

GIS, Expert Systems and Interoperability

Linda Lilburne¹, George Benwell² & Roz Buick³

¹ Landcare Research
P. O. Box 69
Lincoln
New Zealand
Ph +64 (3) 325 6700
Fax +64 (3) 325 2418

² Department of Information
Science
Otago University
P. O. Box 56
Dunedin
New Zealand

³Trimble Navigation
P. O. Box 8729
Christchurch
New Zealand

Email: lilburnel@landcare.cri.nz

1. Introduction

How should geographic information systems be developed? There is a strong demand from users for enhanced functionality and power. Vendors can and do respond to these demands. But where will this lead? Will the result be one all-embracing and all-conquering program or geographic information system (GIS)? A GIS could grow to incorporate all statistical functions, all visualisation techniques, all data management functions etc. It is possible to perceive a scenario in which GIS is developed to 'bloatware' proportions.

An alternative scenario is one in which a GIS is interfaced with other software systems. Embedding database bridges and other product-specific links, providing data import and export routines, and system calls are all ways of interfacing GIS with other systems. GIS vendors could opt to produce a 'linkware' GIS, interfaced to as many third party systems as possible.

Given these two alternatives to GIS development, an interesting set of questions arises. How far do vendors go with enhancing their systems compared with interfacing with third party systems? Is there a balance? Or do GIS users just keep calling for 'more', regardless of the solution set?

There is a balance. GIS is likely to be developed by being enhanced AND by being interfaced with third party software. In a way, this is a third developmental track leading to an increasingly functional GIS whose ability to interact with other systems is greatly improved. This interoperable GIS allows flexible combinations of system components while still providing a comprehensive range of spatial operations and analytical functions. Figure 1 depicts the three developmental tracks, leading to the 'bloated' GIS, the linked GIS, or the interoperable GIS in an environment in which systems can cooperate.

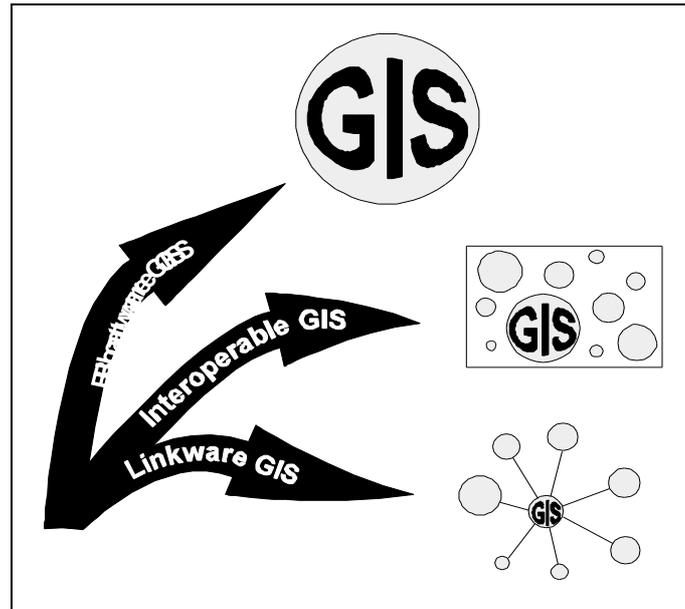


Figure 1 Three developmental tracks

Of these three developmental tracks, this paper presents an example of what can be achieved with the interoperable GIS. Expert systems are introduced along with the client/server and object-oriented paradigms. By using these paradigms, a generic, spatial, rule-based toolbox called SES (spatial expert shell) has been created. SES is described using examples and contrasted with other documented expert system – GIS linkages. But first integration is modelled in three dimensions to highlight the need for improvements in how GISs can interact with other systems.

2. The integration cube

Integration has been described by Fedra (1993), Goodchild (1992), and Nyerges (1992) from two perspectives: data and user interface. A more comprehensive model of integration can be described by adding a third perspective: interoperability. This relates to the ability of two systems, or processes, to cooperate with each other. Processes, which can exchange a variety of dynamically determined requests and information, have high interoperability. The interwoven nature of the requests allows the functions of each system to be highly integrated. More commonly however, integration is achieved by starting, executing and exiting a second process either after, or at some predefined point within, the first process. The flow of functionality between the systems is sequential. One process is followed by a second process, after which control may return to the first system. Interoperability is low.

By using the three dimensions of user interface, data and interoperability, integration can be portrayed as a cube (Figure 2). This gives a more complete picture of a coupling between two systems. The data axis progresses from no exchange of data, transfer of data, to sharing data. The user interface axis extends from a position of no link between two interfaces, to a trigger (e.g. a menu option or button) between two interfaces, to two interfaces which have a similar look and feel, through to a single interface. The

interoperability axis ranges from static, sequential integrations with limited functionality to dynamic, interleaved linkages with full functionality. The degree of integration is represented by its position in the cube. The front bottom left corner represents no integration; the back top right corner of the cube represents a fully integrated system.

