University of Otago

Te Whare Wananga O Otago
Dunedin, New Zealand

Using Data Models to Estimate Required Effort in Creating a Spatial Information System

George L. Benwell
Stephen G. MacDonell

The Information Science Discussion Paper Series

Using Data Models to Estimate Required Effort
in Creating a Spatial Information System

Assoc. Prof. George L. Benwell[1]
Dr Stephen G. MacDonell
Computer and Information Science
University of Otago

January 1996

Abstract

The creation of spatial information systems can be viewed from many directions. One such view is to see the creation in terms of data collection, data modelling, codifying spatial processes, information management, analysis and presentation. The amount of effort to create such systems is frequently under-estimated; this is true for each aspect of the above view. The accuracy of the assessment of effort will vary for each aspect. This paper concentrates on the effort required to create the code for spatial processes and analysis. Recent experience has indicated that this is an area where considerable under-estimation is occurring. Function point analysis presented in this paper provides a reliable metric for spatial systems developers to assess required effort based on spatial data models.

---

[1] Address correspondence to: Assoc. Prof. G.L. Benwell, Department of Information Science, University of Otago, P.O. Box 56, Dunedin, New Zealand. Fax: +64 3 479 8311 Email: gbenwell@commerce.otago.ac.nz

1   Introduction

A taxonomy for the use of spatial information systems has been espoused (Calkins and Obermayer 1991). This work was related to, and developed from, earlier work by Calkins (Calkins 1989). Both these papers were addressing the issues of use, value and benefits for spatial information. The taxonomy developed by Calkins (*ibid*) is shown in Figure 1. This taxonomy is one of the plethora of views of the use of spatial information and can be seen to reflect the development process of spatial information systems (SIS). Other views of systems development have been frequently presented in the information science (or systems) literature. The reader is referred to (Carswell and Navathe 1987, Finkelstein 1989, Mahmood 1987) as typical examples.
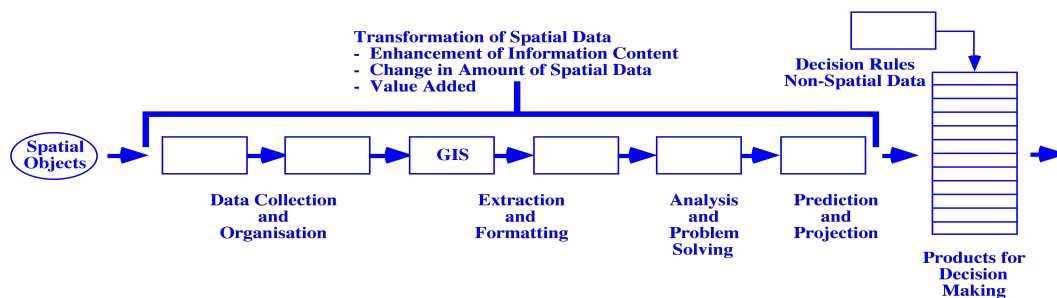
Figure 1: One view of spatial systems development (after (Calkins 1989))

A degree of isomorphism is apparent between Figure 1 and another view presented in Figure 2. The latter is more an information system's view and is considered appropriate to the discussion here. Figure 2 represents one view of the process of system development (Hawryszkiewycz 1988), moving left to right from specifying user requirements to eventually presenting results. The view is purposely data driven and therefore in line with contemporary understandings of system development.

Figure 2: A more generic perspective on systems development

This bias serves to match contemporary research and is a starting point to discuss the central theme of this paper. For a system to be of use, it obviously must match the user needs and accordingly, the data must be structured in a form within an appropriate database (see a summary by Mounsey (1991)). Added to this is the need for appropriate algorithms for data entry, maintenance and analyses. Recent experience (Glassey *et al.* 1994) has indicated that there may be a considerable degree of under-estimation of the effort required to develop this code. This observation is not seen to contradict research findings that a significant percentage (Mackaness 1989) of effort in developing a system could be attributed to data related collection and analysis. This finding is supported by the present authors; the work here is to subdivide this effort still further. The particular interest is the area of effort that may be best described as modelling and codification.

