

**A KNOWLEDGE-BASED APPROACH TO THE DESIGN  
AND IMPLEMENTATION OF SPATIAL DECISION  
SUPPORT SYSTEMS**

Xuan Zhu

Ph.D. Thesis

The University of Edinburgh

1995



## **DECLARATION**

I hereby declare that this dissertation represents my own work and where the work of others has been used it is duly acknowledged.

## **Abstract**

Geographical information systems (GIS), expert systems (ES) and spatial decision support systems (SDSS) are becoming important tools for supporting managers and planners in making decisions for resource and environmental management. In recent years, attention has turned to the integration of existing GIS, ES systems and other problem-solving techniques to develop more powerful SDSS systems. Such systems should lead to significant competitive advantages, such as cost savings, the ability to couple analytical modelling with heuristic reasoning, and automated explanation facilities for interpreting and justifying the results of modelling studies. However, early attempts have also demonstrated a number of drawbacks, such as user unfriendliness, lack of flexible model management capabilities and poor adaptation to users' needs. To try and overcome some of these problems, this research establishes a new approach to the development of spatial decision support systems within an integrated framework of GIS, spatial modelling and expert systems techniques and technologies. In this approach, knowledge-based techniques are introduced into the design of knowledge-based spatial decision support systems (KBSDSS), with emphasis on the design of a representation scheme based on spatial influence diagrams and mechanisms for structuring, representing and formulating spatial problems, together with automation of the solution process.

Spatial influence diagrams are graphic knowledge representations for resource and environmental problems, consisting of information about all problem variables or parameters and their relationships. They can be seen as spatial analogues of influence diagrams developed for decision analysis. However, spatial influence diagrams are deterministic cases of influence diagrams without decision components. Algorithms are developed to formulate and evaluate spatial influence diagrams using

domain-specific knowledge in the system to represent and evaluate specific spatial problems according to the decision maker's perspective.

Based on the spatial influence diagram-based representation scheme and mechanisms, the thesis proposes an architecture for KBSDSS systems. The KBSDSS architecture is composed of six components: user interface, query processing subsystem, modelling subsystem, problem processor, knowledge base and back-end subsystem. The main functions of a KBSDSS system include query, formulation of spatial problem models and evaluation of spatial problems by integrating GIS models (GIS analysis functions), analytical models (mathematical equations or functions) and rule-based models (if-then rule sets).

The thesis provides a KBSDSS development environment, which is built through integration of a GIS package (ARC/INFO), a hypertext diagramming tool (HARDY) and an expert system tool (CLIPS). A prototype KBSDSS system called ILUDSS (the Islay Land Use Decision Support System) is developed within the KBSDSS development environment to demonstrate the KBSDSS technology. It is designed to aid planners in strategic planning of land use for the development of the island of Islay, off the west coast of Scotland.

## **Acknowledgements**

I would like to thank my supervisors Dr. Richard G. Healey and Dr. Richard J. Aspinall (Macaulay Land Use Research Institute, Aberdeen). I thank them for their constant encouragement, their time and patience in refining my thoughts and for their thorough comments on early drafts of the thesis.

Special thanks are due to Dr. Julian Smart of Artificial Intelligence Application Institute (AIAI), the University of Edinburgh, who provided the HARDY software. I would also like to acknowledge the Scottish Natural Heritage who gave permission for the use of the Islay GIS database.

All equipment and computing facilities for this thesis were provided by the Department of Geography in the University of Edinburgh. I am grateful to Mr. Steve Dowers who has been a source of powerful technical support from the very beginning of my research.

I would like to thank collectively all those people who helped me in different ways: Dr. Ian Harrison (AIAI), Mr. John Fraser (AIAI), Mr. Gavin Park and Dr. Mike Mineter.

I am greatly indebted to my parents Baiming Zhu and Shuzhen Li. Their support has been unconditional and unwavering, but they have given me more than that. The really significant thing they did for me was to show me that education is the primary road to a better and more meaningful life. I am more grateful to them than I can say.

Last and not least, thanks go to my wife Ni Yu. Her support and encouragement throughout has made this work possible. My seven-year-old son Lingyun Zhu, who can speak both fluent Chinese and English, has made me very happy. I would like to dedicate this thesis to my parents, my wife and my son.

# Contents

	<u>Page</u>
<b>1. Introduction</b>	1
1.1 Traditional Computer-aided Spatial Decision Support Tools and Their Limitations .....	2
1.2 The Integrated Systems Approach to Spatial Decision Support Systems (SDSS): Benefits and Pitfalls .....	8
1.3 The Research Goal – Knowledge-Based Spatial Decision Support Systems (KBSDSS) .....	13
1.4 Summary of Aims .....	16
1.5 Thesis Structure .....	18
<b>2. Expert Systems and Knowledge Representation</b>	19
2.1 Basic Concepts of Expert Systems .....	19
2.2 Elements of Expert Systems .....	22
2.2.1 The Knowledge Base .....	23
2.2.2 The Inference Engine .....	24
2.3 The Representation of Knowledge .....	27
2.3.1 Production Rules .....	27
2.3.2 Frames .....	28
2.3.3 Other Representations .....	29
2.4 Expert System Development Tools .....	31
2.5 Summary .....	33

<b>3. GIS and Spatial Modelling</b>	34
3.1 Spatial Data Models and Structures .....	35
3.2 Geographical Features and Data Types .....	39
3.3 Taxonomy of GIS Models .....	41
3.4 Spatial Modelling in GIS Systems .....	50
3.5 Summary .....	51
<b>4. Representation and Structuring of Resource and Environmental Problems</b>	54
4.1 Basic Problem Structuring Methodologies .....	55
4.2 Formal Definitions of Spatial Influence Diagrams .....	57
4.3 Representation of Resource and Environmental Problems Using Spatial Influence Diagrams .....	61
4.4 Implementation of the Spatial Influence Diagram-Based Representation .....	64
4.5 Spatial Influence Diagrams and Cartographic Modelling .....	67
4.6 Summary .....	68
<b>5. Formulation and Evaluation of Resource and Environmental Problems</b>	70
5.1 Knowledge Requirements for Formulation and Evaluation of Spatial Influence Diagrams .....	71
5.2 Formulation of Resource and Environmental Problems .....	74
5.3 A Solution Procedure for Resource and Environmental Problems ...	80
5.4 Evaluation of Nodes in Spatial Influence Diagrams .....	84
5.5 Summary .....	93
<b>6. An Architecture for KBSDSS Systems</b>	94
6.1 Characteristics of KBSDSS Systems .....	94
6.2 The Knowledge Base Structure of KBSDSS Systems .....	97
6.3 An Architecture for KBSDSS Systems .....	101

6.4 Summary .....	107
<b>7. A KBSDSS Development Environment: Integrating ARC/INFO, HARDY and CLIPS</b> .....	108
7.1 Overall Structure of the Development Environment .....	109
7.2 System Tools .....	111
7.2.1 HARDY .....	111
7.2.2 CLIPS .....	115
7.2.3 ARC/INFO .....	124
7.3 Inter-Process Communication .....	127
7.4 Data Exchange .....	131
7.5 Development of Model Solvers and Utility Programs .....	138
7.6 Summary .....	144
<b>8. ILUDSS: A Prototype KBSDSS System for Rural Land Use Planning</b> .....	145
8.1 The Need for KBSDSS Technology in Land Use Planning .....	146
8.2 Land Use Problems on Islay Faced by Planners .....	147
8.3 The Problem Domain Addressed by ILUDSS .....	149
8.4 A GIS Database for Islay .....	152
8.5 Model Development .....	154
8.5.1 GIS models .....	155
8.5.2 Physical land suitability models .....	159
8.6 System Structure and Functionality .....	164
8.6.1 The user interface .....	168
8.6.2 The query processing subsystem .....	168
8.6.3 The knowledge base .....	171
8.6.4 The modelling subsystem and the problem processor .....	171
8.6.5 The back-end subsystem .....	174
8.7 Implementation of ILUDSS .....	174
8.7.1 Query .....	175
8.7.2 Automatic modelling .....	177



8.7.2.1	Formulation of land use problem models .....	177
8.7.2.2	Evaluation of land use problem models .....	182
8.7.3	User-assisted modelling .....	185
8.7.3.1	Construction and modification of problem models ..	185
8.7.3.2	Addition of new analytical and rule-based models ..	190
8.7.3.3	Acquisition of meta-data .....	194
8.8	Features and Limitations with ILUDSS .....	195
8.9	Summary .....	200
<b>9.</b>	<b>Conclusions</b> .....	<b>201</b>
9.1	Summary .....	201
9.2	Contributions of the Thesis .....	203
9.3	Perspective: Strengths and Limitations of KBSDSS Systems .....	206
9.4	Future Work .....	208
<b>References</b>		<b>211</b>
<b>Appendices</b>		<b>221</b>
Appendix A	Conflict Resolution Strategies in CLIPS .....	221
Appendix B	The Main Software Modules in ARC/INFO .....	223
Appendix C	Spatial Data Types Supported by ARC/INFO .....	227
Appendix D	File Structure of the ILUDSS Program .....	230
Appendix E	Running ILUDSS .....	233
Appendix F	A Sample Session with ILUDSS .....	235
Appendix G	Arithmetic Expressions in ILUDSS .....	271
Appendix H	Logical Expressions in ILUDSS .....	272
Appendix I	Shade Colour Set Used in ILUDSS .....	274

# List of Figures

<u>No.</u>		<u>Page</u>
Figure 1.1	Components of a traditional SDSS system .....	7
Figure 1.2	Components of an integrated SDSS system .....	9
Figure 2.1	Basic concepts of expert systems .....	20
Figure 2.2	Structure of an expert system .....	23
Figure 2.3	A frame describing the characteristics of a land parcel .....	28
Figure 2.4	A semantic net .....	30
Figure 3.1	Levels of spatial data abstraction .....	35
Figure 3.2	Spatial data models .....	36
Figure 3.3	The regular tessellations .....	38
Figure 3.4	A triangulated irregular network .....	38
Figure 3.5	Types of geographical features .....	40
Figure 3.6	Taxonomy of GIS models .....	44
Figure 3.7	Cartographic modelling. A cartographic model for identifying land use potential for afforestation .....	53
Figure 4.1	A spatial influence diagram constructed for predicting potential areas for development in a certain region .....	59
Figure 4.2	A basic spatial influence diagram .....	61
Figure 4.3	Value node frame of “Potential Areas for Development” .....	65
Figure 4.4	Chance node frame of “Proximity to Nature Conservation Areas” .....	65
Figure 4.5	Border node frame of “Elevation” .....	66

Figure 4.6	A spatial influence diagram for identifying land use potential for afforestation .....	69
Figure 5.1	Domain variable frame of “Proximity to Roads” .....	72
Figure 5.2	Solver frame of <i>union</i> .....	73
Figure 5.3	Solver frame of <i>combine</i> .....	73
Figure 5.4	Meta-data frame of the data layer “landcover” .....	75
Figure 5.5	Process for formulating a spatial influence diagram .....	77
Figure 5.6	Procedure for formulation of a spatial problem .....	79
Figure 5.7	Solution procedure for a spatial problem .....	82
Figure 5.8	Evaluating the land use problem of identifying potential areas for development depicted in Figure 4.1 .....	83
Figure 5.9	Process of evaluating a node .....	88
Figure 5.10	Value node frame of “Potential Areas for Development” (reproduced from Figure 4.3) .....	90
Figure 5.11	The node frame of “Potential Areas for Development” after the values of its predecessors have been propagated to it .....	91
Figure 5.12	The node frame of “Potential Areas for Development” after its value has been obtained .....	92
Figure 6.1	Knowledge base structure .....	98
Figure 6.2	Components of an IDSS system .....	102
Figure 6.3	Architecture for KBSDSS systems .....	103
Figure 7.1	Conceptual architecture of a KBSDSS system development environment .....	110
Figure 7.2	HARDY Diagram Type Manager .....	113
Figure 7.3	A spatial influence diagram in HARDY .....	114
Figure 7.4	The components of CLIPS .....	116
Figure 7.5	A sample dataset in ARC/INFO .....	133
Figure 7.6	A deftemplate definition .....	134
Figure 7.7	A sample dataset to be imported into CLIPS .....	135
Figure 7.8	Sample CLIPS rules for I/O .....	136

Figure 7.9	Process of reasoning on the data imported from an INFO database .....	137
Figure 7.10	A sample input data file to the model solver “forestsuit” .....	142
Figure 7.11	A sample output from the model solver “forestsuit” .....	143
Figure 8.1	Study area .....	148
Figure 8.2	Functions of ILUDSS .....	165
Figure 8.3	Overall structure of ILUDSS .....	167
Figure 8.4	Screenshot of ILUDSS .....	169
Figure 8.5	ILUDSS help system .....	170
Figure 8.6	<b>Data</b> menu in the main window .....	175
Figure 8.7	<b>Model</b> menu in the main window .....	176
Figure 8.8	<b>Diagram</b> menu in the main window .....	176
Figure 8.9	<b>File</b> menu in the main window .....	177
Figure 8.10	Pop-up land use list .....	179
Figure 8.11	Pop-up attribute list .....	179
Figure 8.12	Formation of a spatial influence diagram .....	181
Figure 8.13	<b>Evaluation</b> menu in a diagram window .....	183
Figure 8.14	<b>File</b> menu in a diagram window .....	186
Figure 8.15	<b>Node</b> menu in a diagram window .....	187
Figure 8.16	Spatial influence diagram editor (Diagram window with an attribute editor) .....	188
Figure 8.17	A spatial influence diagram for creating a buffer zone using the GIS model <i>buffering</i> .....	190
Figure 8.18	<b>Add new models</b> menu in the main window .....	191
Figure 8.19	A pop-up window for adding an analytical model .....	191
Figure 8.20	A pop-up window for adding a rule-based model .....	192
Figure 8.21	A pop-up window for adding a reclassification table .....	193
Figure 8.22	A pop-up window for acquisition of meta-data .....	195

Figure 8.23	Land use problem model for identifying potential sites for afforestation according to the physical land suitability for forestry, the proximity to roads, SSSI areas and existing woodlands .....	198
Figure A.1	Screenshot of ILUDSS when it starts up .....	237
Figure A.2	Screenshot of ILUDSS when a land use model has been formulated .....	242
Figure A.3	Screenshot of ILUDSS when the map of physical land suitability for forestry is displayed .....	266
Figure A.4	Screenshot of ILUDSS when the 500-meter buffer zone around roads has been created and is displayed .....	266
Figure A.5	Screenshot of ILUDSS when the 100-meter buffer zones around SSSI areas has been created and is displayed .....	267
Figure A.6	Screenshot of ILUDSS when the final result is reached and displayed .....	267
Figure A.7	Screenshot of ILUDSS when the land use model for evaluating the land use potential for peat cutting has been formulated .....	268
Figure A.8	Screenshot of ILUDSS after the land use model for evaluating the land use potential for peat cutting has been modified .....	268
Figure A.9	Screenshot of ILUDSS when the map of peat deposits is displayed .....	269
Figure A.10	Screenshot of ILUDSS when the 300-meter buffer zone around roads has been created and is displayed .....	269
Figure A.11	Screenshot of ILUDSS when the map of the potential sites for peat cutting has been created and is displayed .....	270

# List of Tables

<u>No.</u>		<u>Page</u>
Table 3.1	Levels of measurement for attribute data of geographical features .....	40
Table 8.1	Data comprising the Islay GIS data base .....	153
Table 8.2	GIS models available in ILUDSS and their usage in spatial influence diagrams .....	156
Table 8.3	Agriculture capability classification .....	160
Table 8.4	Forestry capability classification .....	162
Table 8.5	Rules for assessment of physical land suitability for forestry .....	163

# Chapter 1

## Introduction

Environmental scientists, resource managers and other decision-makers are faced with an increasing number of resource and environmental problems. The decision-making process in environmental and resource management involves a series of steps ranging from the analysis and diagnosis of problems, to determination of alternative solutions, to selection of a “best” solution and finally evaluation of the choice (Dueker 1980). Throughout the phases of the decision making process, complex sets of multivariate information are combined and synthesised with multiple criteria to evaluate problems. At each phase, interpretations and recommendations are based on human experience and expertise in combination with factual and analytical information.

Geographical information systems (GIS) and spatial decision support systems (SDSS) are becoming important tools for supporting managers and planners in making decisions for resource and environmental management. They place the immense spatial data storage, computation and analysis capabilities of modern computers at the finger tips of decision makers. Since the mid-1960s, Artificial Intelligence (AI) has achieved considerable success in the development of Expert Systems (ES). Expert systems are also called knowledge-based systems. They are designed to represent and apply human knowledge in specific areas of expertise to solve real-world problems (Jackson 1990). Expert systems or knowledge-based techniques have proven to be very attractive for a variety of problems, which can be

solved through the use of heuristic methods or rules of thumb. The reasoning and explanation capabilities offered by expert systems are very effective. There has been substantial attention devoted to the use of expert systems as tools for spatial decision support.

In recent years, attention has turned to the integration of existing GIS, ES systems and other problem-solving techniques to develop more powerful SDSS systems. Such systems should lead to significant competitive advantages, such as cost savings, the ability to couple analytical modelling with heuristic reasoning, and automated explanation facilities for interpreting and justifying the results of modelling studies. However, early attempts have also demonstrated a number of drawbacks, such as user unfriendliness, lack of flexible model management capabilities and poor adaptation to users' needs. To try and overcome some of these problems, this research establishes a new approach to the development of spatial decision support systems within an integrated framework of GIS, spatial modelling and expert systems techniques and technologies. It aims to provide decision makers with an interactive and flexible spatial decision making environment which allows the decision maker to evaluate alternative solutions based on his or her objective and subjective judgement.

## 1.1 Traditional Computer-aided Spatial Decision Support Tools and Their Limitations

Within the last decade, a variety of computer-based problem-solving and decision support techniques have been developed and used to solve natural resources and environmental problems. Among them, GIS, ES and SDSS systems have proved to be valuable tools in spatial decision-making (Kessel 1990; Parent and Church 1987; Robinson et al. 1987; Densham and Armstrong 1987; Smith et al. 1987; Peterson



1993; Foster et al. 1992; Rauscher et al. 1991). Each of these tools was developed for a particular type of problem, and has its strengths and limitations.

## **Geographical Information Systems**

GIS systems are capable of integrating data from different sources, managing, analysing and displaying resource and environmental data. They provide spatial decision makers with a means of integrating information to understand and address many spatial problems. Typically, a GIS system has following capabilities:

- extracting and describing spatial problems and their spatial relationships,
- storing and managing large quantities of complex and heterogeneous spatial data,
- structuring the available information using spatial or geographical models,
- effectively manipulating and analysing spatial data, providing spatial data handling facilities from simple map overlay to some complex spatial analyses,
- providing various display facilities.

The areas of application range from the monitoring and modelling of resource and environmental problems to the support of management and planning decisions.

With the capabilities described above, GIS systems can assist in many stages of the spatial decision making process. They can be used to describe the past and present situation of resources and environmental conditions, identify problems when a decision is required, and generate possible solutions. More importantly, GIS systems provide the ability to ask “what if” questions in order to evaluate alternative scenarios (Teicholz and Berry 1983) and examine the future. For example, they can help the land use planner to answer the following questions: What is the current land use pattern like in the area? What will happen if present agricultural management practices remain unchanged? What adverse effects are associated with each type of land use? GIS systems are implicitly designed to support spatial decision making. They offer the possibility of underpinning a sophisticated spatial

decision support system. However, current GIS solutions have a number of limitations, which include the following:

- they only have comparatively simple analytical functions, and do not adequately support spatial analysis or analytical modelling (Openshaw 1991);
- they provide few facilities for explaining analytical results and lack the ability to interpret the meanings of results;
- they implicitly assume that all information encoded is absolutely correct and precise. However, the location and nature of a phenomenon may not be uniquely definable. Exact data models and modelling techniques in GIS systems variously cause loss of information, inconsistency of phenomena and inaccuracy in analysis (Berry 1987b; Burrough 1986; Kollias and Voliotis 1991; Wang et al. 1990);
- they lack user-friendly user interfaces and do not facilitate user interaction during the solution process (Frank et al. 1991). While GIS systems allow spatial databases to be integrated and manipulated through a set of procedures, a comprehensive spatial analysis procedure from initial data entry through to product generation requires extensive user interaction in the sequence of processing steps;
- current GIS systems adopt the procedural paradigm, which solves problems through the implementation of an algorithm in a program. Most commonly, they require problems to be quantified so that some mathematical techniques can be applied to them (Davis et al. 1987). Thus, they are not good at handling qualitative data, and provide few facilities for representation, storage and processing of human knowledge and expertise.

## **Expert Systems**

Expert systems emulate the decision making ability of human experts and perform decision-making tasks by reasoning using domain knowledge that has been judged by an expert in his domain to be true. An ES system typically contains an inference engine and a knowledge base. The knowledge base is composed of the

knowledge required to solve specific problems. It exposes knowledge about some domain explicitly via symbolic data structures. In a rule-based system, the knowledge base contains domain knowledge needed to solve problems coded in the form of rules. The inference engine is a set of procedures which operate upon the knowledge base. Users supply facts or other related information to the ES system and receive “expert” advice or expertise in response. The same knowledge encoded in the knowledge base can be simultaneously used for more than one purpose, such as solving a given problem, explaining the solutions produced by the system and offering advice about the problem. In addition, the power of an expert system can be extended by expanding the knowledge base. It can be easily achieved because knowledge is isolated from control and reasoning procedures. Thus, a large system can be developed incrementally. Typical tasks for expert systems involve automated interpretation, diagnosis, monitoring, and planning sequences of actions (Jackson 1990). They put emphasis on developing and understanding non numerical methods for problem solving. The distinctive strength of ES systems can be summarised as (Jackson 1990; Hayes-Roth et al. 1983):

- handling imprecise data, incomplete and inexact knowledge;
- exploiting knowledge “at the right time”;
- explaining and justifying the reasoning that led to a conclusion;
- changing or expanding knowledge relatively easily.

Since the 1980s, many expert systems for natural resources management have been developed (Davis and Clark 1989; Robinson et al. 1987). Expert systems have now been universally recognised as major problem-solving tools in spatial decision making. The approach to be presented in this thesis takes some advantages of ES systems described above, including their ability to handle incomplete and inexact knowledge and expertise, apply the expertise of the expert within a given problem domain to problem solving in a right time and the ability to develop a system incrementally. However, expert system technology alone does not adequately support spatial decision making. It has the following limitations:

- spatial decision-making requires large volumes of spatial data. These data mainly reside in GIS systems and not in ES systems. ES systems lack facilities for handling large-scale data sets (Stonebraker 1986);
- expert systems are concentrated on symbolic reasoning and do not provide good arithmetic capabilities (Jackson 1990). Yet, arithmetic operations are required in spatial data handling. Mathematical models are often used in spatial analysis as part of a solution process;
- ES systems lack spatial data handling capabilities such as buffering and overlay which are unique and important to spatial analysis. The inference speed is also too slow for many problems typically handled by a GIS system (Webster 1990);
- ES systems do not provide facilities for representation of complex spatial phenomena and for output in a variety of spatial forms such as maps, images and other types.

### **Spatial Decision Support Systems**

SDSS systems provide decision makers with a decision-making environment which incorporates different tools such as spatial database management, spatial modelling, analytical models, graphical display and tabular reporting (Densham 1991). Generally, an SDSS system is implemented for a specific problem domain. Densham and Goodchild (1989) suggested that SDSS systems can be viewed as “spatial analogues of decision support systems (DSS) developed in operations research and management science”. As Geoffrion (1983) described, a DSS system has the following distinguishing characteristics:

- being capable of solving ill-structured problems;
- having a powerful user interface that is easy to use;
- integrating analytical models and data;
- being able to generate a series of feasible alternatives;
- supporting a range of decision-making styles;

- supporting interactive and recursive problem solving.

As the spatial nature of environmental management problems virtually dictates the use of GIS technology, GIS systems have been taken as a major tool in SDSS systems. In recent years, many SDSS systems have been designed and implemented using GIS technology, coupled with specific analytical modelling techniques and models (for examples, see Walker and Moore 1988; Pearson et al. 1991; van der Vlugt 1989; Negahban et al. 1993). The architecture of these traditional SDSS systems can be depicted as in Figure 1.1. A traditional SDSS system puts emphasis on information access and display, and on numeric computation by analytic models. It has the same limitations as those of GIS systems. In addition, the knowledge required for building and working with such systems is hidden in procedures and algorithms, or embedded implicitly as mathematical formulas and analytical models which are handled by model management algorithms, or by the user via the user interface. They can not be easily changed and applied to other similar applications. As part of a decision making process, decision makers use their experience and expertise together with factual knowledge or/and analytical results gained from an SDSS system to develop corresponding recommendations and explanations. Extensive user interaction is required. The user of the system must have knowledge about the problem domain and knowledge about its tools.

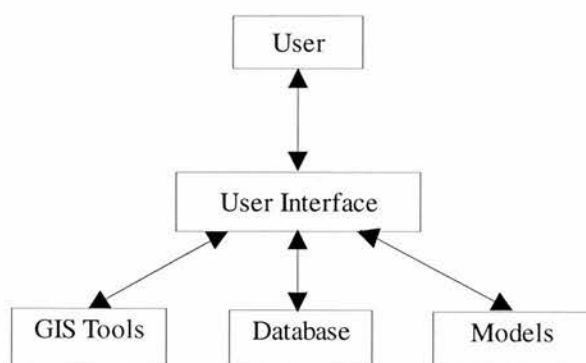


Figure 1.1 Components of a traditional SDSS system

## 1.2 The Integrated Systems Approach to Spatial Decision Support Systems: Benefits and Pitfalls

GIS, ES and SDSS systems have been applied in a wide range of areas, such as locational planning (Densham and Armstrong 1987; Armstrong et al. 1990; Shaw and Maidment 1988), drought management planning (Palmer and Tull 1987), pest management (Ravlin 1991) and estate development (Peterson 1993).

While GIS, ES and SDSS systems can each be used to support spatial decision making as stand-alone systems, they have some limitations in supporting spatial decision making as described above. These differing limitations have promoted the integration of GIS, ES systems and other modelling techniques.

Recently, efforts have been made to integrate existing GIS, ES systems and other problem-solving techniques to develop applications in the area of natural resources management and environmental planning. For example, GEODEX (Chandra and Goran 1986) was built to assist planners in allocation of land to specified land use activities. It integrates ES technology with the spatial data processing capabilities of a GIS system (MAP). The system accepts maps as an input and use the MAP (Map Analysis Package) to perform map overlay. ASPENEX (Morse 1987) was designed to assist management of aspen resources in the Nicolet National Forest, Wisconsin which interfaces an expert system shell (EXSYS) with a GIS system (MOSS). Diamond and Wright (1988) described a system designed for multiobjective land-use planning, which links GRASS (a GIS package), KES (a rule-based system) and a multiobjective programming model. Miller and Xiang (1992) developed a system which coupled ARC/INFO (a GIS package) with VP-Expert (a PC-based ES shell) for visual impact assessment in transmission line siting. INFORMS-TX (Williams 1992) integrates ARC/INFO, CLIPS (an ES tool), ORACLE (a relational database management system), and various forest resource models. It is intended for facilitating project-level planning in forest ranger districts. IRMA (Loh and Rykiel 1992) integrates an expert system, a database management

system and a GIS for supporting resource managers in forest planning. The integration of GIS, ES systems and decision models may avoid some of the limitations and difficulties existing in each of them, and take advantage of their strengths.

The benefits of the integrated systems approach can be realised along several dimensions: the ES contribution, the GIS and spatial modelling contribution, and the synergy resulting from the joint contributions. The integration of GIS, ES and modelling systems offers capabilities for spatial modelling and large-scale spatial data management, as well as heuristic reasoning. Large-scale SDSS development can be expedited through the integration of GIS, ES and modelling systems, each representing the best available technology in a particular application area. The spatial decision process can be made more effective within such integrated systems. It not only automates the storage and manipulation of knowledge, but also affords the opportunity to integrate the analysis of text, numerical and spatial data using the knowledge embedded in an expert system. We call such systems **Integrated SDSS Systems**.

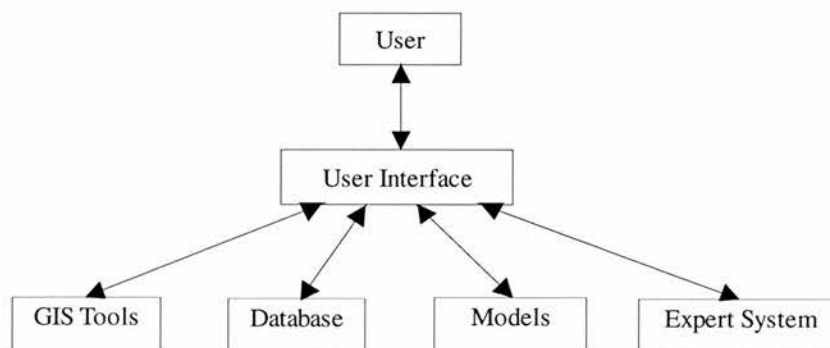


Figure 1.2 Components of an integrated SDSS system

An integrated SDSS system consists of a user interface, a database, a GIS component, an ES component and analytical models (Figure 1.2). Within such a system, the GIS prepares data for use in analytical models and the ES, through access

to the database or by performing some GIS analysis functions, and it displays modelling and reasoning results. The ES component performs symbolic reasoning based on knowledge in the knowledge base and the facts (or data) contained in the GIS or derived from the analytical models. If the above tools can be linked together, two main new capabilities result from the integration which are as follows:

- Integrating analytical models with symbolic reasoning

The integrated SDSS system provides the possibility of coupling mathematical models with symbolic reasoning. This can be done in several ways. For example, we may use the analytical models to compute criteria and then use these in the reasoning process, or we may derive weighting strategies for the model through reasoning. INFORMS-TX is an example. It determines the suitability for forest treatment through reasoning and it evaluates alternatives using models (Williams 1992).

- Justifying the decision

The integrated SDSS system makes it possible to build in an explanatory capability in the same way that decision makers can explain their reasoning, so as to justify their decision and increase the confidence that the correct decision is made.

The SDSS systems which integrate GIS, ES and modelling tools show great promise for supporting spatial problem solving and decision making. However, they do have shortcomings. Existing integrated SDSS systems can be seen as “tool-boxes” which are a collection of GIS, ES and modelling tools.

In early integrated SDSS systems, the user is responsible for integration of all components of the system to perform a task. For example, in order to derive the land suitability for a given land use type using the knowledge base in the ES component, the user has to select and retrieve the relevant data sets from the database in the GIS system and perform appropriate GIS analysis operations to produce the required information. After the information is created by the GIS system, the user has to transform the information into the format required by the expert system before it can be input into it. When the ES reasoning finishes, the results are transferred back to



the GIS system. Then, the user has to design the output format to display the results using GIS display functions. GEODEX (Chandra and Goran 1986) is an example of this type.

The user of such an integrated SDSS system has to cope with the intricacies of the GIS and other software tools it contains, as well as the complexity of the spatial problems themselves. GIS systems are powerful toolboxes which are becoming more and more complex (Burrough 1992). An integrated SDSS system may be even more difficult to use since data has to be exchanged between modules and a series of different processes are interlinked. To use them effectively, requires knowledge about the problem domain, the solution strategy, the methods of selecting and employing system facilities, and detailed knowledge about the system tools and the functions they offer. Therefore, (a) new users need considerable training in running the system; (b) implementation of each tool and its functions and models may be complex due to obscure input formats. In addition, the calling sequences of parameters need to be specified and large amounts of input data may be required.

The second type of integration has been adopted in IRMA (Loh and Rykiel 1992). This approach hides the complexity of data structures and technical details from the user. The integration of all components of the system is accomplished by a user interface shell. The shell consists of a message manager and a graphical user interface handler. When a user makes a request by selecting a menu item, the message manager automatically sends a message to an appropriate subsystem. If the subsystem, say the expert system, requires some information from the GIS system for producing the answer for the user's request, it sends a message to the GIS system. Upon receipt of the request from the expert system, the GIS system will in turn ask the database to retrieve the related data, perform certain analysis functions and send the results back to the expert system. After the expert system completes its inference, the final results are sent back to the graphical user interface handler for display via the message manager. The complex interactions between the components of the system is regulated by the shell and are invisible to the user. This kind of integration gives the user access to a set of stored analyses or models, but limits the flexibility of the system at the same time that the user is freed from having to be acquainted with

the necessary procedural instructions for constructing the analysis. It is also unable to handle decision problems where the preferred solution is sensitive to the specific preferences and desires of one or several decision makers.

Another type of integration was attempted by Djokic (1993). The principle on which this approach is based is very similar to that used in the second type of integration. Both approaches focus on the problem of the data exchange and command control. Djokic attempted to develop a generic SDSS shell. In his system, models are domain-dependent, designed to support a particular class of problems encountered by the intended users. The user interface allows the user to select appropriate data and models to perform an analysis in response to a particular problem. The choice of an analysis is left to the user. Such a system provides little support in use and requires that users have a lot of knowledge about tools, models and their implementation details because he or she must procedurally specify the required inputs to the analysis. The users have to be aware when, how and in what sequences to use the existing models and tools in combination to solve their specific problems, although they are not required to know about complex data structures and other technical details.

The last two types of integration may relieve some problems with the first type of integration, and they make the system more user friendly. The two types of systems can automatically exchange data and pass control from one component to another without the need for intervention by the user. They enable integrated use of tools and data to perform specific analyses. However, just as with the first type of integration, the latter two types of integration are only the physical integration of models and tools. They either restrict the flexibility of the system or require the user to have considerable knowledge about tools, including models, data and other implementation details. Moreover, these systems provide the user with little support in how to select appropriate models and other system tools, and how to use them in combination in order to solve a particular spatial problem. They do not provide support in model formulation and integration. In other words, they lack flexible model management capabilities.

Most decision problems in resource and environmental planning can rarely be solved by single models encompassing all problem aspects. Instead, such problems often require the integration of multiple models, each addressing specific subproblems, as well as heuristic knowledge from human experts. Consider the problem of solving rural land use conflicts within a region. Fundamental trade-offs here are between physical land suitability for each competing land use, the environmental impacts of certain land use patterns, the required investment and economic returns of each land use. There may be numerous suitability models addressing variations of these trade-offs. However, geographical characteristics determine the physical land suitability and environmental impacts of each land use. Market economic models that forecast the prices of agricultural products may play a role. Government planning policies also will enter the decision. This integration is still an art practised by professionals with little or no machine assistance.

The overall trend of SDSS development shows that the decision problems to be solved by SDSS systems are becoming more and more complex (Djokic 1993). Thus, there is a great need for different models to be integrated to analyse and support decisions. Effective computer-based support for spatial decision making requires that the SDSS system have generalised and flexible model management capabilities for model formulation and integration. In order to make SDSS systems more user-friendly and provide flexible model management capabilities, a new approach to the development of SDSS systems is needed.

### 1.3 The Research Goal — Knowledge-Based Spatial Decision Support Systems

The objective of this research is to develop a new approach to the development of spatial decision support systems within an integrated framework of GIS, spatial modelling and expert systems techniques and technologies, to support decision

making in natural resource and environmental management and planning. It aims to overcome the following problems of many existing integrated SDSS systems:

- user-unfriendliness;
- lack of flexible model management capabilities;
- poor adaptation to user's needs and preferences.

SDSS systems need to be user-friendly so that they can be used by planners and managers who are knowledgeable in their domains of expertise, but may have little knowledge of the necessary computer system commands. Generally, SDSS systems were designed for specific domains. They embed domain-specific knowledge. **Domain-specific knowledge** is the knowledge about the problem domain for which the SDSS system is to be used. It involves theories and concepts of a particular problem domain. The focus of most SDSS systems has been on the ability to perform analyses and modelling with data and models. However, the spatial decision process involves more than data interpretation. Before data and results are obtained, planners or managers are faced with a series of tasks: firstly, building the database relations and models; then, deciding modelling strategies; selecting appropriate data sets; choosing sequences of commands for analyses; and finally, displaying the results of the analyses or offering solutions to the problems. Few of these tasks involve domain-specific knowledge of a particular resource and environmental application. Instead, they involve knowledge of how to perform spatial modelling and how to run and use a set of tools for particular analytical purposes. This type of knowledge is called **tools knowledge**. Tools knowledge is knowledge about a set of tools, including GIS functions, decision models and other spatial modelling techniques, needed in problem solving. Traditional SDSS systems lack tools knowledge. Many existing SDSS systems assume that users have expertise with the tools. The user has to act as a domain expert and a tools expert. It was considered that an effective way to overcome the problem was through incorporating tools knowledge and expertise into the SDSS system using knowledge-based or expert systems techniques (Zhu and Healey 1992; Usery et al. 1988; Lein 1992; Armstrong et al. 1986). Tools knowledge

and expertise is encoded in a tools knowledge base in the SDSS system. Therefore, we introduce knowledge-based techniques into the design of an SDSS system, and develop methods for the knowledge representation of GIS tools, models and data and construction of the tools knowledge base, so that tools knowledge and expertise will not be a requirement for the use of the SDSS system.

As we argued above, if an SDSS system is to become a useful tool in the task of spatial problem analysis and decision support, it should provide the user with effective support in selecting appropriate models and solution strategies to solve his or her problems. That is, an SDSS system should have flexible model management capabilities, such as model selection, integration and formulation. However, solution of the problem is possible only after the problem is structured and a suitable model formulated. A decision support system that can structure and simplify the problem and present the complex interactions of the problem parameters in a comprehensible manner, is of great potential value (Banerjee and Basu 1993). Thus, an appropriate technique for spatial problem structuring and representation is needed to facilitate the machine representation and manipulation of spatial problem models. In this light, we develop a spatial influence diagram-based representation scheme and mechanisms for spatial problem representation, formulation, structuring, communication and evaluation. A spatial influence diagram is a graphical representation of a spatial problem, which consists of nodes and directed arcs with no cycles. The nodes represent the problem parameters or variables relevant to a particular spatial problem. The arcs represent relations between the variables. We use spatial influence diagrams as a “front-end” for automated spatial problem-solving aids. The spatial influence diagram-based representation scheme provides an interface through which heuristics and algorithms may be developed to build the capabilities for integrating data and models logically, and driving the solution process for spatial problems. In this research, we distinguish three types of models: **analytical models**, **GIS models** and **rule-based models**. An analytical model refers to a procedure composed of mathematical equations, such as arithmetic equations, probabilistic formula, regression equations and linear programming functions. A GIS model is constructed using GIS analysis functions, which operates on spatial data (digital maps). A rule-

based model is a set of knowledge base rules that perform reasoning to infer a solution to a particular problem.

The spatial influence diagram-based approach provides flexibility in model integration and problem formulation in response to users' problems. As users get more involved in interactive spatial decision support systems, the demand for "what-if" analysis increases. As the problem parameters are modified, the system is needed to identify alternative models to represent and solve the modified problem. Our spatial influence diagram-based approach has such mechanisms. As the problem parameters are modified, a new spatial influence diagram will be automatically formulated to represent the modified problem. Thus, the system can be easily adapted to the user's needs and desires. On the other hand, spatial influence diagrams can be represented in the same form in which they are evaluated. It is natural for the decision makers and experts to understand the process and to be involved.

An evolutionary approach to system development is an important feature of rule-based expert systems techniques. That is, the knowledge base can be incrementally improved by adding or modifying rules. This advantage facilitates advanced prototyping. In this research, we develop methods for reasoning about the structure of spatial problem models and driving the solution process for the problems in a rule-based manner based on domain knowledge.

## 1.4 Summary of Aims

To summarise, our goal is to introduce knowledge-based techniques into the design of **Knowledge-Based Spatial Decision Support Systems (KBSDSS)** within an integrated framework of GIS, spatial modelling and expert systems techniques and technologies, with emphasis on the design of a representation scheme based on spatial influence diagrams and mechanisms for representing, structuring and formulating spatial problems, together with automation of the solution process. A KBSDSS system is an interactive tool that uses knowledge-based or expert systems

techniques and integrates GIS, analytical and rule-based models to make it a model-based expert system for resource and environmental decision support. A KBSDSS system is able to provide the machine assistance for formulation of the spatial problem according to users' preferences over its solutions, design and execution of a solution process by automatically integrating different types of models and data available in the system.

This dissertation has the following objectives:

- development of a methodology for representing and structuring spatial problems based on spatial influence diagrams;
- development of spatial influence diagram-based mechanisms for formulating and evaluating spatial problems, and automating the solution process;
- development of an architecture for KBSDSS systems;
- creation of a KBSDSS development environment through integration of a GIS software package, a diagramming tool and an ES development tool;
- design and implementation of a prototype KBSDSS system within the framework presented in this thesis to substantiate the major principles of the KBSDSS design and demonstrate KBSDSS technology.

This research focuses on the principles of KBSDSS system design rather than those of human-computer interaction. The prototype KBSDSS system called ILUDSS (the Islay Land Use Decision Support System), which is presented in this thesis, is designed to aid planners in strategic planning of land use for the development of the island of Islay, off the west coast of Scotland. However, this prototype is developed mainly for demonstration of the design principles of KBSDSS systems, rather than testing levels of human-computer interaction. Thus, the intended audience of this thesis is more likely to be developers of systems rather than their end users.

## 1.5 Thesis Structure

This dissertation is organised as follows.

Chapter 2 provides an introduction to expert systems, and gives an overview of knowledge representation schemes for expert systems.

Chapter 3 presents general background about GIS functionality and spatial modelling, proposes a GIS model taxonomy and introduces the principle of cartographic modelling in GIS systems.

Chapters 4 and 5 present the spatial influence diagram-based representation scheme and mechanisms. Chapter 4 introduces the concepts of spatial influence diagrams and develops a spatial influence diagram-based representation scheme to represent resource and environmental problems. Chapter 5 develops algorithms and heuristics for the formulation of spatial problem models, solution of the spatial problems by evaluating spatial influence diagrams and selection of different types of models.

Chapter 6 discusses the advantages of KBSDSS systems, proposes a knowledge base structure for KBSDSS systems and presents a KBSDSS architecture.

Chapter 7 presents a KBSDSS development environment, which integrates a hypertext diagramming tool (HARDY), a GIS package (ARC/INFO) and an expert system tool (CLIPS). An integration approach is proposed to create highly integrated system through heterogeneous coupling of different tools.

Chapter 8 describes ILUDSS, a prototype KBSDSS system for decision support in rural land use problems. It is built within the KBSDSS development environment developed in Chapter 7, and implements our framework for KBSDSS design. The implementation includes all of the major KBSDSS features described in this dissertation.

Chapter 9 contains conclusions and directions for future work.



## Chapter 2

# Expert Systems and Knowledge Representations

An expert system is an important component of KBSDSS systems. In the previous chapter, we discussed briefly the features of an expert system, its advantages and disadvantages in spatial problem solving and decision support. In this chapter, we will look more closely at the functional and structural characteristics of expert systems and discuss some of the commonly used knowledge representations that they employ. The tools currently available for development of expert systems are also reviewed.

### 2.1 Basic Concepts of Expert Systems

**Expert systems** are computer systems that use specialised knowledge and general-purpose procedures to solve problems at the level of a human expert (Jackson 1990). An **expert** is a person who has knowledge or special skills to solve problems that most people can not solve efficiently, if at all. Expert systems are also called **knowledge-based systems**. The terms are often used synonymously. Figure 2.1 illustrates the basic concepts associated with expert systems.

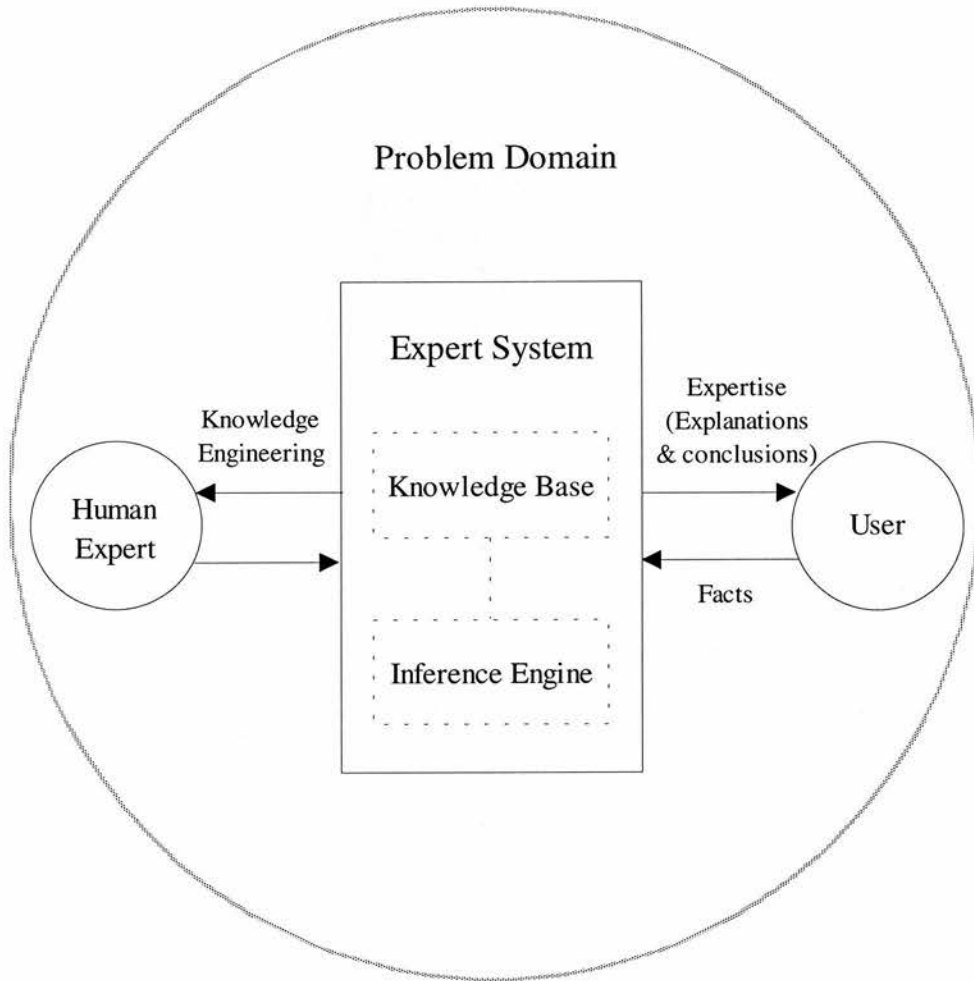


Figure 2.1 Basic concepts of expert systems

Expert systems are generally designed differently from procedural programs (such as C, FORTRAN and PASCAL programs), because the problems they are used for usually have no algorithmic solutions and rely on inferences to achieve reasonable solutions. Here, a reasonable solution is the best we can expect if there is no algorithm available to achieve the optimum solution. An **algorithm** is a method of solving a problem in a finite number of steps. The implementation of an algorithm in

a program is a **procedural program**. The development of expert systems is based on a descriptive theory of human problem solving and focuses on a representation of expertise or knowledge (Jackson 1990). **Expertise** is the knowledge acquired by humans through practice and learning. It is a specialised type of knowledge that experts have, which is not commonly found in public sources of information. Expertise consists of all the particular heuristics and shortcuts that human experts have learned to use. **Heuristics** are rules of thumb or empirical knowledge obtained from experience. They may aid in the solution but are not guaranteed to succeed in the same way that an algorithm is a guaranteed solution to a problem (Giarratano and Riley 1989). The process of eliciting and modelling the knowledge and expertise from experts and other sources, and building an expert system is called **knowledge engineering**, and is carried out by a knowledge engineer (Michie 1973).

Expert systems are mainly designed for specific problem domains. A **problem domain** is a special problem area within which a particular kind of knowledge or expertise works for problem solving. Typical land use problem domains are land evaluation, land economics, environmental assessment and conservation management. The expert system makes inferences or reasons in a manner similar to humans to infer the solution of a problem within a specific problem domain for which it was designed.

An expert system consists of two main components: the knowledge base and the inference engine. It has two main functions: drawing conclusions and explaining reasoning. The user supplies facts or other information to the expert system and receives expert expertise in response. The knowledge base contains the knowledge with which the inference engine draws conclusions. A conclusion is the expert system's response to the user's query for expertise. It can be physical land suitability for a certain type of land use, or a diagnosis of pest problems in forest. It should be noted that the conclusion set is usually specified in advance. What the expert system can do is to find the appropriate element of that set according to the facts. Expert systems work in a consulting mode (Jackson 1990). During the consultation, the user interacts with the system if it requires information for the reasoning. Usually, the system poses the questions and the user provides the answers. The system is always

in control. The user can not interfere the reasoning process directly. Only the information entered during the consultation influences the reasoning. This is in contrast to a conventional SDSS or GIS system where the user controls the operations of the system. The expert system can also explicitly explain in detail the reasoning that led to a conclusion, so that it is understandable.

An expert system replicates and autonomously applies the expertise of the expert within a given domain to problem solving, which would allow the expertise to be provided steadily, unemotionally and completely at all times. Based on its knowledge, an expert system may act as an intelligent advisor for many issues in resources and environmental planning and management, such as land acquisition, nature conservation, forestry operations, urban fringe projects, adverse development control, etc. It can also bring expertise to locations where a human expert is unavailable or situations in which an expert's service would be very expensive. On the other hand, building an expert system involves collecting, combining, organising, codifying, and exploiting human knowledge. In turn, it is very helpful for experts to develop the knowledge itself and improve its quality. Expert systems can act as organised and accessible repositories of the decision-making knowledge accumulated by experts. They have the important function of disseminating knowledge, and giving advice and information on resource and environmental issues, and improving the problem-solving capabilities of non-expert humans.

## 2.2 Elements of Expert Systems

As discussed in Chapter 1, one of the basic feature of expert systems is the separation of knowledge from the problem solving algorithm. Knowledge is domain specific, while problem solving algorithms can be used for several different problem domains. A typical expert system consists of four components: the user interface, the knowledge base, the working memory and the inference engine (Figure 2.2).

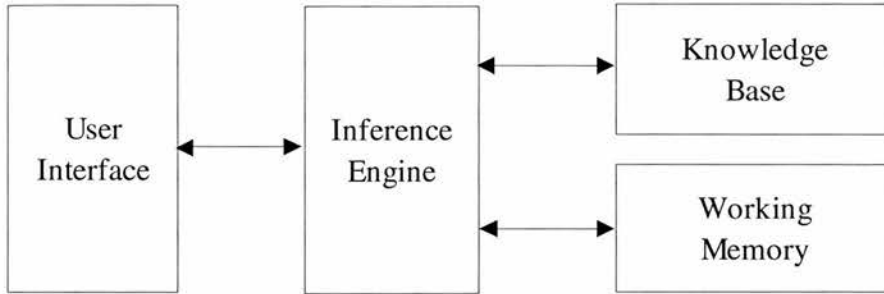


Figure 2.2 Structure of an expert system

The **knowledge base** contains the domain knowledge needed to solve problems. If the knowledge is coded in the form of rules, the system is called a **rule-based expert system**. The knowledge base in a rule-based expert system is also called **production memory**. The **working memory** is a global database containing existing data (also called facts) used by the knowledge or rules in the knowledge base. The inference engine makes inferences. The working memory is continually monitored and updated by executing knowledge in the knowledge base. Given a situation or a particular state of the working memory, the **inference engine** chooses knowledge that can be applied and decides where to look to find data when they are needed according to a control and reasoning strategy (Jackson 1990). The user interface provides for communication between the expert system and the user.

### 2.2.1 The knowledge base

Knowledge bases are collections of expertise or expert knowledge for reasoning in order to solve problems or make decisions. In a rule-based expert system, the knowledge is expressed as a collection of IF-THEN rules. The rules are activated by facts to produce new facts or conclusions. The following is a simple example.

```
Rule: crop-yield
IF
    Precipitation is light
    And soil is sandy loam
    And climate is hot
THEN
    Crop yield is good
```

Each rule is identified by a name. The section of the rule between IF and THEN is known as the **antecedent**, **conditional part** or **left-hand-side (LHS)**. The individual condition such as “Precipitation is light” is called a **pattern** or a **conditional element**. The part of the rule following THEN is called the **consequent** or **right-hand-side (RHS)**, which is a list of actions to be executed. When these actions are executed, we say that the rule **fires**. Rules like this are called the **production rules**. Each rule represents a piece of knowledge. Rules are independent of each other. Thus, it is easy to extend the scope of the expert system by adding further rules to the knowledge base. Rule-based techniques have proven to be very attractive for a variety of problems, which can be solved through the use of heuristic methods or rule of thumb (Giarratano and Riley 1989). While production rules are a popular paradigm for representing knowledge, other types of expert systems use different representations. These will be discussed in Section 2.3.

The knowledge in expert systems may be either expertise, or knowledge which is generally available from books, journals and experts.

### 2.2.2 The inference engine

The inference engine is the heart of an expert system. It is a computer program that guides the manipulation of knowledge in a knowledge base to produce solutions. The inferential reasoning mechanism is distinct from the knowledge base. Different types of expert systems have different inference mechanisms. Rule-based expert systems

can be broadly classified as **forward-chaining** and **backward-chaining** (Jackson 1990). The two kinds of inference mechanisms control how the inference engine determines when rules are needed, which rules to select, and how rules should be processed.

Forward-chaining reasons from facts to the conclusions which are based on the facts. The facts about a problem reside in the working memory, comprising the current state of knowledge about a problem. The inference engine starts with the facts, examines the LHS of rules and evaluates these antecedents by matching them against the facts in the working memory. If all conditions are true, then the RHS actions are executed. The inference proceeds to invoke the rules by working from the data towards a goal or a solution, continuing until the problem is solved, or no solution is possible if after an exhaustive search, no answer is found. Forward-chaining is also termed **data-driven**. It is applicable to most of the applications in spatial problem domains. This data-driven inference process corresponds to the way in which spatial data are typically used (Gardels 1987). A typical spatial analysis usually starts from the basic data to explore possible scenarios, and then draw a reasonable conclusion.

Backward-chaining takes the opposite approach. It reasons from goals (hypotheses or conclusions) to the facts that may justify the goals. Thus, backward-chaining is sometimes termed **goal-driven**. The backward-chaining system searches the consequents of all rules for occurrences of the goal. If at least one rule is found with the desired goal, the antecedents of this rule become the hypotheses and are recursively evaluated. It tests each rule in turn until an answer is found or until all possible rules have been examined and no answer exists. The backward-chaining systems commonly elicit evidence from the user to aid in proving or disproving hypotheses. This contrasts with the forward-chaining systems in which all the relevant facts are usually known in advance. Explanation is facilitated in backward-chaining because the system can easily explain exactly what goal it is trying to satisfy.

The inference engine works in cycles (Giarratano and Riley 1989). A cycle consists of three steps:

- (1) pattern matching
- (2) conflict resolution
- (3) action

As described above, a rule requires a match between the preconditions of the rule and the content of the working memory. This process of matching is called **pattern matching**. In forward-chaining reasoning, it is to match the conditions of the rule with the facts in the working memory, while in backward-chaining reasoning, it is to match the conclusions of the rule with the current goal in the working memory. When the preconditions of a rule match the current content of the working memory, the rule is said to be **applicable**. Multiple rules may be applicable during one circle. In this case, a selection strategy must be used to determine the rule to be fired first. The process of deciding which rule to fire first is called **conflict resolution**. The actions of the selected rule are executed (which may result in changes in the content of the working memory), and then the inference engine selects another rule and executes its actions. This process continues until no applicable rules remain.

The conflict resolution is directed by **conflict resolution strategies** or **search strategies**. There are a number of conflict resolution strategies developed in the field of artificial intelligence (AI). These include depth-first search, breadth-first search, hill-climbing, beam search, best-first search and so on (Winston 1992). Detailed discussions of these conflict resolution strategies can be found in Winston (1992) and other AI texts. An expert system tool may have several conflict resolution strategies available for the knowledge engineer to choose. Appendix A lists the conflict resolution strategies available in CLIPS, an expert system tool to be used in this research.



## 2.3 The Representation of Knowledge

Expert systems are designed based on a certain type of knowledge representation. The way in which an expert system represents knowledge affects the development, efficiency, speed and maintenance of the system (Giarratano and Riley 1989).

**Knowledge representation** in expert systems is a formalism that represents knowledge adequately and efficiently in a symbolic form, which can be interpreted and manipulated by computer programs. A number of different knowledge representation approaches have been developed in AI, including rules, semantic nets, frames, scripts, knowledge representation languages, conceptual graphs and so on (Giarratano and Riley 1989; Bar and Feigenbaum 1981; Brachman and Smith 1980; Winston 1992). Among them, rules and frames are most commonly used.

### 2.3.1 Production rules

Production rules are the most common way to represent knowledge in current expert systems. It is probably an axiom of AI that intelligent behaviour is rule-governed (Jackson 1990). It is often argued that a single rule corresponds well to a unit of human problem-solving knowledge. In other words, rules appear to be a natural way of modelling how humans solve problems. Production rules are usually used to encode empirical associations between patterns of data presented to the system and actions that the system should perform as a consequence.

In this scheme, a rule is the atomic form of knowledge representation canonically represented in the following form:

IF < antecedent conditions >  
THEN < conclusions >

Section 2.2.1 has given an example. If the conditions are logically evaluated to be true, then the conclusions can be said to be logically true. IF-THEN rules in an expert

system differ from similar rules in conventional computer systems, in that they can be modified much more easily to meet changing needs. This makes it easy to encapsulate knowledge and expand the expert system by incremental development. In addition, it is easy to build explanation facilities with rules since the antecedents of a rule specify exactly what is necessary to activate the rule.

Production rules are a good representation for linking conditions with actions. They are especially appropriate when the solution control procedures are not known in advance. They are also well-suited to representing problems that can be decomposed into relatively independent subproblems with no fixed order of solution processing. However, production rules are not good at representing knowledge about objects, events and arbitrary relationships between them (Jackson 1990).

### 2.3.2 Frames

**Frames** serve as a kind of template for holding related clusters of data, facts, rules, hypotheses, graphics, comments, questions for users, or any other knowledge in a single conceptual unit. They can sometimes come closer to mimicking the ways human beings reason and remember.

A frame is basically a group of **slots** and **fillers** that define a stereotypical object. It is analogous to a record structure in a high-level language such as PASCAL. The slots and slot fillers of a frame correspond to the fields and values of a record. An example of a frame for describing the characteristics of a land parcel is shown in Figure 2.3.