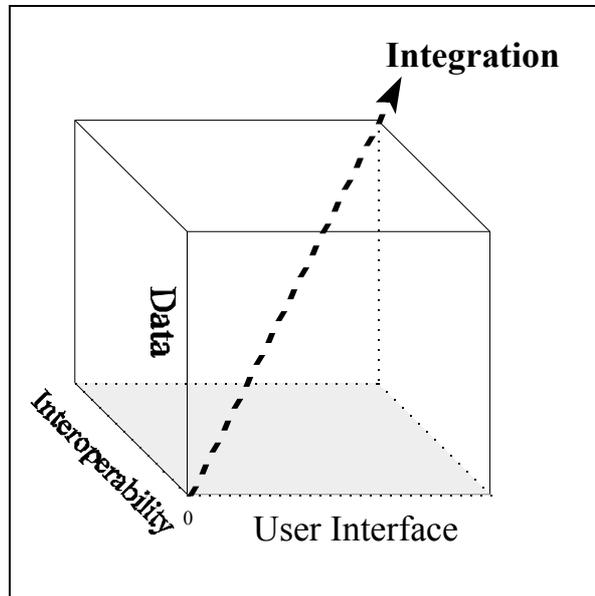


Figure 2: Integration cube

A sequentially integrated system can now be differentiated from a system in which the components of the integrated system can freely interact with each other. This model is defined in detail in Lilburne (1996). Lilburne also demonstrates how most integrations are positioned on the front plane of the cube, i.e. have low interoperability scores.

3. GIS and Expert Systems

An expert system is a “computer program designed to model the problem-solving ability of a human expert” (Durkin, 1994 p. 7). It comprises a knowledge base (KB), an inference engine, and working memory. The expert’s domain knowledge is stored in the KB. The inference engine can process or reason with this knowledge to draw conclusions about a problem. The working memory contains the facts and deductions made in a session. Expert systems are used to identify, monitor, diagnose, predict, control, specify, design, configure, and plan (Jackson, 1990). An expert system can combine many types of knowledge including intuition, experience, qualitative beliefs, heuristics, empirical observations, and expert judgement.

The advantages of integrating a GIS with an expert system have been recognised by a number of authors (Burrough, 1986; Fedra, 1995; Fischer, 1994; Fisher et al., 1988; Lein, 1992; Leung, 1993; Robinson et al., 1987; Smith and Yiang, 1991; Zhu and Healey, 1992). These authors saw the expert system as having the potential to add

intelligence to GIS tasks, e.g. map design, generalisation, automated name placement, feature extraction, spatial query. Fischer (1994) notes the increasing use of AI techniques to represent meta-data. Burrough (1992) envisions how an expert system could be an adviser or tutor for using a GIS. Domain knowledge represented in an expert system together with spatial data can provide a decision support environment in which users are guided by the integrated system towards a recommendation.

Table 1 contrasts the strengths and weaknesses typically observed in expert systems and GISs. This both shows strengths that are complementary and how some of the weaknesses of one system are matched by strengths in the other.

Expert system	GIS
qualitative	quantitative
imprecise data	precise data
uses symbols	uses geometric primitives eg point, line segment
integrates knowledge	integrates data
handles incomplete data and knowledge	does not easily handle incomplete data
suited to unstructured problems	suited to structured problems
no spatial capability	spatially capable
handles incomplete data and knowledge	does not easily handle incomplete data
does not cope well with lots of data	cope with large volumes of data
explanation facility	no explanation facility
can represent knowledge	is not designed to represent knowledge
can manage knowledge	can not easily manage knowledge
has inference engine	no inference or reasoning capability
opportunistic	algorithmic i.e. sequential
no mapping/graphing capability	variety of output maps/graphics
can not efficiently do arithmetic operations	can efficiently perform geometrical ops.

Table 1 Comparison of some expert system and GIS characteristics

The vision of those promoting expert system – GIS linkages was developed five or more years ago. To date, linkages have not fulfilled their vision. In part this is due to over-optimistic rhetoric about the capabilities of expert systems. It also relates to the level of interoperability between a GIS and an expert system.

For example, one application of an expert system – GIS linkage is to assist with solving or understanding environmental problems which are often very complex. GIS allows the real world to be modelled in its spatial context. Knowledge of real world objects and processes can be represented in the expert system. Sometimes the best knowledge

available is in the form of heuristics. Combining spatial data with knowledge offers real opportunities in the management of our natural resources (Fedra, 1995). However, the real world is not a series of sequential processes; rather it is a complex tapestry of interactions between objects and processes. Hence the degree of interoperability between an expert system and a GIS will affect the ability of an integrated system to model the complexity of the real world. Recent technological advances offer new potential for closer interaction between systems, in particular the client/server and object-oriented technologies.

4. Technological paradigms

Client/server technology refers to the software that allows a process to receive messages from another process. These messages request services of the receiving system (the server). The service might be to perform a specified action or to return some information to the requesting system (the client). Both processes remain in memory concurrently, avoiding the loss of performance that occurs when loading a system into memory every time one of its functions is required. There is no limit to the number of requests, nor are there any restrictions on the types of requests that can be made. Both systems must conform to a common client/server protocol. There are incompatible client/server protocols, e.g. DDE, OLE (PC), RPC (UNIX), APPC (IBM).