When designing information systems it is necessary to model data in such a way that it may unambiguously represent reality *and* be efficiently stored in a database (Firns 1990). This implies that a data view of reality is appropriate. Such a view is facilitated using well-established tools such as entity relationship models (ERMs) and data flow diagrams (DFDs) (Chen 1976, DeMarco 1978). In addition it has been suggested (Benwell *et al.* 1991) that other tools, such as Petri Nets (Benwell 1991, Purvis *et al.* 1994) may be useful. This is particularly so if it is deemed prudent to model dynamic and concurrent information processes. An example of a spatial process modelled by these three tools is presented in Figures 3, 4 and 5. The example describes two spatial processes, building a house and buying land, that can occur concurrently.
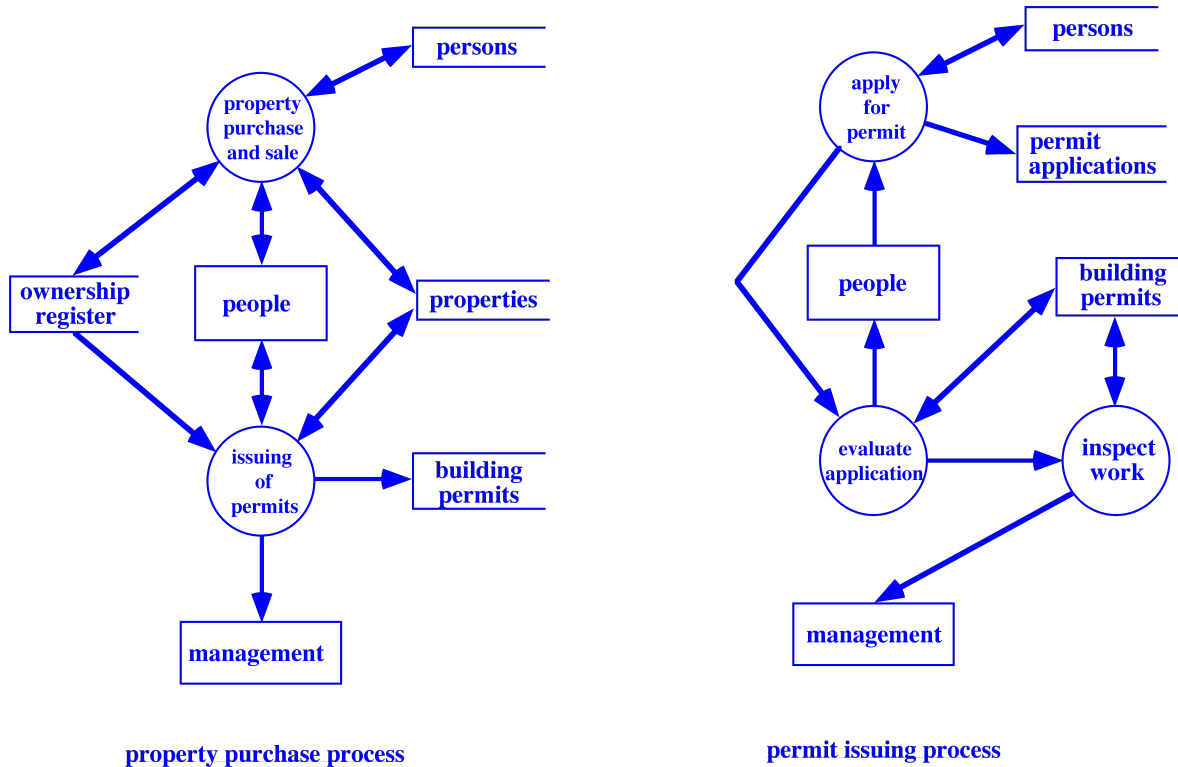
Figure 3: Data flow diagrams of spatial processes
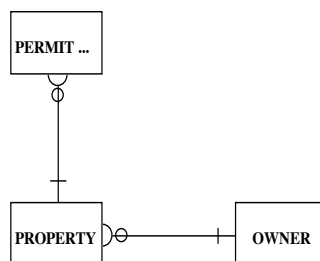
Figure 4: Partial entity relationship model of permit application information
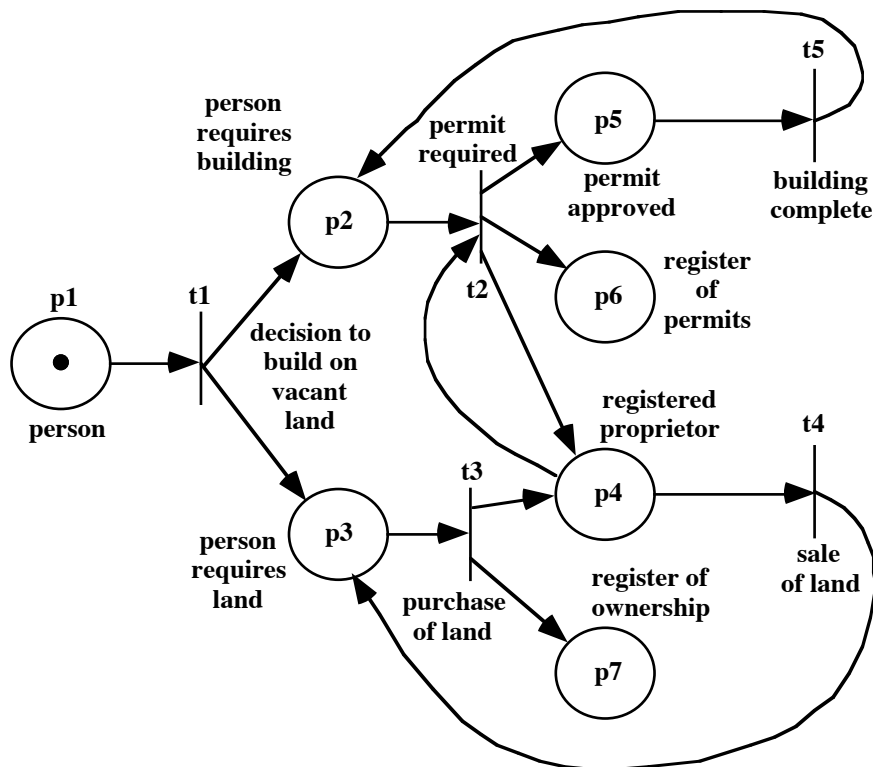
Figure 5: Petri Net of spatial processes relating to land sales and building approval