<i>Slots</i>	<i>Fillers</i>
No. of the land parcel	4
Slope	0.015
Soil texture	silty loam
Water capacity	72.5%
Biomass of vegetation	0.8 kg/m <sup>2</sup>

Figure 2.3 A frame describing the characteristics of a land parcel

Frames use slots to hold information about an item or object. This information may contain the object's attributes and values, procedures, pointers for getting data and information from other frames, and rules and questions involving information in the frame. Many slots of a frame can be pre-filled with values that remain somewhat stable within a class of situations being represented. Defaults are often used to represent commonsense knowledge. For example, a frame describing birds is likely to have a default value: flying, since we commonly think of birds as creatures that can fly. This allows the explicit representation of information that is assumed to be pertinent even when facts are not explicitly stated in the input. Frames can also be linked to other frames to form a frame hierarchy. For example, pointers and procedures can direct the system to search other frames for related data.

Frames are typical of approaches to structured object representations. The use of frames as data structures for storing expectations about typical objects and events has become widespread in AI applications (Winston 1992). Frames are mostly used in conjunction with other representations, such as production rules and semantic nets.

### 2.3.3 Other representations

Expert systems also use other forms of knowledge representations, including semantic nets, scripts, etc..

**Semantic nets** are a kind of network with nodes and links, in which the nodes represent physical objects, concepts or situations, and the links represent relationships between them (Jackson 1990). Descriptive labels attached to the nodes and links ascribe an informal, intuition-based semantics to the network. Relationships provide the basic structure for organising knowledge in the semantic nets. Certain types of relationships have proven very useful in a wide variety of knowledge representations. For example, IS-A and A-KIND-OF links are commonly used relationships. Figure 2.4 shows an example of a semantic net using the two links. A semantic network should present a clear, stable, and unambiguous picture of the represented application. The success of a semantic net is dependent on a well planned, highly structured definition (Giarratano and Riley 1989). Semantic nets are

most suitable for the applications which require a description framework based on a complicated taxonomy and for constructing inheritance hierarchies.

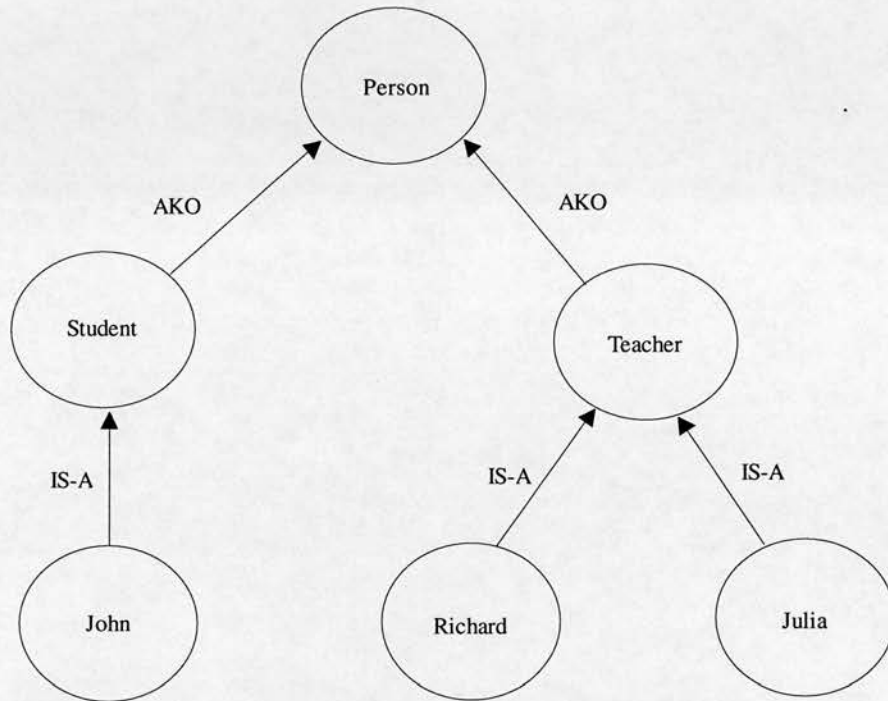


Figure 2.4 A semantic net

A **script** is an expression of a sequence of actions that are relevant to a situation description. It is essentially a time-ordered sequence of frames (Giarratano and Riley 1989). There are also several schemes for representing knowledge based on mathematical logic, such as propositional logic and predicate logic (Jackson 1990). We will not discuss them here.

## 2.4 Expert System Development Tools

An expert system performs problem solving within a domain using encoded knowledge and reasoning that have been programmed using a computer language. Over the years, languages have been developed for writing expert systems. A more recent development in software programming is not only to have a language, but also to have a package of utilities including editors, debuggers, window management, file management and so on. These languages and packages are generally referred to as “programming environments” or “development tools”. According to Giarratano and Riley (1989), there are primarily three categories of expert system development tools: languages, tools and shells.

### **Languages**

Languages used for building expert systems can be procedural languages, such as C, FORTRAN and LISP, and non-procedural specialised expert system languages, such as PROLOG. The main difference between expert system languages and procedural languages is the focus of representation. Procedural languages focus on providing techniques to represent data, while expert system languages focus on providing means to represent knowledge (Giarratano and Riley 1989). Expert system languages specially separate the data from the methods of manipulating the data. In a rule-based expert system, that is the separation of facts and rules. Expert system languages also provide an inference engine to execute the statements of the language. For example, PROLOG has its built-in backward-chaining (Hogrer 1984). LISP is the dominant computer language in AI, but it is not an expert system language. However, it is possible to write an expert system language using LISP. LISP also allows for easy representation of structured knowledge as well as simple facts (Jackson 1990). The specialised expert system languages are very suitable for writing expert systems but not for general purpose programming.

## **Tools**

A tool here refers to an expert system language plus associated utility programs. Some tools may allow the use of different inference mechanisms, such as forward-chaining and backward-chaining in one application. In some cases, a tool may be integrated with all its utility programs in one environment to present a common interface to the user (Giarratano and Riley 1989). There are now many tools available, such as ART (the Automated Reasoning Tool), KEE (the Knowledge Engineering Tool) and CLIPS (the C Language Integrated Production System). CLIPS is a powerful expert system tool, which provides forward-chaining, a rule-based programming language and an object-oriented programming language. We will discuss it in detail in Chapter 7.

## **Shells**

Expert system shells are highly specialised tools for building expert systems in certain domains. EMYCIN shell is an example (Van Melle 1979). This shell is a domain-independent version of the medical diagnostic system MYCIN that uses backward chaining. By simply removing the medical knowledge, EMYCIN was made, which could be used for other diagnostic problems where they are suitable for employing backward chaining. One of the major advantages of an expert system shell is the speed at which an expert system can be built. Utilising all the general functions of a shell, the developer only has to provide the domain knowledge.

Moving from languages and tools to shells implies sacrificing generality and flexibility of programming capabilities. On the other hand, development time and demand on programming skills are reduced. Time and cost-efficient development must be traded off against generality and flexibility.

## 2.5 Summary

Expert systems technology provides a flexible means of representing and processing specific control and domain knowledge. We have introduced some key concepts of expert systems, discussed the basic structure of expert systems and reviewed several knowledge representation schemes, such as production rules, frames and semantic nets. The commonly used development tools for expert systems have also been examined. The rule-based expert system is especially appealing in that expert knowledge and expertise can be represented in a flexible declarative and modular form (Jackson 1990). The reasoning and explanation capabilities offered by rule-based expert systems are also very effective. Moreover, a rule-based approach allows the system to be developed incrementally. This research focuses on the development of an improved tool for spatial decision support using rule-based expert systems techniques, coupled with spatial modelling approaches and the spatial data handling capabilities of GIS systems.

## Chapter 3

# GIS and Spatial Modelling

While the technology of expert systems provides the KBSDSS system with capabilities for simulating reasoning and explaining the reasoning and conclusions, GIS systems serve the needs of the KBSDSS system for spatial data management and analysis. GIS systems are computer systems that can capture, store, manipulate, analyse and display spatial data. Here, **spatial data** are data that are spatially referenced to the earth. They are also called **geographical data**. What distinguishes GIS systems from other information systems is the ability to analyse objects and phenomena where spatial or geographical location is an important characteristic or critical to the analysis. Analysis functions of GIS take into account the spatial positions of spatial objects, their spatial relationships as well as their attribute characteristics. We define a **GIS model** as a model that is built with one or more basic GIS analysis functions to perform a certain type of spatial modelling task. This chapter first briefly reviews basic terminology related to GIS and spatial modelling, then proposes a taxonomy of GIS models, and finally, reviews the principles of cartographic modelling within GIS systems.



### 3.1 Spatial Data Models and Data Structures

Spatial data or geographical data have traditionally been presented for analysis by means of maps. In GIS systems, maps are stored in digital form. The digital map data in GIS systems may be viewed at four abstraction levels (Peuquet 1984): reality (the real world), data model (or conceptual model), data structure (or logical model) and file structure (or physical model) (see Figure 3.1). A **data model** is an abstraction of the real world which incorporates only those properties thought to be relevant to the application or applications at hand, usually a human conceptualisation of reality. A **data structure** is a representation of the data model expressed in terms of diagrams, lists and arrays designed to reflect the recording of the data in computer code. A **file structure** reflects the physical storage of the data on some specific computer storage medium.

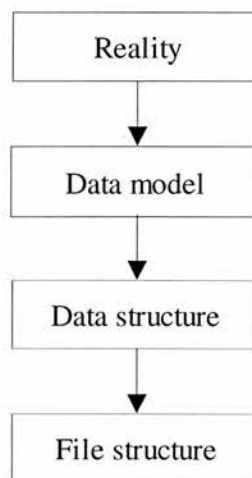
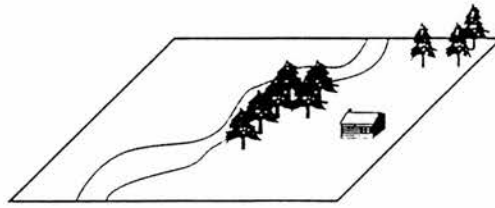


Figure 3.1 Levels of spatial data abstraction (after Peuquet 1984)

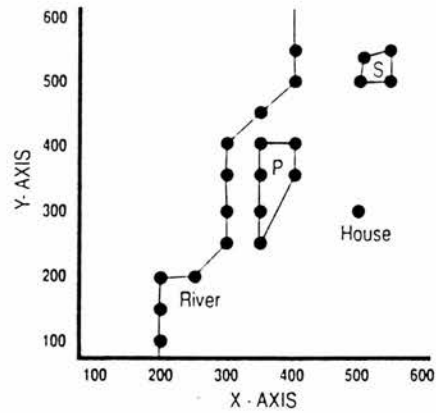
**Spatial data models** represent reality with a series of geographical features (defined by their locations and associated attributes). There are two fundamental spatial data models: **vector** and **tessellation** data models (Figure 3.2).



(a) Real world

	1	2	3	4	5	6	7	8	9	10
1							R			S
2							R		S	S
3						R				
4				R	P	P				
5				R	P	P				
6				R	P			H		
7				R	P					
8		R	R							
9		R								
10		R								

(b) Tessellation



(c) Vector

Figure 3.2 Spatial data models (reproduced from Aronoff 1989).

P - pine forest stand, S - spruce forest stand, R - river

H - house

In the vector data model, geographical features are represented by points, lines and areas, much as if they were being drawn on a map. A point may represent a well, a line may represent a road, and an area may represent a forest stand. A point feature is recorded as a single x, y coordinate pair, a line as a series of x, y coordinates, and an area as a closed loop of coordinate pairs that define the boundary of the area. Indeed, an area is bounded by a closed loop of straight-line segments. So an area is termed a **polygon** in the vector data model. Additional attributes, or descriptive information about any given geographical feature is stored with its x, y coordinate data.

In the tessellation data model, geographical features are described as polygonal units of space in a mesh. The polygonal units may be regular or irregular. There are three types of regular tessellations which have been used as the basis of spatial data models. They are square, triangular and hexagonal meshes (Figure 3.3). The regular square mesh is the most widely used and best known as the **raster** data model. The raster data model divides the space into equal sized grid cells and may be implemented by the use of several data structures such as **quadtrees** and **run-length** encoding (Burrough 1986; Samet 1989). The irregular tessellation also has three commonly used types in spatial data applications. These three are square, triangular and Thiessen polygon meshes. Of these, the **triangulated irregular network (TIN)** is most frequently used as a spatial data model. The TIN is derived from irregularly spaced sample points and breakline features. Each sample point has an x, y coordinate pair and a surface or z-value. These points are connected by edges to form a set of non-overlapping triangles used to represent the surface (Figure 3.4). It is a standard data model for representing terrain data for digital terrain analysis and hydrological applications as well as for representing other continuous data.

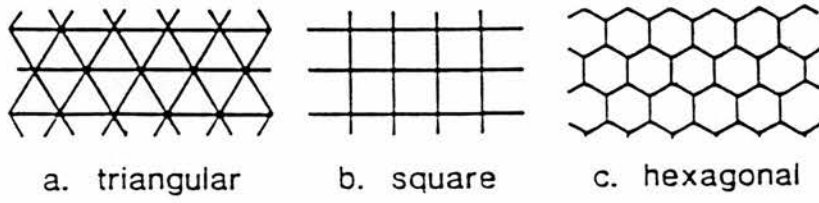


Figure 3.3 The regular tessellations (reproduced from Peuquet 1991)

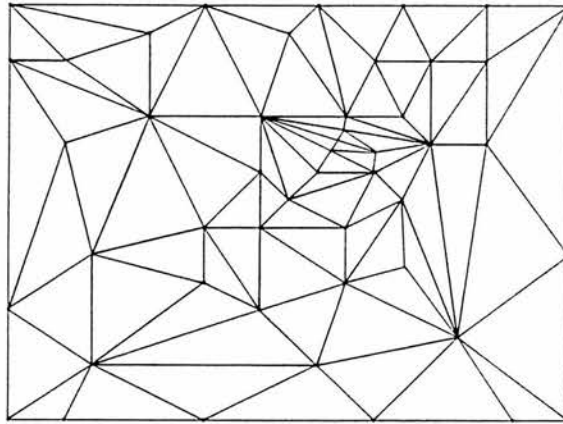


Figure 3.4 A triangulated irregular network (reproduced from Burrough 1986)

## 3.2 Geographical Features and Data Types

Following Maguire and Dangermond (1991), we use the term **geographical features** to refer to the geographical phenomenon which may be defined as a set of spatial locations together with a set of attributes characterising those locations. For example, houses, a river system, a road network and woodlands are all geographical features. In GIS systems, they are represented as either point, line, area or surface features (Unwin 1981; Maguire and Dangermond 1991). Geographical features are also called **spatial entities**.

**Points** are geographical features which are associated with a single location and have no length dimension. **Lines** are geographical features which have a single length dimension. Their location is described by a string of spatial coordinates. **Areas** have two length dimensions, where their locations are represented by a closed string of spatial coordinates. **Surfaces** have three length dimensions. Commonly, their representation on a map is as a set of contours, such as terrain or temperature surfaces.

The representation of the four types of geographical features on maps is depicted in Figure 3.5. In spatial databases, geographical features are represented as collections of locational and attribute data in either the vector or tessellation data models.

**Locational data** describe locations of geographical features. They provide a reference for the attribute data of the geographical features. **Attribute data** describe what is at a point, along a line or within an area. They provide the descriptors necessary to understand the type of geographical features being depicted by the locational data, such as “house” for a point, “river” for a line or “grassland” for an area. Attribute data are used to identify geographical features. They can be classified according to the **level of measurement** (Unwin 1981). The levels of measurement include **nominal**, **ordinal**, **interval** and **ratio** (Table 3.1). Nominal data are classified into categories. Each value is simply a distinct category. Ordinal data are ranked in ascending or descending order. Interval data have the property that distances between

categories are defined as fixed equal size units. Ratio data have in addition an absolute zero.

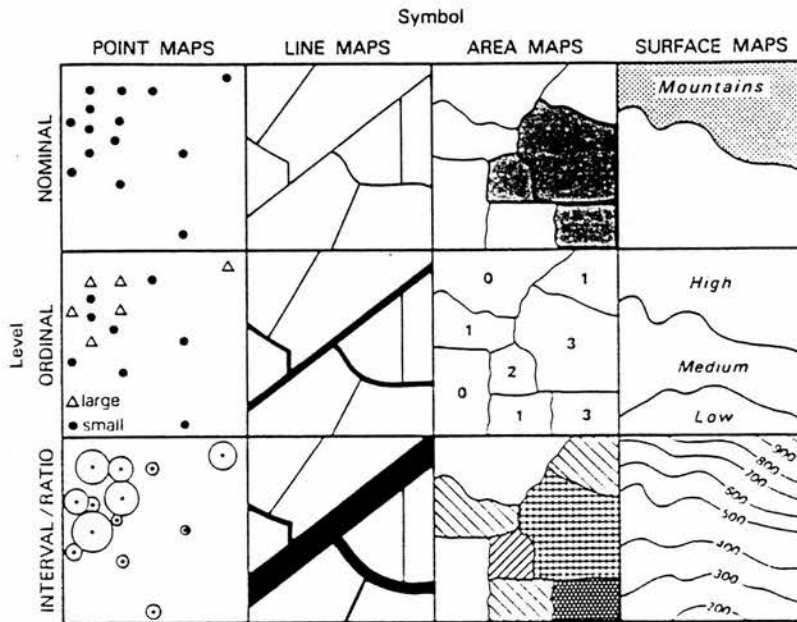


Figure 3.5 Types of geographical features (reproduced from Unwin 1981)

Table 3.1 Levels of measurement for attribute data of geographical features (after Unwin 1981)

<i>Level</i>	<i>Basic operations</i>	<i>Examples</i>
Nominal	Determination of equality of class; counting items	Land use categories
Ordinal	Determination of greater, less, or equality; counting items in class	Land capability classification, city ranking
Interval	Determination of equality or difference of interval; addition, subtraction	Temperature in degree C or degree F
Ratio	Determination of equality or difference of ratio; addition, subtraction, division	Distance, mass, precipitation

Spatial data are organised in spatial databases as a set of themes, such as road network, land cover, soil types, elevation, etc. They are often thought of as **data layers** or **coverages**. A data layer corresponds to a map sheet, which consists of a set of logically related geographical features and their attributes. To facilitate analyses, the data are commonly organised by the type of geographical features they represent. One data layer represents one single theme or factor, or one class of geographical features.

### 3.3 Taxonomy of GIS Models

There is a large range of analysis functions for spatial data analysis and modelling that are available in a GIS system (Burrough 1992). The GIS analysis functions have been documented and classified by many notable authors. Dangermond (1983) described a classification of software components in GIS systems. He grouped GIS analysis functions into ten categories: data retrieval, map generalisation, map abstraction, map sheet manipulation, buffer generation, polygon overlay and dissolve, measurement, grid cell analysis, digital terrain analysis and output techniques.

Burrough (1986) presented a hierarchical classification scheme which groups GIS analysis functions into three categories at the highest abstraction level: operations on the topology or spatial aspects of the geographical data, on the non-spatial attributes of these data, and on non-spatial and spatial attributes combined. Each group is further subdivided into types of functions. For example, operations on non-spatial and spatial attributes combined are divided into four groups: retrieval, overlay and intersection, region analysis and neighbourhood analysis. Aronoff (1989) proposed a similar classification scheme.

Berry (1987a) proposed a three-level classification scheme according to how the computer obtains values for processing. At the first level, the most abstract level, three types of analysis functions are identified based on map components. They are point processors, regional processors and neighbourhood processors. Point

processors consider each geographical features independently. When the analysis involves several data layers, the process can be conceptualised as “vertically spearing” a series of features from a stack of spatially-registered data layers. Regional processors associate each location with a set of other locations having a similar characteristics and can be conceptualised as forming “cookie cutter” templates or regions which are superimposed on other data layers. The features within each of the regions are identified for processing. Neighbourhood processors identify features for processing in terms of their spatial proximity specified by distance and direction. At the second level, four classes of operations are identified based on the processing transformation: reclassifying maps, overlaying maps, measuring distance and connectivity, and characterising neighbourhoods. Each of the four classes of operations can be further subdivided into different functions, which form the third level of the classification.

In a similar way to that of Berry, Burrough (1992) proposed another classification scheme for GIS analysis functions. In this scheme, three groups of GIS analysis functions are identified: (a) functions by which new attributes of geographical features and their values are derived from existing attributes without spatial contiguity considerations; (b) functions by which new attributes and their values at a certain location are derived from existing attributes within the area surrounding that location. Spatial contiguity is taken into account; (c) functions by which new attributes are derived from existing attributes that vary with time. The three groups are subdivided into nine classes of operations, according to exact or non-exact valued non-spatial attributes, feature types (discrete spatial entities or continuously varying surfaces) and other information. Each class is further decomposed into different basic analysis functions.

Each basic GIS analysis function may be seen as a GIS model. Based on the previous work, we propose a classification scheme of GIS models for use throughout the thesis and for development of GIS models in our prototype KBSDSS system to be described in Chapter 8. This scheme classifies GIS models with emphasis on the functional aspects of what the GIS model is capable of doing and not on the procedural aspects of how it is manipulated to solve a problem. However, as pointed



out by Maguire and Dangermond (1991), it is impossible to develop a completely comprehensive classification scheme of GIS models. That is mainly because (a) GIS is a relatively immature and rapidly evolving discipline. The number of GIS analysis functions is constantly growing; (b) GIS applications are very diverse. The problem descriptions involve different terminology. Therefore, we only deal with the most important and often used GIS models.

Our classification scheme organises GIS models in a hierarchy. Figure 3.6 illustrates this taxonomy. At the highest level, four categories of GIS models are identified: location-specific analysis models, regional summary models, neighbourhood analysis models, and geometric measurement models. They can be subdivided into a set of basic GIS models (in *italics*). Together with analytical and rule-based models, these GIS models can be stored in and managed by a KBSDSS system for solving particular spatial problems.

### **Location-specific analysis models**

Location-specific analysis models are used to characterise individual geographical features without consideration of spatial contiguity. They deal with geographical features independently at individual locations (points, lines, areas or grid cells). Location-specific analysis models can be divided into two types: *reclassify* and map overlay.

The *reclassify* model merely reclassifies or changes existing values on a single data layer. It assigns new values to the categories of an existing data layer. These new values may be a function of the initial value, its position or other characteristics of the individual categories. The *reclassify* model can be performed using a **reclassification table**, which defines the parameters for reclassification. Usually, a reclassification table is made up of input values and their associated reclassified values. The general form of description of the relationships between an input value and its output reclassified values in a reclassification table is

input value -> output reclassified value

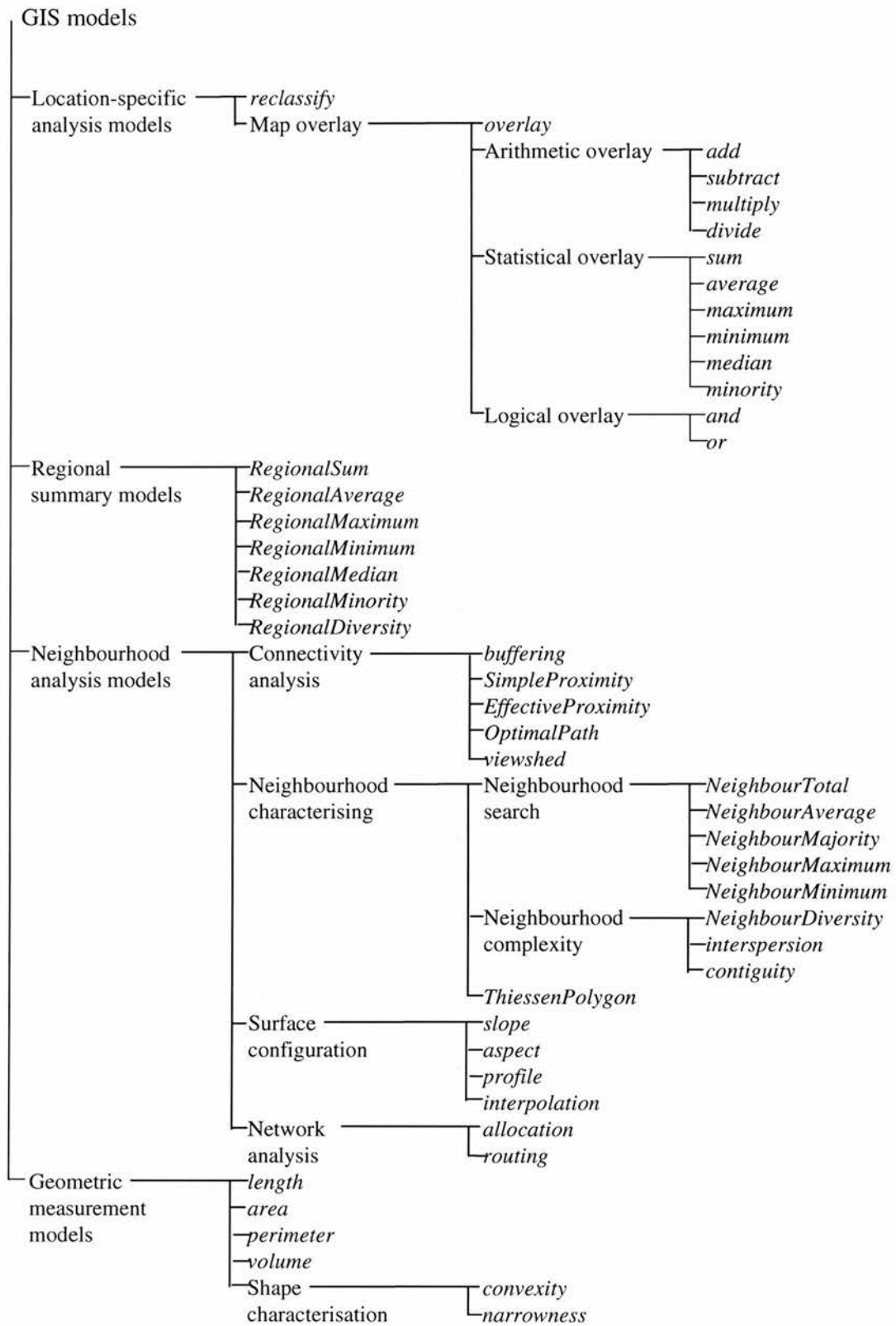


Figure 3.6 Taxonomy of GIS models (after Berry 1987a, Aronoff 1989 and Burrough 1992)

An example is given in Chapter 8. When operating, *reclassify* matches the value of a category on the existing data layer with the input values in the reclassification table, and changes the original value into the reclassified value associated with it. Thus, a new data layer containing the reclassified value is created. It can be thought of as the purposeful “recolouring” of maps, and results in no delineation of new boundaries (Berry 1987a).

Map overlay models combine multiple data layers on a location-by-location basis, so that a unique output value is assigned to each unique combination of input values. They involve the creation of a new data layer on which the value assigned to every location is a function of the independent values associated with that location on two or more existing data layers. Such a function may be a simple logical (Boolean) operation, a simple arithmetical operation (addition, subtraction, multiplication, division, logarithms, exponents and all combinations), simple if-then rules, a weighted logical or arithmetical operation, a statistical model, a probabilistic model or other complex mathematical models and rule-based models (rule sets). Map overlay models can be divided into at least four submodels: *overlay*, arithmetic overlay models, statistical overlay models, and logical overlay models.

The *overlay* model is to combine a set of existing data layers, each representing the geographical distribution of a single attribute, to create a new data layer, identifying the particular attribute combination at each location in the study area. A specific function, such as those described above, is then used to compute values on the new data layer location by location from those attributes being combined.

Arithmetic overlay models may include *add*, *subtract*, *multiply* and *divide*. The *add* model is used to add integer and float values on two number-valued data layers location by location to create a new data layer. It may also be used to add a constant to an existing data layer. The *multiply* model is used to multiply values on two number-valued data layers or multiply a constant with an existing data layer on a location-by-location basis.

The *subtract* model is used to subtract values on one number-valued data layer from values on the other location by location. It may also be used to subtract a constant from values on an existing data layer or vice versa. Similarly, *divide*

performs division operations on two number-valued data layers, or one constant and one data layer on a location-by-location basis.

Statistical overlay models calculate the new values from two or more input data layers location by location according to the statistical functions being used. They may include *sum*, *average*, *maximum*, *minimum*, *median* and *minority*. These models respectively calculate the sum, average, maximum, minimum, median and minority of the values on two or more input data layers on a location-by-location basis. Here, the minority value is the value that appears least often at a certain location.

Logical overlay models perform Boolean operations on two input data layers on a location-by-location basis. The *and* and *or* models are most commonly used logical overlay models. The *and* model performs a Boolean-AND operation on two input data layers, and the *or* model performs a Boolean-OR operation.

Map overlay modelling results in delineation of new boundaries.

### **Regional summary models**

Regional summary models are used to characterise geographical features within homogeneous regions, such as land parcels with certain land cover types and regions defined by county or farm boundaries. Rather than combining information on a location-by-location basis, this group of models statistically summarise the spatial coincidence of entire categories of two or more data layers within each region. The regions on a data layer are also called **templates**. Regional summary models perform statistical summaries on geographical features within the boundaries of all templates. According to the types of statistics, Regional summary models may include the following: *RegionalSum*, *RegionalAverage*, *RegionalMaximum*, *RegionalMinimum*, *RegionalMedian*, *RegionalMinority*, and *RegionalDiversity*. Among the inputs to the models, a template data layer should be specified. These models are respectively used to calculate the sum, average, maximum, minimum, median and diversity values of geographical features within each region on a template data layer. The diversity value here refers to the number of unique values of geographical features within a region.

## Neighbourhood analysis models

Neighbourhood analysis models are used to characterise geographical features within their neighbourhoods. A neighbourhood of a location is defined as a set of locations that bear a specified distance and/or directional relationship to it. Neighbourhood analysis models compute new values as a function of the existing values of geographical features within their neighbourhoods. There are four groups of neighbourhood analysis models: connectivity analysis models, neighbourhood characterising models, surface configuration models, and network analysis models.

### (1) Connectivity analysis models

Connectivity analysis models analyse how locations are connected in space. Target locations (e.g. a road, a school) and a distance definition (e.g. distance in meters, travel time in minutes) should be specified for each connectivity analysis model. Connectivity analysis models mainly include: *buffering*, *SimpleProximity*, *EffectiveProximity*, *OptimalPath*, and *viewshed*.

The *buffering* model identifies all locations within a specified distance of a geographical feature. It forms a zone, also called a **buffer**, around a point, line or an area feature by placing its boundaries at a certain distance or other quantitative measurement from the feature.

The *SimpleProximity* model identifies the shortest straight-line distance from a target location to all other locations in a study area. This is similar to a series of buffers of equal steps emanating from a target feature.

The *EffectiveProximity* model identifies the shortest, but not necessarily straight-line, distance from a target location to all other locations in a study area. Distance is measured as a function of absolute and relative barriers affecting movement in geographical space. It is used to evaluate geographical phenomena that accumulate with distance, such as cost, transportation time, etc. Its operations can be thought of as moving step-by-step outward in all directions from one or more starting

locations and calculating a variable, such as travel time, at each successive step. The output of the model is an **accumulation surface** or **friction surface**, which describes the accumulated value (or running total) at each location from the starting location.

The *OptimalPath* model is used to determine the best route from one location to another over a continuous surface. In the case of a travel-time surface, it determines the quickest path. In the case of a terrain surface, it determines water flow.

The *viewshed* model identifies all locations that are visually connected to target locations. It uses digital elevation data to define the surrounding topography and other data layers describing surface feature locations and heights of obstructions.

## (2) Neighbourhood characterising models

Neighbourhood characterising models are used to evaluate the characteristics of the area surrounding target locations, such as calculating the diversity of vegetation types within 400 meters radius of each target location. There are three types of neighbourhood characterising models: neighbourhood search models, neighbourhood complexity models, and the *ThiessenPolygon* model.

Neighbourhood search models statistically summarise the values within a specified vicinity around each target location, such as finding all residential areas that are in the vicinity of a hospital. According to the statistics being used, they may be classified as the following: *NeighbourTotal*, *NeighbourAverage*, *NeighbourMajority*, *NeighbourMaximum*, and *NeighbourMinimum*. The definition of the neighbourhood and the target locations should be specified for each of the models.

Neighbourhood complexity models include: *NeighbourDiversity*, *interspersion* and *contiguity*. *NeighbourDiversity* calculates the diversity of the values in the neighbourhood, i.e. the number of different classes of geographical features within the neighbourhood. The *interspersion* model calculates the frequency of occurrence of each class in the neighbourhood. The *contiguity* model identifies the classes of adjacent areas.

*ThiessenPolygon* is used to construct **Thiessen polygons** from a set of points. A fundamental feature of Thiessen polygons is that each location within a polygon is

closer to the data point around which the polygon is formed than it is to any other data point.

### (3) Surface configuration models

Surface configuration models are used to characterise the form of continuous surfaces. Any characteristic that has a continuously changing value over an area can be represented as a surface, such as terrain, temperature, noise level, precipitation or concentration of chemicals. The data for surface modelling are generally discrete points either in the vertices of a regular mesh or in irregularly distributed points or in a TIN. Surface configuration models include *slope*, *aspect*, *profile* and *interpolation*.

### (4) Network analysis models

Network analysis models are used to model the flow of resources through the network. Here, **networks** refer to connected linear features, such as street networks, water pipe networks, and power line networks. They provide the capability to follow lines along a network and then process attribute data associated with those lines. There are two main types of network analysis model: *allocation* and *routing*. The *allocation* model is used to divide an area into service districts to optimise the allocation of resources. The *routing* model is used to optimise vehicle routing or find a shortest path along the network in terms of time, cost distance, etc.

## **Geometric measurement models**

Geometric measurement models are used to measure the geometric properties of geographical features. These include *length*, *area*, *perimeter*, *volume* and shape characterisation models.

Shape characterisation models include *convexity* and *narrowness* measurements for an areal feature. The *convexity* model is a measure of the boundary regularity of an areal feature based on the ratio of its perimeter to its area. The *narrowness* model

determines the shortest chord connecting opposite edges of an areal feature, such as a forest stand.

### 3.4 Spatial Modelling in GIS systems

A GIS model operates on both spatial entities or geographical features and their attributes. Typical parameters of GIS models are variables represented in the form of single-factor maps. The basic GIS models described above can be used both singly and in any possible set of combinations. To solve a spatial problem, we may decompose the problem into a set of subproblems so that each subproblem can be solved using a basic GIS model. After all subproblems have been solved, their solutions can be synthesised and the final solution is derived. In other words, by decomposing a complex spatial problem, the fundamental GIS models might be combined to perform complex analyses that provide the solution. This approach is called **cartographic modelling** (Berry 1987a; Tomlin 1991)

Figure 3.7 illustrates a cartographic model of land use potential for afforestation using some basic GIS models. This model is used to identify the potential sites for afforestation according to the proximity to roads and physical land suitability for forestry. The proximity to roads is calculated using the GIS model *buffering* according to roads and a given distance to roads. Assume that the value “1” is to be assigned to all locations within the buffer zones around the roads and the value “0” is to be assigned those outside the buffer zones on the derived data layer “Proximity to Roads”. The land unsuitable for agriculture is derived from the land capability for agriculture through the GIS model *reclassify*, which assigns “1” to the land unsuitable for agriculture and “0” to the land suitable for agriculture on the derived data layer “Land Unsuitable for Agriculture” (LUA). Similarly, the land suitable for forestry is derived from the land capability for forestry through the GIS model *reclassify*, which assigns “1” to the land suitable for forestry and “0” to the land unsuitable for forestry on the derived data layer “Land Suitable for Forestry”



(LSF). The land physically suitable for forestry but unsuitable for agriculture is derived through the GIS model *multiply* from the derived data layers LUA and LSF. Finally, the land potential for afforestation is derived from the data layers “Proximity to Roads” and “Land Physically Suitable for Forestry but Unsuitable for Agriculture” using the GIS model *multiply*. In this cartographic model, nodes represent encoded and derived maps, and arrows represent GIS models. The structure of the model indicates the logical sequencing of modelling operations on the map data that progresses to the desired final map.

Cartographic modelling is a major approach to spatial modelling in GIS systems. Spatial or cartographic models can be constructed by logically sequencing basic GIS models on spatial data to solve specific spatial problems. However, cartographic modelling has some limitations as a basis for spatial decision support. One limitation is that it is inaccurate in analysis because sharp boundaries are imposed between the attributes, such as soils and land cover types. In reality, there is not often a crisp boundary between the attributes (Wang et al. 1990). Cartographic modelling also implicitly assumes that all information encoded on map data layers is correct and contains no error components, and there is no propagation of errors from one stage in the modelling to the next (Burrough 1986). Furthermore, many GIS models are commutative. The user has to be especially careful about the order in which models are carried out. Some other limitations of cartographic modelling will be discussed in the next chapter. KBSDSS systems may incorporate the principles of cartographic modelling, but can avoid some of its limitations, as will be seen from the following chapters.

### 3.5 Summary

GIS systems provide a wide range of functions for spatial data analysis. One or more basic GIS analysis functions can form a GIS model to perform a certain type of spatial modelling tasks. We have given a taxonomy of GIS models for use in this



thesis. This taxonomy can also be used to guide the development of GIS models for KBSDSS systems. GIS models can be broadly classified into four categories: location-specific analysis, regional summary, neighbourhood analysis and geometric measurement. By systematically organising GIS models, complex spatial analyses can be carried out. In the development of the KBSDSS system, knowledge about GIS models will be formalised and made available in a knowledge base.

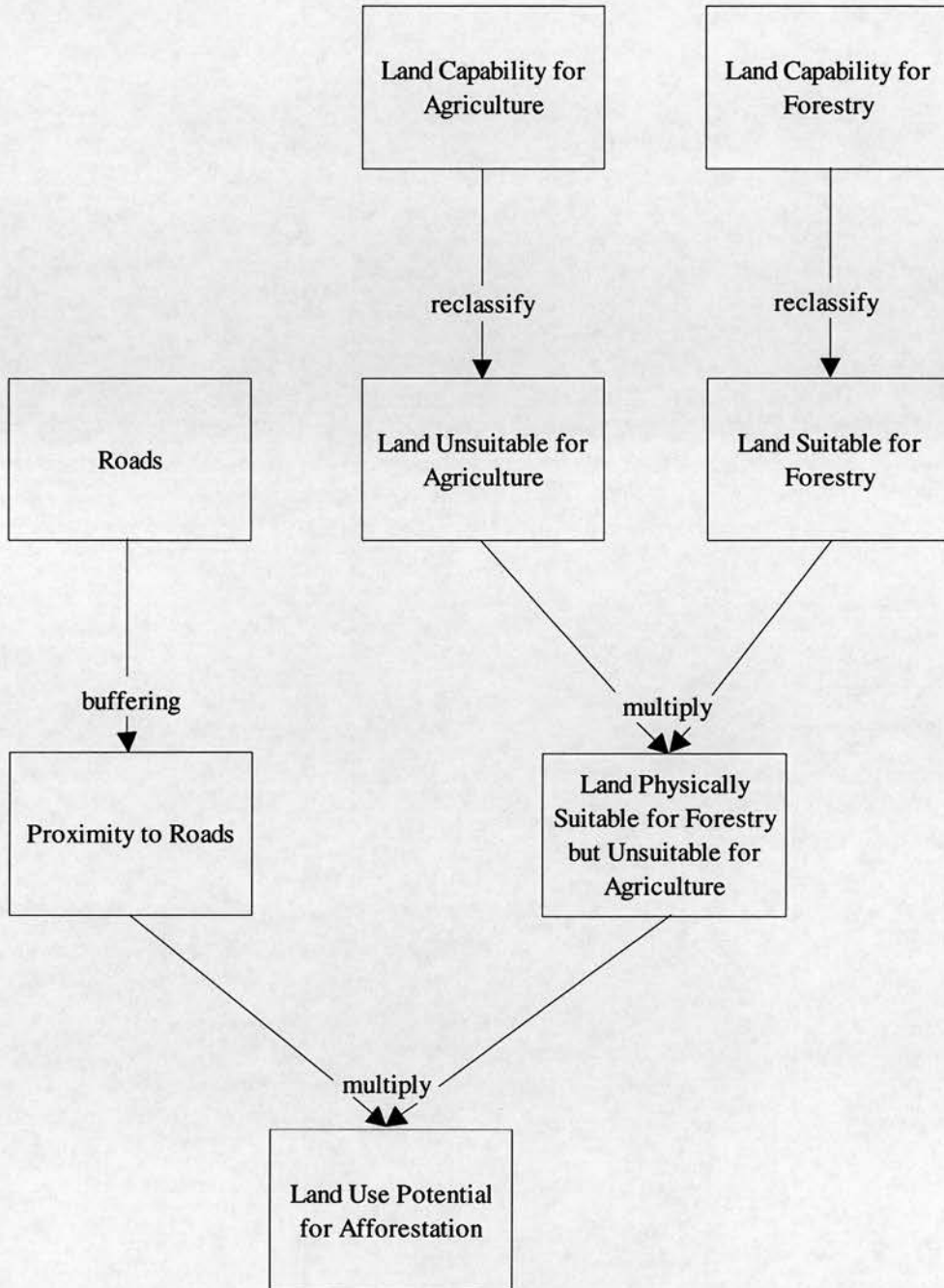


Figure 3.7 Cartographic modelling. A cartographic model for identifying land use potential for afforestation

# Chapter 4

## Representation and Structuring of Resource and Environmental Problems

Resource and environmental problems in a spatial context are often complex and characterised by a large number of interrelated uncertain quantities and alternatives (Franklin 1979). The problem underlying a decision is often articulated by the decision maker in an informal manner with too many components interacting in unspecified ways. As we discussed in Chapter 1, one of the valuable features of a KBSDSS system is to provide machine assistance in the structuring and simplification of a spatial problem and to present the problem parameters or elements in a comprehensible way. Thus, an important issue in the development of a KBSDSS system is to find a means of representing a spatial problem clearly and concisely so as to facilitate machine representation and manipulation of a spatial problem model.

Problem analysis and structuring methods have been studied extensively (Smith 1988). Existing approaches range from purely cognitive descriptions to a mixture of formal and informal descriptions. In this chapter, we examine the basic problem structuring methodologies, and develop a spatial influence diagram-based representation scheme to represent and structure resource and environmental problems.

## 4.1 Basic Problem Structuring Methodologies

Problem structuring can be described as the process of arriving at a sufficient understanding of the components of a particular problem to allow specific research actions (Woolley and Pidd 1981). There are two problem structuring techniques which have been widely used in problem solving: decomposition and modelling.

The decompositional approach to problem structuring (Simon 1973; Smith 1989) suggests that a complex problem can be decomposed into several subproblems, which are less complicated and can be solved separately (“divide and conquer”). After the results of each subproblem are obtained, they are integrated into a solution. Given a way of decomposing problems into finite sets of simple subproblems, decompositional structuring facilitates identification of a successful solution.

The modelling approach sees problem structuring as a part of or a prelude to modelling (Woolley and Pidd 1981). Modelling entails development of a formal representation of the problem, followed by manipulation and solving of this representation and application of the results to the original problem. Problem structuring is essentially a process of identifying a collection of elements from which to build a problem model. The modelling then consists of defining the relationships between the elements. Defining the problem elements and their inter-relations yields the components of a model. The modelling approach also implies a solution strategy.

Graphs are usually employed in problem model-based representation and structuring, as the graphical representation of a problem model is intuitive and can be easily understood by decision makers. Also, an effective graphical representation can facilitate the storage and manipulation of the problem model by a computer.

For example, Markov networks and Bayesian networks are graphical representations of the dependencies embedded in probabilistic models (Pearl 1988). A Markov network is an undirected graph whose links represent symmetrical probabilistic dependencies, while a Bayesian network is a directed acyclic graph whose arrows represent causal influences.

Decision trees and influence diagrams are two well-known graphical representations used in decision analysis. A decision tree is a representation of all scenarios that can possibly result from a given decision. It makes each controllable and uncertain variable explicit and directly gives the value of each possible outcome (Pearl 1988). The root of the tree represents the initial situation, while each path from the root corresponds to one possible scenario. A decision tree can facilitate a search for an optimal option or plan.

**Influence diagrams** are representations of uncertain variables and decisions that explicitly reveal probabilistic dependence and the flow of information (Howard and Matheson 1983; Olmsted 1983; Shachter 1986). They are directed acyclic graphs with three types of nodes and two types of arcs. The three types of nodes are decision, chance and value nodes. **Decision nodes** represent the choices or alternatives available to the decision maker. **Chance nodes** represent uncertain or probabilistic variables. The **value node** (in most cases, there is at most one value node) represents the objective (or utility) to be maximised. The arcs in the influence diagram link the different kinds of nodes. Arcs directed toward the value node and chance nodes are conditional and denote probabilistic dependence, i.e. dependence may be present. The most natural source of dependence is a cause-effect relationship. However, conditional arcs need not represent causality. Arcs pointing to decision nodes are informational and indicate available information. Informational arcs imply time precedence, i.e. the information must be available at the time the decision is made. The idea of influence diagrams was originally the result of a need to communicate with computers about the structure of decision problems (Olmsted 1983). In recent years, they have become an effective tool for developing models and communicating among people (Shachter 1986).

However, influence diagrams can be applied not only for decision analysis, but for any formal description of relationship and thus for all modelling work (Howard and Matheson 1983). They can also be used to represent models in which there are no decisions, i.e. there are no decision nodes (Shachter 1986). In such a case, the influence diagram represents the relationships among the variables, rather than the relationship or dependence of the decisions (or alternatives) and uncertain variables

in a decision problem. In this research, we extend the influence diagram idea to develop a spatial influence diagram-based representation scheme for representation, structuring, formulation and solution of resource and environmental problems. We focus on the representation of the relationships among resource and environmental (RE) variables without concern for decisions, and computation of the expected value in a resource and environmental problem. Thus, spatial influence diagrams can be seen as spatial analogues of influence diagrams, but they are deterministic cases of influence diagrams without decision nodes, which take into account the spatial characteristics of problem variables and their relationships.

## 4.2 Formal Definitions of Spatial Influence Diagrams

In a similar manner to influence diagrams, spatial influence diagrams are both a formal description of the spatial problem that can be stored and manipulated by a computer and an intuitive framework in which to formulate the problem as perceived by decision makers. A spatial influence diagram is a graph consisting of nodes and directed arcs with no cycles. The nodes in the diagram represent the RE variables or factors relevant to a particular spatial problem. There are three types of nodes: chance, value and border nodes, without decision nodes. The directed arcs between nodes represent influences, dependencies or relevance among the RE variables. There is no informational arc in a spatial influence diagram. The spatial influence diagram brings problem considerations together and shows how they are related. A spatial influence diagram is a resource and environmental problem model, or, in short, a spatial problem model. Decision makers can view a graphical display of the model, and understand the overall structure and nature of the dependencies among variables depicted in the graph. The diagram can also be manipulated using specified rules to evaluate the spatial problem (see Chapter 5).

Following some notation of Shachter (1986), we give the formal definitions of spatial influence diagrams below. Here, we borrow the terms “value node” and

“chance node” from influence diagrams, but give them different meanings to define our own spatial influence diagrams.

**Definition 1. Spatial Influence Diagram.** A spatial influence diagram is a simply connected, acyclic and directed graph  $G = (N, A)$ , with a node set  $N$  and an arc set  $A$ .

The set  $N$  contains three types of nodes, partitioned into three subsets  $V$ ,  $C$  and  $B$ . There is at most one value node  $v \in V$ , representing the goal or expected value in solving the spatial problem. For example, a land use planning problem is to predict the potential sites for afforestation. Here, the “Potential Sites for Afforestation” is the goal of the land use planning problem and can be selected as the value node for the problem model. There are zero or more chance nodes in the set  $C$ , representing RE variables, which influence (directly or indirectly) the value node and enable the computation of an outcome of the value node. The border nodes in the set  $B$  represent RE variables that correspond to available or acquirable data. That is, the variables denoted as border nodes already have values, or their values can be acquired from the decision makers.

Figure 4.1 shows a spatial influence diagram constructed to structure a land use problem for predicting the land use potential for development in a certain area.

To differentiate the three types of nodes visually, the value node is drawn as a rounded rectangle, chance nodes are drawn as circles and border nodes are drawn as ellipses.

The decision-maker’s expected value in this land use problem is the potential areas for development, which are directly influenced by the proximity to roads, proximity to nature conservation areas, slope, aspect and visual exposure to water. The proximity to roads is determined by the existing road network and weighted by intervening slopes. However, the proximity to nature conservation areas is determined only by the locations of the designated areas for nature conservation. Slope and aspect are dependent on variations of elevation on the ground. Visual exposure to water is determined by elevation and positions of water bodies. Note that the values of these variables vary from location to location.



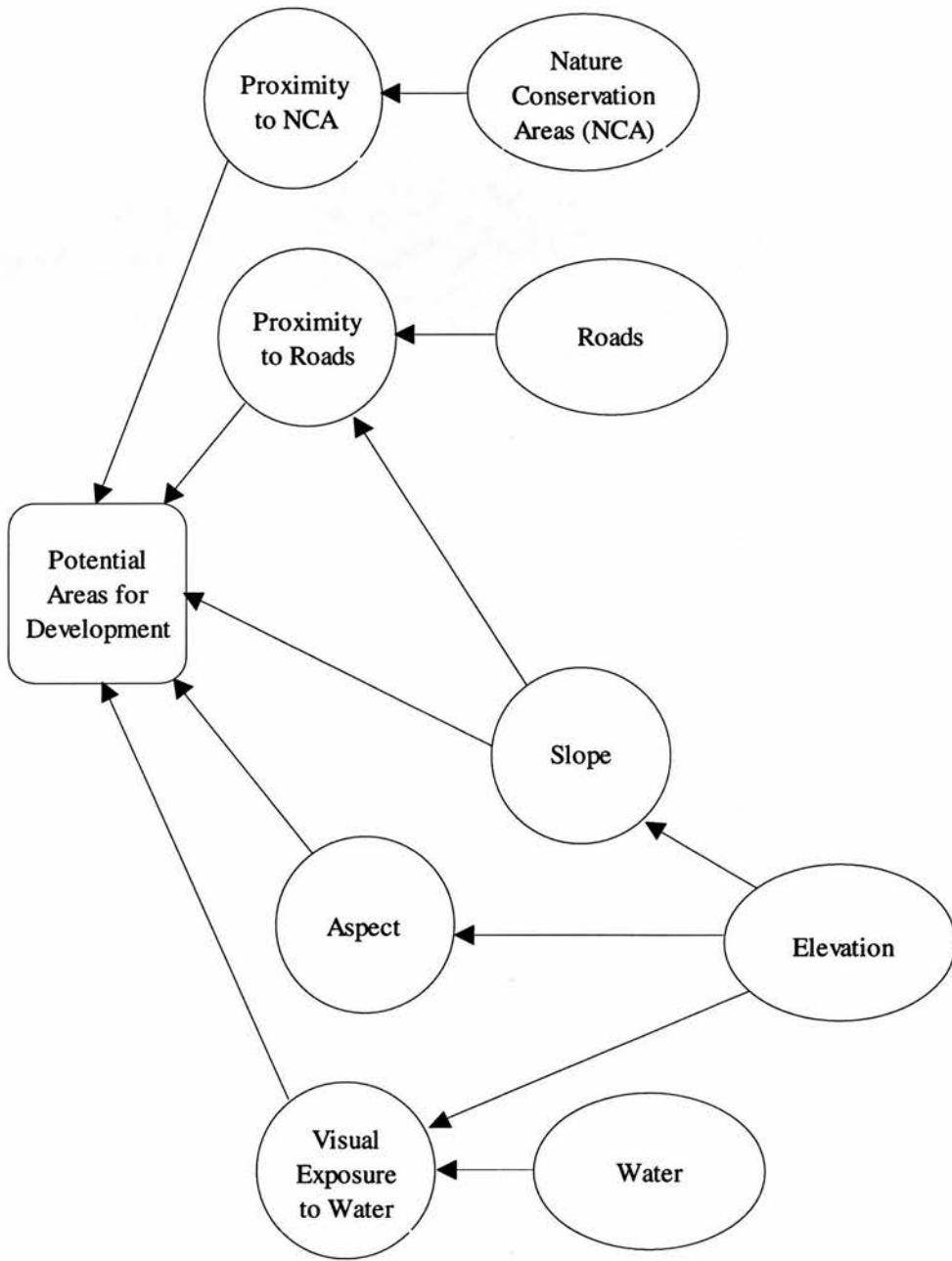


Figure 4.1 A spatial influence diagram constructed for predicting potential areas for development in a certain region

**Definition 2. Path.** A path from one node to another node is a set of arcs that forms a directed line from one node to another.

**Definition 3. Direct Predecessor.** The direct predecessors (or simply predecessors) of node  $i$  are the set of nodes  $P(i) = \{j \in N: (j, i) \in A\}$  with arcs directly connected from  $j$  to  $i$ . Thus, the border nodes are the set of nodes  $B = \{i \in N\}$  without direct predecessors.

**Definition 4. Indirect Predecessor.** The indirect predecessors of node  $i$  are the set of nodes along directed paths into it, but not directly connected to it.

For example, in Figure 4.1, the value node “Potential Areas for Development” has five direct predecessors, which are the chance nodes “Proximity to Roads”, “Proximity to Nature Conservation Areas”, “Slope”, “Aspect” and “Visual Exposure to Water”. Its indirect predecessors are the border nodes “Roads”, “Nature Conservation Areas”, “Elevation” and “Water”.

**Definition 5. Direct Successor.** The direct successors (or simply successors) of node  $i$  are the set of nodes  $S(i) = \{j \in N: (i, j) \in A\}$  with arcs directly connected from  $i$  to  $j$ . The value node has no direct successors.

**Definition 6. Indirect Successor.** The indirect successors of node  $i$  are the set of nodes along directed paths leading from it, but not directly connected to it.

As is apparent from Figure 4.1, the “Elevation” border node has three direct successors “Slope”, “Aspect” and “Visual Exposure to Water”, while its indirect successors are the value node “Potential Areas for Development” and the chance node “Proximity to Roads”.

**Definition 7. Basic Spatial Influence Diagram.** A basic spatial influence diagram is a combination of one node and its direct predecessors with arcs directed from the direct predecessors toward the node. It shows that the value of a variable is directly influenced by the values of a set of other variables.

For example, the value of “Proximity to Roads” is directly influenced by the values of “Roads” and “Slope”. Thus, “Proximity to Roads”, “Roads” and “Slope” form a basic spatial influence diagram with arcs directed from “Roads” and “Slope” to “Proximity to Roads”, as shown in Figure 4.2.

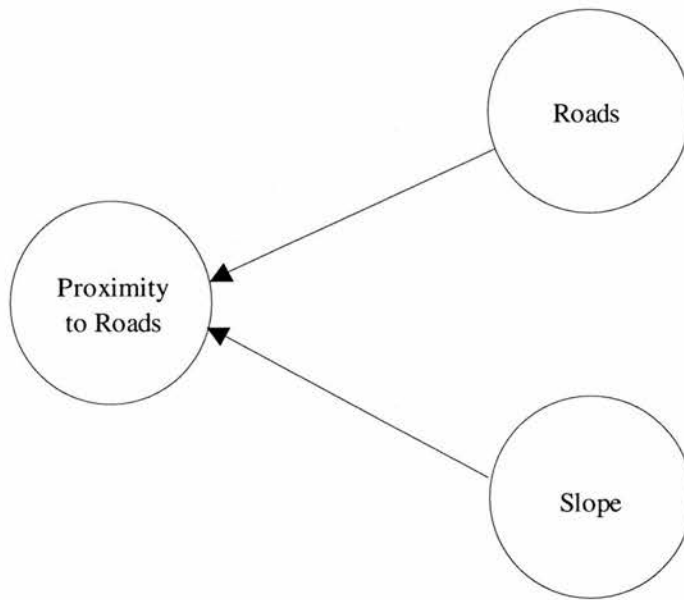


Figure 4.2 A basic spatial influence diagram

### 4.3 Representation of Resource and Environmental Problems Using Spatial Influence Diagrams

A spatial influence diagram can be viewed in terms of a directed acyclic graph consisting of labelled nodes and arcs, as shown in Figure 4.1. It captures the definitional structure of a spatial problem and represents the relationships or interdependence of RE variables relevant to the spatial problem. Chance and border nodes represent sets of problem parameters or variables directly or indirectly

influencing the value node. The values of a RE variable may be related to a location or an area. Therefore, a spatial influence diagram takes into account the spatial position of each problem variable as well as its other characteristics. A node in a spatial influence diagram may denote a variable whose values characterise different locations on the ground. As discussed in Chapter 3, a map is a traditional and effective means of representing the values of such a variable. We call a variable whose values are represented by a single-factor map a **map variable**, other variables would be termed **non-map variables**. So, variables in a spatial influence diagram may be attributes only or maps with different thematic values. The presence of an arc indicates that the values of one node are influenced by or dependent on the values of the other.

However, a spatial influence diagram can represent a spatial problem more fully and not just in terms of its structure. As we discussed in section 4.1, a problem model consists of specifications for all elements or parameters of the problem and their relationships. Thus, a full description of a spatial problem requires that consistent and detailed specifications exist for each node in the spatial influence diagram. A detailed description of each node also includes expressions of relations between the node and the others.

A spatial influence diagram takes into account the attribute or functional relations between variables as well as their spatial relations. The relations among the variables or elements in resource and environmental problems may be not only functional (e.g. logical and probabilistic), but also spatial. Variables of any type can be mathematically or logically characterised. Values of map variables also characterise a location or an area because spatial position is of equal importance to the value of the variable. Their values may depend on their spatial position, spatial shape, spatial distribution pattern and spatial contiguity. Therefore, analyses of resource and environmental problems may require access both to the attributes of the spatial objects under study and to their locational information.

The border nodes in the spatial influence diagram have associated values. The values of other nodes are dependent according to a definite rule on the values of the border nodes. Note that the notion of a “rule” used here is an abstract one which

could represent not only conditional probabilistic distributions as used in influence diagrams, but also other functional relations, such as logical relations, fuzzy logical relations, if-then rules and temporal relations, and spatial relations, such as neighbourhood or inclusion. Here, we differentiate between attribute relations and spatial relations. Attribute relations are represented by analytical models or rule-based models. Spatial relations are represented by GIS models. Every chance or value node (except border nodes) may have an associated functional model or/and a GIS model specified for representing the relations with its predecessors, and through which the values of the node will be derived (see the next section).