Object-oriented (OO) technology is based on objects which have an identity, state and behaviour (Booch, 1994). 'State' refers to the data or values associated with the object at a particular point in time. 'Behaviour' is how an object acts and reacts. Key characteristics of OO technology that are important in a GIS – expert system link are abstraction, encapsulation, inheritance, and polymorphism. Abstraction is a simplified description of an object which encompasses all of its essential characteristics. Encapsulation, or information hiding, is the ability to hide implementation details from the user. Inheritance allows objects to inherit behaviour and state from parent objects. Polymorphism is the ability to redefine or override inherited behaviour from parent classes.

Abstracting the essential qualities that are useful for a given domain and grouping those objects with similar qualities is a powerful way of simplifying the computer representation of a problem. These essential qualities characterise the state and the behaviour of objects. Behaviour is defined by abstracting the essential operations that an object can perform and which can be performed on it.

A *vector* class in an expert system, defines the state and behaviour of GIS vector layers. The state of a *vector* object includes the name of the GIS layer, its description, default colour and its physical location. Behaviour is described in methods which are associated with the *vector* class. These methods are routines which encapsulate the GIS commands to draw, delete, create, modify and manipulate GIS layers. The commands are hidden from the user who does not need to know how GIS operations are implemented. Encapsulation also serves to hide the complexities of GIS data representation.

Objects inherit state and behaviour from parent classes. For example, a *roads* object, representing a GIS road network layer, inherits the state and behaviour of its parent

vector class. Each object representing a GIS layer inherits a constructor method called *create*. This method encodes commands needed to select the required number of features from the layer, create a sub-object for each feature and transfer the GIS attributes to the sub-object. This allows the details of how to create objects that are sourced from a GIS to be associated with the appropriate object, rather than buried in a transformation routine written in a 3GL¹. This facilitates maintenance of data exchange between the two systems.

Multiple inheritance is very useful in a GIS – expert system combination, as it allows an object to inherit behaviour from application specific classes or objects as well as GIS related classes. For example, a *road_segment* object inherits state and behaviour from the GIS *line* class, the specific *road_I* object(s) that the segment is part of and the *roads* vector layer object.

Polymorphism is useful to override inherited behaviour. For example, objects can inherit a *display* method which instructs the GIS to draw the object appropriately. Polymorphism allows a specialised *display* method to override the inherited one during processing.

5. SES design

SES (Spatial Expert Shell) integrates two commercial products: the GIS ARC/INFO (ESRI Inc., 1991), and the expert system shell, Smart Elements (Neuron Data Inc., 1994). ARC/INFO v7 includes some new commands (IAC²request, IACconnect, IACdisconnect, IACreturn) which create a framework for client/server communication with another process. Once a connection has been initialised, messages can be sent between the processes. Smart Elements is a combination of a hybrid frame, rule-based expert shell called Nexpert Object, and a GUI³ developers kit called Open Interface. It has an Application Programming Interface (API) which allows C routines to access Smart Elements functions. SES is developed on a Solaris SUN Workstation platform which both ARC/INFO and Smart Elements support. Both systems use Sun's ONC-RPC client/server protocol. Smart Elements is the client and ARC/INFO is the server. A combination of C and ARC/INFO's macro language AML is used to develop the client/server interface between ARC/INFO and Smart Elements.

Essentially SES is a collection of spatial classes with predefined state and behaviour. The expert system shell is extended to include spatial classes. GIS elements (e.g. the display window, vector data, raster data) are modelled as classes under a top level class *gisObject* (Figure 3). Generalised classifications of spatial data are modelled as classes under the *gisObjectType* class. Appropriate spatial methods are associated with all of these classes. For example, polygon related state (area, perimeter) and behaviour (draw, adjacency, overlap) is defined in the slots and methods of the *polygon* class. An attribute of the state is stored in a 'slot'. Methods describe the operations or behaviour of an

¹ 3GL = Third generation language

² IAC = Inter-application communication

³ GUI = Graphical user interface

object. *Vector* and *grid* classes have slots and methods that define vector and raster layer behaviour, e.g. how layers should be drawn, how features can be selected, and in which colour they should be drawn.