The DFDs in Figure 3 show the processes of house building (on the right) and land purchases (on the left). The ERM in Figure 4 shows a static data view of that information while the Petri Net (Figure 5) represents the two related processes and can be used for simulation, testing and analysis.

These tools are a means of abstracting from reality, the necessary and sufficient information about reality, so that an information system can be designed and implemented. Having accomplished that, it is then necessary to codify rules and algorithms as well as to encode database interactions. This requires an unknown effort but one that, it is contended, can be empirically estimated. This would be a useful operation as it indicates the size of the system and the effort to create it. Furthermore, it may be possible to translate this effort, represented by a factor, into time, dollars or human resource requirements. This concept is supported by the statement;

> *The information gathered during...* [Preliminary Design] *enables a preliminary design for the GIS to be developed. The design will be used for cost-benefit analysis of the proposed GIS, ...*(Clarke 1991 p479)

With reference to Figure 2 again, it is possible to measure the effort to carry out the codifying using techniques from the disciplines of software engineering and software metrics. It has been demonstrated (MacDonell 1993) that it is possible to measure aspects of ERMs and DFDs to derive indicators of effort. In the case of the design and implementation of a spatial information system it would be possible to;

i   determine the user specifications and define the scope of the system
ii   model reality in terms of an ERM and DFDs
iii   measure the ERM and DFDs
iv   use the above metrics to indicate system *size*
v   determine from the metrics the effort required for codification
vi   encode these models into a database and produce the 'lines of code'
vii   implement and use the system

The aim of this paper is to determine the amount of effort involved in encoding (phase vi above) based on data models (phase ii above). It is contended that this effort is considerably underestimated while at the same time it is a non-trivial component of a system's design and implementation. Glassey et al stated (Glassey *et al.* 1994);

> *... problems of implementing a GIS into an organisation ... cannot be emphasised strongly enough, ... it is NOT simple to set up a GIS!* (p108)
> *... Considerable application programming is likely to be required to enable the system to be completed and allow efficient access to the system for the end user* (p115).

While there may be only a degree of consensus with these statements they can be manifest in different forms. There is an amount of anecdotal or circumstantial evidence that supports this view. For example it may be heard;

"This GIS is too hard to use, it is too complex, the learning curve is too steep".. etc.

While this is a complex (and hypothetical) statement to analyse, in part it reflects the difficulty of collecting and structuring spatial data in a database and the encoding of rules to access it. It is therefore important to determine the size of such systems; size in terms of effort to create the code needed to support and analyse the data.

2   Related Work in GIS

It is important to confirm the connection between earlier research and that discussed here. Figure 6 shows this diagrammatically. Research work by Calkins and Dickinson and others with the National Center for Geographical Information and Analysis on value and worth is highlighted in Figure 6 with the hachured horizontal rectangle. This work, while being related to the present paper, must be perceived as essentially different. Worth and value are considered to be factors that relate to a system as a whole. They may be the result of summation of components, but as Figure 6 shows, worth and value are fundamentally derived from a comparison or ratio of costs and benefits. The present research is more concerned with determining a metric which will predict the effort (and eventually dollars) involved with the writing and production of software associated with a spatial information system. It is proposed that this metric is obtainable from an ERM and DFDs which are in turn used for the design and implementation of systems.

The economic evaluation of the implementation of a SIS (Dickinson and Calkins 1988) will involve the assessment of costs and benefits. In both instances there will have to be some forward projections or assumptions. Indeed, Dickinson and Calkins (*op. cit.* p310) state that "the detailed information needed to support the traditional benefit cost analysis is not always available." Further on they report that cost categories such as,

"database entry/transfer

database maintenance (edits, updates, backups)

in-house programming for software enhancements ...