As will be seen in Chapter 8, in order to facilitate the automated selection and execution of models to solve a particular problem by the system, each node should be simple enough so that its attribute relation can be represented by a single functional model and its spatial relation can be represented by a single GIS model. However, it is not a requirement for the end users to conceptualise their problems in this way. A KBSDSS system may be developed in such a way that during the problem-solving process, a node can be automatically decomposed into a number of atomic nodes, each of them having a single functional model to represent its attribute relation and a single GIS model to represent its spatial relation.

Spatial influence diagrams can also represent a decision maker's preferences or needs. These are the decision maker's choices of the attributes, which directly influence the outcome of the value node, and relative rankings of these attributes in terms of desirability for various possible outcomes. In the example shown in Figure 4.1, five attributes, the proximity to roads, proximity to designated nature conservation areas, slope, aspect and visual exposure to water, were selected to evaluate the land use potential for development areas. Indeed, they are the decision maker's concerns for selecting the "best" locations for development.

## 4.4 Implementation of the Spatial Influence Diagram-Based Representation

In the implementation of the spatial influence diagram-based representation, we use frames to represent information for nodes and their relationships.

A node can be described by the following seven categories of information:

- the name of the node;
- the type of the node (value, chance or border);
- its predecessors;
- its successors;
- a functional (analytical or rule-based) model representing the attribute relation between the node and its predecessors;
- a GIS model (composed of GIS analysis functions) representing the spatial relation between the node and its predecessors; and
- the associated data representing its values.

In other words, a node can be represented by a frame with seven slots.

The value node has no successors. Its values will be derived by the specified functional model or/and the specified GIS model from the values of its predecessors. Figure 4.3 is an example, which is a representation of the value node “Potential Areas for Development” in Figure 4.1.

In Figure 4.3, *SuitabilityForDevelopment* is a model name representing the attribute relationship between the node “Potential Areas for Development” and the nodes “Proximity to Roads”, “Proximity to Nature Conservation Areas”, “Slope”, “Aspect” and “Visual Exposure to Water”. The model *overlay* is a GIS model for combining several data layers (see Chapter 3), representing their spatial relationship. The two models will be used together to derive the “Potential Areas for Development” from the values of the variables “Proximity to Roads”, “Proximity to

Nature Conservation Areas”, “Slope”, “Aspect” and “Visual Exposure to Water”.  
 The slots SUCCESSORS and ASSOCIATED\_DATA have no fillers.

NAME	Potential Areas for Development
TYPE	value
PREDECESSORS	Proximity to Roads, Proximity to Nature Conservation Areas, Slope, Aspect, Visual Exposure to Water
SUCCESSORS	
ATTRIBUTE_RELATION	<i>SuitabilityForDevelopment</i>
SPATIAL_RELATION	<i>overlay</i>
ASSOCIATED_DATA	

Figure 4.3 Value Node Frame of “Potential Areas for Development”

Information associated with a chance node includes its predecessors, successors, a functional model representing the attribute relationship between the node and its predecessors and a GIS model representing the spatial relationship between the node and its predecessors. Figure 4.4 is a representation of the chance node “Proximity to Nature Conservation Areas” in Figure 4.1.

NAME	Proximity to Nature Conservation Areas
TYPE	chance
PREDECESSORS	Nature Conservation Areas
SUCCESSORS	Potential Areas for Development
ATTRIBUTE_RELATION	
SPATIAL_RELATION	<i>buffering</i>
ASSOCIATED_DATA	

Figure 4.4 Chance Node Frame of “Proximity to Nature Conservation Areas”

In Figure 4.4, there is no functional model specified to represent the attribute relation. The relationship between the node “Proximity to Nature Conservation Areas” and its predecessor is specified using the spatial relationship alone, which is represented by the GIS model *buffering*. That means the values of the variable “Proximity to Nature Conservation Areas” will be derived from the values of the variable “Nature Conservation Areas” through *buffering*. The slot ASSOCIATED\_DATA is not specified. It will be filled when the value of the node has been derived.

A border node is specified by the information about its successors and associated data. Figure 4.5 is an example, which is a representation of the border node “Elevation” in Figure 4.1.

A border node corresponds to an available or acquirable data set or data layer. It has no predecessors. Thus, no attribute and spatial relationships are specified in the border node frame. We only need to specify its successors and corresponding data. In Figure 4.5, “elevation” is a name of the data layer recording the values of “Elevation”. If the slot ASSOCIATED\_DATA is not specified, the value of the border node will be given by the user.

NAME	Elevation
TYPE	border
PREDECESSORS	
SUCCESSORS	Slope, Aspect, Visual Exposure to Water
ATTRIBUTE_RELATION	
SPATIAL_RELATION	
ASSOCIATED_DATA	elevation

Figure 4.5 Border Node Frame of “Elevation”



## 4.5 Spatial Influence Diagrams and Cartographic Modelling

In the context of GIS analysis, a spatial problem is usually structured using the cartographic modelling approach (Burrough 1986; Berry 1987a; Tomlin 1990). The cartographic modelling approach systematically organises and links fundamental map processing operations or GIS models in sequence into a cartographic model for a particular spatial problem (see Chapter 3).

Figure 3.7 presented a cartographic model of land use potential for afforestation. It was used to identify the potential sites for afforestation. The cartographic model can be easily translated into the spatial influence diagram notation. Figure 4.6 is a restatement of Figure 3.7 in terms of a spatial influence diagram. The two representations are graphically similar. In the spatial influence diagram that describes a cartographic model, every node represents a map variable. Relationships among nodes are simply represented by spatial relations, which are specified using GIS models (i.e. map processing operations in cartographic modelling). However, GIS models in the spatial influence diagram are associated with nodes and not with arcs, as is the case in a cartographic model. Arcs in the spatial influence diagram represent influences, dependencies or relevance. After completion by defining nodes with the information described in Section 4.4, the spatial influence diagram illustrated in Figure 4.6 can be directly and automatically evaluated to identify the potential sites for afforestation, based on the mechanism to be described in the next chapter.

Spatial influence diagrams have two main advantages over cartographic models. First, spatial influence diagrams can accommodate a variety of models, including GIS models, analytical models and rule-based models. Analytical or rule-based models are specified as attribute relations among nodes. GIS models are specified as spatial relations among nodes. Thus, a spatial influence diagram can integrate different types of models as well as heuristic knowledge from human experts in the problem solution procedure. Cartographic modelling uses GIS models alone, hence there are limits to its modelling capabilities.

Second, nodes in a spatial influence diagram may denote map variables as well as non-map variables, while cartographic modelling can only operate on map variables. This feature enables spatial influence diagrams to incorporate different types of data, including map data layers, ordinary database files or data files. Therefore, spatial influence diagrams can be applied to solve a wider range of spatial problems than can cartographic modelling.

These advantages of spatial influence diagrams facilitate the machine representation of spatial problems and the spatial modelling process within the KBSDSS environment, which incorporates GIS, analytical and rule-based models, and aims to solve complex resource and environmental problems by integrating different types of data. The spatial influence diagram reflects an abstraction of spatial problems on which the algorithm and heuristics can be developed to integrate different types of models and data, and to automate the solution process, as will be described in the next chapter.

## 4.6 Summary

Problem structuring and representation is of major importance in KBSDSS systems. We have reviewed some basic problem structuring methodologies and proposed a spatial influence diagram-based representation scheme for structuring and representing resource and environmental problems. The advantages of spatial influence diagrams were also discussed by comparing them with cartographic modelling approaches. Spatial influence diagrams provide an intuitive framework in which to formulate spatial problems as well as a description of information that can be stored and manipulated by a computer.

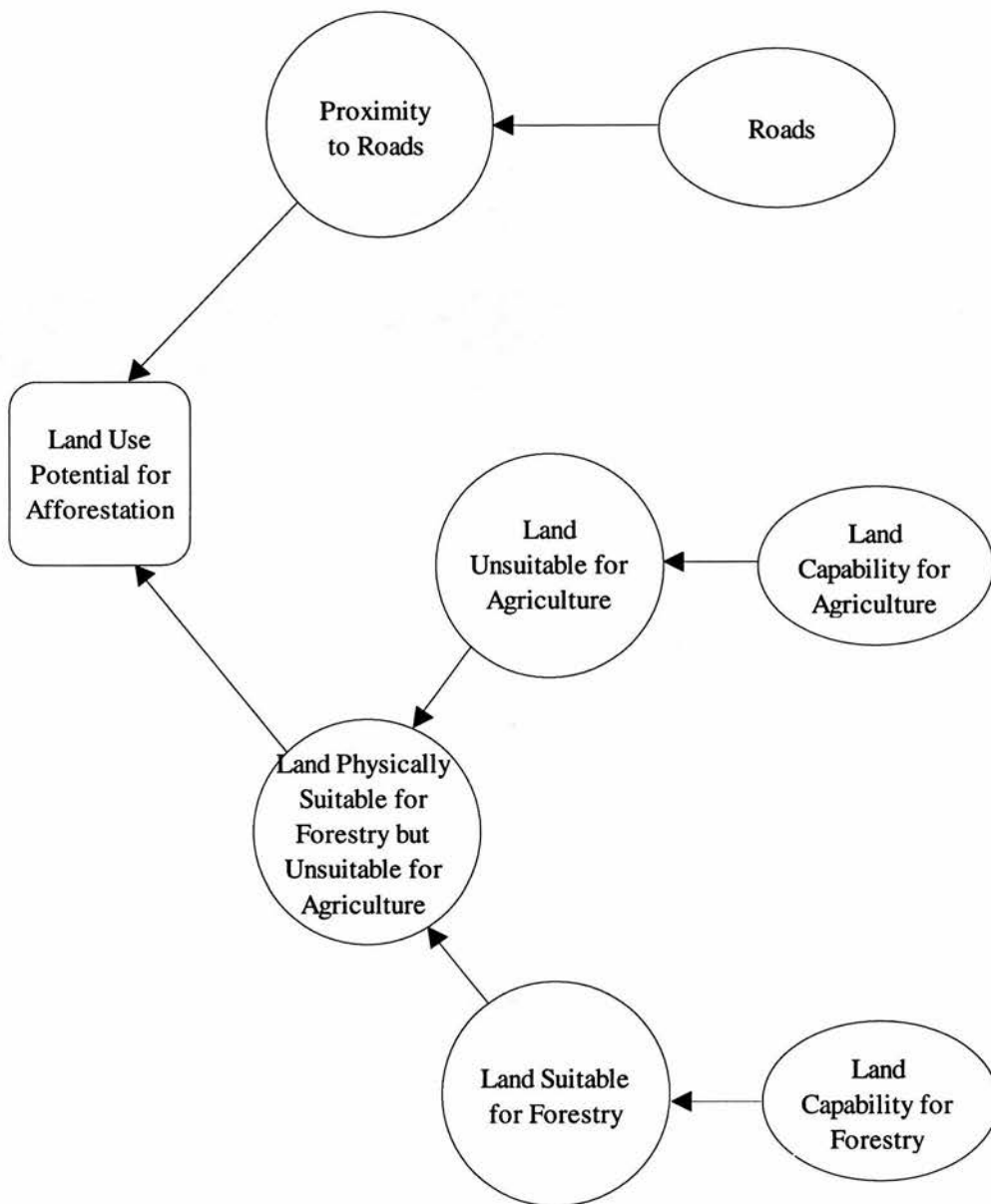


Figure 4.6 A spatial influence diagram for identifying land use potential for afforestation

## Chapter 5

# Formulation and Evaluation of Resource and Environmental Problems

A spatial influence diagram is an intuitive conceptual and operational representation for resource and environmental problems. It is also an effective tool for formulating and solving resource and environmental problems. Indeed, a spatial influence diagram-based representation is an abstraction of a spatial problem on which heuristics and algorithms may be developed to build capabilities for integrating data and models logically and for automation of the solution process.

In this chapter, we focus on the mechanisms for formulation and evaluation of spatial problems using spatial influence diagrams. We first discuss the knowledge requirements for formulation and evaluation of spatial influence diagrams, then describe the algorithms required, and finally, describe the process for evaluating each individual node or problem variable.

## 5.1 Knowledge Requirements for Formulation and Evaluation of Spatial Influence Diagrams

As discussed in the previous chapter, the knowledge representation of a spatial influence diagram consists of a variety of information, including interdependence of problem variables, attribute relations and spatial relations between the variables, and data associated with the border nodes.

In order to formulate and evaluate a spatial influence diagram automatically, a KBSDSS system is required to have a knowledge base containing all the necessary information associated with spatial influence diagrams, as described above. The knowledge base includes three categories of knowledge: domain knowledge, model knowledge and meta-data (knowledge about data), which are respectively stored in different modules.

The domain knowledge module contains domain variable frames, which describe variables relevant to a problem domain. Each domain variable frame represents a stand-alone domain variable, including the descriptions of its relations with other RE variables (or its predecessors in the spatial influence diagram notation) that directly influence it. The relations between the domain variable and its predecessor variables are represented using functional or/and GIS models. A domain variable frame is a basic element for automatic formulation of a spatial influence diagram with regard to a particular problem in a problem domain. In other words, it is a building-block for automatic spatial modelling. A domain variable frame has four slots, NAME, PREDECESSORS, ATTRIBUTE\_RELATION AND SPATIAL\_RELATION, as shown in Figure 5.1.

A domain variable is identified by a unique name. Figure 5.1 represents a domain variable “Proximity to Roads”. Another two variables “Slope” and “Roads” directly influence it. Note that “Roads” represents spatial locations of existing roads in a study area. *EffectiveProximityOnSlope* is a GIS model for identifying the shortest

distance from targets (here roads) to all other locations. This distance is weighted by intervening slopes.

NAME	Proximity to Roads
PREDECESSORS	Roads, Slope
ATTRIBUTE_RELATION	
SPATIAL_RELATION	<i>EffectiveProximityOnSlope</i>

Figure 5.1 Domain variable frame of “Proximity to Roads”

A basic spatial influence diagram (see Chapter 4) can be formulated by a combination of the slot NAME and the slot PREDECESSORS with an arc pointing to the node specified by NAME from each of the predecessor nodes specified by PREDECESSORS. The domain variable frame “Proximity to Roads” corresponds to the basic spatial influence diagram illustrated in Figure 4.2.

The model knowledge module contains information relating to all available analytical, rule-based and GIS models. A model corresponds to one or more system-specific algorithms, procedures or rule sets. We call these system-specific algorithms, procedures or rule sets **solvers**. In many cases, there is exactly one solver for a given model. Some GIS models may have two corresponding solvers in terms of different spatial data models (vector and raster). It is desirable for a KBSDSS system to have both vector- and raster-based spatial data analysis and modelling capabilities, so as to meet the needs of different users. For example, the GIS model *overlay* may have two solvers, which are the ARC/INFO functions *union* and *combine*. The two solvers differ in implementation, data and parameter requirements (see Figure 5.2 and 5.3). Nevertheless, a unique selection can be made by taking their data requirements into consideration. The model knowledge describes every solver available in a KBSDSS system. The descriptions of a solver contain the information about the model type that it belongs to, the tool on which it is executed, the required data model and data

structure of its input data, and the parameters or arguments in the required calling sequences. Figure 5.2 and 5.3 are two examples. Here, an **ARC/INFO coverage** is a data structure used in the ARC/INFO GIS software system for storing a digital map, which consists of topologically linked geographical features and their associated data (ESRI 1991a). It is based on a combined vector-based spatial data model and a relational attribute data model. An **ARC/INFO grid** is another data structure used in the ARC/INFO system for storing a digital map that is divided into discrete units called grid cells. It is based on a combined raster-based spatial data model and a relational attribute data model (ESRI 1991a).

SOLVER_NAME	<i>union</i>
MODEL_TYPE	<i>overlay</i>
TOOL	ARC/INFO
DATA_MODEL	vector
DATA_STRUCTURE	ARC/INFO coverage
PARAMETERS	in_cover, union_cover, out_cover

Figure 5.2 Solver frame of *union*

SOLVER_NAME	<i>combine</i>
MODEL_TYPE	<i>overlay</i>
TOOL	ARC/INFO
DATA_MODEL	raster
DATA_STRUCTURE	ARC/INFO grid
PARAMETERS	out_grid, in_grid1, in_grid2, in_grid3, in_grid4, ...

Figure 5.3 Solver frame of *combine*

In addition, the model knowledge module also contains knowledge in the form of rules for assignment of parameter values of the selected solvers and scheduling of the solver execution sequence.

The meta-data knowledge module consists of knowledge about all data available in the database of a KBSDSS system. The meta-data structure is shown in Figure 5.4, which includes (1) the name of the data set or data layer; (2) its data model; (3) its data structure; (4) the data type (spatial or attribute); (5) the feature type if it depicts spatial objects; (6) its scale if it is a map data layer; (7) the data source; and (8) the encoded attributes.

The domain knowledge module is important to the formulation of a spatial influence diagram. The model and meta-data knowledge modules are important to the selection of model solvers and evaluation of spatial influence diagrams.

## 5.2 Formulation of Resource and Environmental Problems

The process by which a real resource and environmental problem is formulated plays a crucial role in automated spatial modelling using a KBSDSS system. The formulation of a spatial problem is indeed the formulation of a spatial influence diagram.

The formulation of a spatial influence diagram can be viewed as a “backward-chaining” or “goal-driven” generation process (see Chapter 2). That is, the formulation process starts from the goal of the spatial problem solving specified by the decision maker. The decision maker’s goal is also a domain variable. We call it a **goal variable** here. For example, in the land use problem of predicting the land use potential for afforestation, the “Potential Sites for Afforestation” is the goal variable. A goal variable is represented by the value node in a spatial influence diagram. The formulation process involves an extensive search in the meta-data knowledge module and the existing domain variables in the domain knowledge module.



NAME	landcover
DATA_MODEL	raster
DATA_STRUCTURE	ARC/INFO grid
DATA_TYPE	spatial
FEATURE_TYPE	area
SCALE	1:25000
SOURCE	NCC 1:25000 Landcover Map
ATTRIBUTES	<ul style="list-style-type: none"> <li>1 - Natural &amp; semi-natural woodland</li> <li>2 - Coniferous plantations</li> <li>3 - Policy woodlands</li> <li>4 - Heath (wet and dry)</li> <li>5 - Bog</li> <li>6 - Rough grassland (unimproved)</li> <li>7 - Improved and enclosed grassland</li> <li>8 - Marsh, swamp and fen</li> <li>9 - Lochs (open water)</li> <li>10 - Coastal heath, cliffs and dunes</li> <li>11 - built-up (including gardens)</li> <li>12 - Bracken</li> <li>13 - Montane</li> <li>20 - sea</li> </ul>

Figure 5.4 Meta-data frame of the data layer “landcover”

When a goal variable is specified by the user, the process of formulation starts. The system first checks the domain knowledge module. If the goal variable does not exist in the domain knowledge module, the system will report to the user that the problem can not be solved using the existing knowledge stored in the system. Otherwise, the system goes to check the meta-data knowledge module to see whether its value is available in the database. If it is, the process terminates. That means the spatial problem already has a solution. The system will present the solution to the decision maker. No further formulation process is needed. Otherwise, the system will set it to be the value node, create a value node frame (such as that shown in Figure 4.3) for it and then search for its direct predecessor variables in the existing domain variables in the domain knowledge module. Afterwards, the system picks up one of the direct predecessors of the current variable. If its value is available in the database or is to be provided by the user, or it has no predecessors, set it to be a border node and create a corresponding border node frame (such as that shown in Figure 4.5). Otherwise, set it to be a chance node, create a corresponding chance node frame (such as that shown in Figure 4.4) and then search for its direct predecessors in the existing domain variables. The process repeats itself until all border nodes are found. After the formulation process completes, a fully specified spatial influence diagram is formed. Figure 5.5 illustrates the process for formulating a spatial influence diagram as a flow chart. The equivalent procedure to formulate a spatial influence diagram in pseudo-code terms is shown in Figure 5.6.

Creation of a node frame is a process of assigning the values to the slots defined in the node frame, i.e. the name of the node, the node type, the predecessors, the successors, the attribute relation with the predecessors, the spatial relation with the predecessors and the associated data. The information about the node type, the successors and the associated data is obtained through reasoning during the formulation process, as described above. The information about other slots is obtained from the corresponding domain variable frame of the node.

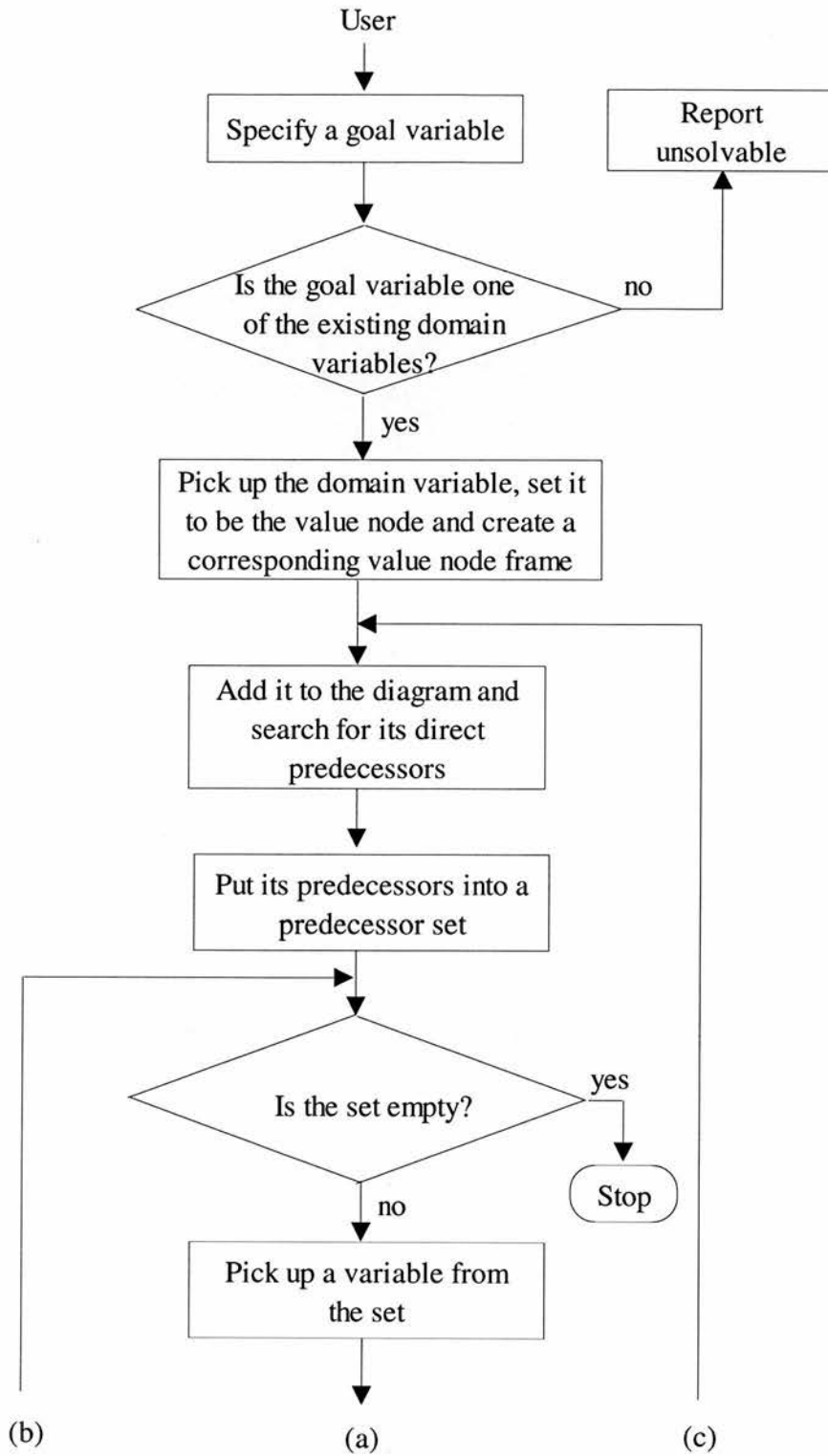


Figure 5.5 Process for formulating a spatial influence diagram

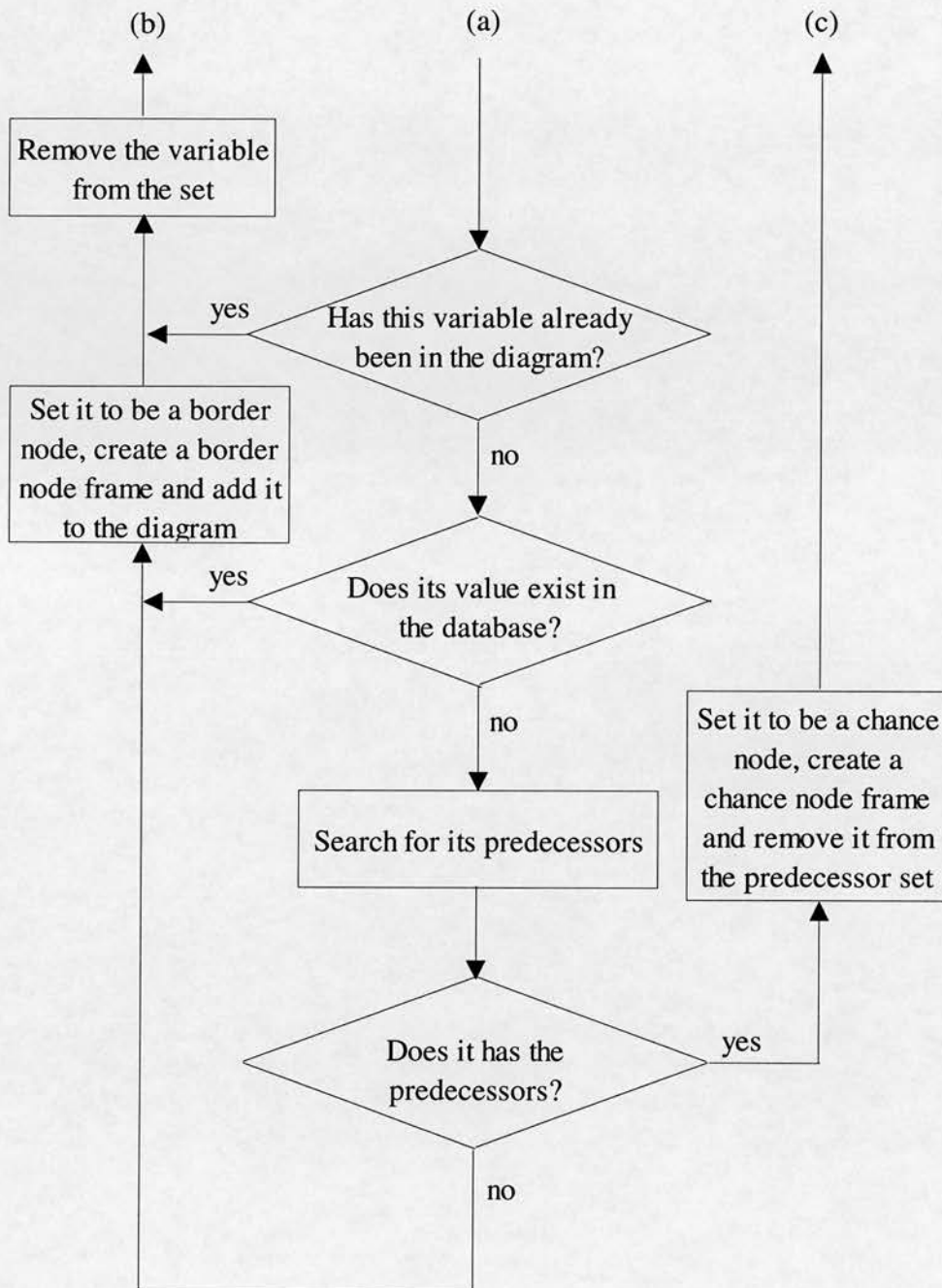


Figure 5.5 (continued) Process for formulating a spatial influence diagram

**Algorithm.** Formulating a spatial influence diagram.

**Input.** A goal variable specified by the user.

**Output.** A spatial influence diagram.

**Assumption.** Suppose the user has specified a goal variable and its value is not directly available in the database.

**Procedure FORMULATION OF A SPATIAL INFLUENCE DIAGRAM**

**begin**

*let  $N$  be the node set of the spatial influence diagram to be generated,  $N = \emptyset$ ;*

*let  $K$  be a set of current nodes under examination,  $K = \emptyset$ ;*

*search a domain variable in the domain knowledge module which corresponds to the goal variable;*

*if there exists a domain variable  $i$  corresponding to the goal variable*

***then** set the variable  $i$  a value node; add  $i$  to  $N$ ,  $N = \{i\}$ ; create a value node frame;*

***else** report that the problem can't be solved using the knowledge in the system; stop;*

*search the PREDECESSORS slot of the domain variable  $i$ ;*

*add all of its direct predecessors to  $K$ ;*

***while**  $K \neq \emptyset$  **do***

***begin***

***for**  $j \in K$  **do***

***begin***

***if**  $j \in N$*

***then** remove  $j$  from  $K$ ,  $K = K \setminus \{j\}$ ; /\*  $j$  already exists in  $N$ . \*/*

***else** search the value of  $j$  in the meta-data knowledge module;*

***if** there exists a value of  $j$  in the database*

***then** set  $j$  a border node; create a border node frame;*

*add  $j$  into  $N$ ,  $N = N \cup \{j\}$ ;*

*remove  $j$  from  $K$ ,  $K = K \setminus \{j\}$ ;*

***else** search the PREDECESSORS slot of the domain variable  $j$ ;*

***if**  $j$  has no direct predecessors*

***then** set  $j$  a border node; /\* the system will prompt the user for its value during the evaluation of the diagram \*/*

*create a border node frame; add  $j$  into  $N$ ,  $N = N \cup \{j\}$ ;*

***else** set  $j$  a chance node; create a chance node frame;*

*add all of its predecessors to  $K$ ;*

*add  $j$  into  $N$ ,  $N = N \cup \{j\}$ ;*

*remove  $j$  from  $K$ ,  $K = K \setminus \{j\}$ ;*

***end***

***end***

*return  $N$ ;*

**end**

(For set  $S$  and set  $T$ , the notation used for set operations is as follows:

$x \in S$   $x$  is a member of set  $S$

$S \subset T$   $S$  is a subset of  $T$

$S \cup T$  union of  $S$  and  $T$

$S = \emptyset$   $S$  is empty

$\{x\}$  a set containing one element  $x$

$T \setminus S$  subtraction of  $S$  from  $T$ )

Figure 5.6 Procedure for formulation of a spatial problem

## 5.3 A Solution Procedure for Resource and Environmental Problems

A spatial influence diagram represents a spatial problem model wherein the major factors influencing the decision maker's objective are explicitly shown. The structure in the spatial influence diagram clearly indicates a logic for obtaining the solution of the spatial problem it represents.

**Definition 8. Evaluation.** The process of calculating the values of all nodes in their natural topological order is called evaluation.

The set of directed arcs in a spatial influence diagram corresponds to an evaluation order. For example, a value of a variable  $i$  could be obtained by first evaluating a variable  $j$ , and then evaluating  $i$  based on the value of  $j$ . This evaluation order would correspond to a two-node spatial influence diagram with an arc from  $i$  to  $j$ . Thus, a spatial influence diagram implies an evaluation order for its solution. Such an order always exists by virtue of the acyclicity assumption (see the definition of spatial influence diagram in Chapter 4).

The relations among variables mean that one variable can depend in a general way on several others. Namely, the value of a chance or value node  $i$ ,  $U(i)$ , is a function of the set of its direct predecessor variables,  $P(i)$ :

$$U(i) = f(P(i)), \quad P(i) \subset N$$

The function  $f( )$  maps every combination of outcomes of the set of its direct predecessor variables to a unique outcome space for the node  $i$ . The function  $f( )$  may be a functional model, or a GIS model, or a functional model plus a GIS model, for evaluating a node in terms of its predecessors. The set of outcomes corresponds to a range of values for the variable being represented.

Spatial influence diagrams provide an interface through which we may develop algorithms for integrating multiple models and automating the solution process of

spatial problems. In this section, we discuss the solution procedure for a spatial problem by evaluating a spatial influence diagram. The focus is on the relationships among the variables and the way to solve the problem. The process for evaluation of each individual node in a spatial influence diagram is left for discussion in the next section. Before going on to our algorithm, we define the assessed nodes and unassessed nodes.

**Definition 9. Assessed Node and Unassessed Node.** A node is said to be an assessed node if it has a value. Otherwise, it is an unassessed node.

Border nodes have no predecessors. In an original spatial influence diagram (before the evaluation process starts), they represent basic factors influencing the value node. Indeed, they are inputs to the spatial problem model. Most border nodes are assessed nodes. For an unassessed border node, the system will prompt the user for its value. Our solution procedure will remove assessed nodes from the diagram until an outcome of the value node is obtained. When a node is removed, it can be dropped from the current node set  $N$ , and all arcs incident to it can be dropped from the current arc set  $A$ . Its value is also propagated to its direct successors. When the values of all the direct predecessors of a node have been obtained, it can be evaluated.

An algorithm for evaluating a spatial influence diagram is outlined in Figure 5.7.

As an example of the algorithm, consider the land use problem of identifying potential areas for development shown in Figure 4.1.

At the beginning, "Nature Conservation Areas", "Roads", "Elevation" and "Water" are border nodes. Their values exist already. They are assessed nodes. The algorithm will propagate their values to their direct successors, and then remove the four border nodes.

After the removal of the nodes "Nature Conservation Areas", "Roads", "Elevation" and "Water", the nodes "Slope", "Proximity to Nature Conservation Areas", "Aspect" and "Visual Exposure to Water" can be evaluated (Figure 5.8a). The algorithm will then evaluate the four nodes one by one, propagate their values to their all direct successors and remove them.

Now, the new diagram has just two nodes (Figure 5.8b). The node “Proximity to Roads” will be evaluated. After its value is inherited by the value node, it will be removed.

Finally, the value node “Potential Areas for Development” is evaluated and the solution of the land use problem is obtained.

The algorithm can evaluate any spatial influence diagram. Every step of the algorithm removes at least one node, so the algorithm will always terminate. The inference on a spatial problem model can be performed by the algorithm, which drives its solution process.

**Algorithm.** Solving a spatial problem through evaluation of a spatial influence diagram.

**Input.** A spatial influence diagram constructed for a particular spatial problem.

**Output.** The outcome of the value node  $v$ .

**Procedure SOLUTION OF A SPATIAL PROBLEM**

**begin**

*let  $N$  be the set of all nodes in the input spatial influence diagram;*

*let  $N'$  be the nodes to be removed,  $N' = N \setminus \{v\}$ ; /\* all nodes but the value node for the spatial problem.\*/*

*let  $K$  be the set of the nodes which can be evaluated currently;  $K = \emptyset$ ;*

**while** *there exist unassessed nodes  $\in N$  do*

*add all the border nodes into  $K$ ;*

**begin**

**while**  $N' \neq \emptyset$  **do**

**begin**

**if** *there exists a node  $i \in K$*

**then if**  *$i$  is an unassessed node*

**then** *evaluate  $i$ ; set  $i$  assessed;*

*propagate the value of  $i$  to its direct successor set  $S(i)$ ;*

*remove  $i$  from  $K$  and  $N'$ ;*

**else** *update  $K$ ;*

**end**

*evaluate the value node  $v$ ;*

*set  $v$  assessed;*

*return the value of  $v$ ;*

**end**

**end**

Figure 5.7 Solution procedure for a spatial problem



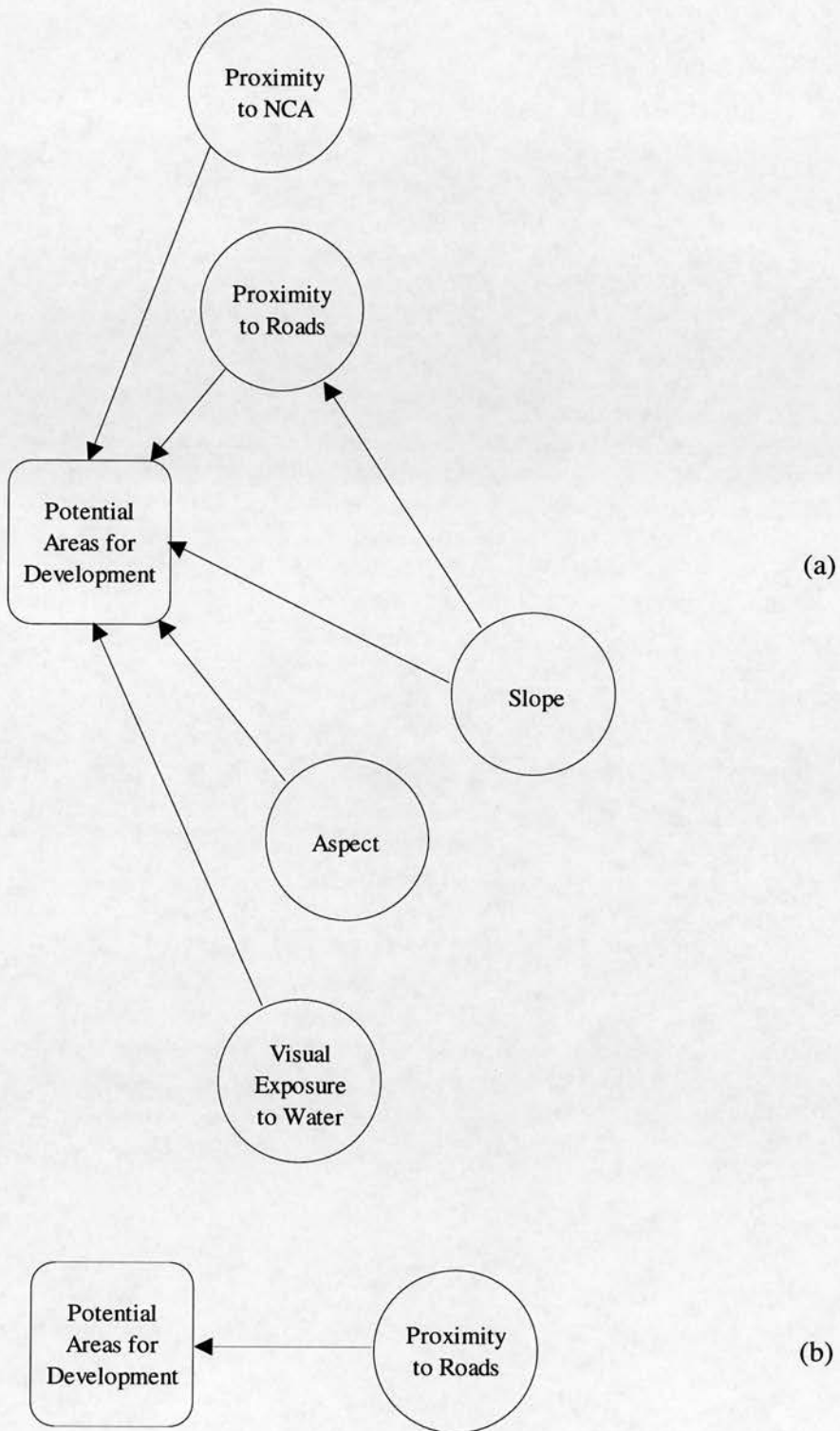


Figure 5.8 Evaluating the land use problem of identifying potential areas for development depicted in Figure 4.1.

## 5.4 Evaluation of Nodes in Spatial Influence Diagrams

Each individual node is evaluated during the evaluation process of a spatial influence diagram driven by the algorithm shown in Figure 5.7. The values of all nodes will be combined to produce the outcome of the value node, i.e. the solution of the specified spatial problem. For an unassessed border node, its value will be given by the user. Therefore, we will mainly discuss the evaluation of chance and value nodes. The evaluation of a node in a spatial influence diagram involves model scheduling if both a functional and a GIS model are used, selection of model solvers, checking of input data, assignment of solver parameter values and execution of the selected solvers.

As described in Section 5.2, after the formulation of a spatial influence diagram, each node has an associated node frame with seven slots: NAME, TYPE, PREDECESSORS, SUCCESSORS, ATTRIBUTE\_RELATION, SPATIAL\_RELATION and ASSOCIATED\_DATA (see Figures 4.3-4.5). For an unassessed node, the ASSOCIATED\_DATA has no filler (e.g. the nodes shown in Figures 4.3 and 4.4). It will be filled with the derived data after the node has been assessed. The meta-data information of the derived data will also be written as a meta-data frame (see Figure 5.4) into the meta-data knowledge module once it is produced. The PREDECESSORS actually represent input data to the models specified in the ATTRIBUTE\_RELATION and SPATIAL\_RELATION slots.

Before the evaluation of a spatial influence diagram starts, a spatial data structure (corresponding to vector or raster) is set as a default spatial data structure to be used for the whole solution process of the problem. During the solution process, data that are stored in other spatial data structures will automatically be translated to the default spatial data structure when necessary. When the evaluation process of a node starts, the system first establishes a model queue for storing the models to be executed for the node in a proper sequence. Initially, the queue is empty, i.e. no models are in the queue. If only one model is attached to the node, the system puts the model into the queue. The model queue contains only one model. If two (one functional and one GIS) models are attached to the node, the system schedules the

models, i.e. decides which model is to be executed first and puts the model into the model queue at the beginning and the other at the end. The model scheduling has to be done by using heuristic rules. For example, the following rule is used to schedule the model *overlay* and a functional model *f* for evaluating a node:

Rule: schedule-overlay-and-a-functional-model

IF the GIS model *overlay* and the functional model *f* are used  
together to derive the value of the current node

THEN put *overlay* into the model queue at the beginning, and *f* at the end.

Since *overlay* is simply to combine multiple data layers, followed by attribute data handling, it is always executed before the associated functional model (see Section 3.3 for the definition of the *overlay* model). Functional models associated with *overlay* can be any analytical models, such as mathematical equations and logical operations, and any rule-based models encoded as if-then rules. These functional models generally accept the outputs from *overlay* as their inputs, and they are executed after it. Therefore, the system always puts *overlay* into the model queue first and its associated functional model second.

Among the GIS models listed in Figure 3.6, *reclassify* may also have functional models associated with it. However, the functional model associated with *reclassify* is normally a reclassification table in a certain format. It is generally non-executable. Instead, it is taken as one of the parameters of *reclassify*. Thus, the system does not recognise a reclassification table as a model, and only puts *reclassify* into the model queue.

After the model or models are put into the model queue, the system gets a model and removes it from the beginning of the queue. Then it selects an appropriate solver for the model by searching the model knowledge module. There may be several solvers for a given model. Selection of an appropriate solver for a given model can be done using heuristic rules stored in the model knowledge module. It should be noted that selection of model solvers is a complex task. It depends on the decision-making circumstance and other factors, and it is hampered by the subjective

preferences of individuals (Banerjee and Basu 1993). Encoding knowledge and expertise for selection of model solvers should take into account the nature of organisation in which the problem has been observed, the characteristics of the problem, the accuracy that can be expected from the problem solution and so on, and some expertise for model solver selection is not yet available for many problem domains. This is one of important areas which need further research in the future. Thus, in our prototype system to be described in Chapter 8, we make a simplifying assumption that there is exactly one solver for a given analytical or rule-based model, but there may be two solvers for a given GIS model in terms of their spatial data models. In such a case, for an analytical model or a rule-based model, solver selection is simply a process of matching the model name with the MODEL\_TYPE slot of solver frames (see Figures 5.2 and 5.3). To select a solver for a given GIS model, the system looks for all solvers for the GIS model in the model knowledge module, and chooses a solver whose required data structure is the same as the default spatial data structure. If there exists no such solver, the system will choose a solver corresponding to the model.

Next, the system checks all input data to the model by searching the meta-data knowledge module. It transforms all data into the data structure required by the selected solver, if their data structures are different. Then, the system assigns the appropriate values to the required arguments (input parameters) of the solver by using heuristic rules according to the parameters defined in the PARAMETERS slot of the corresponding solver frame. For example, the following rule is used to assign the parameter values for the solver *combine* (see Figure 5.3 for its frame structure):

Rule: assign-parameters-to-*combine*

IF the actual solver is *combine*

AND the current node has five predecessors and their corresponding data  
are respectively p1, p2, p3, p4 and p5

THEN out\_grid is temp, in\_grid1 is p1, in\_grid2 is p2, in\_grid3 is p3,  
in\_grid4 is p4 and in\_grid5 is p5.

Here, “temp” is a temporary data layer for storing the result produced by *combine*.

In some cases, the proper choice of parameters critically depends on the actual problem and is left, therefore, to the user. For example, one of the arguments of the GIS model *buffering* is the buffer distance. Different users may have different choices for its value according to their problems and situations. The system will prompt the user for the value of the argument.

Afterwards, the system will activate an appropriate tool to execute the selected solver. A result will be produced. Then, the system checks the model queue. If the queue is not empty, the system will get the remaining model, remove it from the queue and repeat the process above to execute it. If the queue is empty, the evaluation of the node completes. The final output (if it is a map data layer) will be transformed back to the default spatial data structure if they are different. The name of the data set will be filled into the ASSOCIATED\_DATA slot of the node frame, and the meta-data information of the derived data will be written into the meta-data knowledge module.

Figure 5.9 illustrates the evaluation process.

As an example, consider evaluation of the value node “Potential Areas for Development” in the diagram shown in Figure 4.1. Its corresponding node frame has been shown in Figure 4.3, which is reproduced here as Figure 5.10 for convenience.

We assume that all other nodes in the diagram have been evaluated, and their values or associated data have been obtained, which are all encoded in ARC/INFO grids — the default spatial data structure. After the values of all the predecessors of the value node have been propagated to it, the value node frame is updated as shown in Figure 5.11 and the evaluation process starts. Here, roadbuf, conservbuf, slope, aspect and expwater are respectively the data layers associated with the nodes “Proximity to Roads”, “Proximity to Nature Conservation Areas”, “Slope”, “Aspect” and “Visual Exposure to Water”.

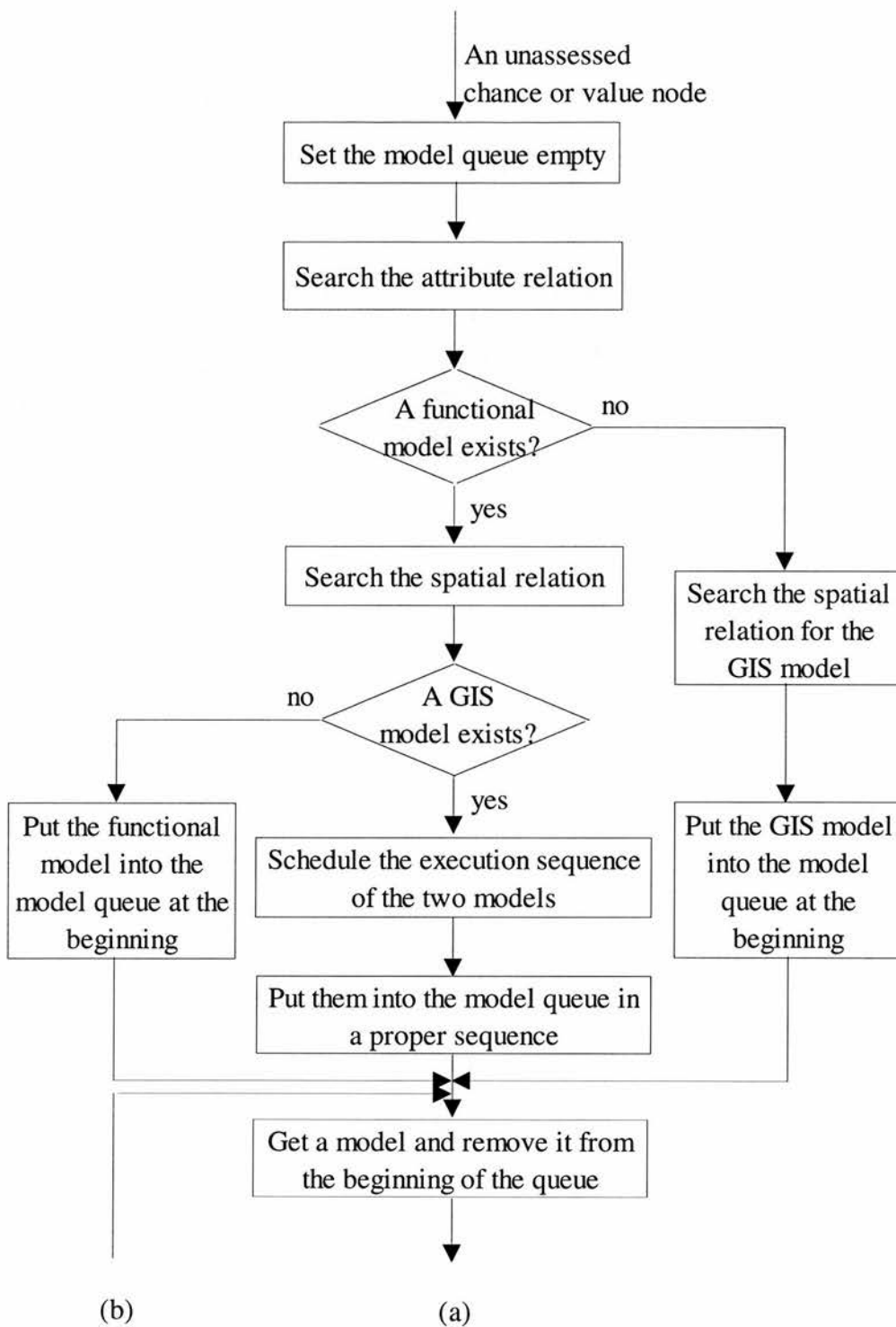


Figure 5.9 Process of evaluating a node

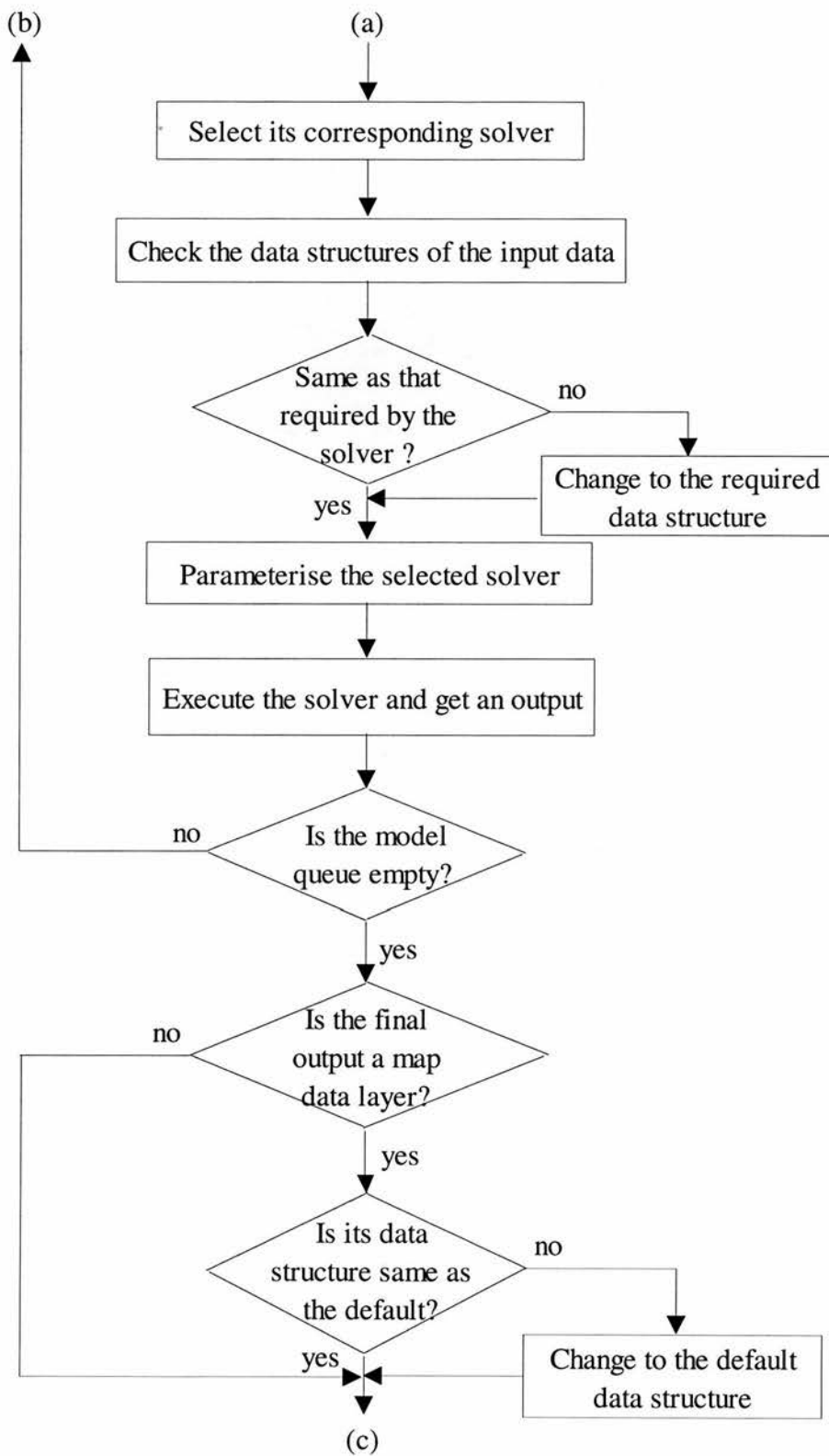


Figure 5.9 (continued) Process of evaluating a node

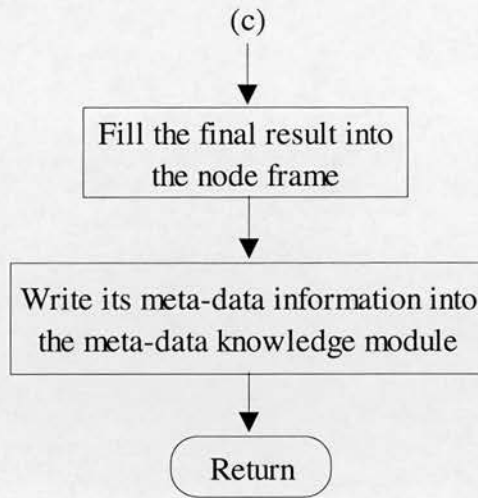


Figure 5.9 (continued) Process of evaluating a node

NAME	Potential Areas for Development
TYPE	value
PREDECESSORS	Proximity to Roads, Proximity to Nature Conservation Areas, Slope, Aspect, Visual Exposure to Water
SUCCESSORS	
ATTRIBUTE_RELATION	<i>SuitabilityForDevelopment</i>
SPATIAL_RELATION	<i>overlay</i>
ASSOCIATED_DATA	

Figure 5.10 Value node frame of “Potential Areas for Development”  
(reproduced from Figure 4.3)



NAME	Potential Areas for Development
TYPE	value
PREDECESSORS	roadbuf, conservbuf, slope, aspect, expwater
SUCCESSORS	
ATTRIBUTE_RELATION	<i>SuitabilityForDevelopment</i>
SPATIAL_RELATION	<i>overlay</i>
ASSOCIATED_DATA	

Figure 5.11 The node frame of “Potential Areas for Development” after the values of its predecessors have been propagated to it

Because two models *SuitabilityForDevelopment* and *overlay* have to be used together in order to evaluate the node, the system first schedule the two models. The rule “schedule-overlay-and-a-functional-model” described above is applicable. It schedules the two models, puts *overlay* into the model queue at the beginning and *SuitabilityForDevelopment* at the end. Here, *f* is the *SuitabilityForDevelopment* model. The *overlay* model should be executed first to combine the data layers describing the values of the proximity to roads, proximity to nature conservation areas, slope, aspect and visual exposure to water. Then, the model *SuitabilityForDevelopment* can be run on the combined data set to derive the suitability for development in the study area.

Then, the system gets *overlay* and removes it from the model queue and starts to execute it. First, it checks the data structures of the data layers “roadbuf”, “conservbuf”, “slope”, “aspect” and “expwater” and determines that they are all in ARC/INFO grids. Thus, the system selects *combine* as the solver of *overlay* because the input data structure it requires is the ARC/INFO grid (see the solver frame shown in Figure 5.3). The rule “assign-parameters-to-*combine*” is then applied to assign the parameters to *combine*. Here, out\_grid is temp, in\_grid1 is roadbuf, in\_grid2 is conservbuf, in\_grid3 is slope, in\_grid4 is aspect and in\_grid5 is expwater. After the

assignment of the parameters, the system executes *combine* and the output data layer “temp” is obtained.

Next, the system gets *SuitabilityForDevelopment* and removes it from the model queue, searches its corresponding solver, and checks its data requirements. Here, “temp” is one of its inputs. If the data structure of “temp” is different from that required by the solver, the system converts it into the required data structure. Then, the system assigns the appropriate parameters to it and executes it. We assume that the final result is written into a data layer called “developareas”. Thus, the value of the node “Potential Areas for Development” is obtained. The node frame associated with the node is updated as shown in Figure 5.12. As can be seen, the ASSOCIATED\_DATA slot is filled with “developareas”, which indicates that the node “Potential Areas for Development” has become an assessed node. The evaluation of the node “Potential Areas for Development” is completed.

NAME	Potential Areas for Development
TYPE	value
PREDECESSORS	roadbuf, conservbuf, slope, aspect expwater
SUCCESSORS	
ATTRIBUTE_RELATION	<i>SuitabilityForDevelopment</i>
SPATIAL_RELATION	<i>overlay</i>
ASSOCIATED_DATA	developareas

Figure 5.12 The node frame of “Potential Areas for Development” after its value has been obtained

## 5.5 Summary

This chapter has described the spatial influence diagram-based mechanisms for formulation and evaluation of spatial problems. We started by discussing the knowledge requirements for formulation and evaluation of spatial influence diagrams. The knowledge required includes domain knowledge about domain variables, model knowledge about model solvers and meta-data about data. The three categories of knowledge form part of the knowledge base of a KBSDSS system. By using the relevant knowledge, the algorithms for formulation and evaluation of spatial influence diagrams were developed. The algorithm for formulation of spatial influence diagrams is a mechanism for the automated formulation of spatial problem models that integrates different types of data and models logically. The algorithm for evaluation of spatial influence diagrams is a mechanism for automating the solution process for spatial problems. We also described the process of evaluation of each individual node in a spatial influence diagram, and discussed the heuristics involved in this process. The mechanisms discussed in this chapter provide a basis for design of KBSDSS systems.