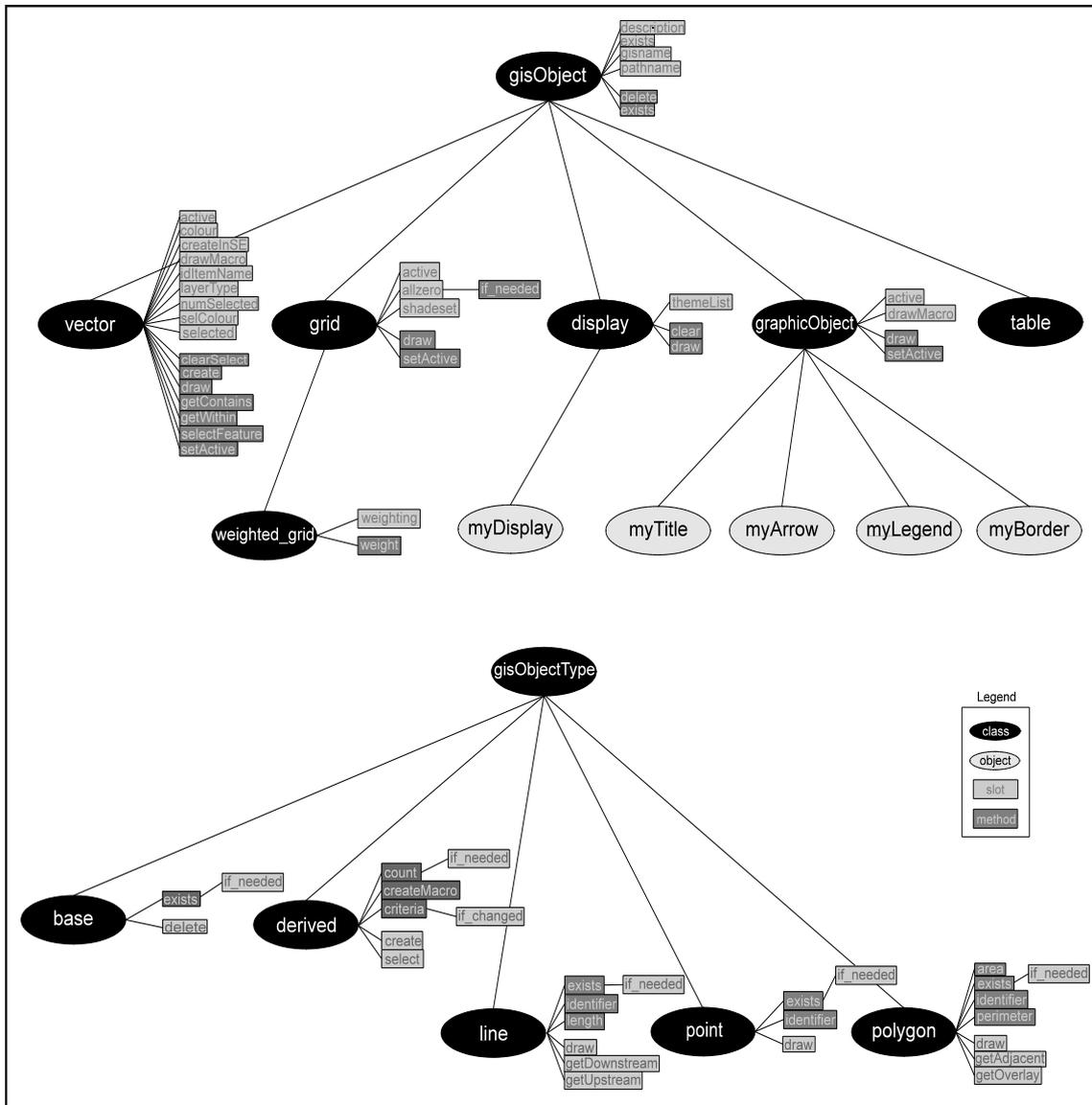


Figure 3 SES class diagram

A spatial expert system application can be developed by creating objects belonging to the spatial classes that make up SES. For example, in an agricultural application, there might be GIS layers of paddocks, tracks, streams, buildings, soil types, and a DTM. Objects would be created for each of these layers, linked to either the *vector* or the *grid* class. The objects inherit slots from these parent classes, e.g. the *paddocks* vector object inherits the *colour* slot and its default value “white” from the *vector* class. The *paddocks* object’s default *colour* value could be redefined to “yellow”. Methods are also inherited. For example, the *paddocks*, *tracks*, *streams*, and *buildings* objects all inherit the *selectFeature* method from the *vector* class. This enables an application expert system

developer to transparently access a GIS function in which GIS features are selected from a map. For example, the DTM object inherits a raster specific *draw* method and default colour scheme from the parent *grid* class.

The application developer creates additional slots defining the domain state of each vector layer object. In the agricultural example, the *soils* object might have inheritable slots which represent the GIS attributes of the soil layer, e.g. soil name, soil code, pH level, soil depth. The *create* method associated with the *soils* object will dynamically create sub-objects of the *soils* object. Each sub-object, *soil_1*, *soil_2* etc., represents a single feature in the vector layer, e.g. a polygon. The sub-objects inherit domain slots, i.e. *soil_name*, *soil_depth* etc. which are populated by attribute values from the GIS. The sub-objects are linked to *gisObjectType* classes so that GIS state and behaviour can also be inherited. In Figure 4, sub-object *soil_1* inherits GIS slots and methods from the *polygon* class and inheritable domain slots and methods from the *soils* object. The *soils* object inherits GIS slots and methods from the *vector* class. These are not inherited by its sub-objects.