in-house support for system users

actual running of applications on the system"

are costs "that will tend to be reflected in actual staff salaries....". Software engineering techniques such as software metrics outlined here in sections 3 and 4 are capable of predicting system complexity, system size and hence the effort required to create them. It is therefore possible to predict, with some confidence, the labour costs, having carried out an adequate system design. The significance of this estimation in the context of overall system cost can be gauged from an analysis for such costs (Smith and Tomlinson 1992).

```
┌──────────────┐
│   System     │
│ Feasibility  │
└──────────────┘
        │
        ▼
┌──────────────┐        ┌──────────┐
│   Use  in    │───────▶│ Benefits │        ┌──────────┐
│  Decision    │        └──────────┘───────▶│ Measured │
│   Making     │        ┌──────────┐        │  Values  │
└──────────────┘───────▶│  Costs   │───────▶└──────────┘
        │               └──────────┘              │
        ▼                                          ▼
┌──────────────┐                          ┌──────────┐
│ Requirements │◀────────────────────────▶│ Economic │
│Specification │                          │Evaluation│
└──────────────┘                          └──────────┘
        │
        ▼
┌──────────────┐      ┌──────────┐      ┌────────────────┐
│  Conceptual  │─────▶│ Detailed │─────▶│ Implementation │
│   Systems    │      │  System  │      └────────────────┘
│   Design     │      │  Design  │
└──────────────┘      └──────────┘
```

▨ **Behavorial analysis for system design**

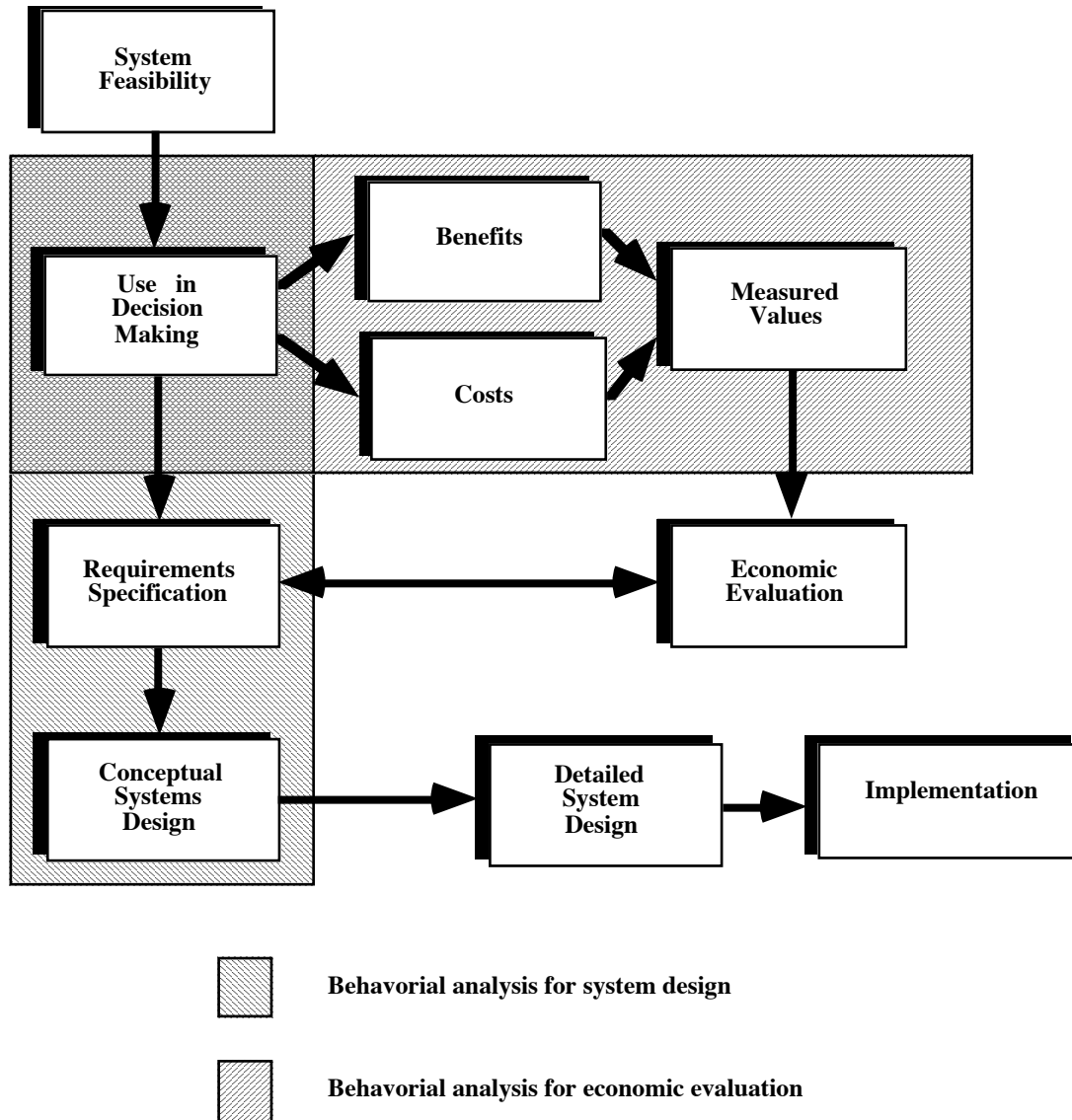▨ **Behavorial analysis for economic evaluation**