# Chapter 6

## An Architecture for Knowledge-Based Spatial Decision Support Systems

An expert system, together with powerful facilities for managing and handling spatial data, spatial modelling, formulating and evaluating resource and environmental problem models, constitutes a **knowledge-based spatial decision support system**. In other words, a KBSDSS system is an interactive tool that integrates expert systems, GIS and analytical modelling as well as spatial problem modelling techniques to make it a model-based expert system for resource and environmental decision support. In this chapter, we will present a framework for the design of a KBSDSS system. First, the features of KBSDSS systems are discussed. Then, the knowledge base structure of a KBSDSS system is described, and finally, an architecture for KBSDSS systems is presented.

### 6.1 Characteristics of KBSDSS Systems

As discussed in Chapter 1, KBSDSS systems extend the capabilities of SDSS systems. In contrast to a conventional SDSS system, a KBSDSS system has three main distinct characteristics:

- knowledge-based support for spatial problem solving;
- structuring and representation of spatial problems;
- integration of GIS, analytical and rule-based models.

### **6.1.1 Knowledge-based support for spatial problem solving**

A KBSDSS system provides the user with a substantial amount of domain-specific knowledge, model knowledge, meta-data and other knowledge. The knowledge within a KBSDSS system can be used to guide the formulation and the evaluation of a spatial problem model for a specific resource and environmental problem, and provide guidance in employing different types of models (GIS, analytical and rule-based models) to solve the problem. Thus, a KBSDSS system should enable the user to exploit the power of spatial problem analysis and modelling, GIS modelling, analytical modelling and rule-based modelling in a relatively simple and fast way. It can provide explanations during the spatial problem solving process, and guide the user in the task of capturing the essential elements of a spatial problem, incorporating them into a spatial problem model, which can be automatically evaluated by access to the database and knowledge base, and by integrating different types of models.

### **6.1.2 Structuring and representation of spatial problems**

A KBSDSS system can structure a spatial problem and present the complex interactions and interrelationships between the problem elements in a comprehensible manner. It does this by means of a spatial influence diagram. As discussed in Chapter 4, spatial influence diagrams provide an effective tool for formulating, structuring and summarising spatial problems. A spatial influence diagram may also identify those factors in a problem solving situation that are of concern and whether a factor indicates a desirable or undesirable outcome. For example, in the land use planning problem shown in Figure 4.1, the decision maker is mainly concerned about the proximity to roads, proximity to nature conservation areas, slope, aspect and visual

exposure to water. He or she may prefer the sites for development that have gentle slopes, good view of water and easterly aspect, and are close to roads but not in the designated nature conservation areas. Thus, such a formal representation of a spatial problem model incorporates expert domain knowledge, a description of the decision-maker's circumstances, and a direct representation of the decision maker's knowledge and preferences for the problem solving approach. The spatial influence diagram-based structuring also implies a solution strategy. The reasoning behind the solution to the problem is explicit and can be reviewed by the decision maker. Formal representation, explicit and precise description of the spatial problem as well as integration of knowledge about the decision-maker's particular preferences and situation into a solution are distinct advantages of a KBSDSS system over an SDSS system.

### **6.1.3 Integration of GIS, analytical and rule-based models**

KBSDSS systems can aid decision makers by providing realistic and useful information derived through effective incorporation of different types of models, including GIS, analytical and rule-based models. As we argued in Chapter 1, for most decision problems in resource and environmental planning, there is a great need for different models to analyse and support decisions. These models need to be integrated automatically to make the system more user-friendly. Current integration approaches either require that the user has a lot of knowledge about implementation details or limit the flexibility of the system. A KBSDSS system uses spatial influence diagrams as an abstraction to integrate GIS, analytical and rule-based models logically, and evaluate spatial problem models by integrating them automatically. Integration of models using spatial influence diagrams is carried out at the logical level. That provides flexibility for a KBSDSS system to integrate existing models to solve specific problems according to the decision maker's perspective. In other words, it enables a KBSDSS system to have generalised and flexible model management capabilities, including model integration and formulation. The mechanism for formulation of spatial influence diagrams discussed in the previous

chapter, plus the extensive knowledge encoded in the system, provides such capabilities.

## 6.2 The Knowledge Base Structure of KBSDSS Systems

The design of a KBSDSS system using knowledge-based techniques requires, as a first step, the development of a knowledge base. The knowledge base contains the knowledge needed for the reasoning process, which follows an expert's line of reasoning through the stages of problem solving. As discussed in Chapter 5, the knowledge required for formulation and evaluation of spatial influence diagrams includes domain knowledge, model knowledge and meta-data. Besides the knowledge above, other knowledge is needed in order to fulfil a spatial problem solving task. We may divide the knowledge in the knowledge base of a KBSDSS system into at least five broad categories: domain knowledge, model knowledge, utility program knowledge, meta-data and process knowledge (Figure 6.1).

### **Domain knowledge**

Domain knowledge contains the knowledge and expertise for finding solutions to spatial problems in a particular problem domain. Most domain-specific knowledge concerns the evaluation of each domain variable relevant to the problem domain. Thus, domain knowledge is largely encoded in the form of domain variable frames. The information associated with each domain variable frame indicates the variables that directly influence it (or its direct predecessors), and its attribute relation and spatial relation with these direct predecessors. The attribute relation between the domain variable and its predecessors is represented by an analytical model or a rule-based model. The spatial relation between the domain variable and its predecessors is represented by a GIS model.

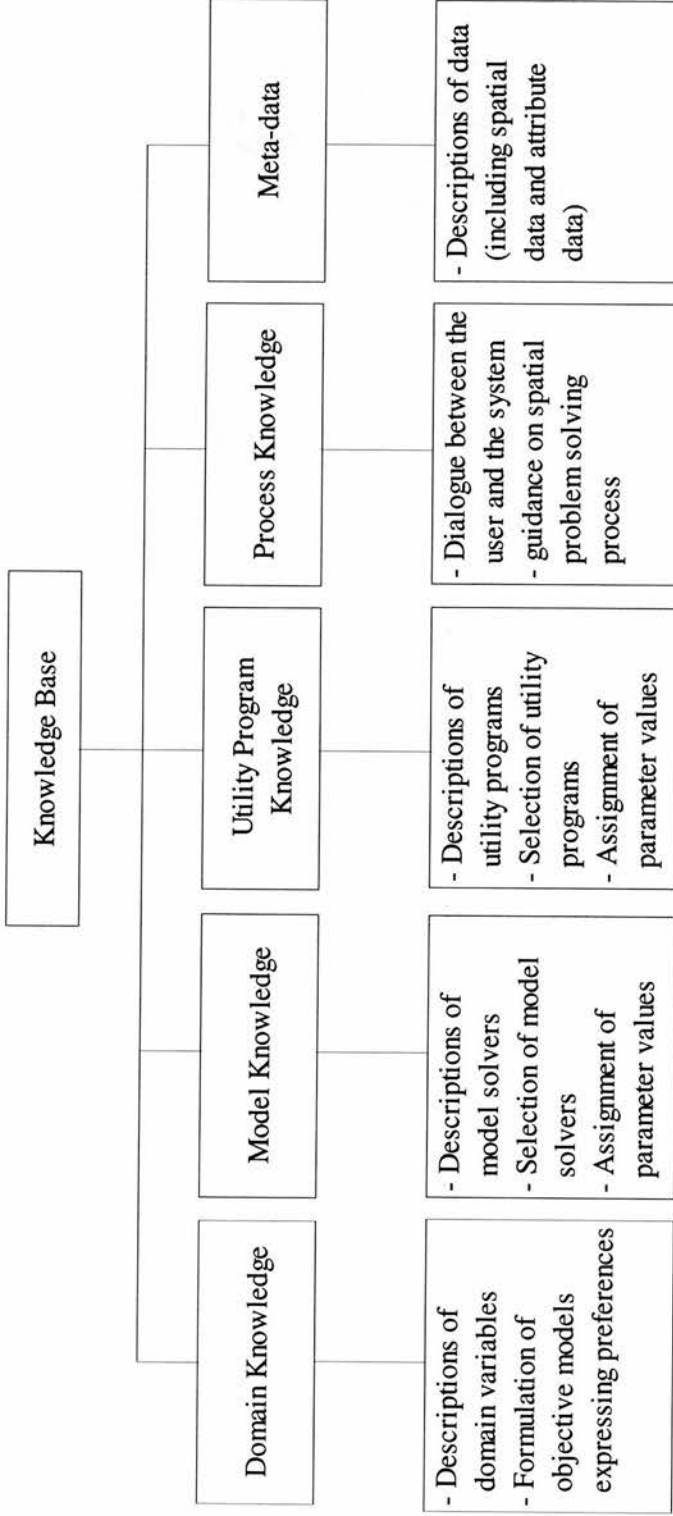


Figure 6.1 Knowledge base structure



In addition to domain knowledge expressed in domain variable frames, a KBSDSS system also contains domain knowledge regarding preferences and circumstances that decision makers could have within a certain decision-making context. This kind of domain knowledge expresses the possible decision-making circumstances and preferences in the form of objective models. An objective model is a model for evaluating the value node or the goal variable in a spatial influence diagram. It is composed of analytical procedures, GIS analysis functions or rule bases that are required for evaluation of the value node. Each objective model accounts for a set of attributes, which are indeed denoted by the direct predecessors of the goal variable. They are formed by indicating what attributes are relevant to the problem in hand. For example, one user may specify the five attributes to be considered in selecting development areas as shown in Figure 4.1. Then, an objective model will be formed accounting for the proximity to roads, proximity to nature conservation areas, slope, aspect and visual exposure to water. Another user may only be concerned about the proximity to roads and slope. In this case, another objective model can be formed accounting for only two attributes. Thus, objective models are preference models representative of decision-maker's preferences for the particular problem solving approach. After an objective model is formed, a particular spatial influence diagram can be formulated, and then evaluation of a spatial influence diagram starts. The domain knowledge regarding preferences and circumstances involves knowledge for forming objective models according to the decision-maker's choices of attributes for evaluating the goal variables of spatial problems. It is used interactively with the decision maker to form an objective model for a specific spatial problem and incorporate the decision-maker's knowledge and preferences for the problem solving into the overall analysis.

### **Model knowledge**

The model knowledge serves to assist the user with the use of different types of models (GIS, analytical and rule-based models) to solve a particular spatial problem. Model knowledge provides meta-information about models and model solvers for

implementing the models. It involves descriptions of model solvers, selection of appropriate model solvers, assignment of the relevant parameter values, and scheduling of execution if two models are needed to evaluate the same node (see Chapter 5).

Model solvers are described using frames. The solver frame contains a solver name (unique), the model type it corresponds to, the tool on which it is executed, its required data model and data structure and a list of parameters that are in the required calling sequences. Figure 5.2 shows a frame representing a solver *union* for the GIS model *overlay*.

Models are specified for each node (except border nodes) in spatial influence diagrams. A model conceptually corresponds to a solver. When evaluating a node with an attached model, the system has to select the appropriate solver to implement the model. The process of solver selection has been described in Chapter 5.

For successful implementation of model solvers, it is important to assign appropriate values to their required arguments (or input parameters) by obeying the calling sequences, the data model and data structure or data type. Model knowledge provides implementation details about all model solvers within the system and rules for assignment of parameter values as described in the previous chapter.

### **Utility program knowledge**

During the evaluation process of a spatial problem, some utility programs may be needed. The utility programs are used to convert the types of data structures, display spatial data in maps and so on. They are also represented by means of frames. The structure of a frame for a utility program is similar to the structure of a solver frame, but without the model type slot. Utility programs are selected and called automatically when they are needed, but they are invisible to the user. The knowledge about selection and execution of utility programs is mostly coded in rules, which decide when and where to call an utility program during the spatial modelling process. Just like model solvers, heuristic rules are used to assign parameter values to selected utility programs for their implementation (see Chapter 5).

## **Meta-data**

The meta-data is knowledge about data. The type of information that is included has been discussed in detail in Chapter 5. It includes name, data model and data structure, geographical feature type (point, line, area and surface) if it is a map data layer, scale if it is a map data layer, definitions of attributes, and data source. The meta-data can be represented using a frame, as shown in Figure 5.4.

## **Process knowledge**

Process knowledge refers to knowledge which provides support during the spatial problem solving process. Generally, the process of spatial problem solving using a KBSDSS system can be typically described as one consisting of specifying a problem, determining preferences (an objective model), formulating a spatial problem model, evaluating the formulated spatial problem model, and generating a solution to the specified problem. Process knowledge is used to guide the actual successful execution of these steps and provide helpful messages in the course of consultation. The help messages provide information about the current modelling process, inform the next modelling operations, and give explanations of GIS and spatial modelling terminology. Thus, the process knowledge can give advice on what to do next, direct the modelling process with effective guidelines, help the user understand the rationale behind the modelling, and control the dialogue between the user and the system.

## **6.3 An Architecture for KBSDSS Systems**

A KBSDSS system can be considered a class of expert systems with an inference engine based on the spatial influence diagram-based mechanisms. Thus, an

architecture different from traditional SDSS architectures (Figures 1.1 and 1.2) must be used to develop a KBSDSS system.

The underlying architecture of KBSDSS systems derives from that of the intelligent decision support systems (IDSS) in management science. Fedorowicz and Williams (1986) described a generic IDSS system, which consists of a user interface, problem processor, knowledge base and model base (Figure 6.2). Our KBSDSS system is composed of a user interface, query processing subsystem, modelling subsystem, problem processor, knowledge base, and back-end subsystem (Figure 6.3).

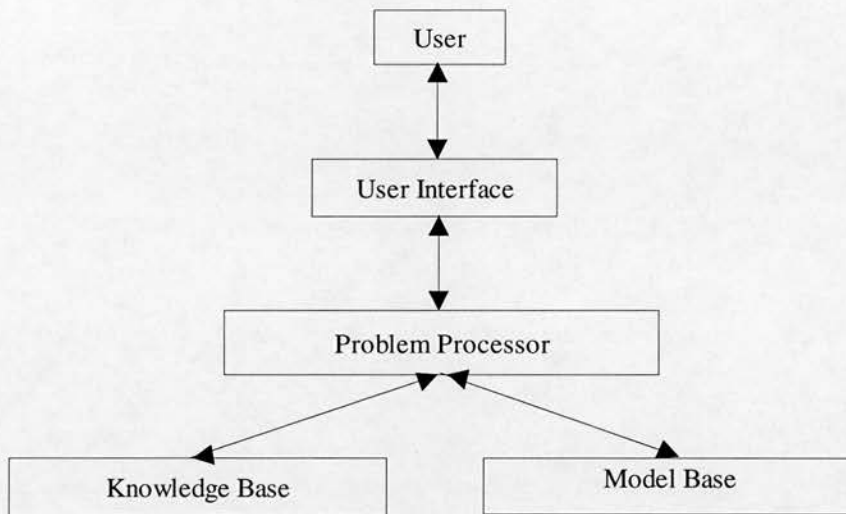


Figure 6.2 Components of an IDSS System (reproduced from Fedorowicz and Williams (1986))

The query processing subsystem is developed to accept user queries, display modelling results and provide on-line help. The subsystem allows the user to retrieve existing data from the database together with data derived during the modelling process, and display them in the forms of maps, images or tables. It also allows the user to retrieve the meta-information about data and models.

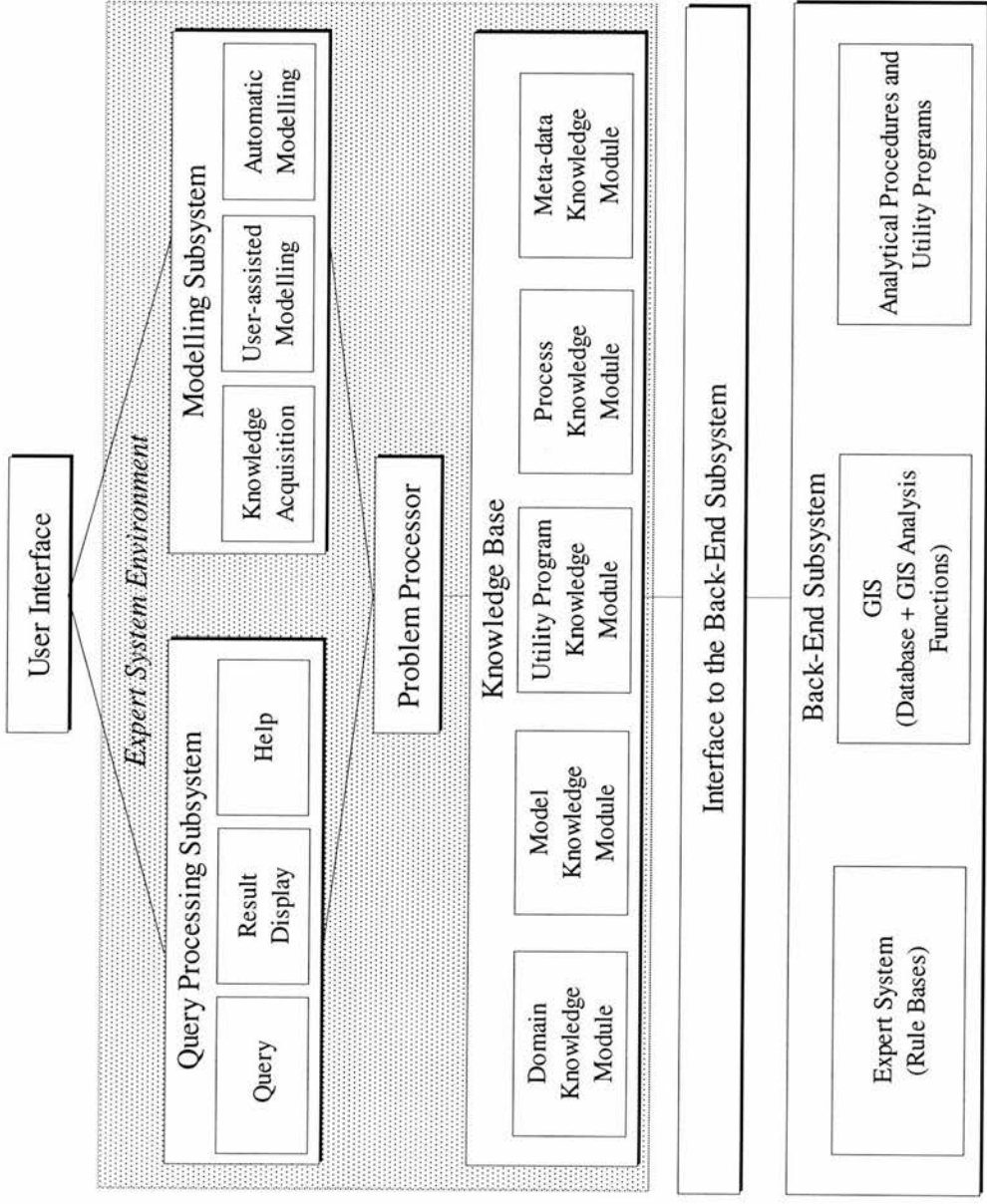


Figure 6.3 Architecture for KBSDSS systems

The modelling subsystem is designed to support the user in developing useful models. The subsystem supports knowledge acquisition, user-assisted modelling and automatic modelling. The system acquires relevant knowledge through the knowledge acquisition module, such as meta-data for new data sets, meta-information about new models, or structures for spatial problems defined by the user. A spatial influence diagram editor can be used to guide the user to capture, elicit and represent the problem structure and develop the problem model.

Within the scope of problem solving for the classes of problems designed for the system, the system can elicit important features of the given situation from the decision maker and can then incorporate these features into a spatial problem model (a spatial influence diagram) specially developed to address the problem at hand as specified by the decision maker. This process is called **automatic modelling**, and it is handled through the automatic modelling module in the modelling subsystem. Note that a **class of problems** refers to a set of problems having some degree of similarity among them. The problems in a class should share a common domain. The system begins the process of automatic modelling by examining the goal for the problem solving. A basic spatial influence diagram is then formulated by building an objective model, which consists of the goal variable and its direct predecessor variables. Afterwards, the basic spatial influence diagram is expanded by rules embodying the domain knowledge, following the algorithm described in Figure 5.6. Further nodes are added into the starting basic spatial influence diagram according to their applicability to the problem. Through such a process, a detailed complete spatial influence diagram is formed.

After a spatial problem model is created, the system allows the user to modify the model. The modification of existing models is termed **user-assisted modelling**. User-assisted modelling also includes creation of new spatial problem models using spatial influence diagrams by the user, addition of new analytical, rule-based and GIS models, and addition of new data sets or data layers in order to meet the requirements of particular analyses.

The problem processor plays a central role in a KBSDSS system. It performs two main functions: inference and control. It is used to drive the evaluation process

for spatial problems, and to integrate the knowledge base, database and different types of models for spatial problem solving. The problem processor accepts a spatial problem model from the modelling subsystem and then evaluates it. The evaluation is based on the algorithm shown in Figure 5.7. This involves the evaluation of each node within the spatial influence diagram through invocation of model solver frames and utility program frames (if necessary) describing model solvers and utility programs provided by the back-end subsystem, database calls and other pertinent knowledge, and assignment of parameters values to the model solvers and utility programs. Actual successful completion of these processes will trigger execution of the selected model solvers and utility programs, and each node gets an outcome. When the node denoting the decision maker's objective (the value node) gets an outcome, a solution to the problem is obtained. The whole process may repeat for the same problem by modifying the problem parameters in the problem model. By comparing the outcomes of different scenarios, the user may make a “best” decision.

The problem processor can also accept commands translated from the request and actions issued by the user through the query processing subsystem and modelling subsystem, execute these commands, control access to the data base and knowledge base, execute models, retrieve knowledge from the knowledge base and make inferences.

The knowledge base is structured into five parts according to the discussion in the previous section. We call each part a **knowledge module**. Thus, there are at least five knowledge modules in the knowledge base of a KBSDSS system. They are the domain knowledge module, the meta-data knowledge module, the model knowledge module, the utility program knowledge module and the process knowledge module. The modular structure of the knowledge base makes the system adaptive to different back-end software tools and different problem domains. It can easily be seen that the inference and control mechanism of the problem processor is not only independent of the specific back-end software tools used for implementation of models but also of the problem domain. To adapt the system to new back-end software tools, only the model and utility program knowledge modules have to be changed. To adapt the

system to a new problem domain, the domain knowledge module has to be changed, and the process knowledge module has to be appropriately adjusted also.

The back-end subsystem consists of the software tools employed in the system for implementing different types of models and utility programs. It contains three basic separate modules: an expert system, a GIS software package (with a database, and spatial data handling and analysis functions) and a collection of analytical procedures and utility programs. The expert system here is a tool for implementing the rule-based models. The subsystem provides appropriate tools for implementing GIS, analytical and rule-based models and necessary utility programs defined in the system. The architecture places no restrictions upon the number of software tools in the back-end subsystem. Different software systems, such as statistical packages, can be added, if required, in order to meet the needs of the problem solving task. When new software tools are added, the meta-information about the model solvers and utility programs they support should be added to the system through the knowledge acquisition module in the modelling subsystem or by modifying the model and utility program knowledge in the knowledge base.

The query processing subsystem, modelling subsystem, problem processor and the knowledge base are built within an expert system environment. The interface to the back-end subsystem provides the link between the expert system environment and the back-end subsystem. It handles inter-process communication between the expert system environment and the back-end subsystem, triggers the immediate execution of the selected model solvers or utility programs, and issues the database calls and vice versa.

The user interface interacts directly with the user. Through the user interface, the user can access the query processing subsystem and modelling subsystem.



## 6.4 Summary

KBSDSS systems can provide knowledge-based support for spatial problem solving, facilitate the representation and formulation of spatial problems according to the decision maker's preferences or situations, and evaluate spatial problems automatically by integrating GIS, analytical and rule-based models. In this chapter, we proposed an architecture for KBSDSS systems, which consists of the following components: a user interface, query processing subsystem, modelling subsystem, problem processor, knowledge base and back-end subsystem. These components can interact with each other and provide the capabilities described above. Later chapters will discuss the implementation of the architecture and the mechanisms of the KBSDSS system.

## Chapter 7

# A KBSDSS Development Environment: Integrating ARC/INFO, HARDY and CLIPS

A KBSDSS system has to integrate closely a variety of techniques and technologies, including expert systems, GIS, analytical modelling and spatial problem modelling, to support decision making effectively. It seems unlikely that monolithic approaches to the development of KBSDSS systems will be successful. Development of KBSDSS systems by writing the whole code for the desired functionality from scratch is a daunting task. Even for development of an SDSS system, such an implementation has not been attempted (Djokic 1993). A more feasible approach is to develop a KBSDSS system through the heterogeneous integration of different software tools. The architecture of KBSDSS systems presented in the last chapter is consistent with such an approach. The underlying rationale is that KBSDSS system development can be expedited through the integration of heterogeneous systems, each representing the best available technology in a particular application area.

This chapter develops a KBSDSS development environment to implement the proposed architecture of KBSDSS systems through integrating HARDY (a hypertext diagramming tool), CLIPS (an expert system development tool) and ARC/INFO (a GIS software package). HARDY provides a diagramming tool and a flexible user

interface. CLIPS provides an expert system environment for developing the query processing subsystem, modelling subsystem, problem processor and knowledge base of a KBSDSS system. At the same time, CLIPS is also used as a component of the back-end subsystem to develop and implement rule-based models. ARC/INFO is used as a back-end to manage and maintain spatial data, develop and implement GIS, analytical models and some utility programs. It also provides facilities for display of spatial data in maps and graphs.

As has been seen, model solver and utility program frames are a higher level abstraction for representing solvers and utility programs. However, the actual implementation of the model solvers and utility programs is carried out by different software tools, here ARC/INFO and CLIPS. In this chapter, we explore the possibility of creating highly integrated KBSDSS systems through the loose coupling of existing software tools, focusing on the inter-process communication between the different software systems, data exchange between ARC/INFO and CLIPS, and the development of model solvers and utility programs.

## 7.1 Overall Structure of the Development Environment

The conceptual architecture of the environment is shown in Figure 7.1. The overall environment consists of three components: HARDY, CLIPS and ARC/INFO.

The diagramming tool, HARDY, is used to display, create, edit and maintain spatial problem models in the form of spatial influence diagrams, and interpret and export the diagrams into the CLIPS representations. It is also used to build a user interface.

The expert system tool, CLIPS, is used to represent and maintain different types of knowledge bases and perform inferences with these to drive the process of spatial problem solving and generate advice in a decision situation. It provides an expert system environment for building the query processing subsystem, modelling subsystem, problem processor and knowledge base in a KBSDSS system. CLIPS is

also taken as a component of the back-end subsystem to build and make inferences on rule bases, each representing a rule-based model.

ARC/INFO is a generic GIS software system for managing, modelling and analysing spatial data. It is used to acquire data describing the spatial entities and their associated attributes, build databases, develop and execute simple analytical models, GIS models or GIS operations such as overlay, buffering and measurements, and some utility programs, such as spatial data transformation, graphic and tabular display, etc. It also prepares data through a series of processing steps for analytical and rule-based models and displays the modelling and reasoning results.

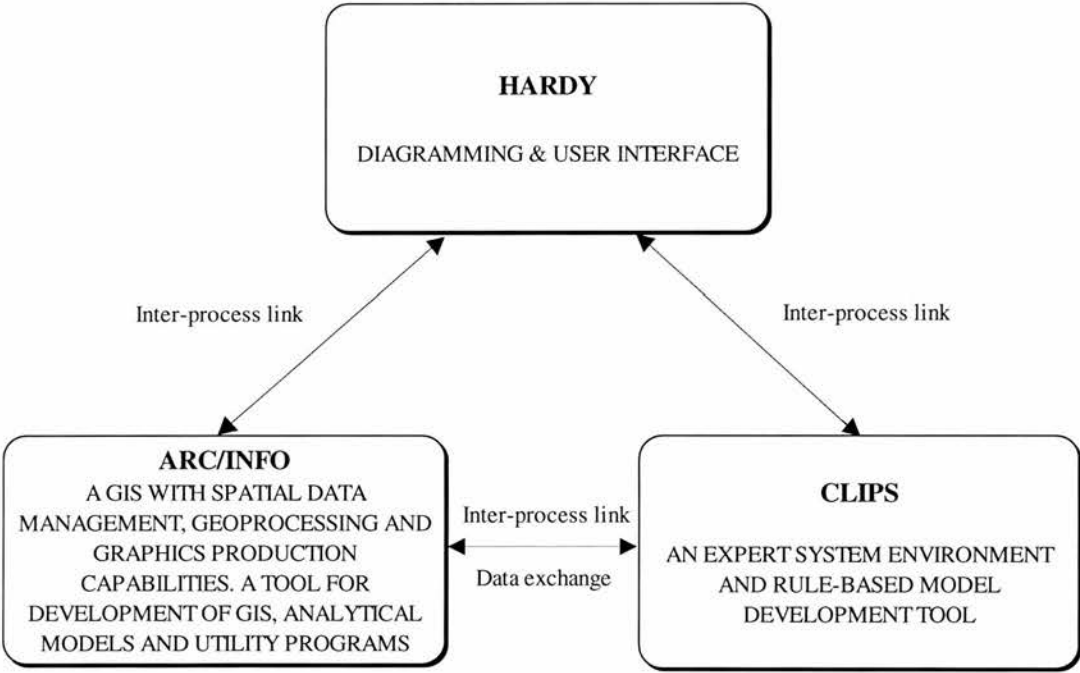


Figure 7.1 Conceptual architecture of a KBSDSS system development environment

This development environment is built on UNIX systems under X.

It is important that the three components operate together to present the user with the appearance of a single system, and that they mutually support one another's operation. For example, HARDY produces spatial influence diagrams and calls CLIPS to evaluate them. CLIPS calls HARDY to interpret spatial influence diagrams and ask the user for data for reasoning. HARDY calls ARC/INFO to allow the user to perform GIS operations. ARC/INFO asks the user for data through HARDY. CLIPS issues a command to activate ARC/INFO operations or get data from the INFO database (a module of ARC/INFO). ARC/INFO performs GIS operations and creates data files in an interchange format read by CLIPS. CLIPS in turn performs reasoning on the data from data files and writes reasoning results to other data files readable by ARC/INFO. The data files are transferred back to ARC/INFO for display or further GIS modelling.

## 7.2 System Tools

HARDY, CLIPS and ARC/INFO are three basic tools integrated in the KBSDSS development environment.

### 7.2.1 HARDY

HARDY (**H**ypertext-based **A**nalysis and **R**eusable **D**esign **s**ystem) is a diagramming tool, which supports a variety of diagram types and hypertext-style expansion cards (Smart 1993a; Smart and Harrison 1992). HARDY has three main features:

(1) It is a flexible hypertext diagramming tool.

HARDY allows the user to build a diagram type, such as a dataflow diagram type, or use an existing diagram type already defined. The user may then choose a diagram type and produce diagrams, which consist essentially of a number of nodes

linked by arcs. HARDY supports nodes and arcs in several different formats (e.g. circles, squares and ellipses). When drawing a diagram, the values for attributes specified in the diagram type may be entered. The facilities of HARDY are well suited to produce spatial influence diagrams.

To create a diagram type, HARDY provides a Diagram Type Manager. A diagram type definition consists of the diagram type name, a list of node definitions and a list of arc definitions. The HARDY Diagram Type Manager makes it easy to generate nodes and arcs with appropriate characteristics for a particular modelling approach (Kingston 1993). For example, we may use the HARDY Diagram Type Manager to define a diagram type for spatial influence diagrams, named *SpatialID* (Figure 7.2). This diagram type defines three types of nodes: *value*, *chance* and *border*, and one arc type for links between nodes: *arc*, following the definitions defined in Chapter 4. Value nodes are drawn as rounded rectangles, chance nodes are drawn as circles and border nodes are drawn as ellipses. Attributes of each type of nodes can be defined using the node type editor within the Diagram Type Manager. In Figure 7.2, the attributes of the chance node type are being defined. These attributes are actually the seven categories of information for describing a node as described in Chapter 4, including the name, node-type, predecessors, successors, attribute-relation, spatial-relation and associated-data. In a similar way, the node type editor can be used to define the attributes of the value and border node types. Whenever a node is created, it automatically has the attributes defined in the diagram type *SpatialID*.

After diagram types are created, the user may select a particular type and draw diagrams. If one chooses the diagram type for spatial influence diagrams, the relevant characteristics of those described above can be drawn. For example, Figure 7.3 is a version of Figure 4.1 drawn within the HARDY environment. When drawing a diagram, values for attributes defined in the diagram type may be entered. Chapter 8 will give more details about how to assign values to the attributes defined in a diagram type.

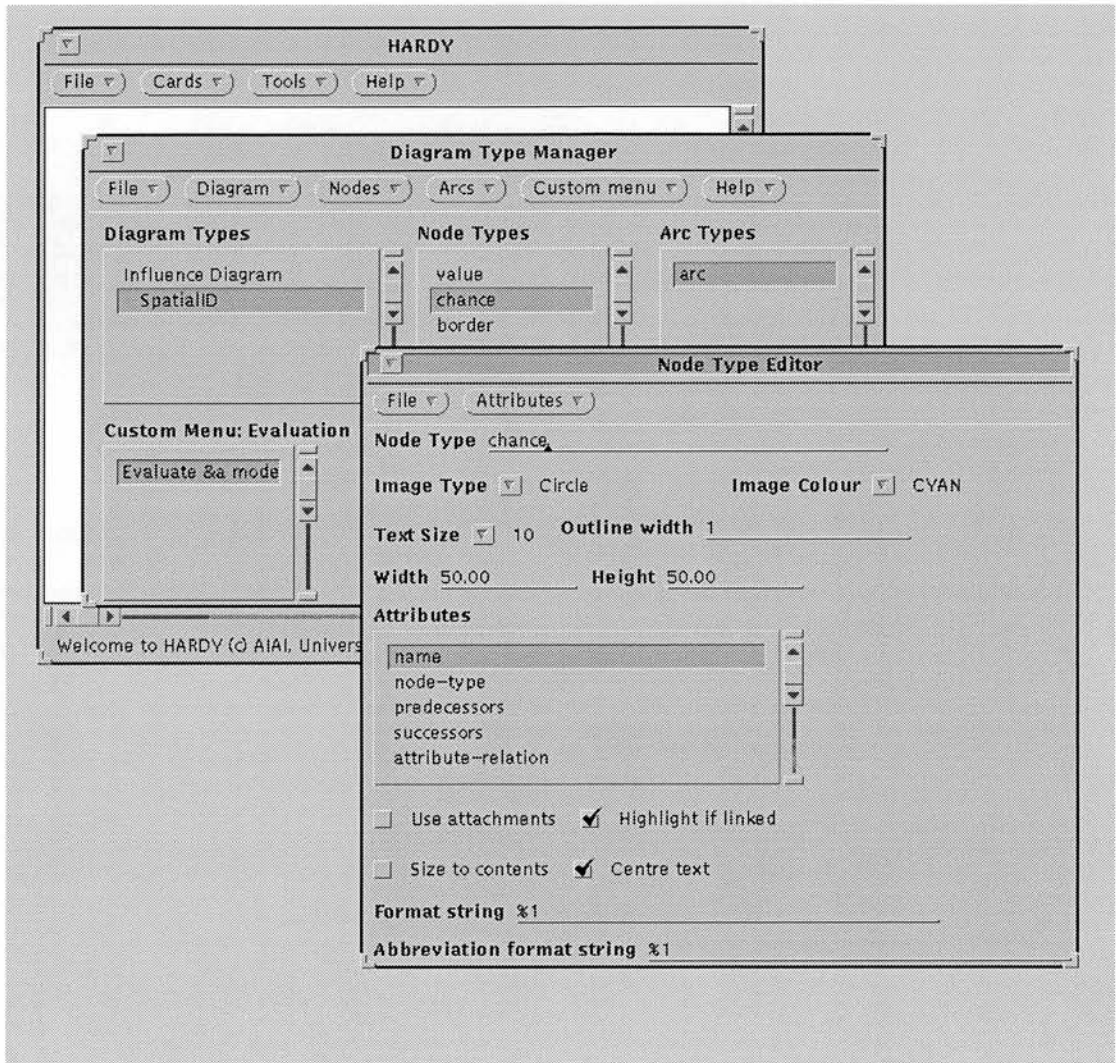


Figure 7.2 HARDY Diagram Type Manager

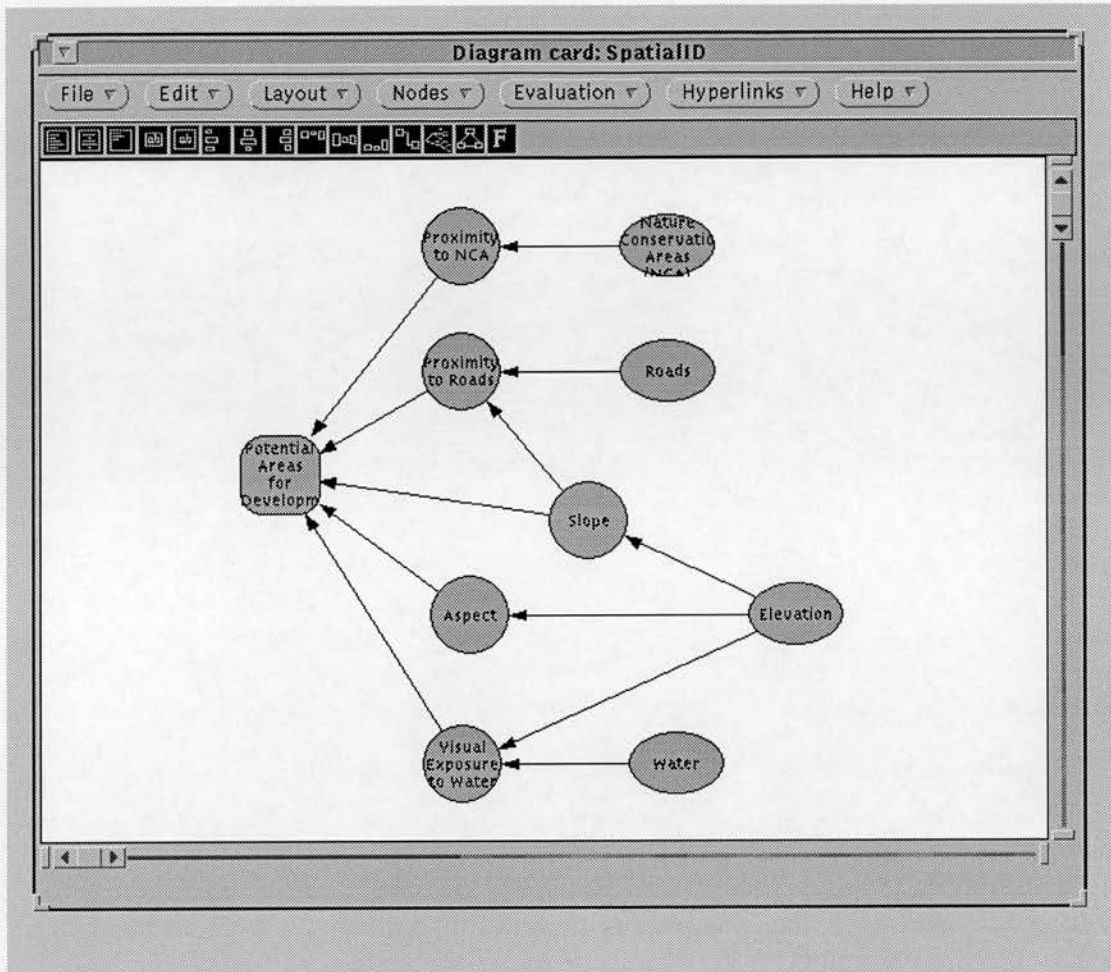


Figure 7.3 A spatial influence diagram in HARDY

In HARDY, each diagram has its own window, as shown in Figure 7.3. A diagram window is also called a **diagram card**. Every card has a file associated with it. After a diagram is produced within a diagram card, it can be saved in a file for later use. Diagram cards can be linked together using the HARDY hypertext facilities to form a tree or network, or to form a hierarchy with a node on a high-level diagram representing an entire diagram at a lower level through expansion cards (Smart 1993a). In addition to diagram cards, HARDY supports text and hypertext cards. These cards can be used document a diagram, display and edit text files, build



hypertext links with other cards and items (nodes and arcs in a diagram card, or blocks of text in a hypertext card) associated with a card (Smart 1993a).

(2) It is portable.

Another feature of HARDY is its portability. HARDY uses a class library called wxWindows to isolate the application graphical user interface code from the hardware platform (Smart 1993b). Thus, the applications developed using HARDY can be X-windows applications as well as Windows 3 applications.

(3) It has an embedded expert system tool.

HARDY is tightly coupled with CLIPS. It has the CLIPS programming language as its built-in language. There are new HARDY-specific functions which may be called from CLIPS. The specialised HARDY CLIPS functions manipulate various structures essential to HARDY and build customised graphical user interfaces. The diagrams of a certain type can be produced and manipulated automatically by programming using the HARDY built-in programming language. In addition, the diagram representation produced by HARDY can be directly translated into the CLIPS representation, fed into CLIPS and processed by the CLIPS inference engine. As will be seen, the KBSDSS has benefited greatly from this HARDY feature.

### **7.2.2 CLIPS**

CLIPS, the **C** Language **I**ntegrated **P**roduction **S**ystem, is a complete environment for developing rule and/or object based expert system programs which are specifically intended to model human expertise or knowledge (NASA 1993). It provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented and procedural. Rule-based programming allows knowledge to be represented as heuristics, or “rules

of thumb”, which specify a set of actions to be performed for a given situation. Object-oriented programming allows complex systems to be modelled as modular components (which can be easily reused to model other systems or to create new components). The procedural programming capabilities are similar to capabilities found in languages such as C and Pascal. One can develop an expert system using only rule-based programming, only object-oriented programming, only procedural programming, or combinations of the three (NASA 1993). Here, we only discuss the major rule-based programming aspects of CLIPS.

CLIPS has three basic components (Figure 7.4):

- the **working memory** (also called the **fact-list**): it is the global memory for data;
- the **knowledge base**: contains knowledge encoded in rules;
- the **inference engine**: makes inferences by specifying which rules are to be satisfied and in what order. An **agenda** is a prioritised list of rules whose patterns are satisfied by facts in the working memory.

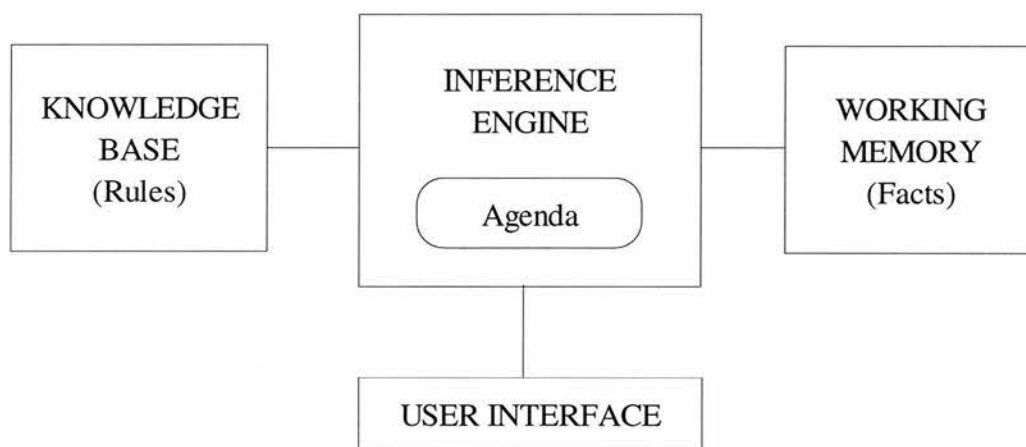


Figure 7.4 The components of CLIPS (after Giarratano and Riley, 1989)

The following is a brief description of the features of CLIPS.

(1) Knowledge representation and data abstraction

Production rules are the main form of knowledge representation within CLIPS. We may recall that a rule is composed of an **antecedent** and a **consequent**. The antecedent of a rule is also referred to as the **left-hand side (LHS)** of the rule. The consequent of a rule is also referred to as the **right-hand side (RHS)** of the rule. The antecedent of a rule is a set of **conditions** which must be satisfied for the rule to be applicable. In CLIPS, the conditions of a rule are satisfied based on the existence or non-existence of specialised facts in the fact-list. One type of condition which can be specified is a **pattern**. The consequent of a rule is a set of **actions** to be executed when the rule is applicable. The following is the rule “crop-yield” described in Chapter 2, which is now expressed in CLIPS syntax:

```
;Rule header
(defrule crop-yield
  ;Patterns
  (precipitation light)
  (soil sandy loam)
  (climate hot)
;THEN arrow
=>
  ;Actions
  (assert (crop-yield good)))
```

In CLIPS, the entire rule must be surrounded by parentheses. Also, each of the patterns and actions of the rule must be surrounded by parentheses. Comments begin with a semicolon and continue until a carriage return. They are ignored by CLIPS. The header of the rule starts with a *defrule* keyword. Following *defrule* must be the name of the rule. In the rule above, the rule name is crop-yield. After the rule header are zero or more patterns. In the “crop-yield” rule, there are three patterns: (precipitation light), (soil sandy loam) and (climate hot). Each

pattern consists of one or more fields. For example, the pattern (precipitation light) contains two fields: precipitation and light. The arrow => represents the beginning of the THEN part of an IF-THEN rule. The LHS of the rule precedes the =>. The RHS is placed after =>. The last part of a rule is the list of actions that will be executed when the rule fires. In our example, the action is to assert the fact (crop-yield good). The *assert* command is used to add a new fact to the fact-list.

**Facts** are data or information with which CLIPS can reason. Generally, facts in CLIPS consist of a relation name followed by zero or more slots and their associated values. The following is an example.

```
(land_feature (land_ID 44)
              (slope 0.15)
              (soil_texture SANDY LOAM)
              (erosion SLIGHT)
              (soil_depth 55)
              (flooding NEVER))
```

The entire fact and every slot are enclosed in matching left and right parentheses. The symbol `Land_feature` is the relation name. The fact has six slots: `land_ID`, `slope`, `soil_texture`, `erosion`, `soil_depth` and `flooding`. The value of the `land_ID` slot is 44, the value of the `slope` slot is 0.15, the value of the `soil_texture` slot is `SANDY LOAM`, and so on.

Before facts can be created, the list of valid slots for a given relation name must be defined using the *deftemplate* construct. The *deftemplate* construct is analogous to a record structure or a frame. The following example is a *deftemplate* for describing the “land\_feature” fact above.

```
(deftemplate land_feature
  (slot land_ID)
  (slot slope)
  (multislot soil_texture)
  (slot erosion)
  (slot soil_depth)
  (slot flooding))
```

Slots of a fact that have been specified with the *multislot* keyword in their corresponding deftemplate are allowed to contain zero or more values, such as the *soil\_texture* slot in the example above. Facts with a relation name that has a corresponding deftemplate are called **deftemplate facts** or **non-ordered facts**. CLIPS also supports **ordered facts**. An ordered fact consists of one or more fields enclosed in matching left and right parentheses. For example, the facts (precipitation light), (soil sandy loam) and (climate hot) are ordered facts. Ordered facts do not have a corresponding deftemplate.

Facts may be added to the fact-list (the current list of facts) using the *assert* command or removed from the fact-list using the *retract* command through an explicit user interaction or as a CLIPS program executes.

CLIPS can store values using **variables**. Variables in CLIPS are written in the syntax of a question mark followed by a symbolic name, such as ?node, ?phase, ?status and ?data. The *bind* command is used to assign a value to a variable. For example, the following command will assign 100 to the variable ?cost.

```
(bind ?cost 100)
```

CLIPS also supports such features as global variables, multi-field variables, wildcards, a set of predicates for constraining variables, representation of logical dependencies, the negation of LHS conditions, procedural-programming constructs as RHS actions, definitions of user-defined functions and so on (NASA 1993).

## (2) Inference mechanism

CLIPS is a forward-chaining data-driven system. A forward-chaining system often involves a large set of data and numerous rules, and attempts to match all the data against all the antecedents of all the rules. CLIPS uses a very efficient algorithm, the Rete algorithm (Forgy 1985), for matching facts against the antecedents in rules to determine which rules have their conditions satisfied. If all the patterns of a rule match facts, the rule is activated and put on the agenda. As described in Chapter 2, an inference engine operates in cycles. The tasks of a cycle for CLIPS can be described in pseudo-code terms as follows (Giarratano and Riley 1989):

WHILE not done

**Conflict Resolution:** If there are activations, then select the one with highest priority, else done.

**Act:** Sequentially perform the actions on the RHS of the selected activation. Those which change working memory have immediate effect in this cycle. Remove the activation which has just fired from the agenda.

**Match:** Update the agenda by checking if the LHS of any rules are satisfied. If so activate them. Remove activations if the LHS of their rules are not satisfied any more.

**Check for Halt:** If a halt action is performed or break command given, then done.

END-WHILE

**Accept a new user command**

Multiple rules may be activated and put on the agenda during one cycle of rule execution. The inference engine selects one of the rule activations for execution. CLIPS determines the order of precedence of reasoning by the **salience** of the rule and the current **conflict resolution strategy**. The agenda acts similarly to a stack. That is, the most recent activation placed on the agenda is the first to fire. Salience

allows more important rules to stay at the top of the agenda regardless of when the rules were placed on it. Rules of lower salience are pushed onto the agenda below all rules of higher salience. If rules have equal salience, then the current conflict resolution strategy is used to determine an order of rule execution. A single rule (along with several other rules) can produce multiple activations. In this case, the rule is arbitrarily ordered in relation to the other rules with which it was activated.

CLIPS provides several conflict resolution strategies, including **depth**, **breadth**, **simplicity**, **complexity**, **lex**, **mea** and **random** (NASA 1993). The default strategy is depth. Using the depth (depth-first) strategy, newly activated rules are placed above all rules of the same salience. For example, given that fact-a activates rule-1 and rule-2, and fact-b activates rule-3 and rule-4, then if fact-a is asserted before fact-b, rule-3 and rule-4 will be above rule-1 and rule-2 on the agenda. However, the position of rule-1 relative to rule-2 and rule-3 relative to rule-4 will be arbitrary. For more information about other conflict resolution strategies available in CLIPS, refer to NASA (1993) or Appendix A.

### (3) Modular construction of knowledge bases

CLIPS allows a knowledge base to be partitioned and provides support for the modular development and execution of knowledge bases. CLIPS modules allow a set of constructs (such as *defrule* and *deftemplate*) to be grouped together such that explicit control can be maintained over restricting the access of the constructs by other modules. By restricting access to *deftemplate* constructs, modules can permit only certain facts to be seen by other modules. Modules are also used by rules to provide execution control.

In order to partition a knowledge base, the various modules must be defined using the *defmodule* construct. For example, the following is a *defmodule* construct, which defines a module SOLVERS.

```
(defmodule SOLVERS)
```

The name of a module must follow the *defmodule* keyword. The module in which a construct is placed can be specified when the construct is defined. The *deftemplate*, *defrule* and other CLIPS constructs all specify the module for the construct by including it as part of the name. For example, the following construct would be placed in the SOLVERS module.

```
(deftemplate SOLVERS::solver
  (slot name)
  (slot model_name)
  (slot output_type)
  (slot tool)
  (slot data_model)
  (slot data_structure)
  (multislot parameters))
```

Here, `SOLVERS::solver` refers to the solver *deftemplate* construct in the SOLVERS module. The `::` symbol is called the **module separator**. To its right is the name of the construct; to its left is the name of the module. With modules, it is possible to have a construct with the same name in different modules. By default, CLIPS defines a module called the MAIN module. The constructs in the MAIN module may be defined without putting `MAIN::` before their names, such as the “crop-yield” *defrule* and “land\_feature” *deftemplate* constructs defined above.

Normally, the constructs of one module may not be used by another module. However, *deftemplate* constructs (and all facts using that *deftemplate*) can be shared with other modules. A module can export *deftemplate* constructs so that they can be used by other modules and the facts using them can be seen by other modules. A module can also import *deftemplate* constructs from another module in order to use them and all facts associated with them. This allows a knowledge base to be partitioned such that rules and other constructs can only “see” those facts which are of interest to them. Modules that export and import *deftemplates* must use the export and import attributes in their *defmodule* definitions, such as



```
(defmodule SOLVERS
  (export deftemplate solver)
  (import DATA deftemplate data))
```

This definition indicates that the deftemplate “solver” is exported by the SOLVERS module and the deftemplate “data” defined in the DATA module is being imported.

Each module defined in CLIPS has its own agenda. Execution can then be controlled by selecting which module’s agenda is selected for executing rules. When a *run* command is given within CLIPS, the agenda of the module which is the current focus is executed. When CLIPS is initially started or cleared, the current module is automatically the MAIN module. Before a rule executes, the current module is changed to the module in which the executing rule is defined (the current focus). The *focus* command is used to change the current focus.

A modular design of a knowledge base may reduce complexity of knowledge base design and facilitate modification of the knowledge base. In addition, using the *focus* command, the execution of a program can be controlled by implementing a series of separate modules in a certain sequence. It also facilitates the development and execution of rule-based models, as will be seen in Section 7.5. A rule-based model may be seen as a functional independent knowledge base module, having a well-defined purpose.

#### (4) Portability, integration and extendibility

CLIPS was written in C and was designed specially to provide high portability and easy integration with existing or conventional software systems. The development environment is designed to run on any system which supports any C compiler compatible with the ANSI C standard. CLIPS can be embedded within procedural code, called as a subroutine, and integrated with languages such as C and FORTRAN. It can be easily extended by a user through the use of several well-defined protocols.

Moreover, CLIPS has facilities to provide a cross-reference listing of relations, to check correctness of programming style and verify consistency of rules in the knowledge base (NASA 1993).

### **7.2.3 ARC/INFO**

ARC/INFO (ESRI 1991b; Morehouse 1989) is a software package with a large set of spatial operators that manipulate, model, analyse and display geographical data. It stores digital cartographic data in operating system files and enables the user to manipulate attribute information pertaining to map features, by means of a relational type file manager. It is designed in a modular structure and supports a variety of geographical data types. ARC/INFO is a command-driven system. It also provides the ARC Macro Language (AML) for building the application-oriented user interface and grouping ARC/INFO commands to form macros for efficient execution of strings of commands.

#### **(1) Modular Design of ARC/INFO**

The current version of ARC/INFO consists of a number of software modules. They include ARC, GRID, INFO, ARCPLOT, ARCEDIT, MAP LIBRARIAN, TIN, COGO, NETWORK, etc.

ARC is the main program environment in ARC/INFO. It contains commands that start each of the other software modules and a set of vector-based geoprocessing tools that handle the digital cartographic data for geographical features and perform various spatial data processing tasks. GRID is a raster-based geoprocessing toolbox. It provides efficient spatial-storage facilities and a powerful modelling environment for analysing geographical features. GRID utilises the map-algebra spatial language (see Tomlin 1990) and supports cartographic modelling. Both ARC and GRID provide spatial data analysis and manipulation capabilities, such as buffer generation, map overlay, neighbourhood analysis, geometric measurement and so on. While ARC and GRID are geoprocessing toolboxes, INFO is a relational type database

management system that stores and manages attribute data for geographical features. It is used to manipulate and update each feature's attributes by performing logical and arithmetic operations on the rows and columns of the table. ARCPLOT is used for map display and query. It provides tools for drawing maps and graphs. ARCPLOT also provides interactive query functions allowing the user to use maps displayed on the screen to retrieve data directly from the database. More information about the functionality of these and other main ARC/INFO software modules is given in Appendix B.

## (2) Data models supported in ARC/INFO

ARC/INFO adopts a hybrid data model, i.e., locational data are described using the vector or raster data model and attribute data are represented using the relational data model. The basic unit of data storage is the coverage or grid, which may correspond to a single map sheet.

A **coverage** is the data structure used by ARC and some other modules for vector-based geoprocessing (ESRI 1991a; Morehouse 1985). It is a digital version of part of a map sheet layer and generally describes one type of map feature such as roads, land use types, wells or soil units. A coverage contains both the locational data and thematic attributes for map features in a study area. In a coverage, map features are stored as simple points, arcs, or polygons. Coverages use a combined vector-based spatial data model and a relational attribute data model.

A **grid** is the data structure used by the GRID software module (ESRI 1991c). A grid is made up of **cells**. Each cell is a square that represents a specific portion of an area. All cells in a grid are the same size. A grid is similar to an ARC/INFO coverage, representing a spatial variable or theme. Cell values represent themes or values of a spatial variable. Grids use a combined raster-based spatial data model and a relational attribute data model.

In addition to coverages and grids, ARC/INFO supports other data types, including TINs, lattices, images, CAD and DBMS tables (ESRI 1991a). They are described in Appendix C.

### (3) ARC Macro Language (AML)

The most powerful tool provided by ARC/INFO is the ARC Macro Language. AML allows the user to automate frequently performed actions, create user-defined commands and develop graphical user interfaces (ESRI 1991d).

AML programs may be composed exclusively of commands from ARC or any of the other modules. In this way, AML can save time by isolating command sequences that are typed in often, and provide new or inexperienced users with a method for performing complex operations without knowing the specific command sequences required to do so.

AML has three elements: directives, functions and variables.

AML directives are AML commands that perform specific actions or determine the flow of control. An AML directive begins with an ampersand “&”. Directives instruct AML to perform the desired AML operations. The following are two examples of AML directives. The first directive is to run an AML program “giscom.aml”, and the second one is to type the message “Loading files ...” to the terminal.

```
&run giscom.aml
&type Loading files ...
```

AML variables are used to store data or values. The `&setvar` or `&s` directive can be used to assign values to variables. For example, the variable `feature` is assigned a value `line` as follows:

```
&s feature = line
```

To reference a variable, percent signs “%” must be used to surround the name of the variable. For example, the following command is to type a message with the value of the variable `feature` to the terminal.

&type The feature type is %feature%.

AML functions perform the defined operations that return values. An AML function is enclosed in square brackets “[ ]”. Shown following is an example of an AML function, which returns a Boolean value TRUE if the data file “quitgis” exists, or returns a Boolean value FALSE if the data file does not exist.

```
[exists quitgis]
```

AML provides full programming capabilities to create flexible AML programs using the three elements. Similar to command languages such as DCL in the VAX/VMS operating system, and the Bourne and C shells in the UNIX operating system, AML provides facilities for execution of sophisticated actions, including logical branching and loops, variable manipulation, arithmetic and trigonometric operations and program argument transfer (ESRI 1991d). An AML program is simply a text file created using a text editor.

