GIS systems still suffer from certain deficiencies that prevent them from being used as full-fledged spatial decision support systems (SDSS). These deficiencies include the absence of explicit analytical and modelling capabilities, the absence of a logical structure and a low level of intelligence in terms of declarative and procedural knowledge.

To overcome these deficiencies certain extensions to current GIS have been advocated, primarily through the integration with decision-making tools drawn from other disciplines. The remedy from the GIS developers is essentially related to substantial improvements of their products, which are rapidly moving towards true distributed, object-oriented tools, sufficiently modular and programmable to permit their integration with other decision supporting tools.

Various logical ways of coupling GIS with other decision support tools have been identified, ranging from the simplest "loosely coupled integration" that only exchange files between systems, up to a so-called full integration of problem specific models into the GIS environment and vice versa. The loosely coupled integration has by far been the most frequently adopted approach in both the research and application environment, while only a limited number of attempts of coupling problem specific modelling tools and GIS within a single application have been found. It appears, however, that exploration and practical application of this "fully integrated approach" is gaining widespread attention especially in circumstances where GIS software packages are becoming sufficiently open and programmable to permit their full integration with other tools.

The concept of an intelligent GIS as a feasible solution for improving complex spatial problem solving tasks obviously exceeds the capability of present day GIS and therefore calls for the integration of expert system methods with GIS. The concept of linking the two systems has also emerged as an important research area although this has not yet reached maturity. As argued by many authors the majority of intelligent spatially enabled systems based on the integration of GIS and KBS are still in the research and development stage and the number of operational system seems to be rather small.

This research can therefore be regarded as appropriate and timely since it aims to attend to the deficiencies mentioned above and apply the concepts to a practical problem-solving situation. Firstly it aims at presenting a practical example of using a GIS to automate an existing decision-making situation. Secondly it examines how the usefulness of a "conventional" GIS in supporting the decision-making situation can be improved by incorporating (embedding) elements of artificial intelligence and knowledge engineering. Thirdly, it represents an effort to illustrate a practical example of the actual integration of the elements of artificial intelligence and knowledge engineering into a GIS. This level of GIS-KBS integration is known as the fully integrated approach and it seems quite possible even when using desktop-GIS. These desktop-GIS provide various types of utilities for file transfer and macro programming that makes it possible to extend their functionality. It is hoped that this attempt to develop a prototype KBGIS system will play a modest part in extending the knowledge and experience in integrating GIS and KBS for spatial decision-making.