Figure 6: A framework for assessing system worth and value
(after Benwell and Dickinson (1991))

Smith and Tomlinson examined costs of systems and attribute 20.7% to *system implementation and maintenance*. While this figure is not exceptionally high, another category, staff training amounts to 26.7%. It seems reasonable to attribute some of this amount to software writing and the encoding of rules. Under such an assumption it is not unreasonable to expect that upwards of 25-30% of a system's costs are associated with software development. If this amount can be predicted or assessed with a confidence limit it follows that cost-benefit analyses could be improved. Such would be the case in the instance cited by Smith and Tomlinson (*op. cit.* p253), "As the system is altered to perform more functions, benefits increase as more products are produced, but costs are also incurred out of these activities." It may well be possible to predict these additional costs. If costs are incurred outside the software domain (such as extra consumables or hardware purchases) these, naturally, will not be determined by the metric.

The importance of data modelling is recognised by the SIS community (Rhind *et al.* 1991). While this recognition may not be wide spread at present it is one of a number of prominent agenda items being examined (*op. cit.* p315). The same authors (*op. cit.* p320) note the need for improved functional requirements as they relate to costs, "The results of research currently under way will emerge in improved products. Of particular significance will be ... improved techniques for conducting functional requirements studies, evaluating costs and benefits, ...". It is held that while, no direct reference is implied between that statement and the research presented in this paper, function point analysis will be of benefit in the determination of costs. This being the case more practitioners and systems analysts will be in a position to determine the scale and scope of systems via these techniques. This cause will be advanced further with developments in spatial data modelling, particularly when the SEER model developed by Firns (Firns 1994) finds general acceptance.

3 Related Work in Software Engineering

Software engineering is concerned with the timely and cost-effective production of computer-based information systems to an acknowledged level of quality. To this end, those responsible for the management of software development have been most interested in understanding, modelling, monitoring, controlling and improving many aspects of the software process, including systems development schedules, development effort projections, and product and process quality.

Clearly these issues are not important solely in the domain of spatial information systems. They are of *equal* importance to the development of spatial systems as for any other system type. This is becoming increasingly so as the costs of data collection, for so long the dominant cost driver in spatial systems development, are reduced in relative terms when compared to the costs of other development tasks and activities.

3.1 Development time schedules

In order to efficiently allocate resources (personnel, computing time, money) to a project, managers need to be aware of the schedule that the development process is likely to follow, as well as the pressures that are often exerted on the schedule. This is especially important in the later stages of software development, when extra personnel may be drafted in to work on a project in the hope that this might ensure delivery on time. In fact it is now widely recognised that this particular strategy tends to actually delay delivery of a system, as co-ordination

difficulties override the benefits attained with extra personnel (Brooks 1975, Abdel-Hamid and Madnick 1989).

Among the most widely cited methods for project scheduling is the work of Putnam (1978). Based on extensive analyses of large software projects, Putnam developed the 'software equation' in order to illustrate the empirical relationship between software product size and the resultant development schedule:

$$\text{Size} = CK^{1/3}t_d^{4/3}$$

      where   Size is the number of delivered source lines of code
                       C is a technology factor
                       K is total project effort (in person-years)
                       t_d is development time (in years)

Solving the software equation for K, the development time, results in a fourth power trade-off with development effort; that is, development effort is said to be inversely proportional to the fourth power of development time. Under this model, then, a decrease in delivery time of just 5% (perhaps as a result of customer demands) leads to a 23% increase in development effort.

4   Software Metrics for Effort Estimation

Accurate estimation of development effort has been a long-time goal of those concerned with software project management. Research and practice have progressed from lines-of-code based estimates, through measures derived from design representations, to present-day methods, through which effort can be estimated (to within specified bounds of accuracy) from functional models. It is this final class of estimation techniques that is the focus of this paper. There are several reasons for this concentration on function-based methods:

(i) functional models, including data structure models, data flow models and the like, are among the first software products available, thus predictions developed from them can be derived at an early stage of the process;

(ii) estimates obtained from functional models are relatively independent of the specific implementation language and technology, but may be calibrated as necessary;

(iii) if the models use common specification notations, measures are easy to extract automatically and objectively from CASE (computer-aided software engineering) environments;

(iv) as functional requirements are added or modified, refined estimates can be produced as needed.

Software development effort is said to be a function of a system's input model information content. Given that the size of a functional specification (the system input model) should approach invariance with respect to decisions of individual modellers, estimates developed from such specifications should be consistent and objective (DeMarco 1982).

One approach to function-based estimation is now considered. It should be noted that this case study does not entail a full evaluation of the method itself. Rather, the purpose of this discussion and case study is to increase the awareness of project managers in the spatial

systems domain as to the potential of such a method, with the hope that similar techniques might then be considered in the management of their projects. An associated aim of this work is the promotion of the use of functional models in the analysis and design of spatial information systems. Effective use of these models, particularly in an automated (CASE) environment, can result in significant improvements in the management and control of development projects.