### 7.3 Inter-process Communication

As shown in Figure 7.1, there are three inter-process links between the software components. They are HARDY-ARC/INFO communication, HARDY-CLIPS communication and CLIPS-ARC/INFO communication. CLIPS is embedded in HARDY and is taken as the HARDY built-in language. Thus, HARDY already supports interaction with CLIPS, because the link between them already exists and does not need further development. In addition, HARDY uses the CLIPS programming language, supports most CLIPS functions and extends CLIPS functionality. HARDY and CLIPS are running at the same time in a single process. Therefore, we view HARDY and CLIPS as a single system. To develop our KBSDSS systems, we use HARDY as an extension to CLIPS with diagramming and graphical

user interface building capabilities. In this view, the HARDY-ARC/INFO and CLIPS-ARC/INFO inter-process communications operate virtually through one link, i.e. the HARDY-ARC/INFO link.

The requirements for communication between HARDY and ARC/INFO are that:

- HARDY be able to issue a function call to ARC/INFO and make ARC/INFO active to execute the function when required during the process of problem solving;
- ARC/INFO return to HARDY, sending it a message in reply, and reactivating HARDY continuing as if it had returned from a function call; and
- the two systems be able to interact asynchronously without loss of control.

The strategy adopted here is to run ARC/INFO and HARDY in two separate processes and have them communicate via the file system. The CLIPS programming language and the ARC Macro Language are used to build the inter-process link. An inter-process protocol has been written as described below.

HARDY actually controls ARC/INFO. HARDY works with a set of predefined rules, which are fired according to the state of the analysis. When a GIS operation is needed, HARDY will generate an AML command file for its execution, send the command file to ARC/INFO, and then run into a loop and wait for a result to be generated by ARC/INFO. ARC/INFO is started by HARDY through a UNIX shell script, and begins to run just after HARDY starts up. ARC/INFO only needs to start once for the entire session. After it starts up, ARC/INFO runs in a loop and waits for an AML command file to be generated by HARDY. Once an AML command file is generated by HARDY, ARC/INFO terminates the loop, and runs the command file. After a result is generated, ARC/INFO deletes the command file, runs in a loop again and waits for HARDY to generate the next AML command file. HARDY checks for the existence of the AML command file. When the command file does not exist anymore, HARDY terminates its loop and continues its inferencing process until the next GIS operation is required, and then repeats the process above.

A fundamental problem here is “waiting for something to happen”. This problem can be solved by using the HARDY and standard *sleep* functions. If we just loop without a *sleep* call, we could use up all available processor cycles. The advantage of the above type of interaction between HARDY and ARC/INFO is simplicity — we do not have to worry about complex inter-process communication details.

To implement this protocol, we have a function `waiting-for-gis-completion( )` on the HARDY side, which performs a loop. It is written in the CLIPS language and expressed as

```
;deffunction is a CLIPS construct for defining a
;function within CLIPS by a user.
(deffunction waiting-for-gis-completion ( )
  (while (open "giscom.aml" cmd)
    (close cmd)
    (sleep 1))
  (return))
```

The function checks for the existence of the AML command file “giscom.aml”. Whenever the command file exists, HARDY sleeps 1 second. Otherwise, HARDY returns to continue its inferencing process.

During its inferencing process, HARDY creates the command file “giscom.aml” that contains an ARC/INFO command to execute a certain GIS model or other functions required. After the command file is sent to ARC/INFO, HARDY calls `waiting-for-gis-completion( )` and sleeps until “giscom.aml” is deleted. The following is an example of how this operates when HARDY is to run a GIS model “buffering”.

```

;?distance is a variable storing the buffer distance.
;"<-" is an operator for getting the address of a fact.
(defrule buffering
  ?f <- (executing-buffering-around roads ?distance)
=>
  (retract ?f)
  (open "giscom.aml" cmd "w")
  (printout cmd "&run buffering roads " ?distance crlf)
  (close cmd)
  (waiting-for-gis-completion)
  (assert (going-on-to-next-stage)))

```

When the CLIPS rule is fired, the command file "giscom.aml" is created, which contains an ARC/INFO command line. This command is to run an AML program "buffering.aml" (a GIS model solver) with a data layer "roads" and a specified buffer distance as arguments. It will produce buffer zones around roads. After the command file is generated, the rule calls the function `waiting-for-gis-completion( )`. HARDY then runs in a loop and waits for the completion of the execution of the command file. Once the command file in the ARC/INFO process is completed and it is deleted, the rule asserts a fact (`going-on-to-next-stage`) into the working memory of CLIPS in HARDY, and instructs HARDY to continue its inferencing process.

On the ARC/INFO side, ARC/INFO runs in a loop after it starts up. Here is the loop in ARC/INFO, which is written in AML.

```

&do &while not [exists quitgis]
  &do &while [exists giscom.aml]
    &run giscom.aml
    [delete giscom.aml]
  &end
  &system sleep 1
&end

```



The indicator for termination of the whole loop is the existence of the file "quitgis". The file "giscom.aml" is created by HARDY. When the file exists, ARC/INFO executes it and then deletes it. If "giscom.aml" does not exist, ARC/INFO sleeps. Thus, ARC/INFO runs in a loop and waits for commands from HARDY. If HARDY sends it the command file "giscom.aml", ARC/INFO executes it. Otherwise, ARC/INFO sleeps. After a session is finished, HARDY generates the file "quitgis", stops the ARC/INFO process and then quits.

## 7.4 Data Exchange

One requirement for integration is that CLIPS and ARC/INFO be able to share and exchange data between them. As described in Section 7.2, INFO stores and manages attribute data for geographical features. CLIPS rule-based models need INFO data for reasoning processes. At runtime, data from the INFO database must be mapped into the CLIPS facts, and the result of the rule-based modelling must be written back to the INFO database. The INFO data can be accessed either directly via the ARC/INFO programming language interface or via ASCII output files. In this study, ASCII files are taken as the bridge for the data transfer. The reason is simplicity, because it is easier to write a simple data conversion program.

The INFO data file holds several predefined standard attributes about the geographical features. An INFO file is a feature attribute table. For each geographical feature, there is one record. For each record, there are several items. A record in an INFO table is analogous to a frame defined by the `deftemplate` construct in CLIPS. The item definitions can be used to define a frame with the items as slots. The values on the defined items are used to define specific instances of the frame. Figure 7.5 shows a sample dataset developed in INFO. A `deftemplate` may be defined for it as in Figure 7.6. Figure 7.7 shows the fact dataset converted from this INFO dataset, which is to be imported into CLIPS. Before the reasoning starts, the related INFO data are firstly converted into the fact data. The conversion is implemented

automatically by a specific program written in the C programming language, and involves no post-editing. Once the fact data file is generated, CLIPS reads one line from the file each time for reasoning until the end of the file is reached.

Assuming that the fact data file is named "land.dat", the file I/O provided by CLIPS is illustrated in Figure 7.8. The first rule (lines 1-6) is called "open-file", which is used to open the file "land.dat". The second rule named "read-file" (lines 8-17) is to read a line of data from the file "land.dat", concatenate the string "land\_feature" and that line of data, and put them into the fact-list. The rule "close-file" (lines 19-23) is used to close "land.dat". The three rules use the patterns: (phase open\_file), (phase read\_file) and (phase close\_file). These patterns are used to indicate their applicability only when specific facts are in the fact-list, which are called **control patterns**. **Control facts** are used to trigger control patterns. In this case, the control facts to trigger these rules must be (phase open\_file), (phase read\_file) and (phase close\_file).

In this example, a fact (land\_feature (\$recno 1) (slope 1.5) (soil\_texture LOAM) (erosion SLIGHT) (soil\_depth 85) (flooding OCCASIONAL)) is first asserted into the working memory in CLIPS for successive reasoning processes. Once the reasoning processes are finished, the results are written to a prespecified output file, and a second fact is read and reasoning begins again. This process continues until all fact data have been processed. Figure 7.9 illustrates this process. The output file created by CLIPS rule-based modelling is an ASCII file in a format required by ARC/INFO (an example will be given in the next section). It will be imported into the INFO database and related to the associated map data layer attribute table for display or further analysis.

<i>\$RECNO</i>	<i>SLOPE</i>	<i>SOIL_TEXTURE</i>	<i>EROSION</i>	<i>SOIL_DEPTH</i>	<i>FLOODING</i>
1	1.5	LOAM	SLIGHT	85	OCCASIONAL
2	7.0	SANDY_LOAM	SIGNIFICANT	60	OCCASIONAL
3	3.5	SANDY_CLAY	SLIGHT	30	NEVER
4	15.4	GRAVEL_CLAY	SIGNIFICANT	20	NEVER
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Figure 7.5 A Sample Dataset in ARC/INFO

```
(deftemplate land_feature
  (slot $recno (type INTEGER) (default 1))
  (slot slope (type FLOAT) (default 0))
  (slot soil_texture (type SYMBOL) (default LOAM))
  (slot erosion (type SYMBOL) (default SLIGHT))
  (slot soil_depth (type FLOAT) (default 0))
  (slot flooding (type SYMBOL) (default NEVER)))
```

Figure 7.6 A Deftemplate Definition

```
($recno 1) (slope 1.5) (soil_texture LOAM) (erosion SLIGHT) (soil_depth 85) (flooding OCCASIONAL)
($recno 2) (slope 7) (soil_texture SANDY_LOAM) (erosion SIGNIFICANT) (soil_depth 60) (flooding OCCASIONAL)
($recno 3) (slope 3.5) (soil_texture SANDY_CLAY) (erosion SLIGHT) (soil_depth 30) (flooding NEVER)
($recno 4) (slope 15.4) (soil_texture GRAVEL_CLAY) (erosion SIGNIFICANT) (soil_depth 20) (flooding NEVER)
...
...
...
```

Figure 7.7 A Sample Dataset to be Imported into CLIPS

```

1      (defrule open-file
2          ?phase <- (phase open_file)
3      =>
4          (retract ?phase)
5          (open "land.dat" data)
6          (assert (phase read_file)))
7
8      (defrule read-file
9          ?phase <- (phase read_file)
10     =>
11         (retract ?phase)
12         (bind ?f0 (readline data))
13         (if (eq ?f0 EOF)
14             then (assert (phase close_file))
15             else (bind ?f (str-cat "land_feature " ?f0))
16                 (assert-string ?f)
17                 (assert (phase land_classification))))
18
19     (defrule close-file
20         ?phase <- (phase close_file)
21     =>
22         (retract ?phase)
23         (close data))

```

Figure 7.8 Sample CLIPS rules for file I/O

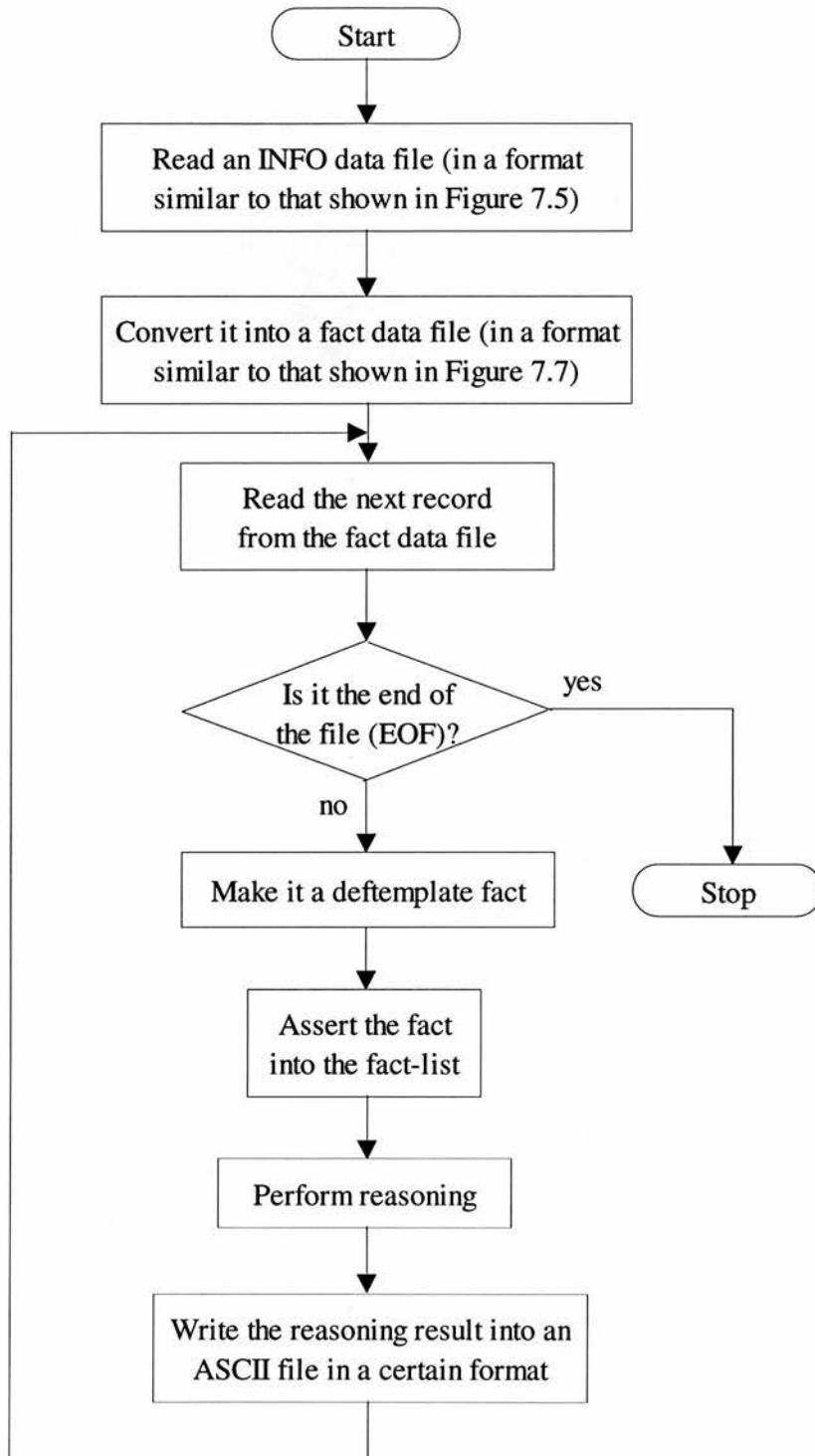


Figure 7.9 Process of reasoning on the data imported from an INFO database

## 7.5 Development of Model Solvers and Utility Programs

ARC/INFO and CLIPS can be used as two components of the back-end subsystem in a KBSDSS system. ARC/INFO is used to develop and implement GIS, analytical model solvers and some utility programs. CLIPS is used to develop and implement rule-based model solvers.

The ARC Macro Language (AML) of ARC/INFO provides a tool to develop GIS model solvers, simple analytical model solvers and utility programs. They can be created by grouping a number of ARC/INFO commands and other AML elements, which perform a certain modelling or computing task. Therefore, a GIS model solver or an analytical model solver or an utility program exists as an AML program. When a GIS model solver is needed, the system will activate ARC/INFO to run the corresponding AML program. The same applies to analytical model solvers and utility programs. For example, the following lines make up an AML program that creates a solver for the GIS model "buffering". It is called by ARC/INFO using a command line, such as `&run buffering roads 100`. Comments in AML begin with `/*` and continue until a carriage return. By convention, an AML program is named using the `.aml` extension.

```
1  /*=====
2  /* File:          buffering.aml
3  /* Purpose:      A solver for GIS model "buffering".
4  /*              Creating buffer zones around specified
5  /*              input coverage features.
6  /* Inputs:       incover - name of the coverage containing
7  /*              features to be buffered
8  /*              outcover - name of the coverage to be
9  /*              created
10 /*              distance - the distance used to create
11 /*              buffer zones around incover feature
12 /* Author:       Xuan Zhu
13 /* Created:      10/9/94
14 /*=====
15
16 /*Define the arguments of the program
17 &args incover outcover distance
18
19 &terminal 9999 /* Set terminal type
20
```



```

21 /*Check if the output coverage exists, give warning if so.
22 &if [exists %outcover% -cover] &then
23     &menu existcov &stripe Warn &position &cc
24
25 /* Pop up a window to inform the user that the buffering
26 /* model is being executed
27 &thread &create working$buffer &menu buffer.menu ~
28 &stripe working... &position &cc
29
30 /* Get the feature type of the input coverage
31 &if [exists %incover% -arc] &then
32     &s feature = line
33 &else
34     &if [exists %incover% -polygon] &then
35         &s feature = poly
36     &else
37         &s feature = point
38
39 /* Execute the ARC/INFO function "buffer"
40 buffer %incover% %outcover% # # %distance% # %feature%
41
42 &thread &delete working$buffer
43 &return

```

As described in Section 7.2, CLIPS supports modular design and partitioning of a knowledge base. Modules can be used by rules to provide execution control. This feature facilitates the construction and execution of rule-based model solvers. A rule-based model solver can be seen as an independent module in a CLIPS knowledge base and can be defined using the defmodule construct. Assume that the rules controlling the execution of all modules defined as rule-based model solvers reside in the SOLVERS module. The following CLIPS code can then be written for a solver of an example rule-based model “forestsuit”.

```

1  ;;;=====
2  ;;; Module:      FORESTSUIT
3  ;;; Purpose:    A solver for the rule-based model
4  ;;;            "forestsuit" that evaluates the physical
5  ;;;            land suitability for forestry of each
6  ;;;            location on the island of Islay.
7  ;;; Inputs:    An input data file with the structure
8  ;;;            defined by the "deftemplate" statement
9  ;;;            below.
10 ;;;            $RECNO - the record number of the data
11 ;;;            line in the data file.
12 ;;;            FEATURE-ID - the ID number of the feature
13 ;;;            in the coverage or grid of ARC/INFO.

```

```

14 ;;;          ALTITUDE - altitude
15 ;;;          AGRICAP - land capability class for
16 ;;;                agriculture
17 ;;;          FORESTCAP - land capability class for
18 ;;;                forestry
19 ;;; Author:    Xuan Zhu
20 ;;; Created:   12/9/94
21 ;;;=====
22
23 ;;; Define the module with the name FORESTSUIT,
24 ;;; representing a solver of the rule-based model
25 ;;; "forestsuit".
26 (defmodule FORESTSUIT
27   (import SOLVERS deftemplate CLIPS-modelling args))
28
29 ;;; The template for each fact holding a list of data
30 ;;; associated with a location required for the model.
31 ;;; Their definitions must be the same as defined in the
32 ;;; input data.
33 (deftemplate FORESTSUIT::facts
34   (slot $RECNO (type INTEGER))
35   (slot FEATURE-ID (type NUMBER))
36   (slot ALTITUDE (type NUMBER))
37   (slot AGRICAP (type NUMBER))
38   (slot FORESTCAP (type NUMBER)))
39
40 (defrule FORESTSUIT::start-up
41   (CLIPS-modelling (status start))
42 =>
43   (assert (phase open_file)))
44
45 (defrule FORESTSUIT::open-file
46   ?phase <- (phase open_file)
47   ;;; The following fact is asserted by a rule in the
48   ;;; SOLVERS module. The variable ?indata stores the
49   ;;; name of the input data file, while ?outdata stores
50   ;;; the name of the output data file.
51   ?args <- (args (infile-outfile ?indata ?outdata))
52 =>
53   (retract ?phase ?args)
54   (open ?outdata clipsout "w")
55   (open ?indata data)
56   (assert (phase read_data)))
57
58 (defrule FORESTSUIT::read_data
59   ?phase <- (phase read_data)
60 =>
61   (retract ?phase)
62   (bind ?f0 (readline data))
63   (if (eq ?f0 EOF)
64     then (assert (phase close_file))
65     else
66       (bind ?f (str-cat "(facts " ?f0 ")))
67       (assert-string ?f)
68       (assert (phase analysis))))
69

```

```

70 (defrule FORESTSUIT::close-file
71   ?phase <- (phase close_file)
72   ?status <- (CLIPS-modelling (status ?))
73 =>
74   (retract ?phase)
75   (modify ?status (status end))
76   (printout t " Reasoning finished." crlf crlf)
77   (close clipsout)
78   (close data))
79
80 ;;*****
81 ;;* The following are domain-specific *
82 ;;* rules for the given modelling task *
83 ;;*****
84
85 (defrule FORESTSUIT::forest-suitability-1
86   ?phase <- (phase analysis)
87   ?f <- (facts ($RECNO ?)
88         (FEATURE-ID ?id)
89         (ALTITUDE ?alt&:(< ?alt 450))
90         (AGRICAP ?agri&:(> 5.0))
91         (FORESTCAP ?forest&:(< ?forest 4)))
92 =>
93   (retract ?phase ?f)
94   (printout clipsout ?id ":suitable" crlf)
95   (assert (phase read_data)))
96
97   .
98   .   ;;; Any other domain-specific rules
99   .   (see Table 8.5 in Chapter 8)

```

This model solver works in a similar way to that shown in Figure 7.9. It takes a data file in the format as shown in Figure 7.7 as an input. Assume that a data file to be input to the model solver, whose logical name is “data” (see lines 55 and 62), is shown as in Figure 7.10. It was transformed from the corresponding INFO table file. The domain-specific rules in this solver are those listed in Table 8.5 in Chapter 8. They perform reasoning on the input data and write the reasoning results into an output data file. For example, when the fact (facts (\$RECNO 4) (FEATURE-ID 2) (ALTITUDE 100) (AGRICAP 6.3) (FORESTCAP 3)) is asserted, the “forest-suitability-1” rule (lines 85-95) will fire, which will assign the land parcel with the ID number of 2 the value “suitable” for forestry, and write the result as a data line “2:suitable” into the output data file whose logical name is “clipsout” (see lines 54 and 94). After the execution of the solver or reasoning on the input data, the

```

($RECNO 1) (FEATURE-ID 4) (ALTITUDE 120) (AGRICAP 5.2) (FORESTCAP 4)
($RECNO 2) (FEATURE-ID 1) (ALTITUDE 160) (AGRICAP 6.2) (FORESTCAP 4)
($RECNO 3) (FEATURE-ID 6) (ALTITUDE 80) (AGRICAP 4.2) (FORESTCAP 4)
($RECNO 4) (FEATURE-ID 2) (ALTITUDE 100) (AGRICAP 6.3) (FORESTCAP 3)
($RECNO 5) (FEATURE-ID 3) (ALTITUDE 110) (AGRICAP 5.3) (FORESTCAP 4)
($RECNO 6) (FEATURE-ID 15) (ALTITUDE 120) (AGRICAP 6.1) (FORESTCAP 5)
($RECNO 7) (FEATURE-ID 7) (ALTITUDE 80) (AGRICAP 4.1) (FORESTCAP 3)
($RECNO 8) (FEATURE-ID 9) (ALTITUDE 200) (AGRICAP 6.1) (FORESTCAP 3)
($RECNO 9) (FEATURE-ID 10) (ALTITUDE 180) (AGRICAP 6.3) (FORESTCAP 3)
($RECNO 10) (FEATURE-ID 5) (ALTITUDE 100) (AGRICAP 5.2) (FORESTCAP 3)
($RECNO 11) (FEATURE-ID 13) (ALTITUDE 400) (AGRICAP 5.2) (FORESTCAP 7)
($RECNO 12) (FEATURE-ID 8) (ALTITUDE 80) (AGRICAP 4.2) (FORESTCAP 3)
($RECNO 13) (FEATURE-ID 11) (ALTITUDE 160) (AGRICAP 5.2) (FORESTCAP 3)
($RECNO 14) (FEATURE-ID 12) (ALTITUDE 150) (AGRICAP 5.3) (FORESTCAP 3)
($RECNO 15) (FEATURE-ID 14) (ALTITUDE 320) (AGRICAP 7) (FORESTCAP 4)
. . .
. . .
. . .

```

Figure 7.10 A sample input data file to the model solver “forestsuit”

```
4:MODERATE
1:MODERATE
6:UNSUITABLE
2:SUITABLE
3:MODERATE
15:LIMITED
7:UNSUITABLE
9:SUITABLE
10:SUITABLE
5:SUITABLE
13:UNSUITABLE
8:UNSUITABLE
11:SUITABLE
12:SUITABLE
14:MODERATE
.
.
.
```

Figure 7.11 A sample output from the model solver “forestsuit”

output data file is created as shown in Figure 7.11. It can be converted into an INFO data file and imported into the INFO database by an utility program.

The model solver can be executed using the CLIPS *focus* command issued by a rule from the SOLVERS module. Rules in the SOLVERS module also assign parameters to rule-based model solvers. In our example, the parameters are the input data file storing facts to be used for reasoning and the output data file for writing the results. The two parameters are passed from the SOLVERS module to the model solver above through asserting a fact in the SOLVERS module that can be “seen” from the model solver (a module). The pattern `(args (infile-outfile ?indata ?outdata))` on line 51 in the model solver is to match that fact asserted by a rule from the SOLVERS module and get the two parameters through the variables `?indata` and `?outdata`. We use this as a method of assignment of parameter values to rule-based model solvers.

The “forestsuit” model solver described above also provides a model for creating other rule-based model solvers in a KBSDSS system. Apart from the module name having to be changed, the *defmodule* statement, the rules “start-up”, “open-file”, “read-data” and “close-file” are common to every rule-based model solver.

They may be copied to create other rule-based model solvers. The domain-specific rules in one solver are different from those in another. The “facts” deftemplate should be redefined according to the definitions in the input data.

## 7.6 Summary

This chapter has presented a KBSDSS system development environment, which integrates HARDY, CLIPS and ARC/INFO. HARDY provides facilities for building diagramming capabilities and flexible user interfaces. CLIPS is used to develop knowledge bases and perform inferences. CLIPS is also used to develop and execute rule-based models. A rule-based model can be constructed as a CLIPS knowledge module. ARC/INFO provides capabilities for spatial data handling. The AML language can be used to develop and execute GIS, analytical models and utility programs. A GIS model corresponds to an AML program. So do analytical models and utility programs. Two links, HARDY-ARC/INFO communication and ARC/INFO-CLIPS data exchange, have been established. Through the two links, the three systems can operate together and mutually support one another's operation. By developing an appropriate database, a modularised knowledge base, suitable GIS, analytical and rule-based models, and necessary utility programs, a powerful KBSDSS system can be built within this software environment. The next chapter will present such a KBSDSS prototype.

## Chapter 8

# ILUDSS: A Prototype KBSDSS for Rural Land Use Planning

The availability of a KBSDSS development environment makes it feasible to construct a prototype KBSDSS having the structure shown in Figure 6.3. This chapter describes a prototype KBSDSS called ILUDSS (the Islay Land Use Decision Support System), which is implemented in the development environment presented in the previous chapter. Namely, ILUDSS is built using ARC/INFO, HARDY and CLIPS through the two links: HARDY-ARC/INFO communication and ARC/INFO-CLIPS data exchange. It is designed to assist planners in strategic planning of land use for the development of the island of Islay, off the west coast of Scotland, and it is developed primarily as a demonstration of the feasibility of KBSDSS technology.

In this chapter, we first discuss the need for KBSDSS technology in rural land use planning, then present the main features of ILUDSS, including the contents of its database, the models available, the system structure and functionality, and its implementation, and finally discuss the strength and limitations of ILUDSS.

## 8.1 The Need for KBSDSS Technology in Land Use Planning

Almost every human activity uses land. With increasing pressure on land, the need for rational land use planning is brought about. Decisions on land use may lead to great benefits or great losses, sometimes in economic terms, sometimes in less tangible environmental changes (McRae and Burnham 1981). **Land use** is the assembly of different activities (e.g. agriculture, forestry, nature conservation and industry) performed within particular areas. **Land use planning** is a process of decision making as to how land resources should be utilised. Its function is to guide planning decisions on land use in such a way that the land resources are put to the most beneficial use to meet the needs of the present, while at the same time, conserving the resources for the future (FAO 1976).

Tomlin and Johnston (1990) listed three major roles of GIS systems in land use planning:

- the maintenance of general-purpose data;
- the generation of specific-purpose information from such data; and
- the utilisation of such information in decision-making contexts.

However, land possesses both physical and economic characteristics. Land use planning must be based on an understanding of the natural environment as well as of the kinds of land use envisaged. The study of land use is a comprehensive applied science, spanning a wide diversity of disciplines ranging from geography, agriculture, environmental and biological sciences to economics and sociology (Davidson 1986; Maxwell 1993). Thus, in a land use planning decision-making context, there is a need to combine a wide variety of relevant knowledge. It is unreasonable to expect most decision makers to acquire the multidisciplinary background needed for land use science to make land use planning decisions.

On the other hand, decision-making about land use often raises strong emotions and is much influenced by the social and economic situation (McRae and Burnham



1981). Therefore, decisions on land use should account for the specific circumstances and preferences of decision makers.

As discussed in Chapter 1, GIS and traditional SDSS systems do not fully capture the knowledge and expertise needed for land use studies. In addition, GIS and SDSS systems are toolboxes, which are sophisticated and complex to use. The research conducted by Davidson et al. (1993) indicates that however useful the output in terms of evaluating land development proposals, cost and time constraints, and requirements of considerable technical expertise would exclude routine usage of GIS systems by planners and land managers for land use planning purposes. They also suggested that “the ideal would be for planning authorities to be able to receive and input GIS files as part of environmental statements and to be able to compare immediately development options using their own systems”. The emergence of knowledge-based support systems presents an opportunity to achieve this objective. KBSDSS systems enable planners and land managers to exploit the power of GIS and other spatial modelling and problem solving techniques to solve their land use problems in a relatively simple and fast way.

KBSDSS systems that are specially designed to support land use decision making offer a number of advantages. In addition to those advantages discussed in Chapter 6, use of these systems may foster timely decisions in land use planning, as well as a clear understanding of the process and the criteria used to arrive at the land development recommendations. Thus, it becomes relevant, and also necessary, to apply the KBSDSS techniques to build a land use decision support system. ILUDSS is such an attempt.

## 8.2 Land Use Problems on Islay Faced by Planners

The island of Islay is off the west coast of Scotland (Figure 8.1). It covers an area of more than 60,000 ha.. About 25% of the island has been designated by the Nature

Conservancy Council (NCC) (now the Scottish Natural Heritage) as Sites of Special Scientific Interest (SSSI), largely because of its importance for birds. The island has a wide range of natural topography and a high proportion of semi-natural vegetation, with the government obliged to protect habitats of the 17 bird species found there that are in Annex One of the European Community Birds Directive (Bignal et al. 1988).



Figure 8.1 Study area

Islay exhibits a combination of land use interests: nature conservation, agriculture, forestry, peat-cutting, tourism and sport management (Selman 1989). The land management is dominated by stock rearing, and by sporting management of the uplands. These management systems have destroyed less of the natural environment than more intensive farming and commercial afforestation has done over most of Britain (Bignal et al. 1988). However, while the effects of changing land use pressures have not yet been widely felt on the island, many changes and land use conflicts have occurred and the rate of change is accelerating. For example, Scottish Malt Distillers plc proposed a plan to extract peat for malting from the Duich Moss SSSI areas on the island (GWFGST 1986). This conflicted with conservation management objectives.

Therefore, as Bignal et al (1988) pointed out, there is a urgent need to assess the potential impact of future changes on the island and enable decisions on land management compatible with conservation. The underlying goals of land use planning for the future land management on Islay are (1) to encourage land resource management which is compatible with maintaining wildlife value, and (2) to integrate farming and other land management with nature conservation. In order to achieve these goals, it is necessary to identify those areas of the island which are of prime, intermediate and little importance to various land use interests. This will help resolve conflicts between competing objectives for land use, and help formulate feasible land use planning strategies.

### 8.3 The Problem Domain Addressed by ILUDSS

To formulate land use planning strategies successfully requires a characterisation and comparative analysis of competing land use interests (Healey et al. 1988). Two principles should be followed in the formulation of land use policies (McRae and Burnham 1981):

- land should be used for the purposes for which it is well suited;
- land of high value for an existing land use should be protected against changes which are difficult to reverse.

A decision making process in land use planning can be described as follows. First, the requirements for each land use type are formulated based on socio-economic demand and supply. According to the land use types under consideration, the potential of land for each competing land use type is evaluated. Various suitability models are developed at this stage. Each suitability model presents one alternative scenario of land use in the region and determines the areas well suited for that land use. By comparing these different suitability scenarios, the planners identify areas of compatible and conflicting land use, then assess the patterns of conflicting land use interests, and finally determine land use allocations. Here, evaluation of land use suitability and potential is a major step forward in land use planning. It also acts as a vehicle for identifying and resolving potential conflicts. It is important that planners have the ability to develop and compare various land use potential or suitability models so that feasible strategies for land use alternatives can be recommended to achieve community goals and objectives (Cowen and Shirley 1991). Allocation of land for a certain type of land use should be preceded by an evaluation of land suitability and potential for that type.

ILUDSS is to help planners, including land managers and those formulating and implementing land use policy, in this decision-making context, focusing on evaluating land use potential and identifying suitability for different land uses, i.e. formulating indicative land use strategies. Land use potential for a certain type of land use can be evaluated according to different factors or attributes. For example, the land use potential for afforestation may be evaluated according to the physical land suitability for forestry and impacts on wildlife. It can also be evaluated according to the physical land suitability for forestry and proximity to roads. It depends on the planner's needs and preference of conditions for the proposed land use.

The purpose of ILUDSS is to assist planners in evaluation of land use potential for various land use types on the island of Islay, according to their preferences and assessments relating to the various criteria and related evaluation factors. It can also be employed by model builders who are responsible for developing land use models. Thus, the following requirements are imposed on the design of the ILUDSS functionality:

For planners who may have limited experience in spatial modelling and GIS, the system can

- guide the user to specify his or her land use interests and preferences relating to the evaluation factors or attributes;
- automatically formulate land use models to evaluate the land use potential according to the user's land use interests and preferred evaluation factors;
- evaluate the formulated land use models by integrating different types of models and data automatically;

for model builders who have spatial modelling and analysis experience in the GIS context, the system allows

- the user to modify existing land use models or produce new land use models;
- to add new analytical, rule-based and GIS models;
- to add new data sets or data layers;
- to acquire meta-data and model knowledge;

and for both types of users, the system is able to

- access the database, retrieve and display required data graphically;
- retrieve meta-data about the existing data, and information about the existing models;

- provide helpful information about the modelling process, system's functionality and use;
- generate cartographic displays and tabular reports.

As a prototype, ILUDSS can not claim to be a comprehensive spatial decision support system, but is an application aimed at illustrating a number of design principles of a KBSDSS. Before going to discuss the system structure and functionality, we will describe the ISLAY database and the models developed for ILUDSS.

## 8.4 A GIS Database for Islay

ILUDSS is built based on a GIS database for Islay, which was created and implemented as a result of a collaborative project established in 1987 between the Nature Conservancy Council (NCC) and the Department of Geography of the University of Edinburgh. The database has been used to examine the impact of designating areas as SSSI on Islay and in the formulation of land use strategies (Healey et al. 1988; Rideout and Holbrook 1992a). The Islay database consists of the following components (Rideout and Holbrook 1992b) (see Table 8.1):

- basic topography
- socio-economy
- environment
- conservation
- ecological survey.

Table 8.1 Data comprising the Islay GIS data base

(From Rideout and Holbrook 1992b)

<b>Basic Topography</b>	
1:10,000 Coast	The most detailed high and low water coastline
25m Generalised Coast	High water coastline generalised to 25m resolution
50m Generalised Coast	High water coastline generalised to 50m resolution
50m Contours	50m interval contours from the 1:50,000 OS (Ordnance Survey) sheet 60
1:50,000 Rivers	All rivers and streams from the 1:50,000 OS sheet 60
Lochs	Lochs from the 1:50,000 OS sheet 60
O.S. Sheet No. Grid	Location and coverage of OS 1:10,000 map sheets
1km Grid	A 1km by 1km cell size grid in National Grid units
10km Grid	A 10km by 10km cell size grid in National Grid units
<b>Socio-economy</b>	
Roads 1:50,000	Roads and tracks from the 1:50,000 OS sheet 60
Farm Boundaries 1:10,000	Farm boundaries from DAFS/OS 1:10,000 sheets
<b>Environment</b>	
Soils 1:50,000	Macaulay Land Use Research Institute (MLURI) 1:50,000 Provisional Soil Map
Woodland 1:50,000	Woodland/scrub from the 1:50,000 OS sheet 60
Woodland 1:25,000	Woodland/scrub from the 1:25,000 OS sheet
Land Capability for Agriculture	MLURI 1:50,000 map of Land Capability for Agriculture
Land Capability for Forestry	MLURI 1:50,000 map of Land Capability for Forestry
<b>Conservation</b>	
SSSI Areas	SSSI areas from 1:10,000 NCC/OS maps
Landcover	NCC 1:25,000 Landcover Map

The database was developed using ARC/INFO and ORACLE (a proprietary relational database management system). The data about basic topography, socio-economy, environment and conservation are encoded in ARC/INFO coverages. The ecological survey data are stored in tables in the ORACLE relational database management system. The ecological data includes habitat information and wildlife species information. Habitat information covers data from the NCC Landcover Map, and a Fresh Water Loch Survey. Species information includes a Balfour Browne Club Coleoptera Survey, the NCC/CSD Islay Birds Survey, the British Trust for Ornithology Atlas Survey and a Royal Botanic Garden Survey. This thesis uses some of the topographic, socio-economic, environmental and conservation data in the Islay database, including 1:10000 Coast, 50m Contours, 1:50000 Roads, 1:50000 Rivers, 1:50000 Soils, 1:50000 Land Capability for Agriculture, 1:50000 Land Capability for Forestry, 1:10000 SSSI Areas, and 1:25000 Landcover. Except Coast, all other data layers used in ILUDSS were rasterised into 50m × 50m grid cells in order to use ARC/INFO GRID analysis and modelling functions, which speeds up the data processing in comparison to manipulation of vector map coverages, and facilitates the demonstration of the system.

## 8.5 Model Development

ILUDSS design takes advantage of several simplifying assumptions. These assumptions make the task of programming the system feasible within the constraints imposed by the limited resources and time available for its development. One simplification is that there is exactly one solver for a given analytical or rule-based model. Second, there may be two solvers for a given GIS model in terms of their spatial data models. Another important assumption made in designing ILUDSS is that no backtracking is allowed during a modelling session. The only way to correct erroneous entry of data is to restart the session.



ILUDSS can incorporate GIS, analytical and rule-based models. The current ILUDSS contains the following models.

### 8.5.1 GIS models

The GIS models currently available in ILUDSS are a subset of GIS models defined in Chapter 3, plus two new models *formula* and *if-then*.

The *formula* is a model that derives a new map data layer by combining several other map data layers and computes the new values location by location using a user-defined mathematical formula. It can be seen as an integration of the GIS model *overlay* and the analytical model represented by the user-defined mathematical formula. When *formula* is specified as a spatial relation in a spatial influence diagram, the system will prompt the user to enter a formula using an arithmetic expression with the arithmetic operators listed in Appendix G.

The *if-then* is a model that derives a new map data layer by combining several other map data layers and computes the new values location by location using a user-defined if-then rule. It can be seen as an integration of the GIS model *overlay* and the rule-based model represented by the user-defined if-then rule. The format of an if-then rule is as follows:

```
if (condition) statement_list
{else if (condition) statement_list} . . .
{else statement_list}
```

The *condition* should be in parentheses, which may be any logical expressions with the logic operators listed in Appendix H. The contents in the braces {} are optional. The *statement\_list* consists of the arithmetic expressions with the arithmetic operators listed in Appendix G. If there are two or more expressions in the *statement\_list*, they have to be enclosed using the braces {}. The following are two examples:

```
if (soil_depth > 5 & slope < 0.05) suitability = 1
```

```
if (soil_depth > 5 & slope < 0.05) suitability = 1
else if (soil_depth <= 5 & slope < 0.05) suitability = 2
else suitability = 3
```

Table 8.2 lists all currently available GIS models and their usage in spatial influence diagrams.

Table 8.2 GIS models available in ILUDSS and their usage in spatial influence diagrams

<i>Model Name</i>	<i>Usage in Spatial Influence Diagrams</i>
<b>Location-specific analysis</b>	
<i>reclassify</i>	Node i with the model as its spatial relation has one predecessor (denoting a map variable) . The value of Node i is derived by reclassifying the value of its predecessor. A reclassification table should be specified as the attribute relation of Node i.  (Note: the spatial relation represents the spatial relationships between a node and its predecessors; the attribute relation represents the functional relationships between a node and its predecessors, see Chapter 4.)
<i>overlay</i>	Node i with the model as its spatial relation has two or more predecessors (all denoting map variables). The value of Node i is derived through combining all map data layers of the predecessors. A functional model should be specified as the attribute relation of Node i.
<i>add</i>	Node i with the model as its spatial relation has two predecessors. The value of Node i is derived by combining two map data layers of the predecessors and adding their values, location by location. One of the predecessors may

	be a constant. In this case, the model simply adds the constant to the map data layer of the other predecessor. No attribute relation is specified for Node i.
<i>subtract</i>	Similar to <i>add</i> , but perform the subtraction operation on two predecessors.
<i>multiply</i>	Similar to <i>add</i> , but perform the multiplication operation on two predecessors.
<i>divide</i>	Similar to <i>add</i> , but perform the division operation on two predecessors.
<i>formula</i>	Node i with the model as its spatial relation has two or more predecessors. The value of Node i is derived by combining all map data layers of the predecessors and computing new values location by location using a formula to be specified by the user. No attribute relation is specified for Node i.
<i>if-then</i>	Similar to <i>formula</i> , but perform analysis using a single if-then rule to be specified by the user on the predecessors.
<b>Regional summary</b>	
<i>RegionalSum</i>	Node i with the model as its spatial relation has two predecessors. The value of Node i is derived by computing the sum of the values of land features on the map data layer of one predecessor that belong to the same region depicted on the map data layer of the other predecessor. The region here refers to a homogeneous region, such as a land parcel with a certain land cover type and a region defined by county or farm boundaries. No attribute relation is specified for Node i.
<i>RegionalMean</i>	Similar to <i>RegionalSum</i> , but calculate the mean of the values.
<i>RegionalMaximum</i>	Similar to <i>RegionalSum</i> , but calculate the maximum of the values.

<i>RegionalMinimum</i>	Similar to <i>RegionalSum</i> , but calculate the minimum of the values.
<b>Neighbourhood analysis</b>	
<i>buffering</i>	Node i with the model as its spatial relation has one or two predecessors. If there are two predecessors, one represents the land features to be buffered, the other represents the buffer distance. The value of Node i is the buffer zone around the land features represented by one of its predecessors. No attribute relation is specified for Node i.
<i>SimpleProximity</i>	Node i with the model as its spatial relation has one predecessor representing target locations. The value of Node i is the shortest straight-line distance from target locations to all other locations in a study area. This is similar to a series of buffers of equal steps emanating from target features. No attribute relation is specified for Node i.
<i>slope</i>	Node i with the model as its spatial relation has one predecessor representing the continuous surface (in TIN or ARC/INFO lattice, see Appendix C) to be calculated. The value of Node i is the resultant slope. No attribute relation is specified for Node i.
<i>aspect</i>	Similar to <i>slope</i> , but the resultant value is aspect.
<b>Geometric measurement</b>	
<i>area</i>	Node i with the model as its spatial relation has one predecessor representing regions to be calculated. The value of Node i is the area of each region represented by the predecessor. No attribute relation is specified for Node i.
<i>perimeter</i>	Similar to <i>area</i> , but the resultant value is the perimeter of each region.

The corresponding solvers of all these GIS models have been created and exist as AML programs. Section 7.5 has given an example, which is a solver of *buffering*. Many of these models have been used in developing applications of ILUDSS, such as *overlay*, *reclassify*, *buffering*, etc., as will be seen from the sample session of ILUDSS described in Appendix F. The list of GIS models are designed for use not only by the programmer of ILUDSS to develop applications, but also by model builders (a type of ILUDSS user as described in the previous section) to develop their own land use models. Therefore, we provide more GIS models than we need for developing our specific applications such that model builders can have a wide range of GIS models for them to choose from to build their land use models in ILUDSS. This list of GIS models will be augmented in the future.

In ILUDSS, almost all modelling activities use spatial influence diagrams as a “front-end”. To use a GIS model to perform a particular analysis or modelling tasks, a spatial influence diagram should be created. That is, a GIS model should be used in a spatial influence diagram. Section 8.7.3.1 will give details about how to use GIS and other types of models.

## **8.5.2 Physical land suitability models**

“A tract of land is defined geographically as a specific area of the earth's surface: its characteristics embrace all reasonably stable, or predictably cyclic, attributes of the biosphere vertically above and below this area, including those of the atmosphere, the soil and underlying geology, the hydrology, the plant and animal populations, and the results of past and present human activity to the extent that these attributes exert a significant influence on present and future uses of land by man” (Brinkman and Smyth 1973).

The assessment of physical land suitability involves the comparison of the requirements of land use with the resource offered by the land. It translates the opportunities and limitations presented by the relatively permanent environmental factors, such as soil, climate, vegetation, geology and topography, into suitable land uses. Currently, ILUDSS contains three physical land suitability models.

(1). Physical land suitability for farming

Physical land suitability for farming is determined based on the land capability for agriculture (LCA). The land capability classification for agriculture classifies land by integrating information on soil, climate and relief (Bibby et al. 1982). The system of land capability classification for agriculture provides a means of assessing the physical value to agriculture of land. The agriculture capability classification comprises seven classes in Scotland. A brief description of the classes is shown in Table 8.3 (Bown et al. 1982).

Table 8.3 Agriculture capability classification (from Bown et al. 1982)

<i>CLASS</i>	<i>DESCRIPTION</i>
Class 1	Land capable of producing a very wide range of crops. Cropping is highly flexible and includes the more exacting crops such as winter-harvested vegetables. The levels of yield are consistently high
Class 2	Land capable of producing a wide range of crops. Cropping is very flexible and a wide range of crops can be grown but difficulties with winter vegetables may be encountered in some years. The level of yield is high but less consistently obtained than in Class 1.
Class 3	Land capable of producing a moderate range of crops.
Class 3.1	The land is capable of producing consistently high yields of a narrow range of crops (cereals and grass) or moderate yields of a wider range (potatoes, field beans and other vegetables and root crops). Grass leys of short duration are common.
Class 3.2	The land is capable of average production but high yields of grass, barley and oats are often obtained. Grass leys are common and of longer duration than in Class 3.1.
Class 4	Land capable of producing a narrow range of crops.
Class 4.1	Long-ley grassland is commonly encountered but the land is capable

Class 4.2	of producing forage crops and cereals for stock. The land is primarily grassland with some limited potential for other crops.
Class 5	Land capable of use as improved grassland.
Class 5.1	Land well suited to reclamation and to use as improved grassland.
Class 5.2	Land moderately suited to reclamation and to use as improved grassland.
Class 5.3	Land marginally suited to reclamation and to use as improved grassland.
Class 6	Land capable only of use as rough grazing.
Class 6.1	Land with high grazing value.
Class 6.2	Land with moderate grazing value.
Class 6.3	Land with low grazing value.
Class 7	Land of very limited agriculture value.

From Table 8.3, we can see that land in LCA Classes 1 to 4 is suited to arable use and that in LCA Classes 5 to 7 is unsuited to arable use. Land in LCA Classes 1 to 3 is not found on Islay. Only a relatively small proportion of land on the island is managed as more intensive ley and rotational arable cropping. The arable land is mainly in Class 4 and is predominantly lowland or coastal and on river soils. The main goal of land use planning on Islay is to protect the existing character of agricultural land use and prevent unnecessary intensification and inappropriate change (Bignal et al. 1988). According to this principle, we define land in LCA Class 4 on the island as that physically suitable for farming, and other land (LCA Classes 5-7) as physically unsuitable for farming. The physical land suitability model for farming contains this rule, and is encoded as a reclassification table.

## (2). Physical land suitability for forestry

The potential sites for afforestation are determined by the physical environmental conditions, such as climate and soil, the economics of forestry and

alternative land uses, land use policies and individual decisions of decision makers (Aspinall 1993). The physical land suitability for forestry is mainly concerned with the physical site conditions. We determine the physical land suitability for forestry according to land capability for forestry (LCF), land capability for agriculture (LCA) and altitude.

The forestry capability classification classifies land according to the limitations imposed by soil, climate and topography on the growth of trees and on silvicultural practices, the principal tree species considered being those broadleaves and conifers commonly grown in Britain (Bibby et al. 1988). It provides an assessment of the afforestable land resource. The LCF is classified into seven classes in Scotland. Table 8.4 is a brief description of the classification system (Futty et al. 1989). A full description can be found in Bibby et al. 1988.

Table 8.4 Forestry capability classification (from Futty et al. 1989)

<i>CLASS</i>	<i>DESCRIPTION</i>
Class 1	Land with excellent flexibility for the growth and management of tree crops. Suited to a wide range of broadleaved and coniferous species.
Class 2	Land with very good flexibility for the growth and management of tree crops. Suited to broadleaves and conifers, but the choice is more restricted than in Class 1.
Class 3	Land with good flexibility for the growth and management of tree crops. Suited to a wide range of conifers and a restricted range of broadleaved species.
Class 4	Land with moderate flexibility for the growth and management of tree crops. Suited to many coniferous species and in places the less demanding broadleaves.
Class 5	Land with limited flexibility for the growth and management of tree crops. Suited to conifers such as spruces, larches and pines, and to birch, alder or other hardy broadleaves.
Class 6	Land with limited flexibility for the growth and management of tree crops.



	Suited to lodgepole pine and Sitka spruce and to amenity broadleaves such as birch and alder.
Class 7	Land unsuitable for producing tree crops.

Tree growth is known to be strongly affected by altitude, while recent government policy has been against the planting of forests on good quality agricultural land (Aspinall 1993). Therefore, we include these factors in our physical land suitability model for forestry. As described above, LCA Class 4 is the only land suitable for arable cropping available on Islay, and its extent is quite small. The land of this type forms important and essential components of the winter feeding ground of migratory geese, golden plovers and choughs, while in summer, it is important for choughs, corncrakes and raptors (Bignal et al. 1988). Thus, LCA Class 4 land will retain the protection of a presumption for agricultural production.

The physical land suitability model for forestry is a rule-based model which consists of a set of rules, as listed in Table 8.5. This model is encoded as a CLIPS knowledge base module called FORESTSUIT, part of which has been shown and explained in Section 7.5.

Table 8.5 Rules for assessment of physical land suitability for forestry

<i>CLASS</i>	<i>RULE</i>
Suitable	LCA Class > 5 and LCF Class ≤ 3 and Altitude < 450m
Moderate	LCA Class > 5 and LCF Class = 4 and Altitude < 450m
Limited	LCA Class > 5 and 4 < LCF Class < 7 and Altitude < 450m
Unsuitable	LCA Class < 5 or LCF = 7 or Altitude ≥ 450m

### (3). Physical land suitability for peat-cutting

The peatlands of Scotland are an important resource. Peat can be extracted and used for fuel and horticultural applications, and the land can be converted to

agriculture or forestry, managed for sheep grazing or recreational sport shooting, and for conservation of wildlife and amenity (Ward 1990). The physical land suitability model for peat-cutting here identifies location and extent of peat deposits from the Macaulay Land Use Research Institute soil map. All peat soils over 50 cm in depth are identified as areas physically suited to peat-cutting. Just as with the physical land suitability model for agriculture, the physical land suitability model for peat-cutting consists of one rule, encoded as a reclassification table.

## 8.6 System Structure and Functionality

The spatial influence diagram-based mechanisms, a development environment integrating HARDY, CLIPS and ARC/INFO, a GIS database and the development of GIS and physical land suitability models described in the previous chapters and sections provide a basis on which the framework for KBSDSS systems presented in this thesis can be implemented. ILUDSS is an implementation of the framework.

Based on the user requirements described in Section 8.3, ILUDSS is initially designed to support three groups of functions (Figure 8.2):

- Query

Data query & display — retrieving spatial data from the database and displaying them in maps.

Meta-data query — retrieving information or knowledge about the data.

Model query — retrieving existing land use problem models (spatial influence diagrams)

Model-information query — retrieving information or knowledge about models.

On-line help — providing information about the system's functionality and use.

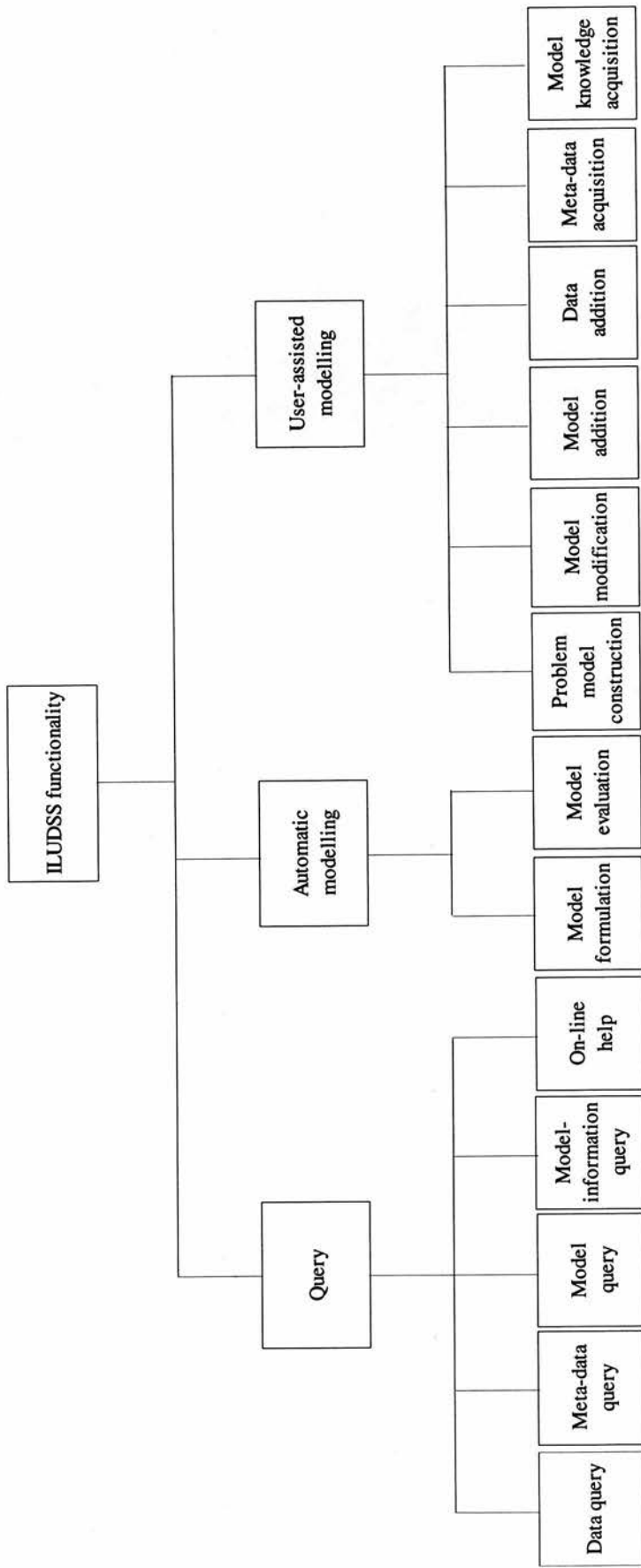


Figure 8.2 Functions of ILUDSS

- Automatic modelling

Model formulation — formulating land use problem models for evaluation of land use potential for certain types of land uses according to the users' land use interests and their preferences relating to the evaluation factors.

Model evaluation — evaluating the formulated land use problem models automatically by integrating GIS, analytical and rule-based models as well as data available in the system.

- User-assisted modelling

Problem model construction — structuring the land use problems which can not be solved by automatic modelling, representing the problems using spatial influence diagrams and building the problem models by employing different types of existing models and data.

Model modification — modifying the existing land use problem models formulated either by the system or by users.

Model addition — adding new analytical or rule-based models.

Data addition — adding new data sets or data layers.

Meta-data acquisition — acquiring meta-data about the new data layers.

Model knowledge acquisition — acquiring information about the new analytical or rule-based models

The spatial influence diagram-based mechanisms drive the formulation and evaluation of spatial problems. The architecture of ILUDSS is quite similar to that illustrated in Figure 6.3, i.e. it is composed of a user interface, query processing subsystem, modelling subsystem, problem processor, knowledge base and back-end subsystem. Figure 8.3 presents these components and their relationships (control, data and knowledge flows).