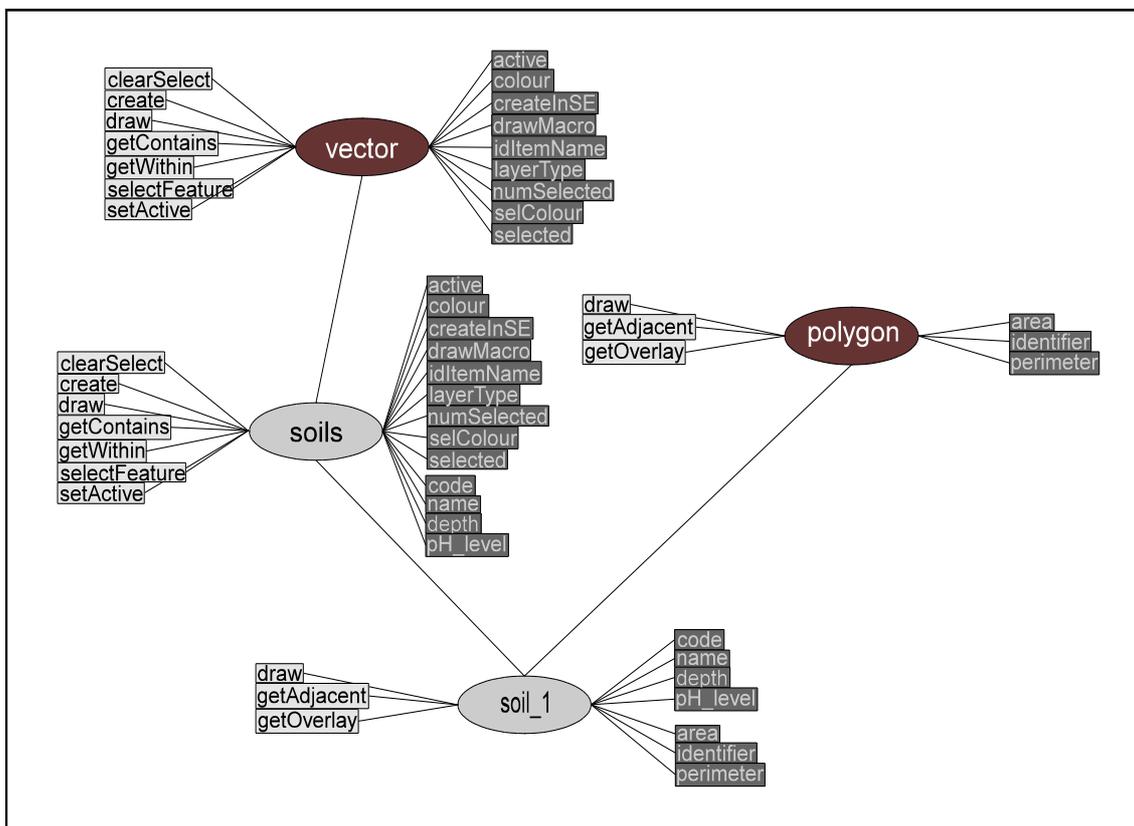


Figure 4 Soil example object diagram

The spatial methods attached to spatial classes access C procedures which perform error checking, creation and management of the information request which is sent to ARC/INFO, and manipulation of the reply. All spatial operations are executed in ARC/INFO. No locational data is manipulated in the expert system. Methods are used to access spatial relationships between objects. This enables the spatial context of an object

to be referenced in heuristic knowledge. For example, a SES rule base could dynamically and transparently determine the truth of the following conditions:

- if forest stand is adjacent to road
- if the nearest firestation = “Albany”
- if bus route is longer than 5km
- if parcel is within the Christchurch District boundary
- if paddock contains sandstone
- if site is at least 200 m from the nearest water source
- if habitat is above 1000m

Each of the conditions requires the GIS to perform either an adjacency, nearest neighbour, route distance, contained within, overlay, buffer and/or a raster map algebra operation. These operations are encapsulated in generic spatial methods which are inherited by the domain objects (e.g. *parcel*, *forest_stand*) from spatial classes. In the first example, the *forest_stand* object sends its inherited *getAdjacent* method to the *roads* object.

In addition to the GIS providing information about spatial data and relationships, the expert system can dynamically access the full presentational and analytical functionality of a GIS. This is possible through provision of a global *gisExecute* method in Smart Elements. This method takes a string argument. First the method determines and substitutes GIS names and locations for any spatial objects referred to in the string. The string, now a valid GIS command that runs a macro, is passed to the ARC/INFO process which then executes it. The expert system can request the GIS to display the results of an inference session on a map, or complex spatial analyses such as pattern analysis, multivariate analysis, location/allocation, hydrological or viewshed analysis can be requested. The results might then be interpreted by the expert system, upon completion of the request.

Both raster objects and vector objects can be manipulated in a rule base. The Smart Elements network diagram in Figure 5 demonstrates how sample heuristics defining suitability of land for forestry can utilise combinations of raster and vector data. Forest suitability depends on the land's aspect, elevation, soil type, landuse capability code, and distance from roads. The *lri*⁴ object refers to a soil polygon layer and the *track* object refers to a line layer. *Aspect*, *elevation*, *nw_high*, *nw_high_sunny*, *pukaki*, *luc*⁵, *6e_6c*, *luc_lessthaneq_5*, *mid*, *road_at_least_200* and *not_nw_high* objects all represent raster maps. The *road_at_least_200* object represents land that is at least 200m from a track. It is generated by executing its inherited *create* method with an argument string “> 200 track”. This string is stored in the *criteria* slot. Similarly the raster *pukaki* object is generated by executing the *create* method with an appropriate argument which selects, then rasterises all the polygons in the *lri* GIS layer with a soil type of Pukaki, i.e. “lri soil cn Pukaki”. The raster objects are weighted according to their relative importance. The raster objects and weights are passed to the GIS by the

⁴ lri = land resource inventory which is a national GIS layer of soil and vegetation data

⁵ luc = landuse capability code e.g. 6e

weight method which combines the GIS layers appropriately. Finally the *suit_forest* object is displayed by the GIS by sending an inherited *draw* method to itself.