4.1 Mark II Function point analysis

Function point analysis (FPA), first developed in 1979, is a widely used function-based productivity assessment and effort estimation approach (Albrecht 1979). Since its introduction the approach has evolved to the point where, although not without its faults, it is regarded as a *de facto* industry standard. Given that the basis of effort estimation in this research is the set of data-centred functional models that make up a system specification, the Mark II version of FPA (Symons 1988, 1991), with its more contemporary view of systems, has been adopted here.

The number of function points (a unitless measure of functionality or value) in a system is the product of two components: one, the information processing size of the system, as calculated from the decomposition of logical transactions into weighted inputs, processes and outputs; and two, an adjustment for the technical complexity of the software and the operating environment.

Calculation of the information processing size of a system in unadjusted function points (UFP) is performed with the following equation:

$$\text{Size in UFP} = (W_I * N_I) + (W_E * N_E) + (W_O * N_O)$$

where N_I is the number of input data elements
          N_E is the number of entities referenced
          N_O is the number of output data elements
          W_n is the empirically calibrated weighting for component *n*
          (Industry standard values: $W_I = 0.58$, $W_E = 1.66$, $W_O = 0.26$)

This equation is said to account for the size of a system required to cope with formatting and validating input and output data items, and with accesses to and from a database.

The technical complexity adjustment (TCA) factor is computed as the sum of the values of 20 characteristic measures, which assess the contributions of data communications, transaction rate, operational ease and several other factors to the overall complexity of the system. Each factor F_i is assigned a value between 0 and 5, illustrating its degree of influence on development. A value of 0 indicates no influence, a value of 5 indicates strong influence throughout. These values are summed and then scaled according to the following calculation (Symons 1991):

$$\text{TCA} = 0.65 + (0.005 * SF_i) \quad i = 1... 20 \qquad 0 \le F_i \le 5$$

The final equation in determining overall functionality is therefore:

$$MkIIFP = UFP * TCA$$

Symons (1991) recommends that, in order to obtain the most useful estimates, specification and effort data should be collected and analysed in each software development environment, so as to ensure that the weightings derived are indeed appropriate for that environment. Similarly, this would also form the basis for the calculation of average productivity rates (in MkIIFP per effort unit) for that environment, essential for further effort estimation. For example, if a development group produces, on average, x function points per work-hour, and they are to build a new system whose size is y MkIIFP, then development effort for that system will be estimated at xy work-hours.

5   Case Study

A hazards information system is being developed (Aldridge *et al.* 1993) for what, it is believed, will be a pilot for a national implementation. In 1992 the Institute of Geological and Nuclear Sciences (IGNS) obtained funding from the Foundation for Research, Science and Technology for a pilot hazards register using a spatial information system. At the invitation of the Institute, the Dunedin City Council, the Otago Regional Council, and Simes Dunckley Valuation (representing the interests of the Institute of Valuers, New Zealand) agreed to participate in the pilot study. The Spatial Information Research Centre (SIRC) at the University of Otago contributed specialist advice in the areas of systems design and development and, in particular, data structure design. The initial SIRC work was for the compilation of a conceptual design and a data model for the hazard system, and included the preparation of the system proposal.

The hazard register is a computer based information system for the recording, maintenance and reporting of an up-to-date consolidated record of existing and potential natural and/or physical hazards. Part of Dunedin City is being used as the pilot study area. The system is composed of two sub-systems.

The *hazards register* sub-system will support local authority routine operations: the processing of applications for building permits, resource consents, etc; and the processing of information requests. These need definite information about a particular property. The hazards register will serve as a single repository for a council's knowledge of the existence of hazards on particular properties. Since it will probably be uneconomic, and in some cases, impossible, to go back through all council records when compiling such a repository, some limit will need to be placed on the period covered by the sub-system. Council clients who require that a search for hazards information extends beyond the stated scope of the sub-system will need to be served by a manual search, presumably at much greater cost.

The second sub-system, a *hazards analysis and modelling component* will be concerned with spatial data which has a resolution too coarse to be used in any valid manner for individual properties. It will contain data on the physical determinants of, principally, natural hazards. These data will be used to infer the existence and extent of hazards. The sub-system will use relevant parts of the national digital cadastral database as a base map and will also draw on digital terrain model data for a topographic base. The physical inventory of hazard

determinants which will be held in the sub-system closely follows the guidelines for urban land use capability surveys. The types of hazards which are assessed during urban land capability surveys are very much the same as those which are of interest for the proposed hazard system. These require the accumulation, at a suitable map scale, of an inventory of physical factors including (but not limited to): rock type, soil type, landform type, slope class, erosion type and intensity, land drainage quality, and land use.

These and other hazard-determining factors are then used to model (analyse) the hazards which result in constraints on land use. The analysis of contributing physical factors is only used to assist with making the final assessments (Jessen 1987). It is proposed that the hazards analysis and modelling system be used in a similar way by developing models which can predict, or help predict, the types of land hazards.