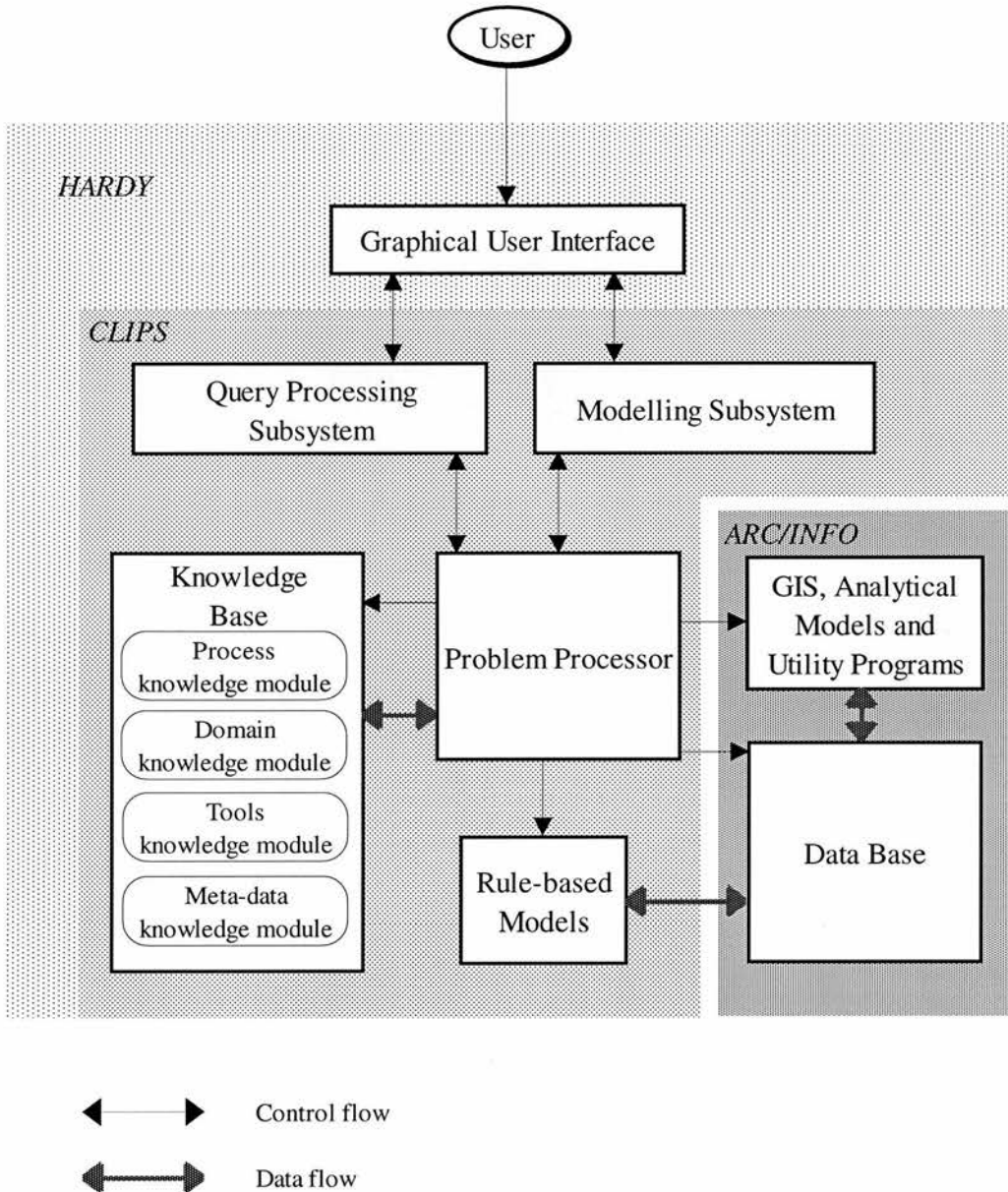


Figure 8.3 Overall Structure of ILUDSS

As can be seen, the level of integration is high. The user can interact with the system through a graphical user interface. The query processing subsystem and modelling subsystem receive the user's queries and actions, and send relevant messages to the problem processor. In response, the problem processor then uses

knowledge from the knowledge base to make inferences, control access to the database and the knowledge modules in the knowledge base, select appropriate models and utility programs, and activate the execution of the selected models and utility programs to answer the user's queries or perform the actions required by the user. The models and utility programs can be fed with data from the database. The results of their execution can also be written back to the database. Generally, the rule-based and analytical models use the INFO attribute data. The GIS models use the ARC/INFO coverages or grids. The following are descriptions of each system component.

### **8.6.1 The user interface**

A graphical user interface for ILUDSS has been designed and built on top of the X-windows system using the CLIPS window functions of HARDY. Figure 8.4 is a screenshot of ILUDSS, which shows five basic windows in a typical ILUDSS session. Going clockwise, the top-left window is the main window, through which a user can access all ILUDSS functions. To the right is a diagram window or a spatial influence diagram editor for displaying, creating or editing a spatial influence diagram. Beneath this is the map display window for map display and query. The map display window is created by ARC/INFO. To the left is a terminal window for activating ILUDSS and monitoring ILUDSS operations. In the middle of the screen is a pop-up window which is used to display the relevant information at the different stages of a session. For example, it can be used to inform the user of the current or next operations, to ask the user questions and receive the user's responses and so on.

### **8.6.2 The query processing subsystem**

All queries can be made through menus in the main window. The subsystem allows the user to access the database, retrieve the required data and display them in the map window. It also allows the user to retrieve the meta-data about data and

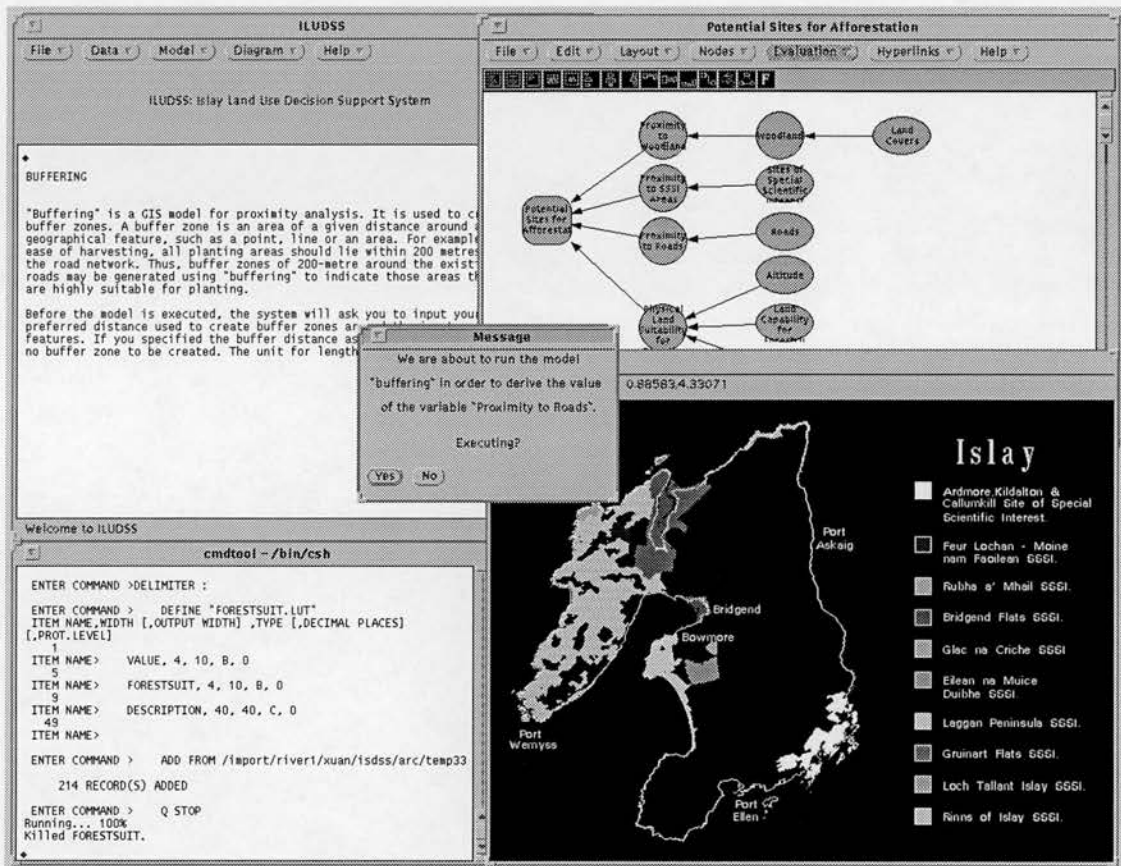


Figure 8.4 Screenshot of ILUDSS

knowledge about models available in the system. In addition, the user can retrieve existing spatial influence diagrams representing land use problem models. The query processing subsystem links the user query through a menu to a specific function or program, sends a message to the problem processor to call and execute it to perform a certain query operation. For instance, when the user makes a query on data, the system calls an appropriate AML program (an utility program) to pop up a list of data sets, and then retrieves the data selected by the user from the database in ARC/INFO,

and displays it in the map display window. The user can retrieve data and other information without the need to identify where the information is stored beforehand.

The help module is built using the wxHelp software delivered with HARDY. The wxHelp software is the platform-independent help system for the class library wxWindows (Smart 1993b). Our help system provides information about the system and its operations if requested. The **contents** page of our help system is shown in Figure 8.5.

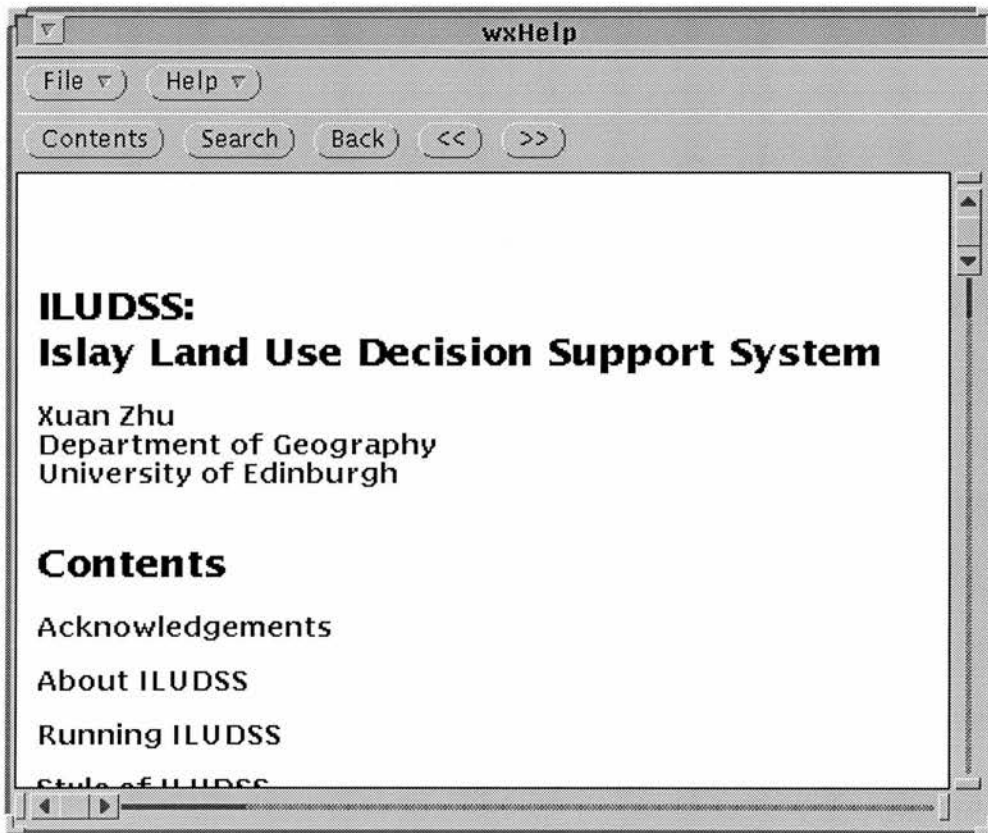


Figure 8.5 ILUDSS help system



### **8.6.3 The knowledge base**

ILUDSS uses five types of knowledge: domain knowledge about the problem domain, model knowledge about the GIS, analytical and rule-based models available in the system, utility program knowledge about the utility programs developed in the system, meta-data about the data in the GIS database and process knowledge about the land use problem solving process. The knowledge representation of the five categories of knowledge is as described in Chapter 6. They are encoded in CLIPS as separate knowledge base modules. The knowledge base structure of ILUDSS is similar to that shown in Figure 6.1. However, model knowledge and utility program knowledge are encoded in the same module in ILUDSS, i.e. the knowledge base of ILUDSS contains four separate knowledge base modules: the domain knowledge base module called VARIABLES, the tools knowledge base module called SOLVERS (containing the knowledge about models and utility programs), the meta-data knowledge base module called DATA, and the process knowledge base module called MAIN (the default module of CLIPS, see Section 7.2). The contents of each knowledge base module are the same as those described in Section 6.2.

### **8.6.4 The modelling subsystem and the problem processor**

ILUDSS employs spatial influence diagram-based mechanisms for the formulation and evaluation of land use problems, and for controlling selection and integration of data and models. Since the modelling subsystem and the problem processor are built and implemented in CLIPS, the spatial influence diagram-based mechanisms are on top of the CLIPS inference mechanism (see Section 7.2).

ILUDSS supports both automatic and user-assisted spatial modelling. For automatic modelling within the problem domain described in Section 8.3, the user can specify his or her land use interest and select a set of the attributes (indicating his or her preferences relating to the relevant evaluation factors) to identify land use potential for the land use of interest. The system will then formulate a land use problem model using its own knowledge, and existing data and models, present the

problem model graphically to the user in a spatial influence diagram, and evaluate it automatically.

The main land use interests on Islay include farming, forestry, peat-cutting, nature conservation, tourism and sport management. In the present implementation, the automatic modelling module in ILUDSS can deal with three land use types: farming, afforestation and peat-cutting, and three classes of attributes or factors: *physical land suitability*, *proximity* (to desirable and undesirable land features) and *area* (the required minimum area of each land parcel). **Land suitability** refers to the fitness of a given type of land for a specified kind of land use (FAO 1976). A proximity consideration may be the desire to locate afforestation sites near roads but a certain distance away from SSSI areas and apart from existing woodlands.

When the user chooses *proximity* as one of the factors or attributes for analysis, the system will prompt the user to specify which desirable land features he or she wishes the intended use to be near, and which undesirable land features he or she wishes the intended use to avoid. At present, the automatic modelling module allows the user to take into account proximity to roads, rivers, SSSI areas and existing woodlands. When the user specifies *area* as one of the attributes to be considered, the system will ask for the minimum area of each land parcel he or she requires.

The situation handled by the automatic modelling module can be any of the three possible intended land uses with an arbitrary combination of the three classes of attributes pertinent to the selected use. In other words, the user can choose one of the three land use types, and indicate the evaluation factors by selecting a set of attributes from the three classes available to evaluate the land use potential for the intended use. For example, if a user wants to identify the land use potential (the suitable sites) for future afforestation on Islay, and prefers the afforestation sites near roads, he or she can specify afforestation as the intended use, and the physical land suitability and the proximity to roads as two pertinent attributes. Note that the physical land suitability is assumed to be of concern in every case. After the land use type and the set of attributes have been specified by the user, the system will perform the reasoning and automate the formulation and evaluation of a land use problem model representing the user's land use problem. The results produced by the automatic

modelling module are land use potential for various land use interests specified by the user. They may be seen as indicative land use strategies formulated according to the user's preference of conditions for the intended uses.

ILUDSS works in a consultant mode, the expert system style, when performing automatic modelling. The application of automatic modelling requires absolutely no user knowledge about the use of different types of models and data employed for the problem solving. Neither does it require knowledge about HARDY, ARC/INFO and CLIPS, or expertise in evaluation of land use potential under the conditions pertaining on Islay. All that is required is a basic comprehension of the field together with understanding of the terminology of rural land use planning and spatial analysis in the GIS context, which is needed for a successful dialogue with the system.

The user-assisted modelling module in ILUDSS supports user participation in defining problems and developing models. For user-assisted modelling, the system provides the diagramming tool for the user to structure and represent the land use problems using spatial influence diagrams by employing existing GIS, analytical and rule-based models, and existing data, or by adding new data sets. The current system also allows the user to add simple analytical and rule-based models and use them in formulating land use problem models in spatial influence diagrams. The problem processor can evaluate any spatial influence diagram formulated using the data and models available in the system by the user. The user can also modify the land use problem models formulated by the automatic modelling module described above.

To develop an analytical model, the user-assisted module provides a facility for the user to enter a mathematical equation using the arithmetic operators (listed in Appendix G). The system will then translate the equation into an AML program. The user-assisted module also allows the user to add rule-based models which consist of an if-then rule using the logic operators listed in Appendix H and the arithmetic operators listed in Appendix G. In addition, the user can build reclassification tables to be used together with the GIS model *reclassify* or to be used for map display. Similar to adding analytical models, the added rule-based models and reclassification tables will be translated into AML programs or ARC/INFO remap tables (see ESRI 1991c) by the system and will be executed by ARC/INFO. The relevant information

about the models added is written into the tools knowledge base module (the SOLVERS module) automatically when they are added.

ILUDSS has no data acquisition facility. The user can create new data sets outside of ILUDSS, put them into the directory for data (see Appendix D) and add their meta-data into the system. For user-assisted modelling, the user should have knowledge and experience in spatial modelling in the GIS context.

### 8.6.5 Back-end subsystem

The subsystem contains the GIS database (encoded in ARC/INFO coverages and grids), the solvers for the GIS models and land suitability models described in Sections 8.4 and 8.5, and the necessary utility programs. All GIS model solvers are independent AML programs, which can be called by their names. The solvers of the physical land suitability models for agriculture and peat-cutting are reclassification tables encoded in INFO tables. The solver of the physical land suitability model for forestry is a CLIPS knowledge base module encoded in the CLIPS rule-based language. Currently, ILUDSS contains the utility programs for attribute data conversion from INFO table data to CLIPS facts (written in C), data conversion from CLIPS facts to INFO table data (written in AML), spatial data conversion between ARC/INFO coverages and ARC/INFO grids (AML programs) and map query and display (AML programs).

## 8.7 Implementation of ILUDSS

In this section, we will discuss the implementation of the main ILUDSS functions described in the previous section. Appendix E explains how to run ILUDSS. A sample session with ILUDSS is presented in Appendix F in order to give a more concrete idea of its operations. In the following, the notation **Menu:Option** refers to the menu item **Option** on the menu **Menu**.

### 8.7.1 Query

ILUDSS allows the user to query data, meta-data, model information and land use problem models.

For queries on data, the user may go to the **Data** menu on the menu bar in the main window (Figure 8.6). By selecting the **Data:Draw and query** menu item, the user gets a list of data, from which one data layer can be selected for query and display. The selected data layer will be displayed in the map display window. If the data layer has no associated key file, the user can use the map displayed to retrieve data directly from the database. Namely, the user can list attribute values for a feature on the map by pointing to it using the mouse. By choosing the **Data>Delete data** menu item, the user can delete any data layer in the database. At the same time a data layer is deleted, its associated meta-data is also deleted from the meta-data knowledge module. The meta-data can be retrieved by selecting the **Data:Show metadata** menu item. The contents of the meta-data to be displayed are exactly similar to those defined in Chapter 5.

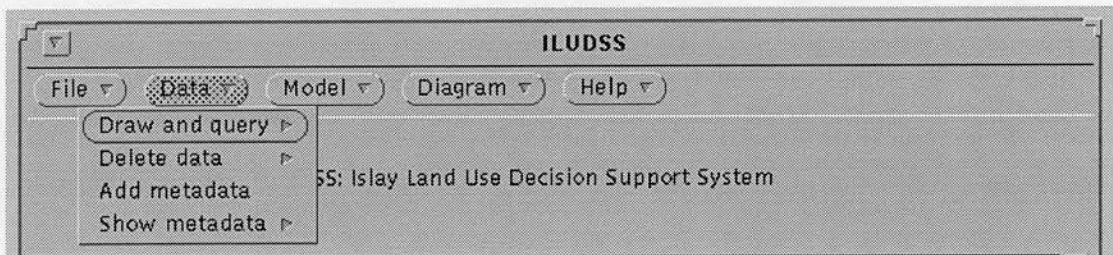


Figure 8.6 **Data** menu in the main window

The information about all existing models can be accessed through the **Model** menu on the menu bar in the main window (Figure 8.7). Models are grouped into three categories: GIS, analytical and rule-based models. Reclassification tables are taken as a special form of rule-based models. GIS models include all of those listed in Table 8.2. They are divided into four groups: location-specific analysis, regional summary, neighbourhood analysis and geometric measurement (see Chapter 3 and

Table 8.2). By choosing an appropriate item from the **Model** menu, the user can get a list of the existing models corresponding to that group. Selecting one from the list, the user can retrieve the information about the model. Information about models describes their functions and usage in a spatial influence diagram.

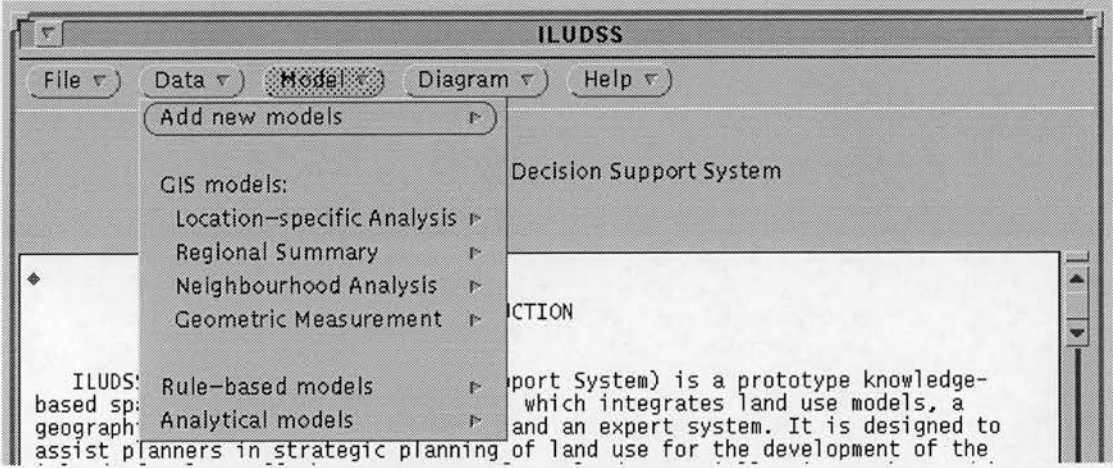


Figure 8.7 **Model** menu in the main window

ILUDSS also allows the user to retrieve and display the existing land use problem models represented in spatial influence diagrams through the **Diagram** menu (Figure 8.8). By choosing the **Diagram:Select** item, the user can get a list of land use problem models. After the user makes a selection from the list, a diagram window will pop up with the selected model displayed in it. The user can view the structure of the problem model, retrieve the attributes associated with each variable depicted in the model, and modify or evaluate the problem model if he or she would like (see below).

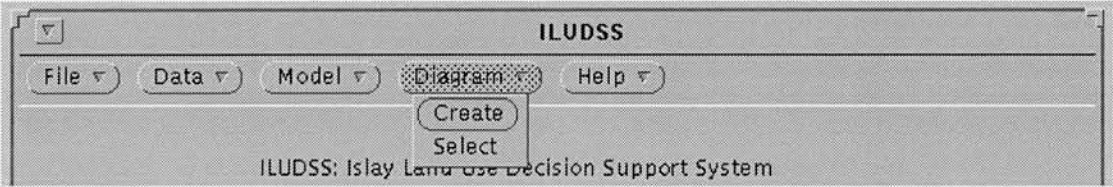


Figure 8.8 **Diagram** menu in the main window

## 8.7.2 Automatic modelling

Automatic modelling is the key feature of ILUDSS, and it is a central requirement for designing effective KBSDSS systems. It involves the automatic formulation and evaluation of land use problems.

### 8.7.2.1 Formulation of land use problem models

Within the problem domain and situations described in Sections 8.3 and 8.6, ILUDSS can formulate a land use problem model using the domain-specific knowledge in the knowledge base. The formulation of a land use problem model in ILUDSS involves three phases: specification of a land use type and a set of attributes (or factors) to be used to evaluate the land use potential for the land use type, building of an objective model and formation of a spatial influence diagram. It starts by selecting the **File:Run** item from the **File** menu in the main window (Figure 8.9). The sample session in Appendix F will be referred to in this and the next sections for illustrative purposes.

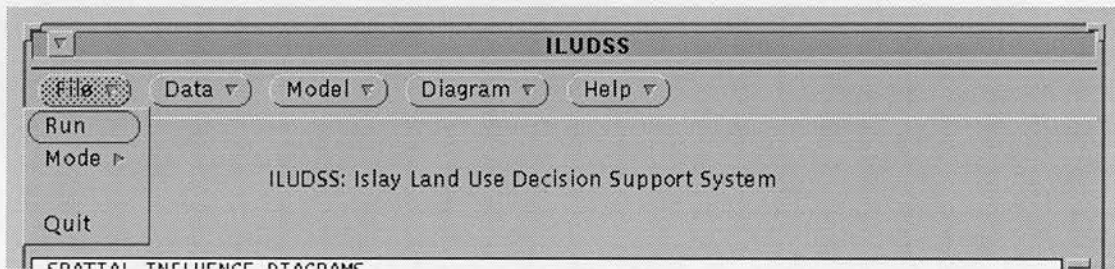


Figure 8.9 **File** menu in the main window

#### (1) Specification of a land use type and a set of attributes for evaluation

As described in the previous section, ILUDSS can currently deal with the three land use types: farming, forestry and peat-cutting. The system displays a list of land

use types, as shown in Figure 8.10. If the user's land use interest is not in the list, the item **Quit** should be selected. The system will then return to its previous status. After the user selects a land use type from the list, the system will give an attribute list to the user, as shown in Figure 8.11. At present, ILUDSS takes account of three classes of attributes: *physical land suitability*, *proximity* (to desirable and undesirable land features) and *area* (the required minimum area for each land parcel). The user can choose any combination of the attributes for determining the potential sites of the specified land use type, according to his or her preference relating to the evaluation factors. The selections of the land use type and a set of the attributes are made by the user through simply clicking menu items. If the user selects *proximity* as one of the attributes for analysis, the system will further ask him or her for the desirable and undesirable land features to be considered in the proximity analysis. At present, only four types of land features can be considered in the proximity analysis. They are roads, rivers, SSSI areas and existing woodlands. In our sample session, the user specifies *afforestation* as the land use interest, and *physical land suitability*, *proximity to roads* and *SSSI areas* as the attributes being considered in identifying the potential sites for afforestation.

## (2) Building of an objective model

We may recall that an objective model represents the decision maker's preferences over the set of possible outcomes that could result from any of the choices being considered. Here, an outcome is land use potential for the specified land use type. The choices are the attributes or factors to be used in identifying the land use potential. Therefore, an objective model takes the formal specification of an outcome and produces a numerical score or a ranked category that corresponds to the desirability of that outcome. In other words, an objective model evaluates the value of a goal attribute (or the value node in a spatial influence diagram) according to the attributes that directly influence it.



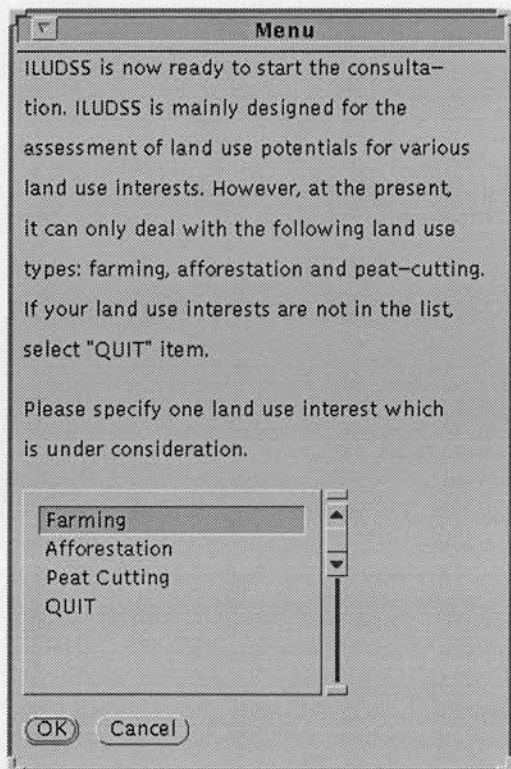


Figure 8.10 Pop-up land use list

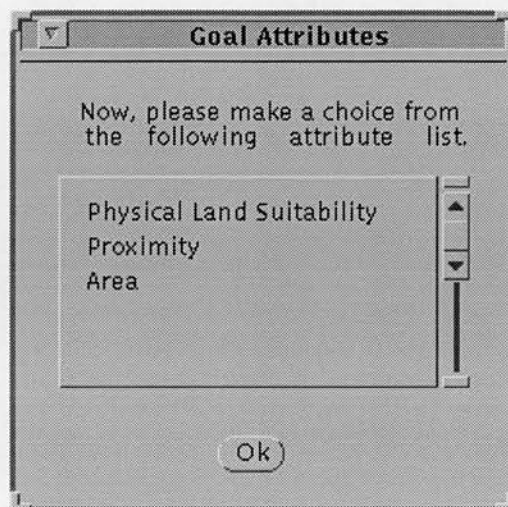


Figure 8.11 Pop-up attribute list

Each objective model accounts explicitly for a different set of attributes for a certain type of land use. ILUDSS builds an objective model using the domain-specific rules in the knowledge base for a given consultation by indicating which of the three classes of attributes are pertinent to the land use interest at hand. In the sample session, an objective model called *forestsitel* is built, which accounts for the physical land suitability for forestry, proximity to roads and proximity to SSSI areas. Indeed, the *forestsitel* is a model containing lines of AML that combines data about the physical land suitability for forestry, the proximity to roads and the proximity to the SSSI areas, and identifies the potential areas for afforestation by including the land physically suited to forestry and within a certain distance (to be specified by the user) to roads, but excluding the SSSI areas. The AML program is written by ILUDSS according to the user's choices of the land use type and the attributes under consideration. Rules in the domain knowledge base module (the VARIABLES module) instruct ILUDSS to write an appropriate objective model using AML or

other languages according to specific circumstances. The process of building an objective model is invisible to the user.

Once an objective model has been built, the process of formulating a land use problem model begins.

### (3) Formation of a spatial influence diagram

Formulating a land use problem model is indeed a process of forming a spatial influence diagram. The process follows the formulation mechanism described in Chapter 5 that starts from the value node until all border nodes are created and the problem model is completed, but it is implemented on top of the CLIPS inference mechanism.

In the sample session, at the beginning, the system sets the variable “Potential Sites for Afforestation” as the value node, the attributes “Physical Land Suitability for Forestry”, “Proximity to Roads” and “Proximity to SSSI Areas” as the direct predecessors of the value node, and puts these predecessors into a predecessor set. Thus, a starting-point spatial influence diagram is formed, as shown in Figure 8.12(a). Then, the system picks up one of the predecessors of the value node from the predecessor set, say “Physical Land Suitability for Forestry”, and asserts it into the fact-list. When it is asserted, a rule controlling the search of the predecessors of a node is activated and placed onto the agenda. When the rule is fired, it activates another rule to check the meta-data knowledge base module (the DATA module), which determines that there is no data (value) about the physical land suitability for forestry. Thus, the third rule is activated to look for its predecessors in the domain knowledge module and put them into the predecessor set. Here, “Altitude”, “Land Capability for Agriculture” and “Land Capability for Forestry” are its predecessors. Thus, the three variables are added to the starting-point spatial influence diagram and a new spatial influence diagram is formed as shown in Figure 8.12(b). After that, the

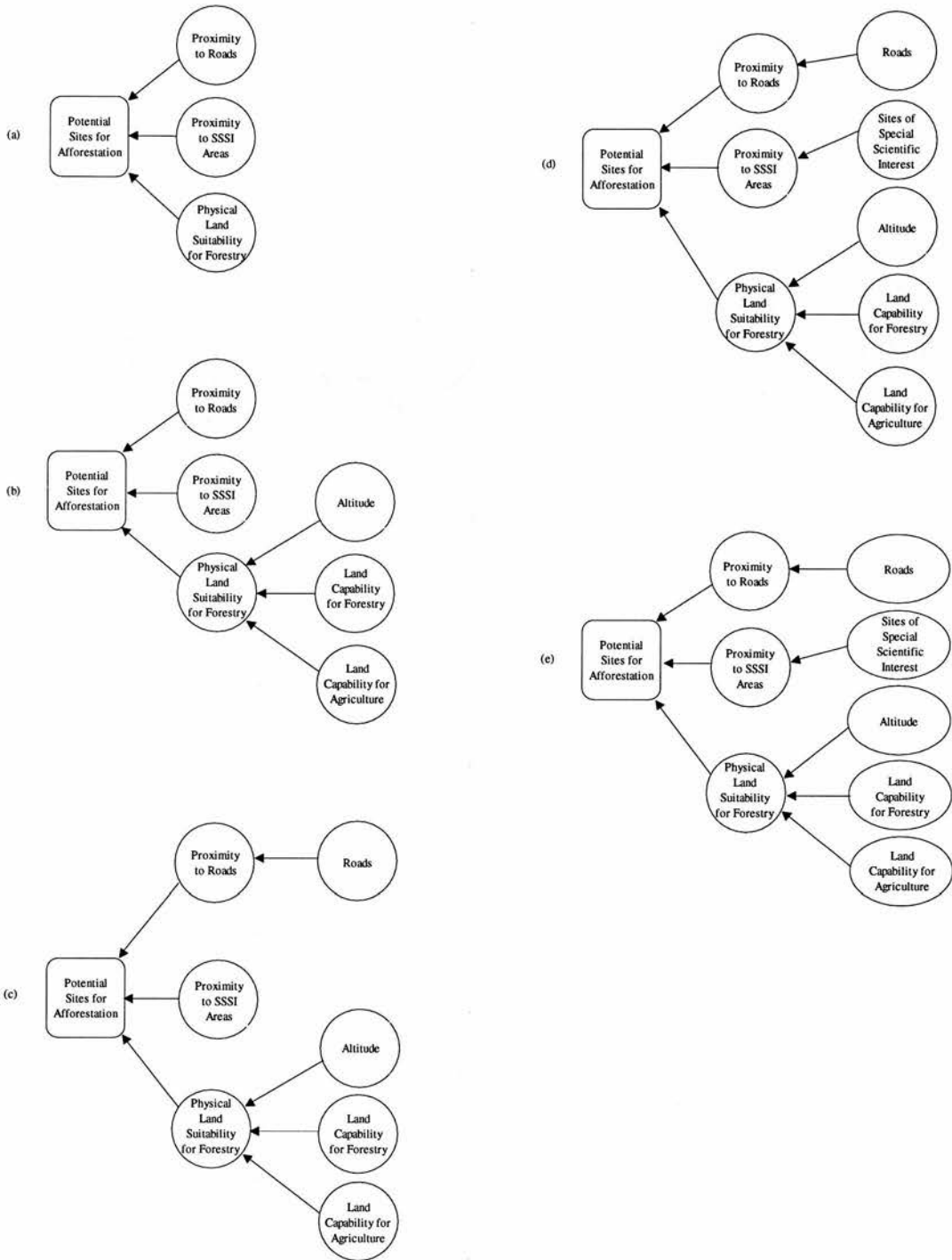


Figure 8.12 Formation of a spatial influence diagram

system picks up another variable from the predecessor set, say “Proximity to Roads”, asserts it to the fact-list and repeats the process above. The node “Proximity to Roads” is expanded with “Roads” (Figure 8.12(c)). Next, the node “Proximity to SSSI Areas” is expanded with the node “Sites of Specific Scientific Interest” (Figure 8.12(d)). Afterwards, the system looks for the values for “Altitude”, “Land Capability for Agriculture”, “Land Capability for Forestry”, “Roads” and “Sites of Specific Scientific Interest” in succession. They all exist in the database. Thus, the system sets them as the border nodes. Finally, a complete spatial influence diagram is obtained, as shown in Figure 8.12(e).

During the formulation process, all values of the attributes associated with each type of node defined in Chapter 4 are assigned to each node (with an additional attribute *internal-name* to be discussed in Section 8.7.3.1). As the formulation process does not produce any question for the user, it is not directly visible in the sample system.

At the start of the formulation process, ILUDSS calls a HARDY function to create and display a diagram window. Once a node is added to the spatial influence diagram, its image is created and displayed in the diagram window using the HARDY diagramming functions. After all node images have been created, they are appropriately connected by arcs. Finally, a HARDY built-in layout algorithm is called to place all images (including nodes and arcs) in suitable positions and form a graph representing the spatial influence diagram. Thus, a land use problem model is formulated and presented graphically to the user.

#### 8.7.2.2 *Evaluation of land use problem models*

After a land use problem model (a spatial influence diagram) has been formulated and presented in a diagram window, the user may go on to evaluate it. The evaluation of a land use problem model is activated by clicking the **Evaluation** menu on the menu bar in the diagram window (Figure 8.13) in which the problem model is displayed.

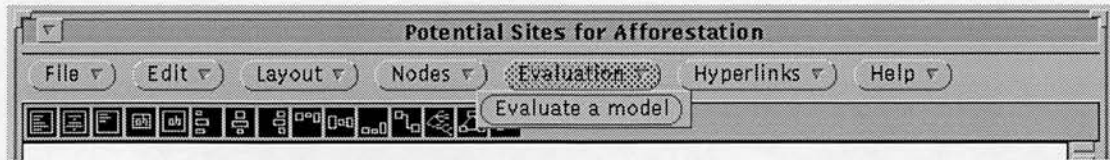


Figure 8.13 **Evaluation** menu in a diagram window

Before the evaluation process actually starts, ILUDSS calls the appropriate HARDY functions to capture the attributes of each node from the graphical representation of the spatial influence diagram, convert them into a set of CLIPS facts and assert them into the fact-list. Each fact describes a node with the information including the *name*, *node type* (value, chance or border), *predecessors*, *successors*, *spatial-relation*, *attribute-relation* and *associated-data*. The evaluation process is characterised by a dialogue with the user. In such a dialogue, the system asks the user to input data when required, informs the user about the status of the operations, gives explanations and helpful messages where necessary. The whole process is controlled by the evaluation mechanism described in Chapter 5. As with the formulation of a spatial influence diagram as described above, it is governed by rules and is implemented on top of the CLIPS inference mechanism. The system is responsible for the overall flow of the evaluation, i.e. it determines the order in which each node in the spatial influence diagram is evaluated, and triggers selection, parameterization, execution of model solvers and utility programs, and interpretation of results.

The evaluation starts from the border nodes. In our sample session, at the beginning, there are five border nodes in the spatial influence diagram. They are “Altitude”, “Land Capability for Agriculture”, “Land Capability for Forestry”, “Roads” and “Sites of Specific Scientific Interest”. Since their values are stored in the database, the system first transfers their values to each of their direct successors, and then deletes them from the fact-list (i.e. they are deleted from the diagram). However, this process is invisible to the user. Once the five border nodes are removed, a rule that controls the evaluation of a node is activated three times, since

the three nodes “Physical Land Suitability for Forestry”, “Proximity to Roads” and “Proximity to SSSI Areas” have got the values of all their predecessors and they can be evaluated. Each time the rule is fired, a node is to be evaluated. The order of evaluation of the three nodes is determined by the current conflict resolution strategy of CLIPS. We set the current conflict resolution strategy as the **breadth strategy** (see Appendix A), so that the node which first gets the values of all its predecessors will be evaluated first. Here, the system first evaluates the node “Physical Land Suitability for Forestry” using a rule-based model *forestsuit* and a GIS model *overlay* based on the data of “Altitude”, “Land Capability for Agriculture” and “Land Capability for Forestry”. Some utility programs are used during the evaluation for converting INFO data to CLIPS facts and vice versa. The operations of utility programs in the system can only be monitored in the terminal window, and are not reported to the user. The system automatically selects the solvers of *forestsuit* and *overlay*, and utility programs for data conversion, assigns appropriate values to the required arguments of the selected solvers and utility programs, and executes the solver of the model *overlay* first and then the solver of the model *forestsuit*. Before the execution of the solver of the model *forestsuit*, the utility program for transforming INFO data into CLIPS facts is executed. After the execution of the solver of the model *forestsuit*, the utility program for transforming CLIPS outputs into INFO data is executed. All these are controlled by rules in the tools knowledge base module (the SOLVERS module) and are not directly visible to the user.

After the value of “Physical Land Suitability for Forestry” is obtained, it becomes an assessed node, and its value can be displayed in a map in the map display window if the user requests. Then the system propagates its value to the value node, removes it from the fact-list, and proceeds to evaluate the node “Proximity to Roads”. The value of “Proximity to Roads” is derived from the data for “Roads” using the GIS model *buffering*. The system first looks for the appropriate solver of the GIS model, and assigns the values to its arguments. One of the arguments of the *buffering* solver is the buffer distance, which has to be specified by the user. Thus, the system prompts the user to input his or her preferred distance to roads. After the value has passed to the solver, the system activates the solver and creates a buffer

zone around the roads. The evaluation of the node “Proximity to SSSI Areas” has a similar process. When all the predecessors of the value node “Potential Sites for Afforestation” obtain the corresponding values, it is evaluated. The value of the value node is derived by the objective model *forestsitel* built during the formulation of the land use problem model (see the previous section). Since the objective model is composed of several GIS operations (here ARC/INFO), it is also called a GIS model. The result of the execution of the model *forestsitel* is the outcome required by the user. In other words, it is a solution to the land use problem specified by the user. The final result is interpreted using a map and presented to the user in the map display window, as shown in Figure A.6. This completes the session run.

The land use problem model displayed in the diagram window can be run repeatedly. Each time the user alters the values of some parameters of the problem model, such as the preferred distance to roads and the preferred distance to SSSI areas, an alternative solution can be obtained. The land use problem model can also be saved for later use by going to the **File:Save as** menu item in the diagram window (Figure 8.14). As a convention, a data file storing a spatial problem model (or a spatial influence diagram) is named with the extension *.dia*.

### 8.7.3 User-assisted modelling

In addition to automatic modelling described above, ILUDSS supports user participation in defining problems and developing models without the need to have the knowledge about the implementation tools in the system. User-assisted modelling provides the user with the facilities for construction and modification of problem models using spatial influence diagrams, addition of new analytical and rule-based models, and acquisition of meta-information about new data and new models.

#### 8.7.3.1 Construction and modification of problem models

ILUDSS employs the HARDY diagram card, the diagram window in ILUDSS, as a spatial influence diagram editor for the user to construct and modify problem models.

A spatial influence diagram represents a problem model. So, construction of a problem model is to create a fully specified spatial influence diagram.

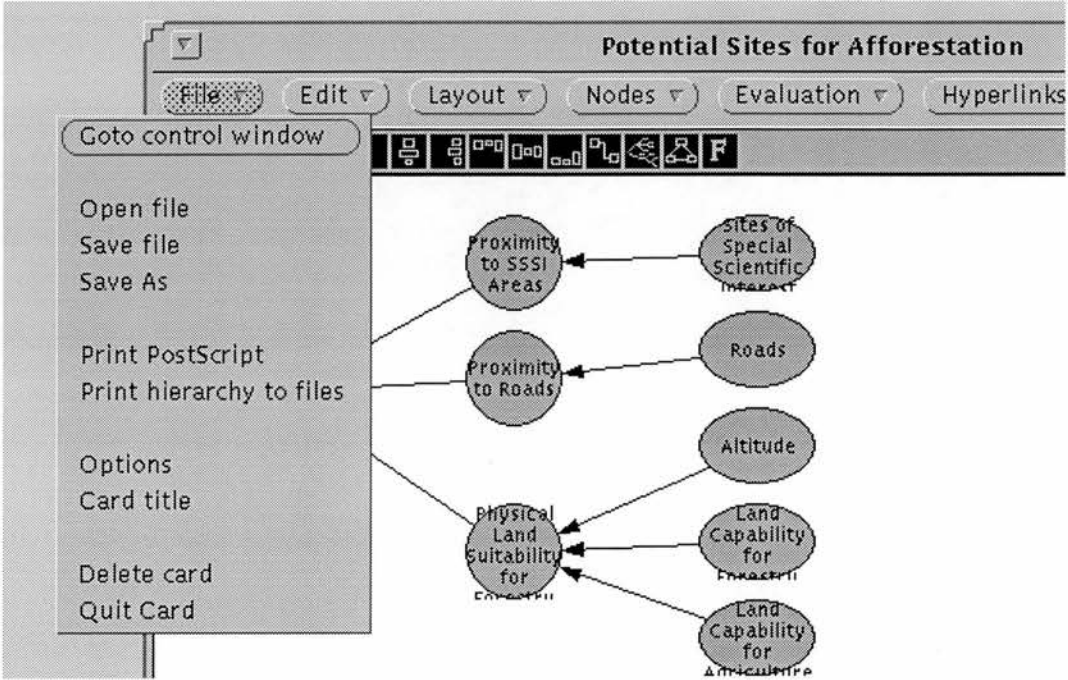


Figure 8.14 File menu in a diagram window

The diagram type for spatial influence diagrams is pre-defined using the HARDY diagram manager as shown in Figure 7.2. The user of ILUDSS does not need to define it. When the user selects the **Diagram:Create** menu item from the main window (see Figure 8.8), a diagram window pops up and a spatial influence diagram editor is activated. In a diagram window, the **Evaluation** menu on the menu bar is the ILUDSS specific function for activating the evaluation process of a spatial influence diagram. The others are HARDY functions (for details, see Smart 1993). The **Nodes** menu is used to create images of three types of nodes: value, chance and border (Figure 8.15).



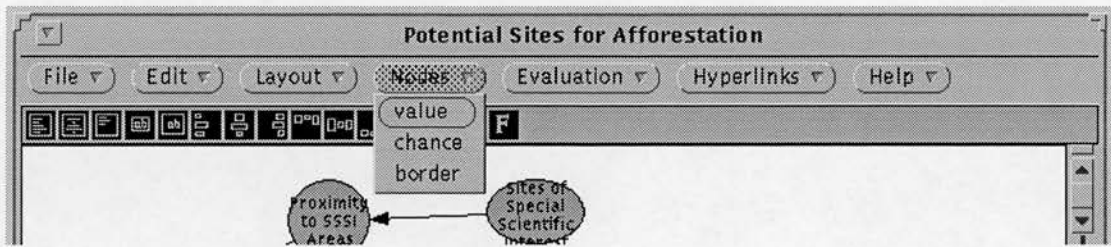


Figure 8.15 Node menu in a diagram window

Each node has several attributes attached. These attributes are the seven categories of information defined in Section 4.4, plus *internal-name*. An internal name is a word that starts with a letter and is followed by zero or more printable ASCII characters. However, a word should not be longer than 12 characters and may not contain any of the following characters

< | & ( ) ;

An internal name names a node to facilitate computer processing.

The attributes attached to a chance node include its *name* (the label displayed in the diagram), *node-type* (chance), *predecessors* (indicated using their internal names), *successors* (indicated using their internal-names), *attribute-relation*, *spatial-relation* and *internal-name*. The attributes attached to a value node include its *name* (the label displayed in the diagram), *node-type* (value), *predecessors* (indicated using their internal names), *attribute-relation*, *spatial-relation* and *internal-name*. For a border node, the attributes include its *name* (the label displayed in the diagram), *node-type* (border), *successors* (indicated using their internal-names) and *associated-data*. The *associated-data* is the name of the data file storing the data representing the value of the border node. It can be any data set or data layer existing in the system. If it is not specified, the system will ask the user to input the data, when it is being evaluated. The *attribute-relation* and *spatial-relation* can be specified using any analytical, rule-based and GIS models available in the system.

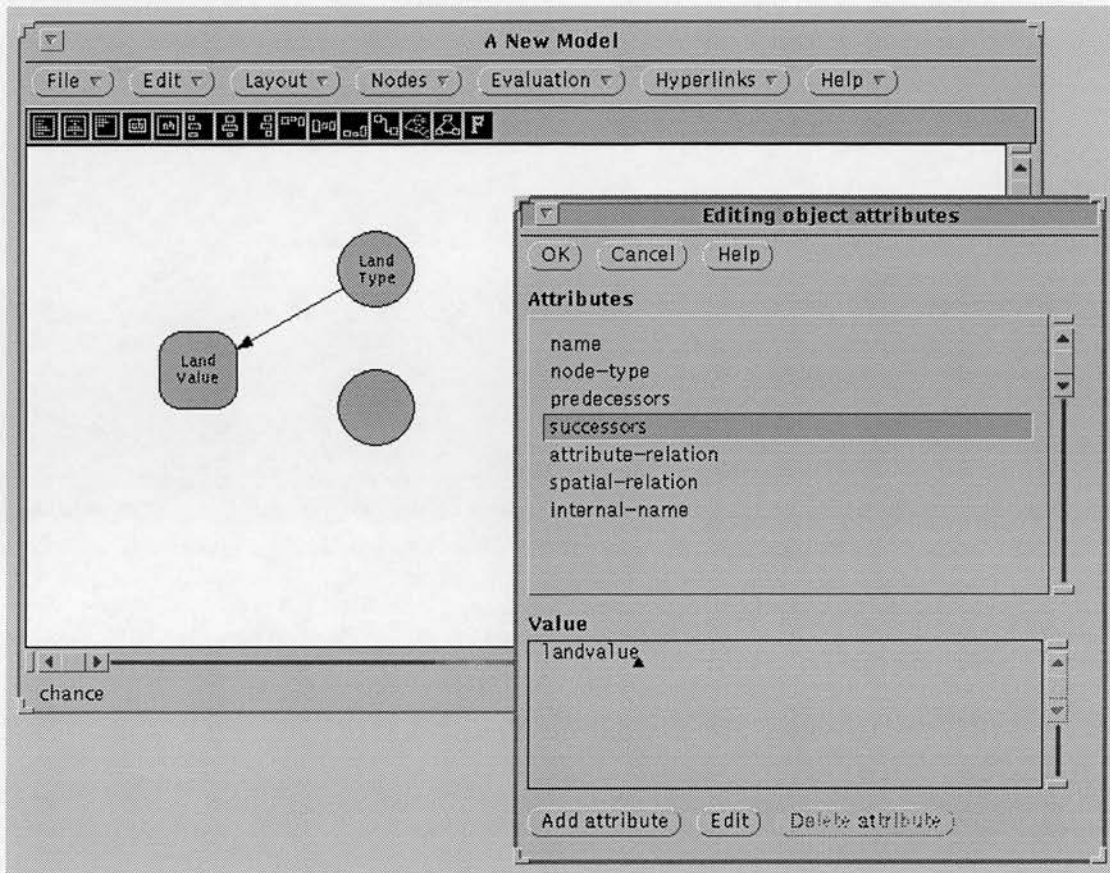


Figure 8.16 Spatial influence diagram editor (Diagram window with an attribute editor)

To create a spatial influence diagram, choose an item from the **Nodes** menu (see Figure 8.15) according to the type of the node to be created. A new node appears at a default position on the screen with no label. Drag the shape to the desired point on the screen by holding down the left button over the shape, moving the mouse and releasing. Then, edit the attributes of the node. To edit the attributes associated with a node, point to the node using the mouse, hold the control key and click left-hand button. A window pops up with all the available attributes and a text editing window for assigning a value for each attribute (Figure 8.16). Having clicked on an attribute, a string can be typed in to represent its corresponding value. After finishing assignment of values to all attributes, the user may press the **OK** button and return to the diagram window. Having created two nodes, hold down the right mouse button,

drag the mouse from the source node to the destination node and release the button, and then an arc between the two nodes is created. In this way, a complete spatial influence diagram can be created and modified. After finishing forming the spatial influence diagram and making sure that there is no errors in the diagram (note that the system can not detect any errors), the user may press the **Evaluation** menu to start to evaluate the spatial influence diagram (see Section 8.7.2.2).

Each individual model (analytical, rule-based and GIS models) can also be used by constructing a spatial influence diagram. For example, consider that one wants to create a 200-meter wide buffer zone around the roads using the model *buffering*. Assuming the map data layer “roads” depicting the roads already exists, a two-node spatial influence diagram corresponding to this operation may be drawn in a diagram window as shown in Figure 8.17. Here, “Roads” is drawn as a border node (representing the map “roads”), and “Road Buffer” is the value node (representing the derived map depicting the buffer zone around the roads). When drawing the spatial influence diagram, the values of the attributes associated with each node can be assigned using the attribute editor following the procedure described above. For the node “Roads”, the value of *name* is Roads, the value of *node type* is border, the value of *successors* is roadbuf (the internal-name of the node “Road buffer”), the value of *associated data* is roads, and the value of *internal-name* is roads. For the node “Road Buffer”, the value of the attribute *name* is Road buffer, the value of *node type* is value, the value of *predecessors* is roads, the value of *spatial-relation* is buffering, the value of *attribute-relation* is unspecified, and the value of *internal-name* is roadbuf. After the spatial influence diagram has been fully specified as above, the system will evaluate this spatial influence diagram, execute the GIS model *buffering* and create a map data layer showing a 200-meter wide buffer zone around the roads.

Thus, the use of GIS and other types of models in ILUDSS is to create a fully specified spatial influence diagram that consists of one value node and a set of border nodes. After such a spatial influence diagram has been created, the system will evaluate it and execute the model or models specified as the spatial relation or/and attribute relation of the value node. In fact, the value node represents an output of the

model (or models), and the border nodes represent the required inputs of the model (or models).



Figure 8.17 A spatial influence diagram for creating a buffer zone using the GIS model *buffering*

By using spatial influence diagrams to execute models, the user is only required to create a fully specified spatial influence diagram as described above. The user does not need to worry about the proper sequence of parameters and other implementation details of the models. ILUDSS will then use its own knowledge to select appropriate model solvers and necessary utility programs, assign the required parameters to the selected model solvers and utility programs, and activate appropriate software tools (e.g. ARC/INFO or CLIPS) to execute them.

### 8.7.3.2 Addition of new analytical and rule-based models

ILUDSS allows the user to develop simple analytical models using arithmetic expressions, and develop simple rule-based models using logic expressions and arithmetic expressions, in order to meet the needs of the problem-solving task.

An analytical model can be developed within ILUDSS. This can be done by selecting the **Model:Add new models:Analytical models** menu item in the main window (Figure 8.18). The *Add a new analytical model* window shown in Figure 8.19 will pop up. The user first gives the model a name, and then enters a formula using an arithmetic expression with the arithmetic operators listed in Appendix G. The system will use its default tool (ARC/INFO) to execute it. Note that the system can allow only one arithmetic expression on a single line.

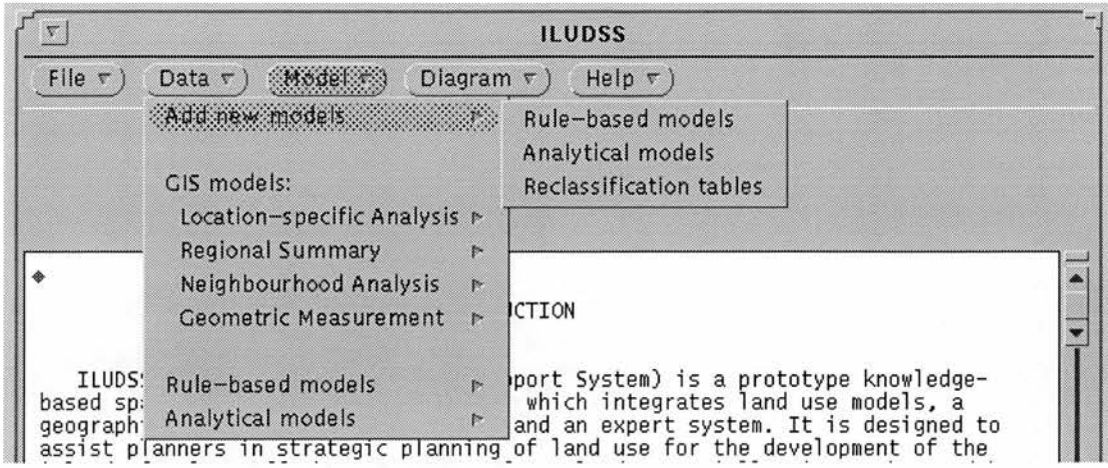


Figure 8.18 Add new models menu in the main window

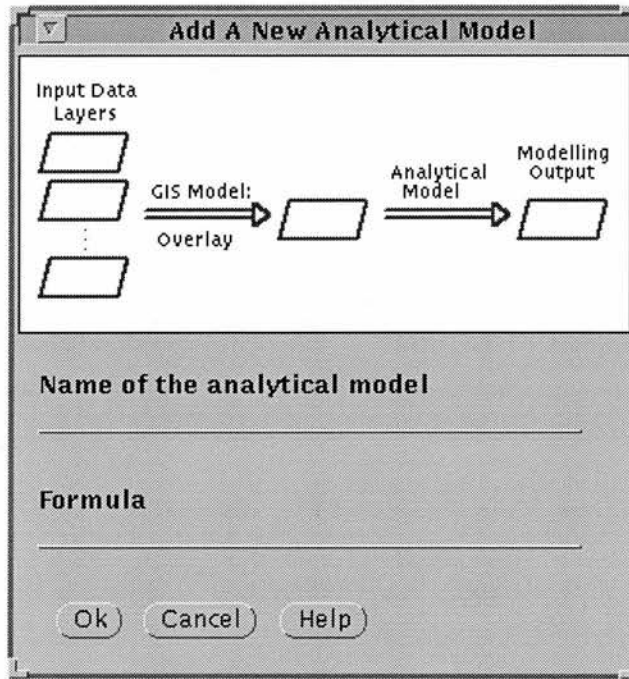


Figure 8.19 A pop-up window for adding an analytical model

A rule-based model can be developed by selecting the **Model: Add new models: Rule-based models** menu item in the main window (see Figure 8.18). The

Add a rule-based model window shown in Figure 8.20 will pop up. Through this window, the user may specify the name of the model, and enter a single if-then rule by following the format described in Section 8.5. The system will then convert the rule into an AML program and execute it by calling ARC/INFO.

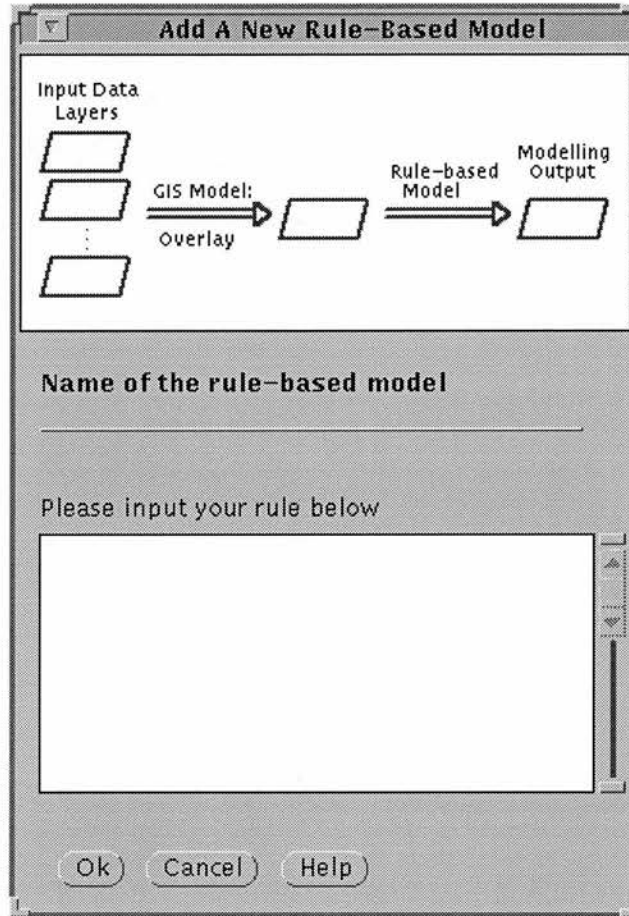


Figure 8.20 A pop-up window for adding a rule-based model

Reclassification tables can be created by selecting the **Model:Add new models:Reclassification tables** menu item in the main window (see Figure 8.18). The window for creating a reclassification table is as shown in Figure 8.21. The type specification *number* is equivalent to specifying *integer* and *float*. A reclassification table is used by the GIS model *reclassify* or for display of values in the form of maps. A reclassification table allows the user to change values of an attribute to alternative

values to create a new attribute or to specify the symbols to be used for values when displaying them using a map.