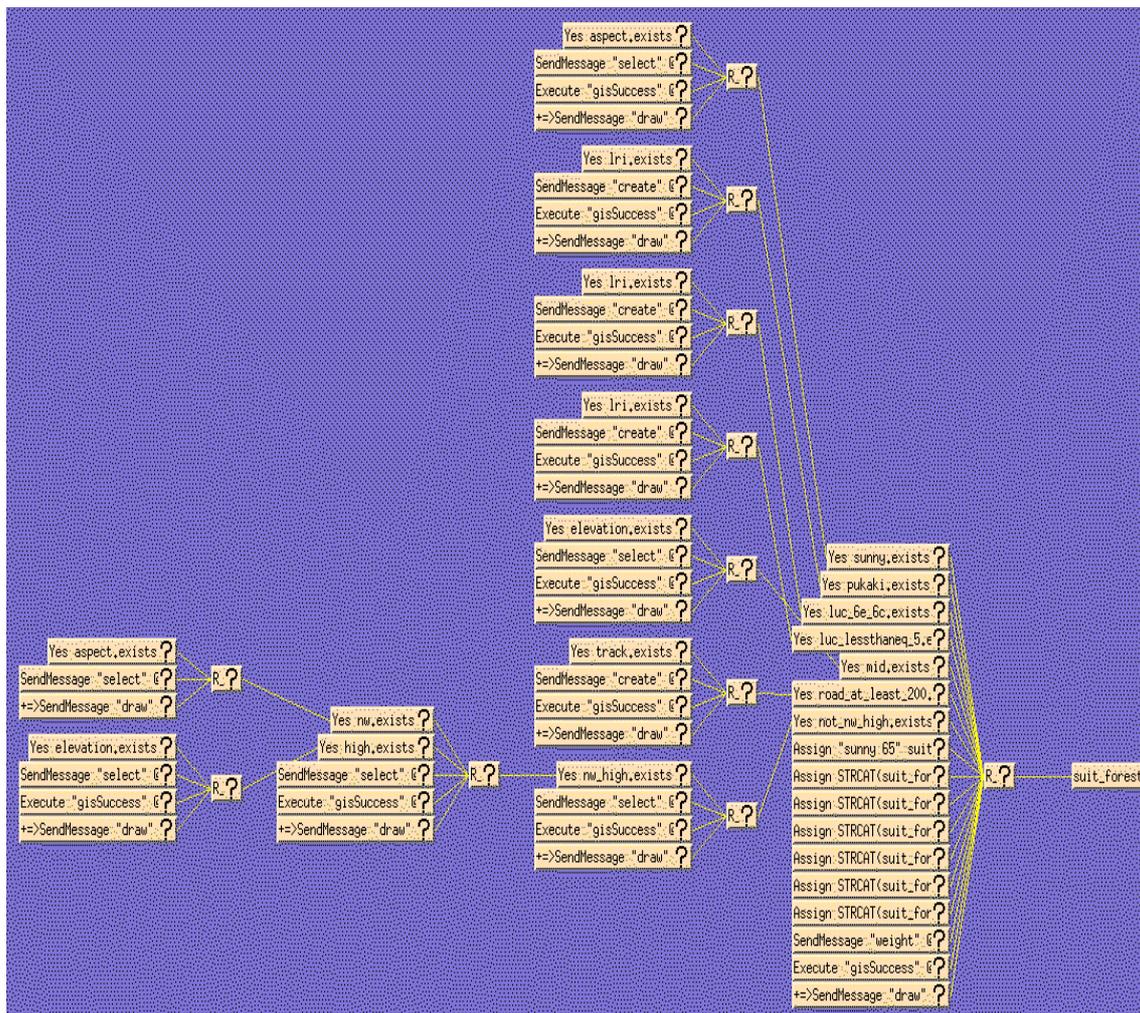


Figure 5 Network diagram modelling rule-based suitability of land for forestry

6. Discussion

The key advantage of an approach with high interoperability is flexibility. In SES, the GIS linkages required by a knowledge-based application are determined at runtime. They are not hardwired into the application. Spatial information is only accessed when it is required by the inference engine. The alternative approach is to calculate all the spatial relationships that might be required, and preload them as facts into the knowledge base, e.g. (Bleecker et al., 1990; Loh and Rykiel, 1992). This approach is only suitable for a limited range of spatial information, e.g. an adjacency matrix, stream connectivity.

The dynamic and generic nature of the linkage allows easy update of the domain knowledge base. The interwoven nature of the communication between the two processes allows the spatial and knowledge-based functions of the two systems to be effectively integrated. Figure 6 shows where SES is positioned in the integration cube.

There is very high interoperability between ARC/INFO and Smart Elements. Data is shared and there are two interfaces which can be designed to have a similar look and feel.