For simplicity, only the hazard register subsystem is used in the case study. Data diagrams for the entire system have been prepared but space dictates that only the register should be used here. Figures 7 and 8 represent the functional hierarchy and the ERM for the hazards system which along with the data flow diagrams, will be subjected to Mark II function point analysis to determine (predict or indicate) development effort.
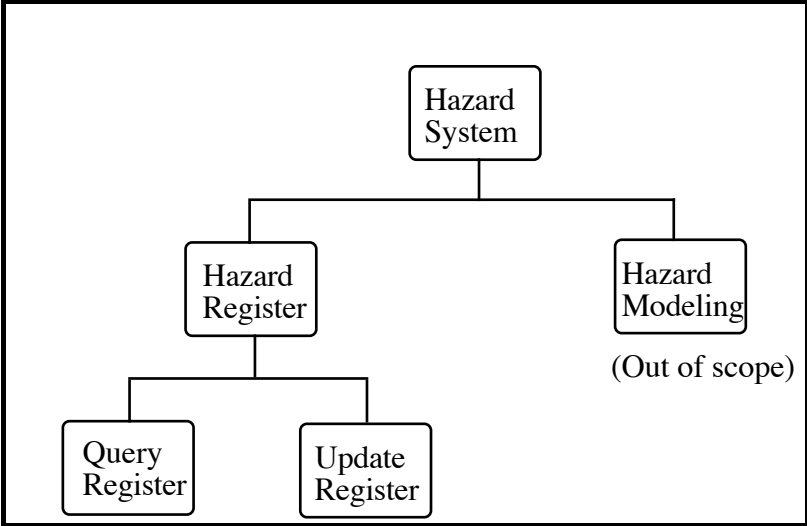

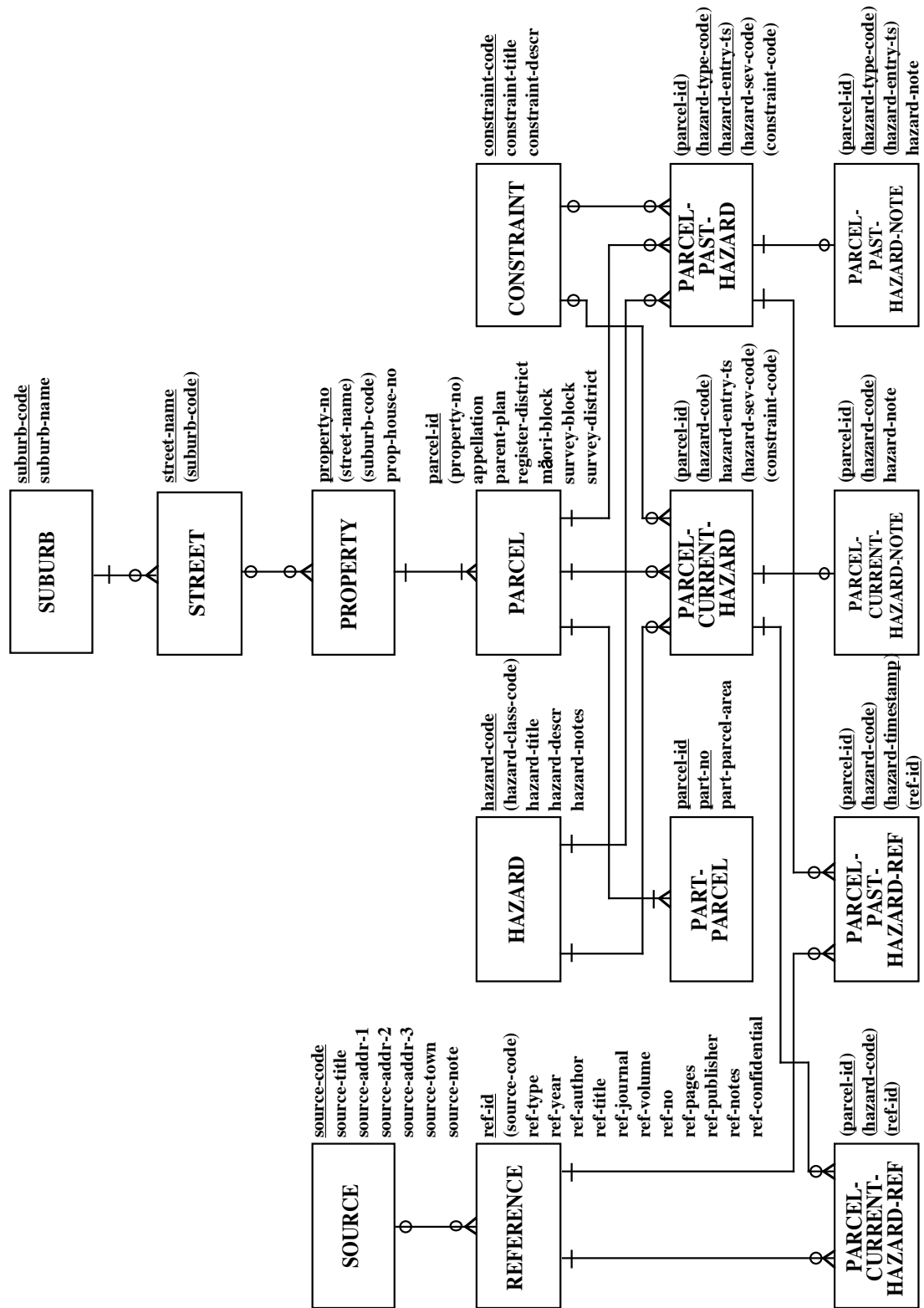
Figure 7: Functional model of the pilot Hazard System