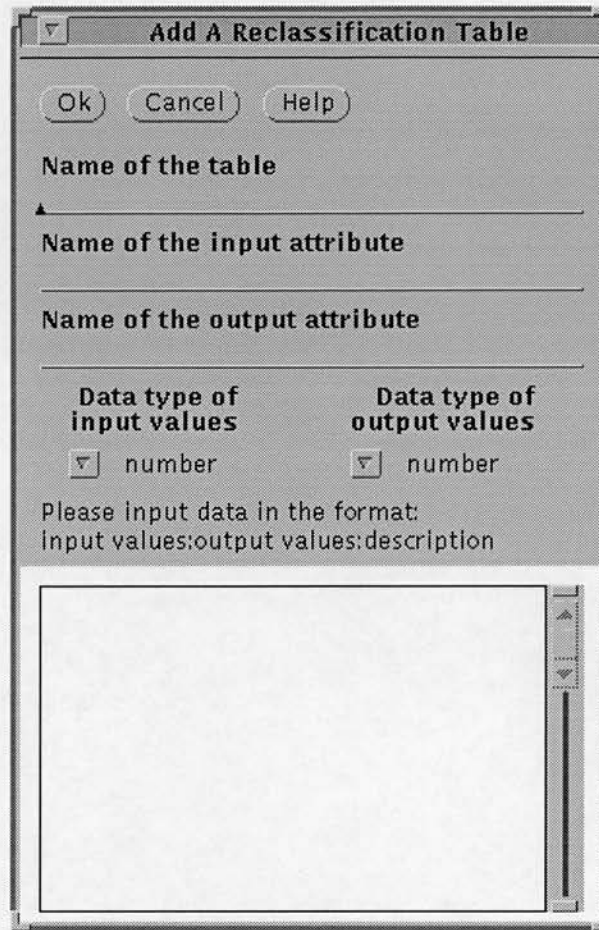


Figure 8.21 A pop-up window for adding a reclassification table

A reclassification table consists of three parts. For GIS modelling, the first part identifies the values of an attribute to be reclassified to create a new attribute of a geographical feature; the second part is the reclassified values for the new attribute. The third is the description of the reclassified values. For map display, the first part is the values to be displayed; the second part is the symbols to be used for displaying these values and the name of the output attribute should be SYMBOL. The shade

colour set used in ILUDSS is described in Appendix I. The three parts are separated using a delimiter “:”. The following is an example:

2 : 1 : mostly suitable  
5 : 1 : mostly suitable  
7 : 2 : suitable  
9 : 3 : moderately suitable  
10 : 4 : poor  
11 : 4 : poor

Note that a reclassification table in this system can only perform single class assignment. For raster data processing, the input and output values should be integers, and the first part should be named VALUE.

Currently, newly added rule-based and analytical models (except reclassification tables) are supposed to be used together with the GIS model *overlay* for deriving a new attribute and its values from several map data layers. So, it is recommended to use the GIS model *formula* to get a user-defined analytical model and use the *if-then* model to get a user-defined rule-based model. They will automatically combine an analytical model or a rule-based model created by the user with the GIS model *overlay* into a single GIS model.

#### 8.7.3.3 Acquisition of meta-data

ILUDSS has no data acquisition function. However, when a new data set (in ARC/INFO coverages or ARC/INFO grids) is added to the system, its meta-data should be added to the meta-data knowledge module. The meta-data can be acquired by selecting the **Add metadata** menu item from the menu **Data** on the menu bar in the main window (see Figure 8.6). A pop-up window as shown in Figure 8.22 will be displayed. The user can enter the information as required in the window. All the information will be translated into the meta-data knowledge module.



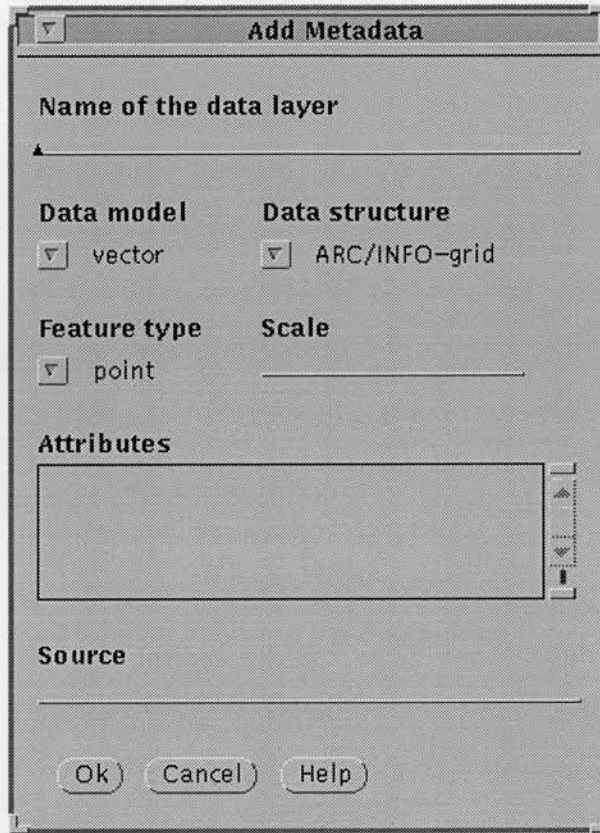


Figure 8.22 A pop-up window for acquisition of meta-data

## 8.8 Features and Limitations with ILUDSS

Within the overall framework for KBSDSS systems presented in this thesis, ILUDSS concentrates on the spatial influence diagram-based representation and mechanisms for spatial modelling. Land use problems addressed by ILUDSS are represented in spatial influence diagrams. Spatial influence diagrams are a “front-end” for almost every modelling activity in ILUDSS.

Within the problem domain addressed by the system, ILUDSS can automatically formulate a land use problem model according to a user's needs and preferences for the assessment of land use potential, and then present the problem

model graphically to the user in a spatial influence diagram. The spatial influence diagram here provides an effective vehicle for communication with the user so that the user can view a graphical display of the problem model and readily apprehend the overall structure and relevance among the land and environmental variables depicted in the graph. The directed arcs in the graph also indicate the order for the solution of the problem. The system will then evaluate the problem model automatically by integrating the database, knowledge base and models, and finally present the modelling results to the user in maps. The spatial influence diagram remains visible throughout the consultation. It can help the user understand the nature of the land use problem and clarify the modelling process. ILUDSS supports automatic land use problem modelling, and is intended to be used by planners and managers who may have limited experience in GIS and spatial modelling, but have basic comprehension of the terminology of land use planning and the way spatial analysis works in a GIS context.

On the other hand, ILUDSS allows user participation in defining problems and in developing models. It can support user-assisted land use problem modelling so that the users who have general GIS and spatial modelling knowledge can build land use models for other land use problems. For such a purpose, the system provides facilities for the user to create and modify the land use problem models graphically by following the definitions of spatial influence diagrams. Here, the spatial influence diagram provides a tool for the user to represent and structure land use problems and build land use problem models. The user may firstly structure the land use problem using a spatial influence diagram, describing the major land and environmental variables involving in the problem and their relevance, then define the values of all attributes attached to each node of the diagram, according to the relationships between nodes, the data, GIS models, analytical models and rule-based models available in the system, or by adding new data and adding new models. After a land use problem model has been created or modified by the user, the system will automatically evaluate the model and produce the results.

Moreover, ILUDSS allows not only the solution of a single land use problem, but also the solution of two or more related problems within one consultation. Any

land or environmental variable that is needed for more than one of the land use problems is evaluated only once. For example, after solving the land use problem presented in the sample session in Appendix F, the user may like to identify the potential sites for afforestation according to the physical land suitability for forestry, proximity to roads and proximity to SSSI areas as well as the proximity to woodlands. In this land use problem, the user also wants to exclude the existing woodlands from the future afforestation areas. Then, the system will formulate a land use model as shown in Figure 8.23. Since the variables “Physical Land Suitability for Forestry”, “Proximity to Roads” and “Proximity to SSSI Areas” have been evaluated in the last session, their values are already in the database. Thus, they are border nodes in the new problem model. They will not be evaluated again. However, if the user wants to evaluate the same variable repeatedly for different land use problems, the data corresponding to the variable has to be deleted before starting formulation of another land use problem model.

Besides its flexibility in modelling, ILUDSS provides an interactive and learning environment. The graphical display of a spatial influence diagram gives the structure of a land use problem and provides an explanation of the modelling process. In addition, ILUDSS provides the relevant information for the user during the modelling process. Pop-up dialogue windows are used to get information from the user, report the status of the reasoning, and inform the user of the model operations which are about to be undertaken or are currently in progress. Thus, at all times, the user can be made aware of how any given inference was made. The text window in the main window acts as an information window. The information offered by ILUDSS in the information window is mainly for explanation of the current modelling process and related GIS and spatial modelling terminology at appropriate times. Moreover, the map query function provides a facility for retrieving the values of attributes at every location in the study area if the map has no keys.

Another feature is that ILUDSS has two spatial data processing modes: vector and raster. In the vector data processing mode, ILUDSS uses vector data (e.g. ARC/INFO coverages) and GIS model solvers operating on vector data (if they exist) for all modelling tasks within a consultation, while in the raster data processing

mode, it uses raster data (e.g. ARC/INFO grids) and GIS model solvers operating on raster data. The raster data processing mode is the default. The user can choose or change the spatial data processing mode by going to the **File:Mode** item in the main window (see Figure 8.9).

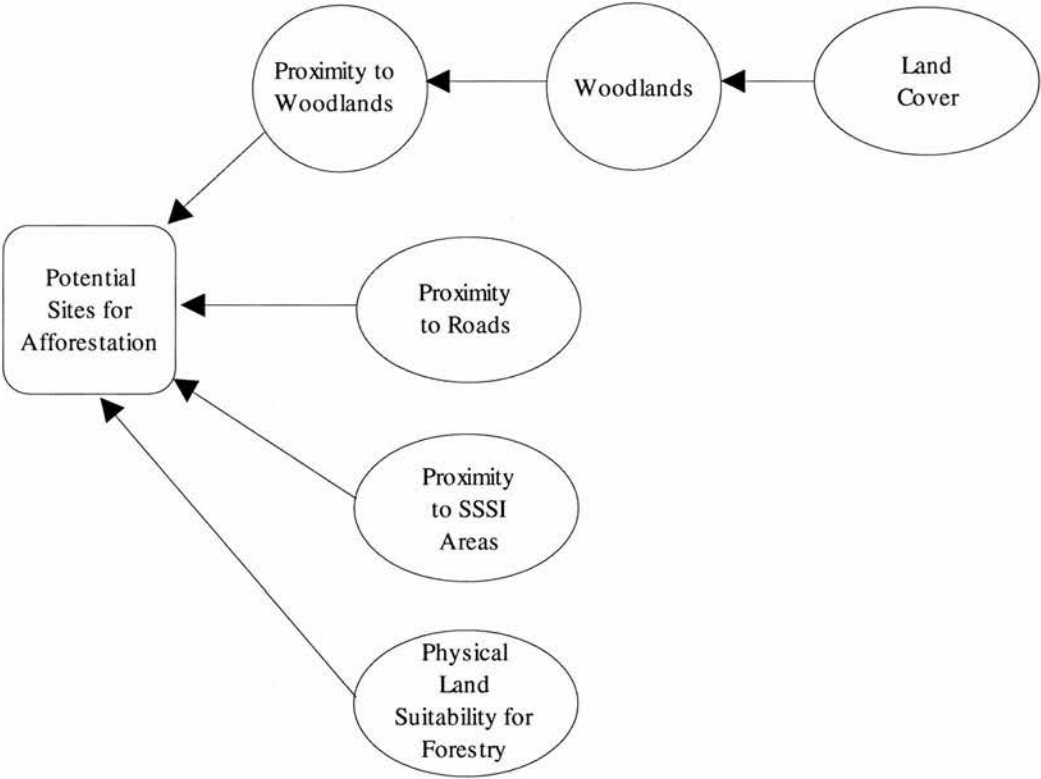


Figure 8.23 Land use problem model for identifying potential sites for afforestation according to the physical land suitability for forestry, the proximity to roads, SSSI areas and existing woodlands

Generally, a user of ILUDSS needs neither to understand the spatial influence diagram-based mechanisms nor to have knowledge about the HARDY, CLIPS and ARC/INFO software on which the system is implemented.

ILUDSS is fully operational. However, its current stage of development is that of a prototype rather than a production system. It is mainly for demonstration of the flexibility of KBSDSS technology. The current implementation of ILUDSS has the following limitations:

- For automatic modelling, ILUDSS can only deal with three types of land uses. The evaluation of land use potential is limited to three classes of attributes or factors including physical land suitability, proximity to desirable and undesirable land features and minimum area of each land parcel. Some other important factors influencing land use potential on Islay, such as potential impacts on wildlife, have not been included because of lack of appropriate data and models.
- The land suitability models are quite simple. They provide a general indication of which areas may be suitable for a certain type of land use, but do not give an indication of which areas are most profitable from an economic viewpoint.
- Only some of the GIS models defined in Chapter 3 and of the ARC/INFO analysis functions are actually supported. All GIS models in the system have their corresponding solvers for raster-based analysis and modelling. However, only three GIS models *overlay*, *buffering* and *reclassify* have the solvers for vector-based analysis and modelling. No analytical models are available.
- ILUDSS can only handle spatial data conversion between ARC/INFO coverages and ARC/INFO grids. No other spatial data transformation facilities, such as scale change and map projection conversion, are available.
- ILUDSS focus on modelling on map data layers. No report generator is available in the system. The final results can only be displayed in maps.
- Finally, to simplify the design of ILUDSS, no backtracking is allowed during a modelling session. Therefore, the only way to correct erroneous entry of data is to restart the session.

Built and developed primarily as a demonstration of KBSDSS technology, ILUDSS served its purpose well. Despite the limitations of its current implementation, ILUDSS has shown that the KBSDSS paradigm offers a promise for decision support in land use planning, and for helping and fitting with the work, knowledge and operation of planning by planners or land managers.

## 8.9 Summary

ILUDSS is a prototype land use decision support system, which is primarily built and developed as an initial demonstration of KBSDSS technology. This chapter has described the main features of ILUDSS. Its strength and shortcomings were also discussed. ILUDSS should not be yet viewed as a comprehensive KBSDSS system. However, in spite of its limitation, ILUDSS has shown that KBSDSS systems can provide knowledge-based support in planning using spatial modelling, which includes problem structuring and representation, model formulation and integration, and automation of the process of spatial problem solving in land use planning.

# Chapter 9

## Conclusion

This final chapter summarises the thesis and its contributions, assesses the KBSDSS approach and discusses future work.

### 9.1 Summary

The thesis has presented a knowledge-based approach to the development of SDSS systems, and developed a framework for the design of knowledge-based SDSS systems, from the spatial influence diagram-based representation scheme and mechanisms to a prototype KBSDSS system designed for strategic planning of land use in a rural area.

The main goal of the work was to develop a new SDSS development framework through integrating expert systems, GIS and spatial modelling technologies. With this framework and its implementation, this thesis has shown that the KBSDSS system is able to overcome some of the problems of current SDSS systems mentioned in Chapter 1, namely, their user unfriendliness, inflexibility in model management (model formulation and integration) and inability to adapt to users' needs.

The spatial influence diagram-based representation scheme which has been established is an abstraction of spatial problems on which algorithms and heuristics have been developed to build capabilities for model formulation, model integration, and automation of the problem solving process.

Based on the spatial influence diagram-based representation scheme and mechanisms, an architecture for KBSDSS systems has been proposed. The underlying KBSDSS architecture demonstrates a new approach for integrating GIS, expert systems and spatial modelling technologies into a single SDSS system that encompasses the capabilities of analytical modelling (numerical computation), GIS modelling (spatial modelling) and rule-based modelling (heuristic reasoning). In the KBSDSS system, the domain-oriented spatial problem description is depicted in a spatial influence diagram without regard to the final implementation tools (e.g. ARC/INFO and CLIPS). A spatial influence diagram serves as a logical-level model or a conceptual model, within which GIS, analytical and rule-based models are integrated logically, thus enabling the system to provide flexible model management capabilities.

The KBSDSS technology is grounded on the premise that for effective spatial decision support, an SDSS system should have capabilities for spatial problem structuring and representation, and a spatial problem representation must reflect the preferences of the user of the system. The KBSDSS system relies on spatial influence diagrams as a formalism for structuring and representing spatial problems. The choice of the direct predecessors of the goal attribute (the value node) in a spatial influence diagram reflects a certain problem solving strategy. The system is able to formulate alternative spatial problem models according to the user's needs or preferences relating to the problem solving strategies.

The KBSDSS system contains extensive knowledge in its knowledge base. The domain-specific knowledge helps formulate specific spatial problem models; the tools (models and utility programs) knowledge and meta-data free the user from having knowledge about the specific tools and data sets; and the process knowledge can guide the user through the spatial problem-solving process. The spatial influence



diagram-based mechanisms enable the KBSDSS system to evaluate a spatial problem automatically from its formulation to actual execution of the selected models.

A prototype KBSDSS system, ILUDSS, has been developed as an implementation of the KBSDSS framework. It is built in a KBSDSS development environment that couples ARC/INFO, HARDY and CLIPS. ILUDSS is designed for strategic land use planning on the island of Islay. Despite its limitations, the prototype has shown the key features of a KBSDSS system.

## 9.2 Contributions of the Thesis

The contributions of this research include the development of:

- a spatial influence diagram-based representation scheme and mechanism for representation and formulation of spatial problems,
- a spatial influence diagram-based mechanism for evaluation of spatial problems,
- an architecture for KBSDSS systems,
- a KBSDSS development environment, and
- a prototype implementation of a KBSDSS approach to rural land use planning

### **Representation and formulation of spatial problems**

This thesis has developed a spatial influence diagram-based representation scheme for structuring and representation of spatial problems, based on the influence diagram in decision analysis. A spatial influence diagram is a framework in which to formulate spatial problems and to incorporate the knowledge of experts. At the same time, it is a description of information that can be stored and manipulated by a computer. Moreover, it can express users' preferences for problem solving.

Formulation of a spatial problem is a process for construction of a spatial influence diagram. An algorithm has been designed to construct a spatial influence

diagram within a certain problem domain according to the user's goal. Starting with the goal attribute representing the user's goal, the formulation process involves an extensive search in the knowledge base until all variables relevant to the problem have been found and the resulting spatial influence diagram can be evaluated.

### **Evaluation of a spatial problem**

The KBSDSS automates the process of evaluation of a spatial problem. An algorithm has been designed to provide the solution procedure for a spatial problem by evaluating a spatial influence diagram, which controls all of the inference and analysis. The algorithm drives the evaluation of nodes in a spatial influence diagram one by one until the value node is evaluated and a solution to the problem is obtained. For evaluating a node in a spatial influence diagram, heuristic rules have been developed for selecting model solvers and utility programs, assigning parameters, and activating the execution of the selected model solvers and utility programs to derive the value of the node.

### **An architecture for KBSDSS systems**

An architecture for KBSDSS systems has been proposed, which is different from those of traditional SDSS systems. The architecture is composed of six components: a user interface, query processing subsystem, modelling subsystem, problem processor, knowledge base and back-end subsystem. Spatial influence diagrams are a "front-end" for spatial modelling activities. The problem processor employs the spatial influence diagram-based mechanisms to make inferences and control the spatial modelling process. The knowledge base contains knowledge about domain variables relevant to the specific problem domain, data, models, utility programs, modelling process. etc. The main functions of a KBSDSS system include query, formulation of spatial problems and evaluation of spatial problems by integrating GIS, analytical and rule-based models. It may support both automatic spatial modelling and user-assisted spatial modelling. For automatic modelling, the

system automatically formulates a spatial problem model to represent the user's problem, and then evaluates the formulated model and presents the results to the user. For user-assisted modelling, the system may help the user formulate spatial problem models using spatial influence diagrams and then evaluate the user-formulated model and get the results.

### **A KBSDSS development environment**

An environment for building KBSDSS systems has been developed which shows the feasibility of creating highly integrated knowledge-based spatial decision support systems through the loose coupling of software tools developed independently. The KBSDSS development environment integrates a GIS (ARC/INFO) with a diagramming tool (HARDY) and an inference tool (CLIPS). Simple inter-application protocols were written to handle inter-process communications between ARC/INFO-HARDY and data exchange between ARC/INFO-CLIPS. The system operates as a single system environment from a user perspective.

### **A prototype implementation of a KBSDSS approach to rural land use planning**

A prototype KBSDSS system, ILUDSS, has been built. It not only demonstrates the KBSDSS technology, but also presents a new approach for computer-assisted strategic land use planning in rural areas. ILUDSS allows the user to identify land use potential for different land use types according to a set of attributes or factors. By specifying the land use interest and the factors or attributes to be considered in the evaluation of the land use potential, ILUDSS can formulate a land use problem model corresponding to the user's preferences relating to the related evaluation factors, present the model graphically to the user, evaluate the model automatically, and finally display the modelling results in maps. The user can also structure his or her own land use problems using the spatial influence diagram editor provided in the system, and build land use problem models by employing the

existing GIS, analytical and rule-based models and data. The features of ILUDSS, besides contributing to efficient land use decision making, also enable the user to identify the land use potential according to his or her requirements without the need to have knowledge about the implementation tools employed in the system.

### 9.3 Perspective: Strengths and Limitations of KBSDSS Systems

KBSDSS systems can provide knowledge-based assistance to decision makers. In terms of the broad picture, KBSDSS systems have several advantages.

First, a KBSDSS system can be incorporated into the decision making process. Resource and environmental planning involves a great deal of *ad hoc* decision making based on the evaluation of alternatives. An ideal planning process can be characterised as the following steps (Dueker 1980):

- defining the problems
- determining objectives
- inventing alternative solutions
- evaluating alternatives
- selecting the best alternative
- implementing the systems or plan
- monitoring the results.

ILUDSS has demonstrated that KBSDSS systems can provide planners with the ability to evaluate alternatives and support the decision making process that characterises planners' needs and preferences. The land use potential for various land use types generated by ILUDSS represents the land use options for each location in the study area that takes into account physical land suitability and the user's

preferences of other specific local conditions. It enables the user or a team to examine comprehensive comparisons of the land use options or alternatives for the interested locations, identify conflicts over the competing land uses, and make trade-off to choose the “best” uses. A KBSDSS system offers the potential to provide support for almost every aspects of resource and environmental planning.

Densham and Goodchild (1989) pointed out that an SDSS system “should incorporate knowledge used by expert analysts to guide the formulation of the problem, the articulation of the desired characteristics of the solution and the design and execution of a solution process”. KBSDSS systems can achieve this goal. A KBSDSS system captures domain knowledge that can assist users in their problem formulation process by formulating a problem model (a spatial influence diagram) for the problem using the spatial influence diagram-based mechanism for model formulation. The spatial influence diagram-based mechanism for model evaluation provides an automated solution procedure for the problem. ILUDSS has shown these capabilities in the formulation and solution of land use problems in a certain problem context.

A KBSDSS system may provide not only domain knowledge for problem formulation, but also model knowledge, utility program knowledge, meta-data, process knowledge and other kinds of knowledge. The extensive knowledge base helps users in the appropriate use of GIS, and other modelling and analysis techniques to solve their problems. As Cowen and Shirley (1991) suggested, a good SDSS should be able to make GIS software accessible to users with different levels of technical expertise. A KBSDSS system can help the users who have limited experience in GIS and spatial modelling to formulate and solve their problems without the need to have knowledge about the use of the relevant models and data. It can also help the users who have experience in GIS and spatial modelling to formulate and solve their problems by using the existing models and data without the need to know the implementation details of the models. Thus, a KBSDSS system enables the user to exploit the power of spatial problem analysis and modelling, in particular GIS modelling, analytical modelling and rule-based modelling in a simple and fast way. The spatial influence diagram-based representation scheme and

mechanisms as well as extensive knowledge in the knowledge base make a KBSDSS system very effective in supporting the formulation of the problem, the design and execution of a solution process, and the proper use of modelling and analysis techniques.

Furthermore, since they contain much of the functionality of expert systems, KBSDSS systems can be used to distribute the expertise of domain experts. In addition, a KBSDSS system can be developed incrementally. As with all expert systems, a KBSDSS system can be started with a small scope. Once experience is gained with the system, more and more areas can be supported.

In summary, KBSDSS systems can support the decision making process, guide the formulation of spatial problems, automate the solution process for spatial problems, support the use of GIS, and other modelling and analysis techniques, and maintain the knowledge and expertise.

However, the KBSDSS approach to decision support presented here focuses on the development of representations and techniques which formulate and evaluate spatial problem models. The KBSDSS architecture does not allow users to express their problems and preferences in natural language. In addition, the architecture does not provide facilities to assimilate new knowledge through machine learning. The mechanism for spatial problem formulation is not flexible enough to allow the user to easily incorporate new elements into the problem model to account for aspects of the problem that were either totally or partially unanticipated in the system's knowledge base. There remain other practical issues to be explored, such as model validity, data quality and uncertainty, which are common considerations for designing computer-based systems.

## 9.4 Future Work

Development of the KBSDSS framework and the implementation of the prototype presented in this thesis indicate the feasibility of KBSDSS technology in spatial

modelling and decision support. With the initial success of the KBSDSS prototype, follow-up research needs to be conducted.

The current research focuses primarily on formulating and evaluating the user's spatial problem, providing no support for appraising the resulting spatial problem models and solutions. The system contains previously defined domain variables. As a continuation of this research, methods need to be developed to assess the validity of these domain variables every time they are used in the context of a specific spatial problem.

More work is required to fine-tune the prototype implementation in order to convert the current ILUDSS into a production system. In addition to overcoming the limitations with the system described in Chapter 8, more models have to be developed, such as habitat suitability models for wildlife, land suitability models for tourism and sport management, etc. These can be incorporated as developed.

As discussed in Chapter 4, the spatial influence diagram was designed to incorporate different types of models (including GIS, analytical and rule-based models) and data. With such a feature, the KBSDSS framework can be utilised to explore other application domains, which may involve all three types of models, and both spatial and non-spatial data. The development environment described in Chapter 7 may be extended by integrating other software systems, such as a statistical package for providing powerful statistical analysis functions. Some possible domains which could be explored are rural land use planning at local or farm level, locational modelling, resource allocation and so on.

Selection of model solvers is a difficult and complex task. A methodology for guiding the users to obtain required information and make trade-offs for selecting model solvers is needed. A more effective knowledge representation scheme may be developed to represent model solvers by incorporating knowledge about the characteristics of problems to which they may be applied, the size of the problem that can be solved, the accuracy that can be expected or the time required to solve the problem, etc.

Another area for future research is uncertainty handling. Most decision problems in resource and environmental planning involve many uncertainties. For

example, in rural land use decision making, the factors, such as prices at the time of harvest, government policies, technological change, and weather conditions, all affect land use productivity and income. However, these factors are rarely known precisely before they occur. Introducing the probabilistic description and evaluation of a spatial problem into the KBSDSS framework would enhance the problem solving capabilities of a KBSDSS system. Spatial influence diagrams are able to capture uncertainty and represent probabilistic dependence among the problem variables. For example, chance nodes may denote uncertain variables, while arcs may denote probabilistic dependence between chance nodes. However, the mechanism for probabilistic reasoning with uncertain elements in a spatial context needs to be developed. In addition, for many real problems, most or all probabilities will need to be obtained from expert judgement. Techniques for eliciting and expressing one's knowledge in terms of probabilities need to be investigated. The possible techniques include probability calculus (Farley 1983), certainty factors (Shortliffe and Buchanan 1985), Bayesian formalism (Pearl 1988), Dempster-Shafer theory (Shafer 1976) and so on.

KBSDSS systems should be built for specific domains. Focusing on a particular domain is essential for developing meaningful knowledge bases. How effective the KBSDSS applications will depend on the nature of their problem domains and on how well the applications address the needs of individual decision makers. KBSDSS technology can enhance the capabilities of SDSS systems. However, there is much yet to be done before an SDSS system can provide fully automated, intelligent, reliable and user-friendly decision support in resource and environmental planning and management. Since spatial problem descriptions and potential solution procedures are very diverse, it may be impossible to develop an SDSS system that can fully automate the spatial problem solving process. The aim of research into KBSDSS is to facilitate the problem solving process by solving the more structured components without needing extensive interaction from the decision makers and guiding them by providing relevant information at appropriate stages. The perfect SDSS system may be an unattainable ideal.



## References

- Armstrong, M.P., P.J. Densham and G. Rushton, 1986, Architecture for a microcomputer based spatial decision support system. *Proceedings of the Second International Symposium on Spatial Data Handling*, Seattle, Washington, pp.120-131.
- Armstrong, M.P., S. De, P.J. Densham, P. Lolonis, G. Rushton and V.K. Tewari, 1990, A knowledge-based approach for supporting locational decisionmaking. *Environment and Planning B: Planning and Design*, Vol.17, pp.341-364.
- Aronoff, S., 1989, *Geographic Information Systems: A Management Perspective*. WDL Publications, Ottawa, Canada.
- Aspinall, R., 1993, Use of geographic information systems for interpreting land-use policy and modelling effects of land-use change. In: Haines-Young, et al. (Eds.), *Landscape Ecology and Geographic Information Systems*, Taylor & Francis, pp.223-236.
- Banerjee, S. and A. Basu, 1993, Model type selection in an integrated DSS environment. *Decision Support Systems*, Vol.9, pp.75-89.
- Barr, A. And E.A. Feigenbaum, 1981, *The Handbook of Artificial Intelligence*. Vol. 1, William Kaufmann Inc.
- Berry, J.K., 1987a, Fundamental operations in computer-assisted map analysis. *International Journal of Geographical Information Systems*. Vol.1, No.2, pp.119-136.

- Berry, J.K., 1987b, Computer assisted map analysis: potentials and pitfalls. *Photogrammetric Engineering and Remote Sensing*, Vol.53, No.10, pp.1405-1410.
- Bibby, J.S., H.A. Douglas, A.J. Thomasson and J.S. Robertson, 1982, *Land Capability Classification for Agriculture*, Monograph. Aberdeen: The Macaulay Land Use Research Institute.
- Bibby, J.S., R.E.F. Hesloop and R. Hartnup, 1988, *Land Capability Classification for Forestry in Britain*, Soil Survey Monograph. Aberdeen: The Macaulay Land Use Research Institute.
- Signal, E.M., D.J.Curtis and J.L.Matthews, 1988, *ISLAY: Land Types, Bird Habitats and Natural Conservation. Part 1: Land Use and Birds on Islay*. NCC Chief Scientist Directorate. Report No. 809, Part 1.
- Bown, C.J., B.M. Shipley and J.S.Bibby, 1982, *Soil and Land Capability for Agriculture. South-West Scotland*. Aberdeen: The Macaulay Institute for Soil Research.
- Brachman R.J. and B.C. Smith, 1980, *SIGART Newsletter, Special Issue on Knowledge Representation*. No. 70.
- Brinkman, R. and A.J. Smyth, 1973, *Land Evaluation for Rural Purposes*. Publication 17, International Institute for Land Reclamation and Improvement, Wageningen, Netherlands.
- Burrough, P.A., 1986, *Principles of Geographical Information Systems for Land Resource Assessment*. Oxford:Clarendon.
- Burrough, P.A., 1992, Development of intelligent geographical information systems. *International Journal of Geographical Information Systems*. Vol.6, No.1, pp.1-11.
- Chandra, N. and W. Goran, 1986, Steps toward a knowledge-based geographical data analysis system. In: Opitz, B. (edt.), *Geographic Information Systems in Government*, Vol.2, A.Deepak Publishing, Hampton, Virginia, pp.749-764.
- Cowen, D.J. and W.L. Shirley, 1991, Integrated Planning Information Systems. In: Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (eds.), *Geographical*

- Information Systems: Principles and Applications*, Vol.1, Longman, London, pp.297-310.
- Dangermond, J., 1983, A classification of software components commonly used in geographic information systems. In: Peuquet, D.J. and J. O'Callaghan (eds), *Design and Implementation of Computer Based Geographic Information Systems*. Amherst, NY: IGU Commission on Geographic Data Sensing and Processing, pp.70-91.
- Davidson, D.A. (ed.), 1986, *Land Evaluation*. New York: Van Nostrand Reinhold.
- Davidson, D.A., A.I. Watson and P.H. Selman, 1993, An evaluation of GIS as an aid to the planning of proposed developments in rural areas. In: Mather, P.M. (ed.), *Geographical Information Handling - Research and Applications*. John Wiley & Sons Ltd, pp.251-259.
- Davis, J.R. and J.L. Clark, 1989, A selective bibliography of expert systems in natural resource management. *AI Applications in Natural Resource Management*, Vol.3, No.3, pp.1-18.
- Davis, J.R., P.T. Compagnoni, and P.M. Nanninga, 1987, Roles for knowledge-based systems in environmental planning. *Environment and Planning B: Planning and Design*, Vol.14, No.3, pp.239-254.
- Densham, P. J., 1991, Spatial decision support systems. In: Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (eds.), *Geographical Information Systems: Principles and Applications*, Vol.1, Longman, London, pp.403-412.
- Densham, P.J. and Armstrong, M.P., 1987, A spatial decision support system for locational planning: design, implementation and operation. *Auto Carto-8*, Maryland, pp.112-121.
- Densham, P.J. and M.F. Goodchild, 1989, Spatial decision support systems: a research agenda. *Proceedings of GIS/LIS'89*, Vol.2, ACSM/ASPRS/AAG, Virginia, pp.707-716.
- Diamond, J.T. and J.R. Wright, 1988, Design of an integrated spatial information system for multiobjective land-use planning. *Environment and planning B: Planning and Design*, Vol.15, pp.205-214.

- Djokic, Dean, 1993, Towards a general purpose spatial decision support system using existing technologies. *Proceedings of Second International Conference /Workshop on Integrating Geographic Information Systems and Environmental Modelling*, Breckenridge, Colorado.
- Dueker, K.J., 1980, An approach to integrated information systems for planning. In: Krammer K. and J. King (eds.), *Computers in Local Government Urban and Regional Planning*. Auerbach Publishers Inc., Pennsauken, pp.(2.1.2) 1-12.
- Environmental Systems Research Institute (ESRI), 1991a, *ARC/INFO Data Model, Concepts, & Key Terms*, Redlands, California.
- Environmental Systems Research Institute (ESRI), 1991b, *Understanding GIS - the ARC/INFO Method*, Redlands, California.
- Environmental Systems Research Institute (ESRI), 1991c, *Cell-based Modelling with GRID*, Redlands, California.
- Environmental Systems Research Institute (ESRI), 1991d, *AML User's Guide*, Redlands, California.
- FAO, 1976, A framework for land evaluation. *FAO Soils Bulletin*, No.32.
- Farley, A.M., 1983, A probabilistic model for uncertain problem solving. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13*, Vol.4, pp.568-579.
- Fedorowicz, J. and G.B. Williams, 1986, Representing modeling knowledge in an intelligent decision support system. *Decision Support Systems*, Vol.2, No.1, pp.3-14.
- Forgy, Charles L., 1985, Rete: a fast algorithm for the many pattern/many object pattern match problem, *Artificial Intelligence*, Vol. 19, pp 17-37.
- Foster, M.A., M.C. Saunders, M. Ramirez, N. Duan and B.J. Miller, 1992, The treatment advisor in GypsES: an object oriented, knowledge based system for Gypsy moth management. *The Compiler*, Vol.10, No.3, pp.35-46.
- Franklin, J. F., 1979, Simulation modelling and resource management. *Environmental Management*, Vol.3, No.2.
- Futty, D.W., A.R. Sibbald, R.J. Aspinall, D.C. Macmillan, E.G. Stevenson and D.R. Miller, 1989, Forestry options for land use. *The Macaulay Land Use Research Institute Annual Report 1988-1989*, Aberdeen, Scotland, pp.5-14.

- Gardels, K., 1987, The expert geographic knowledge system: applying logic rules to geographic information. *Auto Carto-8*, Maryland, pp.520-529.
- Geoffrion, A.M., 1983, Can OR/MS evolve fast enough?. *Interface*, Vol.13, pp.10-25.
- Giarratano, Joseph C. and Gary Riley, 1989, *Expert Systems: Principles and Programming*. PWS-KENT Publishing Company.
- Greenland White-Fronted Goose Study Team (GWFGST), 1986, Duich Moss: a minor administrative hitch. *Ecosystem*, Vol. 7, pp.24-31.
- Hayes-Roth, F., D. A. Waterman and D.B. Lenat, 1983, *Building Expert Systems*, Addison-Wesley Publishing Company, Inc.
- Healey, R.G., P. Dowie, A. Mowle and J. Holbrook, 1988, Integrating remote sensing data into a geographical information system: a foundation for rural land use strategies - Nature Conservancy Council project. *Proceedings of IGARSS'88 Symposium*, Edinburgh, pp.111-112.
- Hogrer, C.J., 1984, *Introduction to Logic Programming*, Academic Press, New York.
- Howard, R.A. and J.E. Matheson, 1983, Influence diagrams. In: Howard, R.A. and J.E. Matheson (eds), *Readings on the Principles and Applications of Decision Analysis*, Vol.2, Strategic Decisions Group, California, pp.719-771.
- Jackson, P., 1990, *Introduction to Expert Systems*, Addison-Wesley Publishing Company.
- Kessell, S.R., 1990, An Australian geographical information and modelling system for natural area management. *International Journal of Geographical Information Systems*, Vol.4, No.3, pp.333-362.
- Kingston, J., 1993, Using HARDY to support the use of Checkland's soft systems methodology. *Airing, News and Information from AIAI*. The University of Edinburgh.
- Kollias, V. J. and A. Voliotis, 1991, Fuzzy reasoning in the development of geographical information systems. FRIS: a prototype soil information system with fuzzy retrieval capabilities. *International Journal of Geographical Information Systems*, Vol.5, No.2, pp.209-223.

- Lein, J.K., 1992, Modelling environmental impact using an expert-geographic information system. *Proceedings of GIS/LIS'92*, Vol.1, San Jose, California, pp.436-444.
- Loh, D.K. and E.J. Rykiel, 1992, Integrated resource management systems: coupling expert systems with data-base management and geographical information systems. *Environmental Management*, Vol.16, No.2, pp.167-177.
- Maguire D.J and J. Dangermond, 1991, The functionality of GIS. In: Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (eds.), *Geographical Information Systems: Principles and Applications*, Vol.1, Longman, London, pp.319-360.
- Maxwell, T.J., 1993, An integrated approach to land use research and decision making. *The Macaulay Land Use Research Institute Annual Report 1993*. Aberdeen, Scotland, pp.3-11.
- McRae, S.G. and C.P. Burnham, 1981, *Land Evaluation*. Clarendon Press, Oxford.
- Michie, D., 1973, Knowledge engineering. *Cybernetics*, Vol.2, pp.197-200
- Miller, Ralph and Wei-Ning Xiang, 1992, A knowledge-based GIS method for visual impact assessment in transmission line siting. *Proceedings of GIS/LIS'92*, Vol.2, San Jose, California, pp.577-584.
- Morehouse, S., 1985, ARC/INFO: A geo-relational model for spatial information. *Auto-Carto 7*, Washington, D.C., pp.388-397.
- Morehouse, S., 1989, The architecture of ARC/INFO, *Auto-Carto 9*, Baltimore, Maryland, pp.266-277.
- Morse, B., 1987, Expert interface to a geographic information system. *Auto-Carto 8*, Maryland, pp.535-541.
- National Aeronautic and Space Administration (NASA) , 1993, *CLIPS Reference Manual 6.0*, Software Technology Branch, Lyndon B. Johnson Space Centre.
- Negahban, B., C. Fonyo, K.L. Campbell, J.W. Jones, W.G. Boggess, G. Kiker, E. Hamouda, E. Flaig and H. Lal, 1993, LOADSS: a GIS-based decision support system for regional environmental planning. *Proceedings of Second International Conference /Workshop on Integrating GIS and Environmental Modeling*, Breckenridge, Colorado.

- Olmsted, S.M., 1983, *On representing and Solving Decision Problems*. Ph.D. Thesis, EES Department, Stanford University.
- Openshaw, S., 1991, Developing appropriate spatial analysis methods for GIS. In: Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (eds.), *Geographical Information Systems: Principles and Applications*, Vol.1, Longman, London, pp.389-402.
- Palmer, R.N. and R.M. Tull, 1987, Expert systems for drought management planning. *Journal of Computing in Civil Engineering*, Vol.1, No.4, pp.284-297.
- Parent, P. and R. Church, 1987, Evolution of geographic information systems as decision making tools. *Proceedings of GIS'87*, pp.63-70.
- Pearl, J., 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc., California.
- Pearson, E. J., G. Wadge and A. P. Wislocki, 1991, Mapping Natural Hazards with Spatial Modelling Systems. *Proceedings of European Conference on Geographical Information Systems*, Brussels, Vol.2, pp.847-855.
- Peterson, K., 1993, Spatial decision support systems for real estate investment analysis. *International Journal of Geographical Information Systems*, Vol.7, No.4, pp.379-392
- Peuquet, D.J., 1984, A conceptual framework and comparison of spatial data models. *Cartographica*, Vol. 21, pp.66-113.
- Peuquet, D.J., 1991, Methods for structuring digital cartographic data in a personal computer environment. In: Taylor, D.R.F. (ed), *Geographic Information Systems. The Microcomputer and Modern Cartography*. Pergamon Press, pp.67-95.
- Rauscher, H. Michael; Dale L. Bartos; Stuart M. Davey; Kent Downing; Gregory A. Elmes; George Gertner; Biing T. Guan; J.N.R. Jeffers; Ronald E. McRoberts; Daniel L. Schmoldt; David R.B. Stockwell; Mark J. Twery; and Charles B. Yuill. 1991. The Encyclopaedia of AI Applications to Forest Science. *AI Applications*, Vol.5, No.2, Insert: 592,080 bytes; 235 chunks; and 449 links (electronic).

- Ravlin, F.W., 1991, Development of monitoring and decision-support systems for integrated pest management of forest defoliators in North America. *Forest Ecology and Management*, Vol.39, pp.3-13.
- Rideout, T.W. and J. A. Holbrook, 1992a, Practical geographical information system for conservation and rural management. In: Rideout, T.W. (ed.), *Geographical Information Systems and Urban and Rural Planning*. The Planning and Environment Study Group of the Institute of British Geographer. pp. 129-140.
- Rideout, T.W. and J. A. Holbrook, 1992b, *An Introduction to the NCCS ARC/INFO GIS Interface*, Nature Conservancy Council for Scotland and University of Edinburgh, Unpublished.
- Robinson, V. B., A. U. Frank and H. A. Karimi, 1987, Expert systems for geographic information systems in resource management. *AI Applications in Natural Resource Management*, Vol.1, No.1, pp.47-57.
- Samet, H., 1989, *Applications of Spatial Data Structures*, Addison-Wesley, Reading, MA.
- Selman, P.H., 1989, Conservation, development and landuse planning: Scotland leads the way. *Scottish Geographical Magazine*, Vol.105, No.3, pp.142-148.
- Shachter, R.D., 1986, Evaluating influence diagrams. *Operations Research*, Vol.34, No.6, pp.871-882.
- Shafer, G., 1976, *A Mathematical Theory of Evidence*, Princeton University Press.
- Shaw, D.T. and D.R. Maidment, 1988, An intelligent design support system for computer aided site development. In: Strech, M.(ed.), *Critical Water Issues and Computer Applications*, ASCE, New York, pp.68-71.
- Shortliffe, E.H. and B.G. Buchanan, 1985, A model of inexact reasoning in Medicine. In: *Rule-based Expert Systems*, Addison-Wesley, pp.233-262.
- Simon, H. A., 1973, The structure of ill structured problems. *Artificial Intelligence*. Vol. 4, pp.181-201.
- Smart, J., 1993a, *User Manual for HARDY*. AIAI, the University of Edinburgh.
- Smart, J., 1993b, *wxWindow 1.50 User Mannual*. AIAI, the University of Edinburgh.



- Smart, J. and I. Harrison, 1992, HARDY: A hypertext-based diagramming tool, *Airing. News and Information from AIAI*. The University of Edinburgh.
- Smith, G.F., 1988, Towards a heuristic theory of problem structuring. *Management Science*. Vol.34, No.12, pp.1489-1506.
- Smith, G.F., 1989, Defining managerial problems: a framework for perspective theorizing. *Management Science*. Vol.35, No.8, pp.963-981.
- Smith, T.R., S. Menon, J.L. Star and J.E. Estes, 1987, Requirements and principles for the implementation and construction of large-scale geographic information systems. *International Journal of Geographical Information Systems*, Vol.1, No.1, pp.13-31.
- Stonebraker, M., 1986, Triggers and inference in database systems. In: Brodie, M. and John Mylopoulos (eds.), *On Knowledge base management systems: Integrating Artificial Intelligence and Database Technologies*. Springer-Verlag, New York, pp.297-314.
- Teicholz, E. and B. J. L. Berry, 1983, *Computer graphics and environmental Planning*, Prentice-Hall, Inc.
- Tomlin, C.D., 1990, *Geographic Information Systems and Cartographic Modelling*, Prentice Hall.
- Tomlin, C.D., 1991, Cartographic modelling. In: Maguire, D.J., Goodchild, M.F. and Rhind, D. W. (eds.), *Geographical Information Systems: Principles and Applications*. Vol. 1, Longman, London, pp.361-374.
- Tomlin, C.D. and K.M. Johnston, 1990, An experiment in land-use allocation with a geographical information system. In: Peuquet, D.J. and D.F. Marble (eds.), *Introductory Readings in Geographic Information Systems*, Taylor & Francis, pp.159-169.
- Unwin, D.J., 1981, *Introductory Spatial Analysis*, Methuen, London.
- Usery, E.L., P. Altheide, R.R.P. Deister and D.J. Barr, 1988, Knowledge-based GIS techniques applied to geological engineering. *Photogrammetric Engineering and Remote Sensing*, Vol.54, No.11, pp.1623-1628.

- van der Vlugt M., 1989, The use of a GIS based decision support system in physical planning. *Proceedings of GIS/LIS'89*, ASPRS, Bethesda, Maryland, pp. 459-467.
- van Melle, W., 1979, A domain-independent production rule system for consultation programs. *IJCAI*, pp.923-925.
- Walker, P. A. and D. M. Moore, 1988, SIMPLE. An inductive modelling and mapping tool for spatially oriented data. *International Journal of Geographical Information Systems*, Vol.2, No.4, pp.347-363.
- Wang, F., G. B. Hall and Subaryono, 1990, Fuzzy information representation and processing in conventional GIS software: database design and application. *International Journal of Geographical Information Systems*, Vol.4, No.3, pp.261-283.
- Ward, S.A., 1990, the national peatland database for Scotland: an example analysis for Caithness District. In: Bibby, J.S. and M.F. Thomas (eds.), *Evaluation of Land Resources in Scotland*, The Macaulay Land Use Research Institute and Royal Scottish Geographical Society, pp.85-88.
- Webster, C., 1990, Rule-based spatial search. *International Journal of Geographical Information Systems*, Vol.4, No.3, pp.241-259.
- Williams, S.B., 1992, INFORMS-TX overview. *Paper Presented at the Seventh Forest Service AI Workshop*, Blacksburg, Virginia.
- Winston, P.H., 1992, *Artificial Intelligence, the Third Edition*, Addison-Wesley.
- Woolley, R.N. and M. Pidd, 1981, Problem structuring: a literature review. *Journal of Operation Research Society*. Vol.32, pp.197-206.
- Zhu, Xuan and Richard Healey, 1992, Towards intelligent spatial decision support: integrating geographical information systems and expert systems. *Proceedings of GIS/LIS'92*, San Jose, California, Vol.2, pp.877-886.

# Appendix A

## Conflict Resolution Strategies in CLIPS

(From: NASA , 1993, *CLIPS Reference Manual 6.0*)

There are seven conflict resolution strategies in CLIPS. These are depth, breadth, simplicity, complexity, lex, mea and random.

**Depth** strategy causes newly activated rules to be placed above all rules of the same salience. For example, given that fact-a activates rule-1 and rule-2, and fact-b activates rule-3 and rule-4, then if fact-a is asserted before fact-b, rule-3 and rule-4 will be above rule-1 and rule-2 on the agenda. However, the position of rule-1 relative to rule-2 and rule-3 relative to rule-4 will be arbitrary.

**Breadth** strategy places newly activated rules below all rules of the same salience. For example, given that fact-a activates rule-1 and rule-2, and fact-b activates rule-3 and rule-4, then if fact-a is asserted before fact-b, rule-1 and rule-2 will be above rule-3 and rule-4 on the agenda. However, the position of rule-1 relative to rule-2 and rule-3 relative to rule-4 will be arbitrary.

**Simplicity** strategy puts newly activated rules above all activations of the same salience rules with equal or higher specificity. The specificity of a rule is determined by the number of comparisons that must be performed on the LHS of the rule. Each comparison to a constant or previously bound variable adds one to the specificity. Each function call made on the LHS of a rule as part of the :, =, or test conditional element adds one to the specificity. The Boolean functions **and**, **or**, and **not** do not add to the specificity of a rule,

but their arguments do. Function calls made within a function call do not add to the specificity of a rule.

**Complexity** strategy places newly activated rules above all activations of the same salience rules with equal or lower specificity.

**Lex** strategy places, among rules of the same salience, an activation with a more recent fact-index before activations with less recent fact indices. If two activations have the exact same recency, the activation with the higher specificity is placed above the activations with the lower specificity.

**Mea** strategy determines the placement of the activation according to the recency of the fact-index associated with the first pattern. If the fact-indices of two activations are the same, then the **lex** strategy is used.

**Random** strategy determines the placement of activated rules using a random number assigned to each activation.

# Appendix B

## The Main Software Modules in ARC/INFO

ARC/INFO contains a number of software modules. The main modules are as follows:

### **ARC**

ARC is the main program environment in ARC/INFO. It contains commands that start each of the other software modules and a set of vector-based geoprocessing tools that handle the digital cartographic data for geographical features and perform various spatial data processing tasks. ARC tools offer the following main spatial data handling capabilities:

- Spatial data generation, management and manipulation, including map coverage digitising and editing, error detection and verification, coordinate projection and transformation, rubber sheeting, coordinate generalisation, feature snapping, etc.
- File and workspace management, including operations to list, rename, copy, delete and describe map data layers, and operations to create, delete, rename, list, copy and change workspaces.
- Spatial analysis, including map overlay, buffer generation, nearest neighbour analysis, cartographic measurement, statistical summary, etc.

## GRID

GRID provides a set of operators and functions for processing of raster data. The operators it supports include:

- Arithmetic (plus, minus, multiply, divide, etc.)
- Boolean (TRUE or FALSE)
- Relational (greater than, less than, etc.)
- Bitwise (binary left shift, right shift, etc.)
- Combinatorial (overlay of grids while carrying attributes)
- Logical (contained within a set, etc.)
- Accumulative (sum, divide, multiply. etc. all values within a grid)
- Assignment (output equals)

The functions available in GRID include:

- Local functions: computing an output grid in which the output value at each cell location is a function of the corresponding input value at each location.
- Focal functions: computing an output grid in which the output value at each cell location is a function of the input cells in the specified neighbourhood of each location.
- Zonal functions: computing an output grid in which the output value at each cell location depends on the values of all the input cells on an input-value grid that share the same input value (or zone) defined by an input-zone grid.
- Global functions: computing an output grid in which the output value at each cell location is potentially a function of all the cells in the input grid.

GRID is integrated with ARC/INFO. Many of the capabilities that are common between modules can be used within GRID. GRID is also coupled with the capabilities of the INFO database management system that manages all attributes associated with the cell values. It uses the graphic environments of ARCPLOT and

ARCEDIT for all visual display of grids. In addition, AML can be used as a macro facility and to control the user environment.

## **ARC PLOT**

ARC PLOT is used for map display and query. It provides the cartographic tools for all mapping needs including full cartographic design, display and production capabilities. Within ARC PLOT, map components can be displayed and manipulated on graphics screen. Graphics files can also be created for high quality hardcopy output. Besides the mapping capabilities, ARC PLOT provides interactive query functions which allow the user to use the maps displayed on the screen to retrieve data directly from the database. The query function includes reselecting and displaying features based on some criterion (feature attribute values or spatial context), listing attribute values for all the features in a dataset, or pointing to a feature and retrieving attribute information about the feature. In addition, ARC PLOT supports graph drawing. Graphs can be used to display attribute data without a map, or to augment a map.

## **ARC EDIT**

ARC EDIT is a graphics and database editor within ARC/INFO. It combines all of the facilities for digitising map coverages with a more comprehensive set of editing commands. The user can edit feature attributes, add text annotation, use other database layers as a background display, diagnose and correct digitising errors, perform rubber sheeting and map sheet edge matching, etc.

## **Map librarian**

This module is used to manage large sets of digital cartographic data or the map library. The map library is defined as a collection of coverages which are organised by subject or content into layers and by location into tiles. Map librarian

provides a set of functions to store, update, query and display geographical data in the map library, such as insert, extract, create, replace, browse, etc.

## **NETWORK**

The module supports address geocoding and network analysis. Both are based on networks which are systems of connected linear features composed of arcs (there may be street segments, portions of a river, etc.) and intersections. Network analysis enables the user to model the flow of resources, such as vehicles travelling along a network of streets. Attributes such as direction of movement and impedance can be stored to support the analysis of optimum routes and to relate resources to centres of activity. Address geocoding links address information to geographical locations contained in various network coverages.

## **TIN**

TIN stands for “triangulated irregular network”. TIN is an ARC/INFO module containing a set of tools for storing, managing, and analysing three dimensional surfaces. It supports digital terrain modelling, including contouring, calculating slope, aspect, generating cross-sectional and three-dimensional displays, analysing cut and fill volumes, analysing terrain mobility, identifying drainage networks, analysing visibility and calculating intensity. It can create an elevation model from contours or randomly-spaced points.

## **COGO**

COGO is a software module in ARC/INFO primarily for capturing survey data. It contains a set of tools used for creating and manipulating survey data defined through coordinate geometry. They can be used in applications requiring accurate coordinate geometry for data entry and manipulation, such as the design and layout of subdivisions, roads, and related facilities.



# Appendix C

## Spatial Data Types Supported by ARC/INFO

(From: ESRI, 1991, *ARC/INFO Data Model, Concepts, & Key Terms*)

A number of geographical data types are supported by ARC/INFO, including coverages, grids, tins, images and CAD drawings.

### **Coverages**

A coverage represents the main method for geographical data storage in ARC/INFO. It is a digital version of a single map sheet layer and generally describes one type of map features such as streets, parcels, soil units, or forest stands. A coverage contains both the locational data and thematic attributes for map features in a given area. In a coverage, map features are stored as simple points, arcs, or polygons. Coverages use a combined vector spatial data model and a relational attribute data model.

### **Grids**

A grid is the primary data structure used by ARC/INFO GRID software. The cell is the spatial entity within a grid, and cell values represent theme or layer values. Each cell is square, has the same size as other cells in the grid and contains a numeric value. Grids use a combined raster spatial data model and a relational attribute data model. Each grid represents a spatial variable, theme or layer just like a coverage does.

## **Tins**

This is the primary data structure used by the TIN software module for representing continuous surfaces, such as terrain. A tin is composed of triangles, nodes and edges. Nodes are the xyz locations from which a tin is constructed. Triangles are formed by connecting each node with its neighbours. Edges are the sides of triangles. Tins are useful for representing surfaces that are highly variable, and contain discontinuities and breaklines.

## **Lattices**

A lattice is a regular-spaced sample of points representing a surface. Lattices apply the same data structure as a grid in ARC/INFO. The difference is in how various operators interpret the data. The TIN module contains a rich set of lattice operators that work on grid data sets which represent surfaces.

## **Images**

Images provide both geographical displays as well as picture attributes about spatial features. The set of software tools known as the IMAGE INTEGRATOR contains all the image management and display tools. They are used to display images as maps as well as images as attributes about spatial features.

## **CAD**

Computer-aided drafting (CAD) drawings play similar roles to images in a geographical database - they represent "picture" information about the area of interest. They are useful for enriching a person's knowledge about an area or particular geographical feature. However, they are not useful for modelling and analysis. The applications of CAD drawings are supported through the DXF display

interface in ARC/INFO's graphics programs as well as through ARC/INFO's CAD convertors. ARC/INFO can convert coverage data to and from a number of common CAD formats.

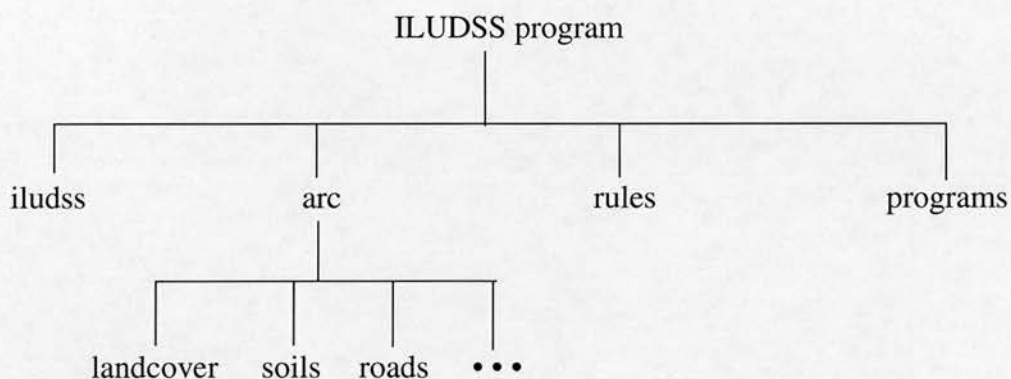
### **DBMS tables**

Attributes about features are stored in DBMS tables that are related to geographical features via a feature ID or value. Both coverages and grids can have associated attribute information stored in tables. One of the major advantages of the coverage and grid data models is that any attribute can be used to view, select, analyse, and display features contained within the coverage or grid.

## Appendix D

### File Structure of the ILUDSS Program

The ILUDSS program is installed under a directory which contains four subdirectories.