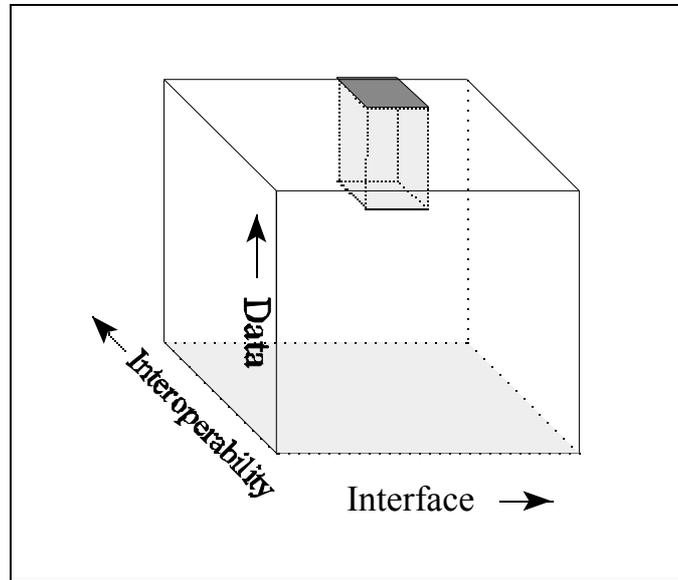


Figure 6 SES in the integration cube

This flexible linkage was achieved with minimal development effort as SES brings together the considerable power of the expert system shell Smart Elements and the GIS ARC/INFO. This was achieved by using a client/server approach.

The strengths of each system are maintained. SES maintains the computational power of a GIS by performing all geometric and raster map operations in the GIS. Raster maps are mapped to objects in Smart Elements which can be managed by the rule base. The expert system does not manipulate individual cell values. Knowledge is represented in Smart Elements using a symbolic representation (objects). This is more intuitive to work with than a quantitative representation (Sharma et al., 1994). Knowledge can be easily modified. A knowledge base developer can modify knowledge on the fly without needing to alter code or the linkage mechanism. Explanation capabilities provided in Smart Elements are available to the integrated system, as is the ability to handle incomplete data and opportunistic reasoning. GIS database management facilities can be used by the integrated system to access and manage data. The full set of GIS presentational functions are accessible in SES.

A limitation of SES is that while the intricacies of GIS operations are encapsulated in spatial classes, the use of spatial objects in rules is not completely transparent. For example, in a conditional statement:

If paddock = high

where *high* refers to the elevation, the syntax of the rule varies according to whether *high* is an aspatial attribute (i.e. “high”, “mod” or “low”), a raster map representation of cells with values over 1000m in an area of interest, or a fuzzy raster representation. The

knowledge base developer must be aware of these differences and structure the rules accordingly.

The design of SES is portable to other combinations of layer-based GISs and hybrid object/rule-based expert system shells that support client/server functionality. For example, Smart Elements could use the DDE protocol to make requests of an ARCVIEW (ESRI Inc., 1994) server by passing appropriate AVENUEJ commands. For SES to be portable between GISs, spatial services must be standardised and made accessible to client processes by vendors. Requests can then be made in a format that is independent of which GIS product is the server. There is a move in this direction by the Open GIS Consortium (1996) which is developing a specification for distributed geoprocessing.

SES is based on a client/server relationship in which Smart Elements is the client and ARC/INFO the server. A reverse architecture is possible in which a client ARC/INFO process requests a Smart Elements server to run an inference session. For example, a rule base of diagnostic heuristics could be accessed from ARC/INFO. The user can be prompted for some information but all other information required by the rule base must be passed to Smart Elements before the rule(s) is processed. This is because a Smart Elements – ARC/INFO linkage is synchronous. In a synchronous linkage, the requesting process must wait for the server process to complete the request before it can continue processing. If in the diagnostic example, a GIS data value or relationship is required, this can not be dynamically accessed. The GIS is too busy waiting for its original request to finish, to service any requests made to it. The need to pass all potentially relevant facts to a rule base inhibits the flexibility of the system and is not very efficient. Spatial facts, especially distance related facts can quickly become quite extensive. An asynchronous link, in which processes do not have to wait for each other, would allow concurrent bi-directional requests.

Knowledge accessed by a client needs to be modularised into discrete reasoning segments. This is usually possible with meta-data and classification knowledge. For example, an expert system server can inform a GIS client of the validity of a value, or an expert system can be requested to fire a rule base to classify a series of data values supplied by a GIS client. The expert system-as-server/GIS-as-client architecture is appropriate in a GIS controlled application where one or more modular rule bases are accessed to perform a specified classification, diagnosis, data validation, and/or recommendation. Meta-knowledge is not appropriate in this architecture.

However, an expert system application is often a system in which many strands of knowledge are interwoven together. Control knowledge is integrated with domain knowledge, meta-data, and an intelligent interface. For example, a DSS⁶ expert system application might combine an intelligent interface including appropriate question windows, recommendations, and explanations, with both knowledge about the process of determining a solution and knowledge about the domain objects. The expert system-as-client/GIS-as-server architecture (i.e. SES) is more suitable for a system in which

⁶ Decision support system

multiple types of knowledge are integrated. GIS tasks can be modularised and thus made accessible to the expert system client which controls the integrated system.

Other integration approaches are defined by Lilburne (1996). One approach taken by some authors is to develop an in-house GIS and/or expert system which is very demanding of resources, e.g. Davis (1991), Lam (1993). Loose and merged/embedded approaches where the systems are sequentially executed have low interoperability and are usually inflexible, but these approaches require minimal resources to implement a link. An enhanced approach where one of the systems is extended to incorporate functions normally performed by the other system results in a subset of the total functionality available to SES. For example, basic GIS display functions were incorporated into an expert system based on PROLOG (Crossland, 1990). A tight approach usually requires access to the source code of the systems being integrated.