#### **The *iludss* directory**

This directory is the main directory of the ILUDSS program. It contains:

<i>ILUDSSinit.clp</i>	CLIPS code to initiate the system and load all knowledge base modules
<i>ILUDSSgui.clp</i>	CLIPS code to create the graphical user interfaces of ILUDSS
<i>processor.clp</i>	CLIPS code to perform the spatial influence diagram-based mechanisms
<i>variables.clp</i>	The domain knowledge base module (the VARIABLES module)

<i>process.clp</i>	The process knowledge base module (the MAIN module)
<i>solvers.clp</i>	The tools knowledge base module (the SOLVERS module)
<i>data.clp</i>	The meta-data knowledge base module (the DATA module)
<i>SpatialID.def</i>	The diagram definition file of spatial influence diagrams
<i>*.dia</i>	The diagram files storing spatial influence diagrams
<i>*.txt</i>	The text files to be used for explanations during an ILUDSS session

### **The *arc* directory**

This directory mainly contains the database, GIS models, analytical models and utility programs in AML, which includes

<i>*.aml</i>	GIS models, analytical models and utility programs, which are AML programs
<i>*.menu</i>	AML programs for creating pop-up menus
<i>landcover</i>	Directory for storing the data layer of Land Cover on Islay
<i>soils</i>	Directory for storing the data layer of Soils on Islay
<i>roads</i>	Directory for storing the data layer of Roads on Islay
•••	Other directories for other data layers

Any new data layers or data sets should be put in this directory.

### **The *rules* directory**

The directory contains rule-based models, encoded in CLIPS knowledge base modules.

<i>*.clp</i>	Rule-based models defined as CLIPS knowledge base modules
--------------	---

## The *programs* directory

This directory contains analytical models and utility programs written in C and other programming languages, including

<i>InfoToClips.c</i>	C program for converting INFO data to CLIPS facts
<i>InfoToClips</i>	The executable file of <i>InfoToClips.c</i>
*.c	Other C programs
*	Other executable files

## Appendix E

### Running ILUDSS

To run ILUDSS, go to the *iludss* directory and type `hardy -clips ILUDSSinit.clp` as a command line. The main window should come up with menus **File**, **Data**, **Model**, **Diagram** and **Help**. At the same time, the map display window should pop up with a map displaying the island of Islay and locations of SSSI areas on it. The HARDY main window also comes up together with the ILUDSS main window. The user may send the HARDY main window behind the ILUDSS main window, or iconise it, but must not close it.

To start a consultation session (for automatic modelling), pull down the **File** menu and left-click **Run**.

The **Data** menu allow the user to query, display, delete the existing data layers or data sets, add and show meta-data.

The **Model** menu offers the user options for adding new models (rule-based, analytical models and reclassification tables) and retrieving the information about all existing models within the system.

The **Diagram** menu provides the facility for the user to create new spatial influence diagrams representing land use problem models or select existing spatial influence diagrams for evaluation.

The **Help** menu allows the user to activate the on-line help system which provides information about the functionality and use of the system.

To quit the system, go to the **File** menu and left-click **Quit**. All processes and windows will be closed gradually. When quitting the system, there may be several

pop-up windows created by HARDY to prompt the user to save index files and diagram cards. Just ignore them by pressing the **No** button on these pop-up windows.



## Appendix F

### A Sample Session with ILUDSS

The sample session demonstrates the application of ILUDSS to the example of Section 8.7.2. The dialogue between the system and the user is through pop-up dialogue windows. The items and buttons on the pop-up dialogue windows are enclosed in square brackets [ ]. Information displayed in the main window is enclosed in frames. Annotations are inserted to give explanatory notes and describe the user's actions or inputs. The system has two modes: *raster* and *vector*. We use the default mode *raster* here. Figure A.1 shows the screenshot when ILUDSS starts up. Below is the introductory information about the system displayed in the main window.

#### INTRODUCTION

ILUDSS (Islay Land Use Decision Support System) is a prototype knowledge-based spatial decision support system, which integrates land use models, a geographical information system (GIS) and an expert system. It is designed to assist planners in strategic planning of land use for the development of the island of Islay, off the west coast of Scotland. Especially, it can be used by planners in assessment of land use potential for various land use interests on Islay.

Land use potential for a certain type of land use can be evaluated according to different factors. For example, the land use potential for afforestation may be evaluated according to the physical land suitability for forestry and impacts on wildlife. It can also be evaluated according to the physical land suitability for forestry and proximity to roads. It depends on the planners needs and preferences of conditions for the proposed land use. The two main functions supported by ILUDSS are automatic modelling and user-assisted modelling. The automatic modelling function is to assist

planners in evaluation of land use potential for various land use types on the island of Islay, according to their preferences and assessments relating to the various criteria and related evaluation factors without the need for them to have much knowledge about GIS and spatial modelling. The user-assisted modelling function allows the user who has general GIS and spatial modelling knowledge to build land use models for land use problems, without the need to know the implementation details of the modelling and analysis techniques used in the system.

Land use problems addressed by ILUDSS are represented in an intuitively simple form, called a spatial influence diagram. A spatial influence diagram is a graphical representation of a resource and environmental problem, which consists of nodes connected by directed arcs such that there are no paths following the arcs that lead in cycles. The nodes represent the land and environmental variables relevant to a particular problem. The directed arcs represent relationships between the variables (for more information, refer to the help system). A spatial influence diagram is a problem model. The automatic modelling function of ILUDSS can formulate a land use problem model using the knowledge in the system according to the user's needs and preferences for the assessment of land use potential, and then present the problem model graphically to the user in a spatial influence diagram. The spatial influence diagram here provides an effective vehicle for communication with the user so that the user can view a graphical display of the problem model and understand the overall structure and relevance of the land and environmental variables depicted in the graph. The directed arcs in the graph also imply the order for the solution of the problem. The system will then evaluate the problem model automatically by integrating the database, knowledge base, and different types of models (GIS models -- GIS analysis functions; analytical models -- mathematical equations; and rule-based models -- rule sets), and finally present the modelling results to the user in maps.

On the other hand, the user-assisted modelling function of ILUDSS supports user participation in defining problems and developing models. It provides facilities for the user to create and modify the land use problem models graphically by following the definitions of spatial influence diagrams and employing the existing models and data, or adding new models or data. Here, the spatial influence diagram provides a tool for the user to represent and structure land use problems and build the land use problem models. After a land use problem model has been created or modified by the user, the system will automatically evaluate the model and produce the results. For more information about the facilities for user-assisted modelling, and other functions and their uses, refer to the help system.

ILUDSS is designed and implemented based on the mechanisms of spatial influence diagrams. It uses HARDY (a hypertext diagramming tool) to display, create, edit and maintain land use problem models in the form of spatial influence diagrams, and interpret the diagrams into a CLIPS

(an expert system tool) representation. ILUDSS couples HARDY, CLIPS and ARC/INFO (a GIS software package), and can be used as a spatial decision support system for land use strategic decision making or other land use modelling tasks for the island of Islay. The user does not need either to understand the spatial influence diagram-based mechanisms or to have knowledge about HARDY, CLIPS and ARC/INFO on which the system is implemented. However, ILUDSS assumes the user has at least basic comprehension of the terminology of land use planning and the way spatial analysis works in a GIS context, for a successful dialogue with the system.

For automatic assessment of land use potential, the user can simply start by going to the FILE menu, and left-clicking the RUN button on it in the main window.

For more information on the mechanisms of spatial influence diagrams, please consult the reference: Xuan Zhu, 1995, A knowledge-based approach to the design and implementation of spatial decision support systems, Ph.D. thesis, the University of Edinburgh.

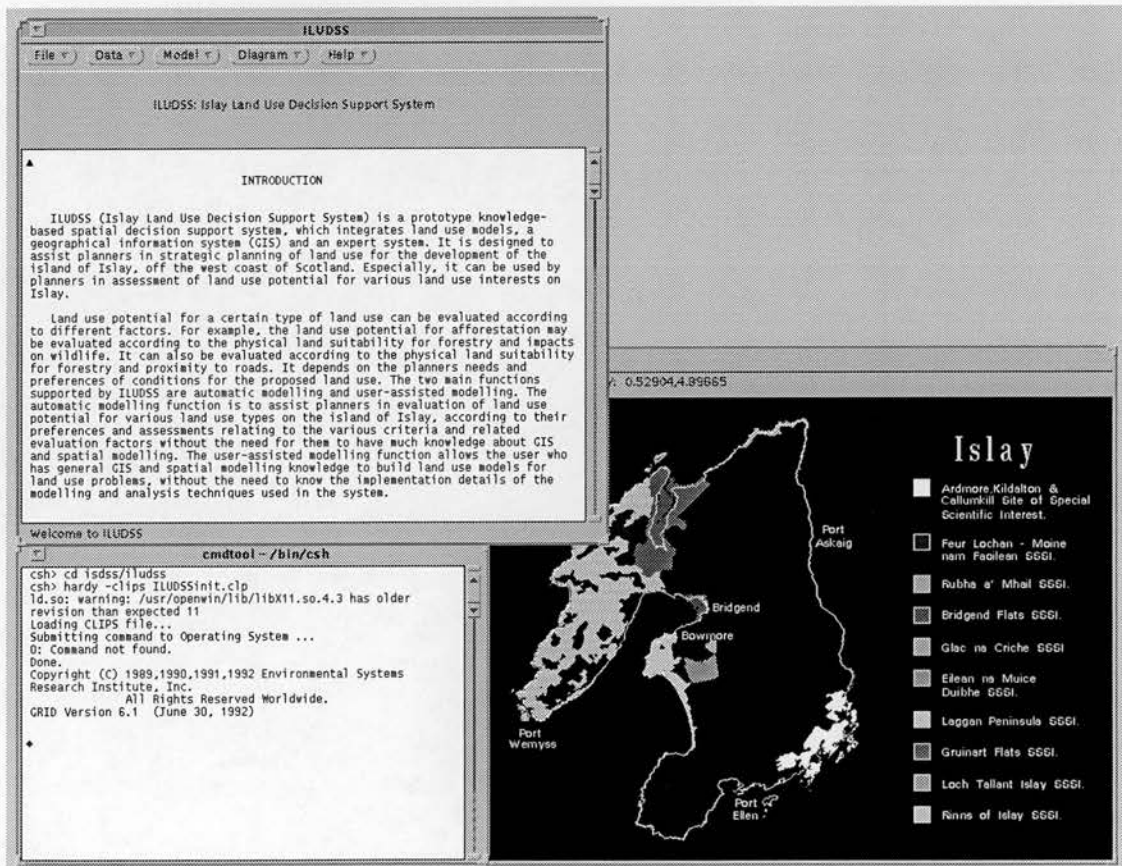


Figure A.1 Screenshot of ILUDSS when it starts up

The user may read the introductory information first if he or she would like, and then start the consultation by selecting the **File:Run** menu item in the main window.

Before proceeding, would you like more information about the assessment of land use potential in the system and the style of the system operations?

[Yes] [No]

The user clicks [Yes].

In order to evaluate land use potential for your land use interest, you are required to specify the land use type under consideration and select a set of attributes (indicating your preference relating to the evaluation factors). The system will then formulate a land use problem model using its own knowledge and existing data and models, present the problem model graphically to you in a spatial influence diagram, and then evaluate it automatically.

In the present implementation, ILUDSS can only deal with three types of land use: farming, afforestation and peat cutting, and three classes of attributes: physical land suitability, proximity (to desirable and undesirable land features) and area (the required minimum area of each land parcel). Land suitability refers to the fitness of a given type of land for a specified kind of land use. A proximity consideration may be the desire to locate afforestation sites near roads but a certain distance away from SSSI areas and exclusive of existing woodlands.

When you choose "proximity" as one of the factors or attributes for analysis, the system will prompt you to specify which desirable land features you wish the intended use to be near, and which undesirable land features you wish the intended use to avoid. At present, ILUDSS allows you to take into account proximity to roads, rivers, SSSI areas and existing woodlands. When you specify "area" as one of the attributes to be considered, the system will ask for the minimum area of each land parcel you require.

The situation that can be handled now by ILUDSS can be any of the three possible intended land uses with an arbitrary combination of the three classes of attributes pertinent to the selected use. In other words, you can choose one of the three land use types, and indicate the preferred evaluation factors by selecting a set of attributes from the three classes available to evaluate the land use potential for the intended use. Given the land use type and your choice of the attributes, ILUDSS will first formulate an objective model accounting for the selected attributes and land use type if it does not exist, and then formulate a land use problem model for the specified goal. Here, an objective model may be a procedure, a GIS analysis function, a set of rules encoded in

a knowledge base or an integration of them. The resulting land use potential will be presented to you in a map.

ILUDSS works in a consultant mode for automatic modelling. During the consultation, ILUDSS provides the relevant information. Pop-up dialogue windows are used to get information, report the status of the reasoning, and inform you of the system operations which are about to be undertaken or are currently in progress. The information given in the main window is mainly for explanation of the current modelling process and related GIS and spatial modelling terminology at appropriate times. The graphical display of a spatial influence diagram representing the model formulated to evaluate the land use potential for your land use interest will remain visible throughout the session so that you can comprehend the structure of the problem as well as the evaluation process.

[OK]

The user presses [OK].

ILUDSS is now ready to start the consultation. ILUDSS is mainly designed for the assessment of land use potentials for various land use interests. However, at the present, it can only deal with the following land use types: farming, afforestation and peat-cutting. If your land use interests are not in the list, select "QUIT" item.

Please specify one land use interest which is under consideration.

[Farming]  
[Afforestation]  
[Peat Cutting]  
[QUIT]

[OK] [Cancel]

The user selects [Afforestation] and then presses [OK].

Now, please make a choice from the following attribute list.

[Physical Land Suitability]  
[Proximity]  
[Area]

[OK]

The user chooses [Physical Land Suitability] and [Proximity] according to his or her preference relating to the evaluation factors, and then presses [OK].

Which land features do you think should be near afforestation sites?

[road]

[river]

[SSSI]

[woodland]

[OK] [Cancel]

The user chooses [road] according to his or her preference and knowledge, and presses [OK].

Which land features do you think should not be near or contain afforestation sites?

[road]

[river]

[SSSI]

[woodland]

[OK] [Cancel]

The user chooses [SSSI] and presses [OK].

Our goal is to determine the potential sites for afforestation according to the physical land suitability for forestry, the proximity to "road" and "SSSI".

Now we are about to formulate an objective model accounting for the attributes above.

[OK]

The user presses [OK], and the system proceeds to build an objective model using the domain-specific rules in the knowledge base. The rules construct the objective model using the AML language, name it *forestsitel* and store it to the file forestsitel.aml.

The objective model "forestsitel" has been formulated. Now we are going to formulate a land use problem model for our goal.

[OK]

The user presses [OK]. The system then proceeds to formulate a land use problem model without any user interaction, and present the structure of the problem graphically to the user. During the model formulation, a diagram window pops up at the top-right of the screen, within which a land use model to be formulated will be displayed as a spatial influence diagram. At the same time, the information about spatial influence diagrams is displayed in the main window.

A land use problem model has been formulated according to our goal. It is represented in the form of a spatial influence diagram shown in the top right diagram window. More information about the spatial influence diagram has been displayed in the main window. You may also go to Help for more information.

Now, you can view the graphical display of this land use problem model. After viewing, left click the <Evaluation> menu in the diagram window. Then the system will proceed to evaluate the problem model. You can modify the model before going to evaluate it, if necessary. The instructions for modification have been given in the main window. After modification, left click the <Evaluation> menu to evaluate it.

[OK]

The user presses [OK]. The screen looks now as shown in Figure A.2. The land use problem model formulated is displayed in the diagram window. The information displayed in the main window is as follows.

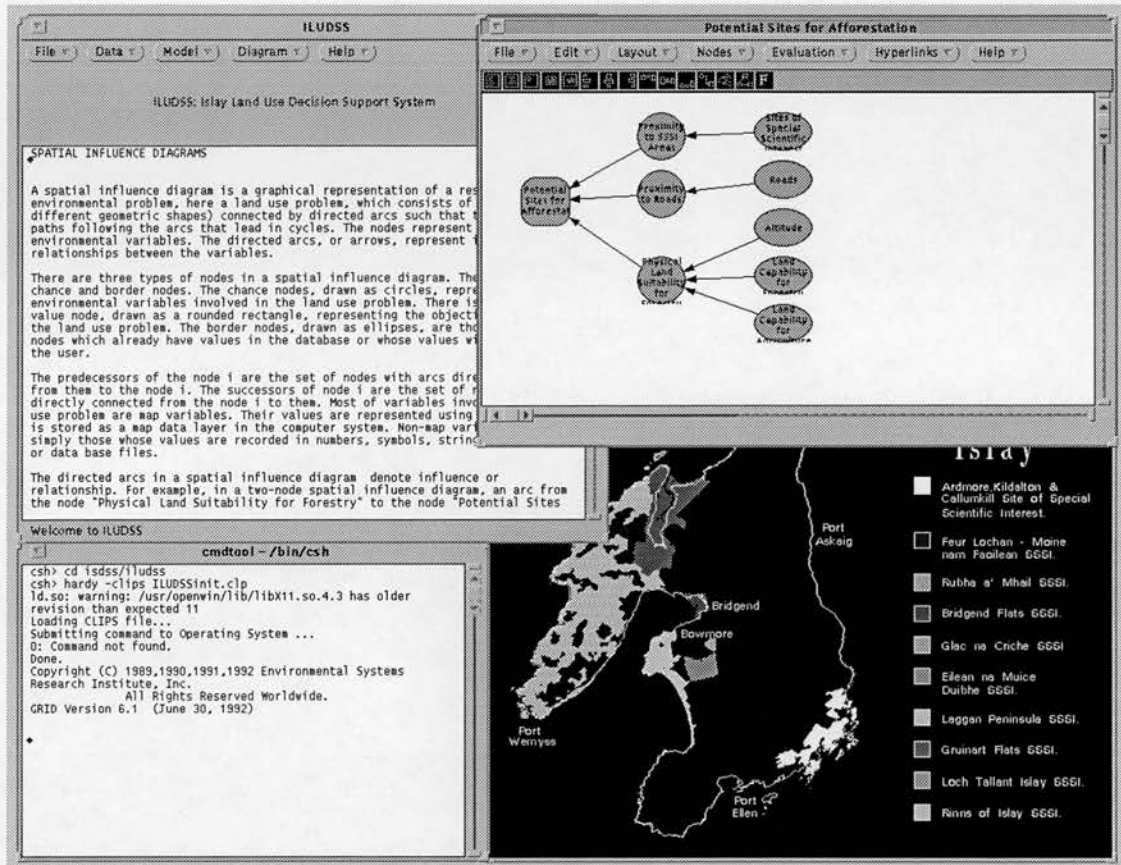


Figure A.2 Screenshot of ILUDSS when a land use model has been formulated

#### SPATIAL INFLUENCE DIAGRAMS

A spatial influence diagram is a graphical representation of a resource and environmental problem, here a land use problem, which consists of nodes (in different geometric shapes) connected by directed arcs such that there are no paths following the arcs that lead in cycles. The nodes represent the land and environmental variables. The directed arcs, or arrows, represent influences and relationships between the variables.



There are three types of nodes in a spatial influence diagram. They are value, chance and border nodes. The chance nodes, drawn as circles, represent land and environmental variables involved in the land use problem. There is at most one value node, drawn as a rounded rectangle, representing the objective in solving the land use problem. The border nodes, drawn as ellipses, are those chance nodes which already have values in the database or whose values will be given by the user.

The predecessors of the node  $i$  are the set of nodes with arcs directly connected from them to the node  $i$ . The successors of node  $i$  are the set of nodes with arcs directly connected from the node  $i$  to them. Most of variables involved in a land use problem are map variables. Their values are represented using maps. One map is stored as a data layer in the computer system. Non-map variables are simply those whose values are recorded in numbers, symbols, strings, data files or data base files.

The directed arcs in a spatial influence diagram denote influence or relationship. For example, in a two-node spatial influence diagram, an arc from the node "Physical Land Suitability for Forestry" to the node "Potential Sites for Afforestation" means that the Potential Sites for Afforestation is related to or influenced by the Physical Land Suitability for Forestry. Or you can consider that the Potential Sites for Afforestation are derived from the Physical Land Suitability for Forestry.

The spatial influence diagram is displayed or created in a diagram window (its default position is in the top-right corner of the screen). The label of each node indicates the name of the variable it denotes. If you point to a node and left click it, the full name can also appear on the status line underneath the diagram window.

Each node has several attributes attached to it. The attributes attached to a chance node include its name (the label displayed in the diagram), node-type (chance), predecessors (indicated using their internal names), successors (indicated using their internal names), attribute-relation, spatial-relation and internal-name. The attribute-relation describes the relationship between the node and its predecessors which does not involve locations. It is represented by an analytical (a procedure written using a procedural programming language) or a rule-based model (a set of rules encoded in the knowledge base). The spatial-relation describes the spatial relationship between the node and its predecessors. It is represented by a GIS model (composed of GIS analysis functions). We can say that the value of a node is derived from the values of its predecessors through an analytical or a rule-based model, and/or a GIS model. The relations between a node and its predecessors may be represented using either a single GIS model or a single analytical or rule-based model, or both. The internal name is used to facilitate the computer processing and store its corresponding data set or map data layer. So, if you would like to add or delete a data set or data layer associated with a node, you should use its internal name.

The attributes attached to a value node include its name (the label displayed in the diagram), node-type (value), predecessors (indicated using their internal names), attribute-relation, spatial-relation and internal-name. For a border node, the attributes include its name (the label displayed in the diagram), node-type (border), successors (indicated using their internal-names) and associated-data. The associated-data is the name of the data set or data layer representing the value of the border node. If it is not specified, the system will ask the user for the required data when it is being evaluated.

To view or edit the attributes associated with a node, point

to the node using the mouse, hold the control key and left click. A window pops up with all the available attributes and a text editing line for each. Having clicked on an attribute, a string can be typed in to represent its corresponding value, or you can view the value. After finishing viewing or assignment of values to all attributes, the user may press the "OK" button and return to the diagram window.

To create a spatial influence diagram, choose an item from the "Nodes" menu of the diagram window. A new node appears at a default position on the screen with no label. Drag the shape to the desired point on the screen by holding down the left button over the shape, moving the mouse and releasing. Then, edit the attributes of the node as described above. Having created two nodes, hold down the right mouse button, drag the mouse from the source node to the destination node and release the button, and then an arc between the two nodes is created. To delete a node or an arc, right click on it and select the pop up menu's <Delete Image> item. You may save a diagram by going to the <File> menu of the diagram window. For modification of a spatial influence diagram, follow the instructions above.

The user can read the information above if he or she would like. At this point, the user may judge the problem model using his or her knowledge. He or she can modify the model graphically by employing the existing models and data. The modification involves adding and deleting nodes and arcs, and changing or assigning values of attributes to related nodes. In this example, the user does to make any change. Then, the user presses the **Evaluation** menu in the diagram window. The system continues to evaluate the land use problem model.

Now, we are about to evaluate the land use problem model that has been built. If you want more information about the evaluation process, see the information displayed in the main window before you press the <OK> button.

[OK]

Below is the information displayed in the main window.

THE PROCESS OF EVALUATION OF A LAND USE PROBLEM MODEL

The evaluation of a land use problem model is carried out by means of access to the database, knowledge base and by integrating analytical models, rule-based models and GIS models. The process of evaluation starts from the existing border nodes (the ellipses). First, check the database to see whether their values exist. If not, ask the user for the data. Then, delete all border nodes which already have values. For each of remaining nodes, check whether all their predecessors have existing values. If yes, evaluate them using appropriate models, then delete all evaluated nodes. The process goes on until the value node (the rounded rectangular) is evaluated. The arcs in the graph indicate the order of evaluation of the variables. Note that the deletion of nodes will not be shown in the diagram window. The complete spatial influence diagram stays visible throughout the consultation so as to help the user understand the nature of the land use problem with ease and clarify the land use problem modelling process.

Now, the user presses the [OK] button and the system continues as follows.

The units used in the system are meters for length and square meters for area.

[OK]

The user presses [OK]. The system checks the existence of the values of the border nodes "Altitude", "Land Capability for Agriculture", "Land Capability for Forestry", "Roads" and "SSSI Areas", and determines that they are all in the database. Thus, the

system propagates these values to the successors of the five border nodes and removes them from the diagram. Afterwards, the nodes of "Physical Land Suitability for Forestry", "Proximity to Roads" and "Proximity to SSSI Areas" get the values of all their predecessors and can be evaluated. Then, the system proceeds to evaluate these nodes one by one as follows.

Now, we are going to evaluate the variable "Physical Land Suitability for Forestry" using the model "forestsuit" and the GIS model "overlay" based on the data: altitude agricap forestcap.

Would you like more information about "forestsuit"?

[Yes] [No]

The user presses [Yes].

The information you require has been displayed in the main window. After reading, press <OK> to continue.

[OK]

The following is the information about the model *forestsuit* displayed in the main window.

FORESTSUIT

"Forestsuit" is a rule-based model for evaluation of physical land suitability for forestry. It uses the following rules:

```
IF Land capability class of agriculture > 5 AND
   Land capability class of forestry <= 3 AND
   Altitude < 450m
THEN Suitable for forestry
```

```
IF Land capability class of agriculture > 5 AND
    Land capability class of forestry = 4 AND
    Altitude < 450m
THEN Moderately suitable for forestry

IF Land capability class of agriculture > 5 AND
    4 < Land capability class of forestry < 7 AND
    Altitude < 450m
THEN Limited for forestry

IF Land capability class of agriculture < 5 OR
    Land capability class of forestry = 7 OR
    Altitude >= 450m
THEN Unsuitable for forestry
```

After reading the information, the user presses [OK].

Would you like more information about "overlay"?

[Yes] [No]

The user presses [Yes].

The information you require has been displayed in the main window. After reading, press <OK> to continue.

[OK]

#### OVERLAY

"Overlay" is a widely-used GIS model for characterising spatial coincidence. It combines a set of existing digital maps, each of them describing the geographical distribution of a single attribute, to create a new digital map, identifying the particular attribute combination at each location in the

project area. The operations of "overlay" can be conceptualised as looking through a stack of acetate map sheets, delineating blobs of colour differences and labelling each parcel with your interpretation. Here, we are asking the computer to simulate the operations. A specific function (mathematical or logical model) is then used to compute values of categories on the new digital map from those of existing digital maps being overlaid. Such a function varies according to the nature of the data being processed and the specific use of those data within a modelling context.

For example, desirable areas for cottages might be defined as those areas that have a forest vegetation cover, have well-drained soils, and have a south-facing exposure. If vegetation, soils, and exposure are presented as separate digital maps in the GIS, then "overlay" operations, together with a logical operation could be used to identify the location where these conditions occur together.

If the model is implemented on raster data, the "reclassify" model will be used in order to create a new data layer depicting the modelling result.

The user then presses [OK]. Since it is running in *raster* mode, the system schedules the sequence of the models and determines to execute *overlay* first, then *forestsuit*, and finally *reclassify*.

We are about to run the model "overlay" in order to derive the value of the variable "Physical Land Suitability for Forestry".

Executing?

[Yes] [No]

The user presses [Yes]. The system is about to select an appropriate solver of the model *overlay*. According to the data structure of the input data, the system automatically selects the ARC/INFO function *combine* as a solver. Then, the system determines the values of the arguments of *combine* without any user interaction, executes it and finally stores the combined map to a data layer. When the solver is executed, the following message appear which tells the user that the solver *combine* has been selected and is being executed.

```
Overlaying multiple data layers using the ARC/INFO function  
"combine"...\nPlease wait...
```

After the execution of the model solver, the message above disappears and the following message pops up.

```
The operation of the model "overlay" is completed.
```

```
[OK]
```

The user presses [OK]. The system determines that the model solver of the model *forestsuit* to be executed next is a CLIPS knowledge base module. Thus, the system calls the utility program *InfoToClips* (a C program) to extract the attribute data from the data layer produced by the model *overlay* in the operation above and convert them into the CLIPS fact data, which is the input data for the model *forestsuit*. The data conversion does not need any user interaction. After the data conversion has completed, the system executes the model *forestsuit*.

```
We are about to run the model "forestsuit" in order to derive  
the value of the variable "Physical Land Suitability for  
Forestry".
```

```
Executing?
```



[Yes] [No]

The user presses [Yes] and the system executes the solver of the model *forestsuit*, i.e. activates the corresponding CLIPS knowledge base module and makes the inference. The result is stored to a file.

The operation of the model "forestsuit" is completed.

[OK]

The user presses [OK]. The system calls the utility program *ClipsToInfo* (an AML program) to convert the result produced by the CLIPS rule-based model *forestsuit* to an INFO file, which will be used as a reclassification table in the following operation to create a new data layer representing the value of the variable.

We are about to run the model "reclassify" in order to derive the value of the variable "Physical Land Suitability for Forestry".

Executing?

[Yes] [No]

The user presses [Yes]. The system is about to execute the GIS model *reclassify*. Similar to the execution of the model *overlay*, the system selects an appropriate solver of *reclassify* (here the ARC/INFO function *reclass*), determines the values of its arguments and executes it.

Reclassifying the values of the input data layer using the ARC/INFO function "reclass" according to a reclassification table...

Please wait...

After the operation above, a new data layer is created, which represents the value of "Physical Land Suitability for Forestry". To save space, the system automatically deletes all intermediate data layers produced during the evaluation of a node.

The evaluation of the variable "Physical Land Suitability for Forestry" is completed.

Would you like to view the result?

[Yes] [No]

The user presses [Yes]. Then, the map depicting the physical land suitability for forestry is displayed in the map display window, as shown in Figure A.3 (see the end of this Appendix).

The result has been displayed in the map display window. Press <OK> to continue.

[OK]

After viewing the result, the user presses [OK] and the system continues to evaluate another variable (node).

Now, we are going to evaluate the variable "Proximity to Roads" using the GIS model "buffering" based on the data: roads.

Would you like more information about "buffering"?

[Yes] [No]

The user presses [Yes].

The information you require has been displayed in the main window. After reading, press <OK> to continue.

The information about *buffering* is as below.

#### BUFFERING

"Buffering" is a GIS model for proximity analysis. It is used to create buffer zones. A buffer zone is an area of a given distance around a geographical feature, such as a point, line or an area. For example, for ease of harvesting, all planting areas should lie within 200 metres of the road network. Thus, buffer zones of 200-metre around the existing roads may be generated using "buffering" to indicate those areas that are highly suitable for planting.

Before the model is executed, the system will ask you to input your preferred distance used to create buffer zones around the input features. If you specified the buffer distance as "0", there would be no buffer zone to be created. The unit for length is meters.

After pressing [OK], the system prompts the user to input a buffer distance.

Please input your preferred distance below (meters) in order to calculate Proximity to Roads.

---

[OK] [Cancel]

The user types 500 and presses [OK].

We are about to run the model "buffering" in order to derive the value of the variable "Proximity to Roads".

Executing?

[Yes] [No]

The user presses [Yes]. The system selects *buffer* composed of ARC/INFO functions as the solver of the model *buffering*, assigns its parameters and executes it.

Creating buffer zones around the input features using the program "buffer" which consists of several ARC/INFO functions...

Please wait...

The system is executing *buffer*.

The operation of the model "buffering" is completed.

The execution of *buffer* has finished. The result is stored as a data layer.

The evaluation of the variable "Proximity to Roads" is completed.

Would you like to view the result?

[Yes] [No]

The user presses [Yes]. The data layer depicting the 500-meter buffer zone around roads is displayed in the map display window, as shown in Figure A.4 (see the end of this Appendix).

The result has been displayed in the map display window. Press <OK> to continue.

After the user presses [OK], the system continues to evaluate the variable "Proximity to SSSI Areas".

Now, we are going to evaluate the variable "Proximity to SSSI Areas" using the GIS model "buffering" based on the data: sssi.

Would you like more information about "buffering"?

[Yes] [No]

The user presses [No]. The following operations are exactly similar to those in evaluation of the variable "Proximity to Roads" as described above.

Please input your preferred distance below (meters) in order to calculate Proximity to Roads.

---

[OK] [Cancel]

The user types 100 and presses [OK].

We are about to run the model "buffering" in order to derive the value of the variable "Proximity to Roads".

Executing?

[Yes] [No]

The user presses [Yes].

Creating buffer zones around the input features using the program "buffer" which consists of several ARC/INFO functions...

Please wait...

The model is being executed.

The operation of the model "buffering" is completed.

The result has been produced.

The evaluation of the variable "Proximity to SSSI Areas" is completed.

Would you like to view the result?

[Yes] [No]

The user presses [Yes]. Then, the map depicting the 100-meter buffer zone around the SSSI areas is displayed in the map display window, as shown in Figure A.5 (see the end of this Appendix).

The result has been displayed in the map display window. Press <OK> to continue.

The user presses [OK], and the system continues to evaluate the value variable "Potential Sites for Afforestation".

Now, we are going to evaluate the value variable "Potential Sites for Afforestation" using the model "forestsitel".

[OK]

The user presses [OK].

We are about to run the model "forestsitel" in order to derive the value of the variable "Potential Sites for Afforestation".

Executing?

[Yes] [No]

The user presses [Yes]. The system calls the objective model *forestsitel* formulated above and executes it.

Evaluating the value variable using the objective model formulated...  
Please wait...

Now the execution has finished. The final result is stored as the data layer "forestsitel".

The operation of the model "forestsitel" is completed.

[OK]

The user presses [OK].

The specified goal of the consultation has been reached. The final map is stored as the data layer "forestsites1". We are about to display the result.  
Press <OK> and wait for a while.

[OK]

The user presses [OK]. The final result is displayed in the map display window as shown in Figure A.6. With this result, the specified goal of the consultation is reached and the session is terminated.

At the moment, the user may continue to start another session to evaluate another land use problem on Islay, or quit the system. Assume that the user would like to evaluate the land use potential for peat cutting on Islay, so as to compare it with the land use potential for afforestation. The user starts the session by selecting the **File:Run** menu item in the main window. The system then runs as follows. Since

the system operation is similar to that in the previous session as described above, some steps of operation are omitted.

ILUDSS is now ready to start the consultation. ILUDSS is mainly designed for the assessment of land use potentials for various land use interests. However, at the present, it can only deal with the following land use types: farming, afforestation and peat-cutting. If your land use interests are not in the list, select "QUIT" item.

Please specify one land use interest which is under consideration.

[Farming]  
[Afforestation]  
[Peat Cutting]  
[QUIT]

[OK] [Cancel]

The user selects [Peat Cutting] and then presses [OK].

Now, please make a choice from the following attribute list.

[Physical Land Suitability]  
[Proximity]  
[Area]

[OK]

The user chooses [Physical Land Suitability] and [Proximity], and then presses [OK].

Which land features do you think should be near peat cutting sites?

[road]  
[river]  
[SSSI]  
[woodland]

[OK] [Cancel]



The user chooses [road] and presses [OK].

Which land features do you think should not be near or contain peat cutting sites?

[road]

[river]

[SSSI]

[woodland]

[OK] [Cancel]

The user chooses [SSSI] and presses [OK] .

Our goal is to determine the potential sites for peat cutting according to the physical land suitability for peat cutting (peat deposits), the proximity to "road" and "SSSI".

Now we are about to formulate an objective model accounting for the attributes above.

[OK]

The user presses [OK]

The objective model "peatsite1" has been formulated. Now we are going to formulate a land use problem model for our goal.

[OK]

The user presses [OK]. A land use problem model is formulated and displayed in the diagram window as shown in Figure A.7 (see the end of this Appendix).

Since the variables "Proximity to SSSI" and "Proximity to Roads" have been evaluated in the previous session, they are border nodes in this model. However, the existing value of "Proximity to Roads" is the 500-meter zones around roads.

According to the user's knowledge, the peat cutting areas should be located within 300-meter zones around roads. Therefore, the node "Proximity to Roads" should be converted into a chance node, so that it can be re-evaluated. One of the methods to do so is to modify the diagram graphically (another method is to delete the data layer associated with "Proximity to Roads", then start the session again to formulate another problem model). The user here chooses to modify the model graphically. The diagram is modified according to the user's knowledge and is shown as in Figure A.8 (see the end of this Appendix). Then, the user presses the **Evaluation** menu in the diagram window. The system proceeds to evaluate the model.

Now, we are about to evaluate the land use problem model that has been built. If you want more information about the evaluation process, see the information displayed in the main window before you press the <OK> button.

[OK]

The user presses [OK].

The units used in the system are meters for length and square meters for area.

[OK]

The user presses [OK].

Now, we are going to evaluate the variable "Peat Deposits" using the model "soil2peat" and the GIS model "reclassify" based on the data: soils.

Would you like more information about "soil2peat"?

[Yes] [No]

The user presses [No].

Would you like more information about "reclassify"?

[Yes] [No]

The user presses [NO].

We are about to run the model "reclassify" in order to derive the value of the variable "Peat Deposits".

Executing?

[Yes] [No]

The user presses [Yes].

Reclassifying the values of the input data layer using the ARC/INFO function "reclass" according to a reclassification table...

Please wait...

After the operation finishes, a new data layer is created.

The evaluation of the variable "Peat Deposits" is completed. Would you like to view the result?

[Yes] [No]

The user presses [Yes]. Then, the map depicting the peat deposits is displayed in the map display window, as shown in Figure A.9 (see the end of this Appendix).

The result has been displayed in the map display window. Press <OK> to continue.

[OK]

After viewing the result, the user presses [OK].

Now, we are going to evaluate the variable "Proximity to Roads" using the GIS model "buffering" based on the data: roads.

Would you like more information about "buffering"?

[Yes] [No]

The user presses [No]. The system prompts the user to input a buffer distance.

Please input your preferred distance below (meters) in order to calculate Proximity to Roads.

---

[OK] [Cancel]

The user types 300 and presses [OK].

We are about to run the model "buffering" in order to derive the value of the variable "Proximity to Roads".

Executing?

[Yes] [No]

The user presses [Yes]. Because the data layer associated with the node "Proximity to Roads" has not been deleted, the following message will pop up.

WARNING! The data layer depicting the values of the variable already exists.

Continue the operation?

[Yes] [No]

The user presses [Yes]. The system first deletes the data layer and then performs the following operation.

Creating buffer zones around the input features using the program "buffer" which consists of several ARC/INFO functions...

Please wait...

After the execution of *buffer* is finished, the result is stored as a data layer.

The evaluation of the variable "Proximity to Roads" is completed.

Would you like to view the result?

[Yes] [No]

The user presses [Yes]. The data layer depicting the 300-meter buffer zone around roads is displayed in the map display window, as shown in Figure A.10 (see the end of this Appendix).

The result has been displayed in the map display window. Press <OK> to continue.

After the user presses [OK].

Now, we are going to evaluate the value variable "Potential Sites for Peat Cutting" using the model "peatsite1".

[OK]

The user presses [OK].

We are about to run the model "peatsitel" in order to derive the value of the variable "Potential Sites for Peat Cutting".

Executing?

[Yes] [No]

The user presses [Yes].

Evaluating the value variable using the objective model formulated...

Please wait...

Now the execution has finished.

The operation of the model "peatsitel" is completed.

[OK]

The user presses [OK].

The specified goal of the consultation has been reached. The final map is stored as the data layer "peatsites1". We are about to display the result.

Press <OK> and wait for a while.

[OK]

The user presses [OK]. The final result is displayed in the map display window as shown in Figure A.11. Now, the user can view the map of "Potential Sites for Peat

Cutting”, retrieve and display the map of “Potential Sites for Afforestation”, and compare the results. The process can be repeated until the user finishes his or her consultation.

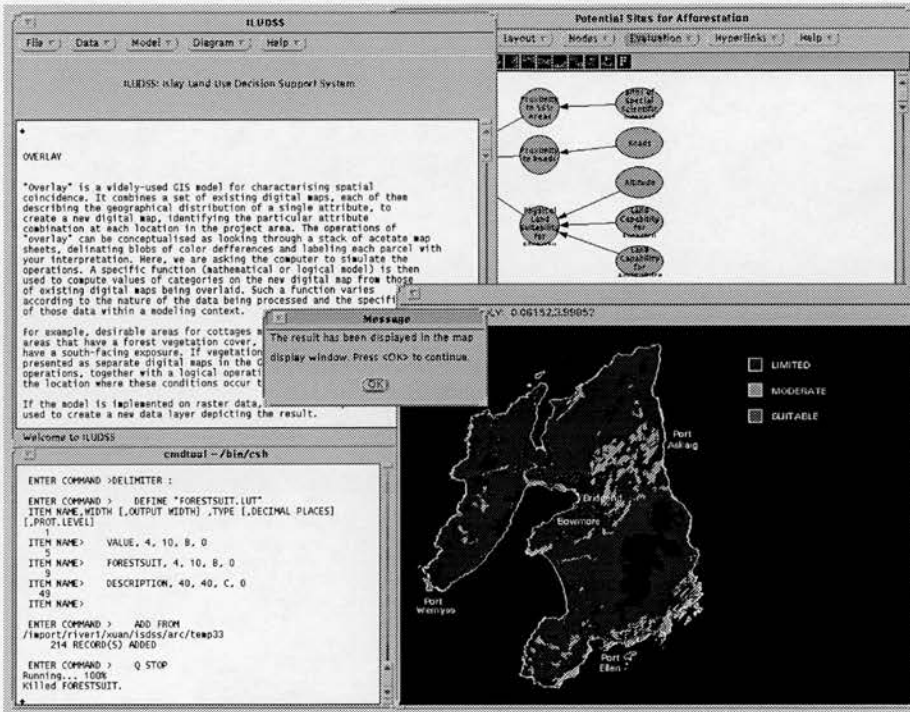


Figure A.3 Screenshot of ILUDSS when the map of physical land suitability for forestry is displayed

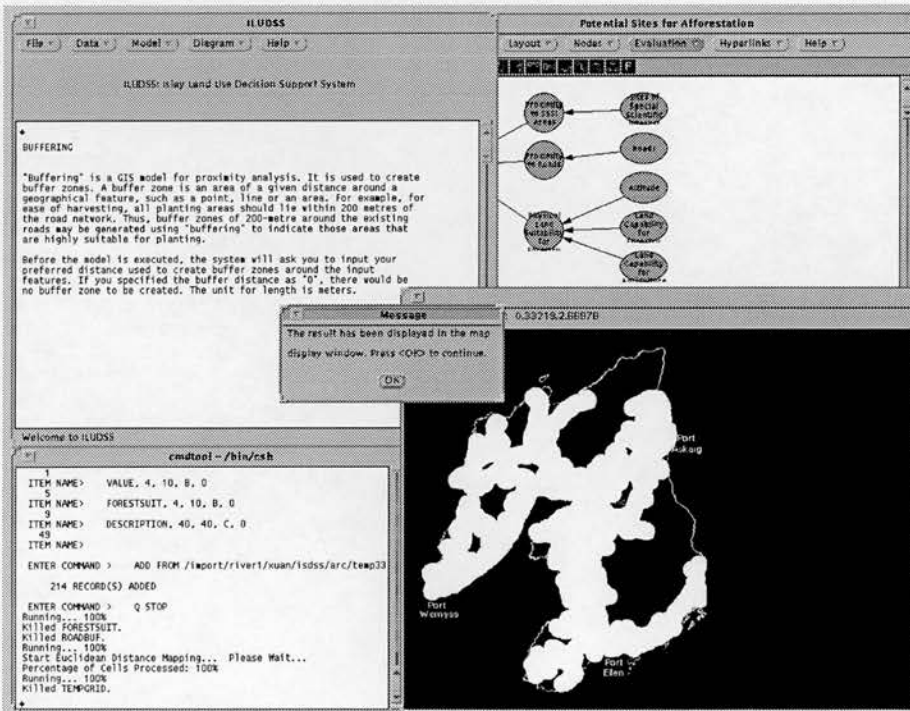


Figure A.4 Screenshot of ILUDSS when the 500-meter buffer zone around roads has been created and is displayed



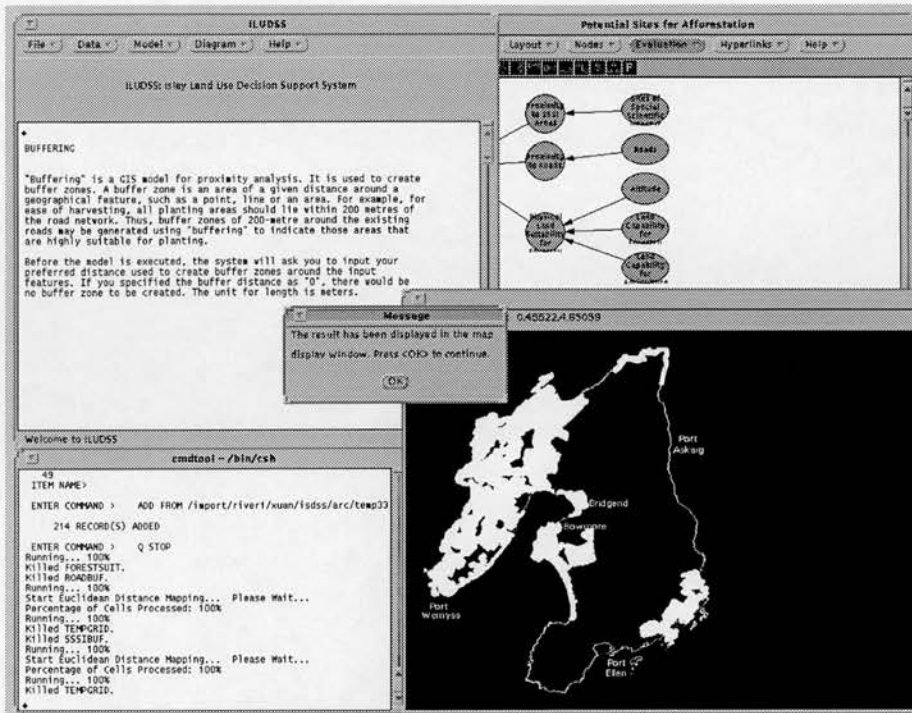


Figure A.5 Screenshot of ILUDSS when the 100-meter buffer zones around SSSI areas have been created and are displayed

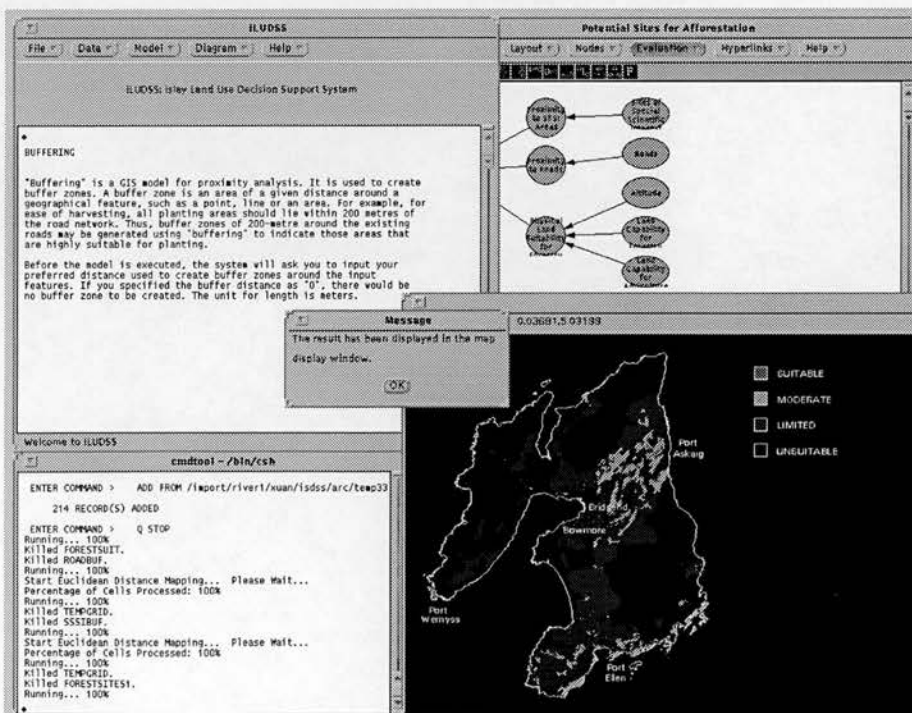


Figure A.6 Screenshot of ILUDSS when the final result is reached and displayed

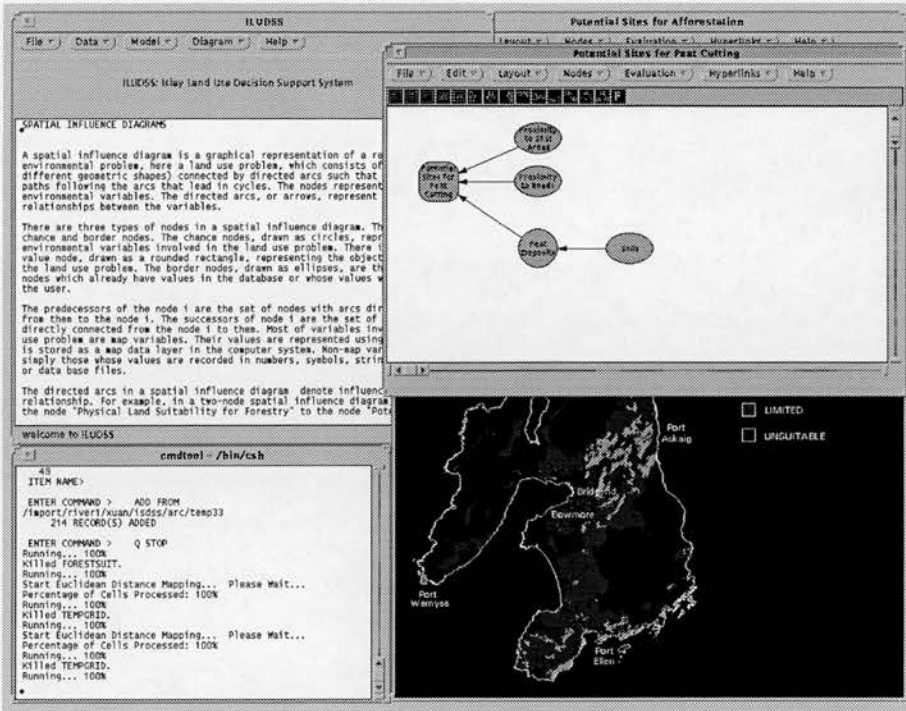


Figure A.7 Screenshot of ILUDSS when the land use model for evaluating the land use potential for peat cutting has been formulated and displayed

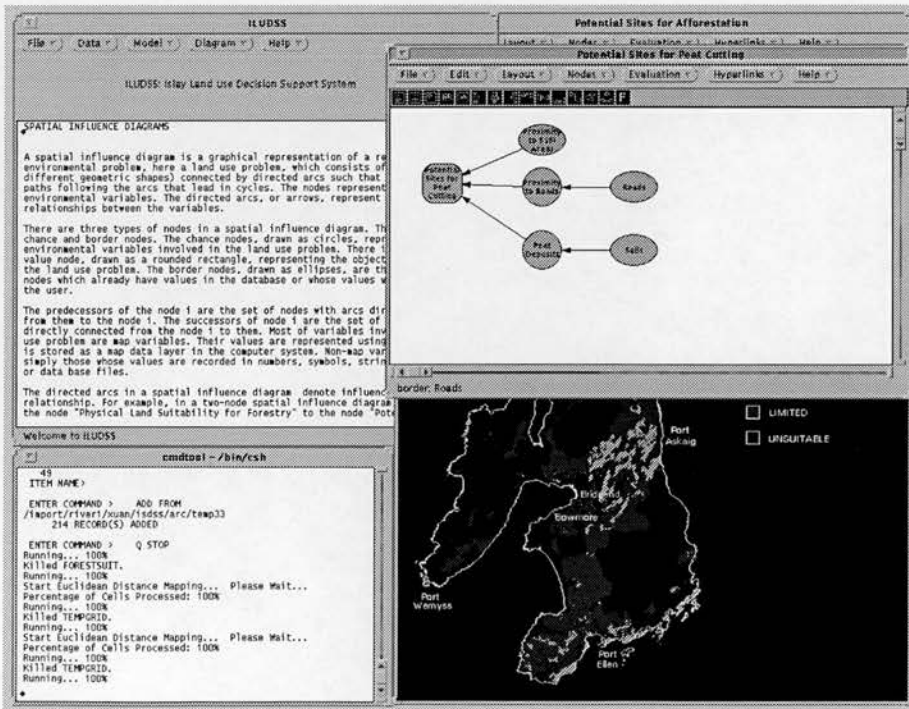


Figure A.8 Screenshot of ILUDSS after the land use model for evaluating the land use potential for peat cutting has been modified

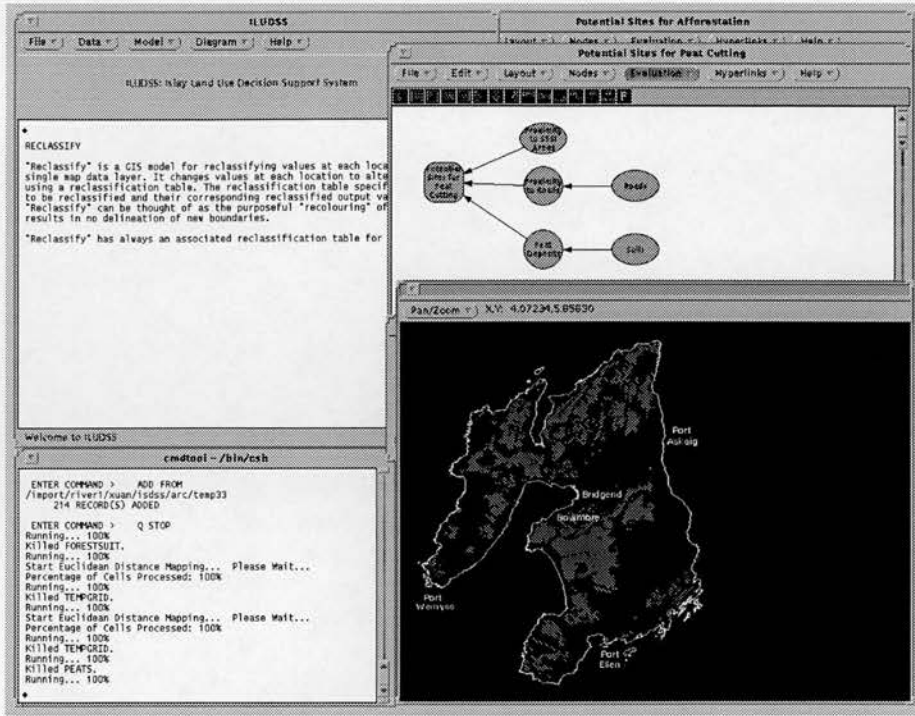


Figure A.9 Screenshot of ILUDSS when the map of peat deposits is displayed

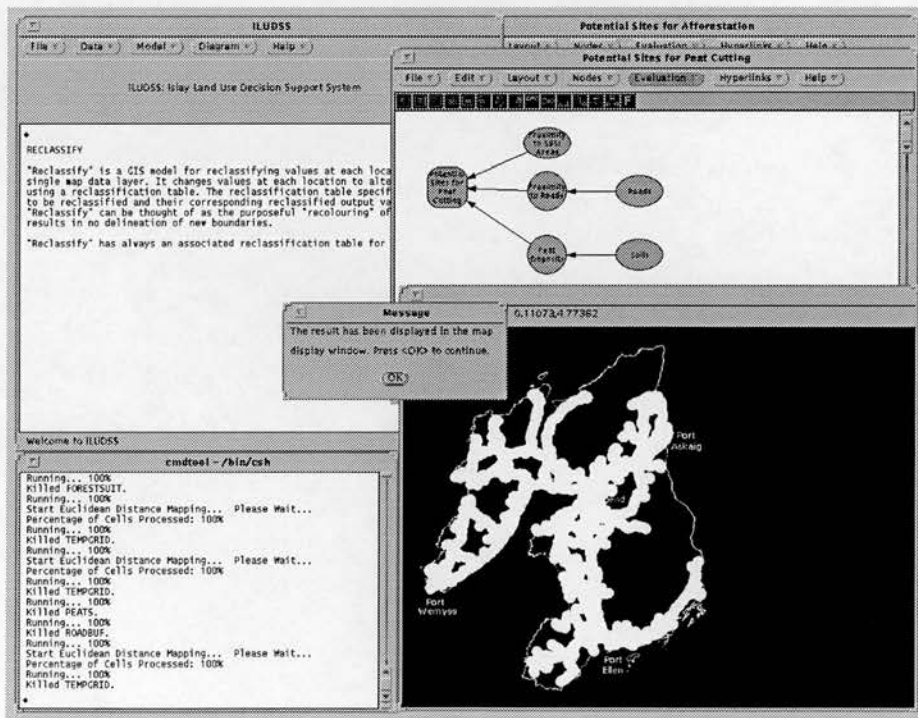


Figure A.10 Screenshot of ILUDSS when the 300-meter buffer zone around roads has been created and is displayed



# Appendix G

## Arithmetic Expressions in ILUDSS

The general arithmetic expression in ILUDSS would be:

*<operand-1> <arithmetic-operator> <operand-2>*

The *operand-1* and *operand-2* can be variables, numbers and arithmetic expressions.

At present, the arithmetic operators that can be used include

+	addition
-	subtraction
*	multiplication
/	division

A mathematical formula can be written using an arithmetic expression with “=” as an assignment operator. The general form would be:

*<output> = <arithmetic-expression>*

Here, the *output* is a variable name.

# Appendix H

## Logical Expressions in ILUDSS

The general form of a logical expression used in ILUDSS is

*<operand-1> <logical-operator> <operand-2>*

The *operand-1* can be a variable or an arithmetic expression with the following operators: +, -, \*, /. The *operand-2* can be a value, a variable, or an arithmetic expression. A value may be a number, or a character string. A string should be quoted using single quotes. The *logical-operator* may be

==	equal to
^=	not equal to
>=	greater than or equal to
<=	less than or equal to
>	greater than
<	less than

Logical expressions can also be connected using the following key words:

&	Boolean AND, the logical expressions on both sides of & must be true
	Boolean OR, the logical expressions on one or both sides of   must be true

! Boolean exclusive OR, the logical expressions on one and only one side of ! must be true.

# Appendix I

## Shade Colour Set Used in ILUDSS

The following colour set is provided in ILUDSS for shading polygon features or drawing point and line features.

1	white	2	red
3	green	4	blue
5	cyan	6	magenta
7	yellow	8	orange
9	pale green	10	chartreuse
11	navy	12	purple
13	deep pink	14	grey
15	dim grey		