7. Conclusion

SES is a powerful, flexible, generic toolbox which can be used to represent knowledge from any domain in a spatial context. Use of the object-oriented and client/server paradigms have enabled a highly interoperable linkage, accessing the full range of functionality, to be established between two powerful systems. Minimal resources were required to achieve this.

SES supports our belief that moving towards a truly interoperable GIS is essential in today's interlinked world of distributed systems. An interoperable GIS allows effective use of external techniques and systems. In particular, an interoperable GIS expands the potential of combining knowledge with GIS. So, returning to the question of the balance between developing functionality versus interfaces, there is a need for vendors to follow the third developmental track and further improve the interoperability of GIS.

8. Acknowledgments

This research was funded by the Foundation for Research, Science and Technology, New Zealand. Rhys Gibson's cheerful assistance with Smart Elements is much appreciated, as are the comments by the reviewers: Paul Luckman, Megan Ogle-Mannering and Grant Hunter.

9. References

- Open GIS Consortium, Inc. 1996. Open GIS. WWW ref: <http://www.opengis.org>.
- Bleecker, M., Hutson, J. L., and Waltman, S. W. 1990. Mapping groundwater contamination potential using integrated simulation modeling and GIS. *Proceedings of Application of geographic information systems, simulation models, and knowledge-based systems for landuse management*, Blacksburg, VA.
- Booch, G. 1994. Object-oriented analysis and design, The Benjamin/Cummings Publishing Company Inc., Redwood City, California.

- Burrough, P. A. 1986. Principles of geographical information systems for land resources assessment, Oxford University Press.
- Burrough, P. A. 1992. Development of intelligent geographical information system. *International journal of geographical information systems*, 6(1).
- Crossland, M. D. 1990. HyrdoLOGIC - a prototype GIS expert system for examining an AI application in a GIS environment. *Proceedings of GIS/LIS '90*.
- Davis, J. R., Cuddy, S. M., Laut, P., Goodspeed, M. J., and Whigham, P. A. 1991. Testing of soil moisture prediction model for army land managers. *Journal of irrigation and drainage engineering*, 117(4), p. 476-489.
- Durkin, J. 1994. Expert systems: design and development, Macmillan Publishing.
- Fedra, D. K. 1993. GIS and environmental modeling. *In Environmental modeling with GIS*, M. Goodchild, B. Parks, and L. Steyaert, eds., Oxford University Press.
- Fedra, K. 1995. Decision support for natural resources management: models, GIS and expert systems. *AI Applications*, 9(3), p. 3-19.
- Fischer, M. M. 1994. From conventional to knowledge-based geographic information systems. *Computers, environment and urban systems*, 18(4), p. 233-42.
- Fisher, P. F., Mackaness, W. A., Peacegood, G., and Wilkinson, G. G. 1988. Artificial intelligence and expert systems in geodata processing. *Progress in physical geography*, 12, p. 371-388.
- Goodchild, M., Haining, R., and Wise, S. 1992. Integrating GIS and spatial data analysis: problems and possibilities. *International journal of geographical information systems*, 6(5), p. 407-423.
- Jackson, P. 1990. Introduction to expert systems, Addison-Wesley Publishing Co.
- Lam, D. C. L., and Swayne, D. A. 1993. An expert system approach of integrating hydrological database, models and GIS: application of the RAISON System. *Proceedings of HydroGIS 93: Application of geographical information systems in hydrology and water resources*, Vienna, IAHS Publication 211, p. 23.
- Lein, J. K. 1992. Modeling environmental impact using an expert-geographic information system. *Proceedings of GIS/LIS '92*, San Jose, CA, USA, p. 436-44.
- Leung, Y. 1993. Towards the development of an intelligent spatial decision support system. *In Geographic information systems, spatial modelling and policy evaluation*, M. Fischer and P. Nijkamp, eds., Springer-Verlag.

- Lilburne, L. 1996. The integration challenge. *Proceedings of SIRC '96*, Dunedin.
- Loh, D. K., and Rykiel, E. J. 1992. Integrated resource management systems: coupling expert systems with database management and GIS. *Environmental management*, 16(2), p. 167-177.
- Nyerges, T. L. 1992. Coupling GIS and spatial analytical models. *Proceedings of 5th international symposium on spatial data handling*, Charleston, South Carolina, p. 534-543.
- Robinson, V. B., Frank, A. U., and Karimi, H. A. 1987. Expert systems for GIS in resource management. *AI Applications*, 1(1).
- Sharma, J., Flewelling, D. M., and Egenhofer, M. J. 1994. A qualitative spatial reasoner. *Proceedings of 6th international symposium on spatial data handling*, p. 665-681.
- Smith, T. R., and Yiang, J. 1991. Knowledge-based approaches in GIS. *In Geographical information systems: principles and applications*, D. J. Maguire, M. F. Goodchild, and D. W. Rhind, eds., Longman Group UK Ltd., Marlow, UK.
- Zhu, X., and Healey, R. 1992. Towards intelligent spatial decision support: integrating geographical information systems and expert systems. *Proceedings of GIS/LIS '92*.