Figure 8: ERM for the pilot Hazard System

```
POST-AMALGAMATION PROPERTY HAZARDS MEMORANDUM

APPLICANT                          DESCRIPTION OF PROPERTY

Mr J.W. Smith                      Location              205 Howard St
Smith and Jones Valuation                                Macandrew Bat
PO Box 18-517                      Property No           123678
Dunedin                            Valuation No          34596-345
                                   Legal Description

                                       LOT 15
                                       UPPER HARBOUR EAST SD

                                   Area in hectares          0.0945
                                   Differential Classification   Residential

HAZARD SUMMARY

A search of the Council's Hazard Register confirms the following at the above property:

        Hazard Type     Land instability: land slip
        Hazard Rank     Definite problem
        Constraint      A building permit will not be issued without title caveat
        References      Soil Bureau Report 28, Otago Peninsula
                        Stone, R., 1990, Land stability at Macandrew Bay, Otago Peninsula.
                            Institute of Geological and Nuclear Sciences, File I44/971.



        Hazard Type     Underground electricity cable
        Hazard Rank     Definite problem
        Constraint      No building within 2 metres of cable
        References      Consult Dunedin Electricity for location details
```

Figure 9: A typical report resulting from a query on the Hazards Register

The information processing size of the Hazard Register system is determined from the system's logical transactions. Based on the functional hierarchy shown in Figure 7 and the system DFDs, two main transactions comprise the Register system. For each, the number of input and output elements and the number of entities accessed must be specified and counted, as in the following example:

Transaction: *Query HR System*

Inputs   -   Applicant details - 4 elements
            Property details - 1 element
            Menu choices - 3 elements $I = 8$

Entities   -

Property (S)	1	
Street (S)	1	
Suburb (S)	1	
Parcel	2	
Parcel-Current-Hazard	3	
Hazard	4	
Constraint (S)	4	
Parcel-Current-Hazard-Note	5	
Hazard-Class-Table (S)	5	
Parcel-Current-Hazard-Ref	6	
Hazard-Type (S)	6	
Reference	7	
Source	8	$E = 8$

Outputs   -   Latest update - 1 element
            Hazards memo - 15 elements
            Reference information - 16 elements
            Error message (Property) - 1 element
            Error message (Menu) - 1 element
            Map - 3 elements $O = 37$

(In the specification of entity references, accesses to a 'System Entity', that is, a look-up table used mainly for validation, are counted just once for the whole transaction. Thus, accesses to entities denoted '(S)' do not result in the incrementing of the entity reference count value except in the first instance.)

A similar decomposition of the Update HR System transaction produces the following component values: $I = 63$, $E = 7$ and $O = 3$. Thus the total values for each component, to be used in the calculation of the size of the Register system, are: $I = 71$, $E = 15$, $O = 40$. This leads to the specification of the following partially completed equation:

$$\text{Size in UFP(Register System)} = (W_I * 71) + (W_E * 15) + (W_O * 40)$$

The weightings associated with each component are normally calibrated to a specific environment, based on data collected in that environment, to reflect the relative impact of each component on the size of the system. This study, however, had no such historical data available to enable the calibration. Industry-standard weightings have been supplied as a starting point ($W_I = 0.58$, $W_E = 1.66$, $W_O = 0.26$) (Symons 1991), but these were generated

from around ninety business systems, so they may be inappropriate for the spatial systems domain. Relevant figures obtained from spatial systems development would clearly be more useful.

In an attempt to collect data from which useful weightings could be determined, two separate approaches were made several months apart to the approximately 650 members of GIS-L, the international listserver for those interested or involved in the use or development of geographical information systems. The request asked for the provision of system specification documents, along with associated development effort records, for spatial systems developed in recent times. Unfortunately, but not unexpectedly, no responses were received. As in the business systems domain, there has until recently been a reluctance to invest in the data collection and analysis tools and procedures necessary for metric analysis, as proof of the effectiveness of the models is demanded first. This reluctance, however, introduces a cyclic pattern of avoidance, as illustrated in Figure 10. Developers and managers will not use metric models because they have no evidence of their effectiveness. If the models are not used, the data cannot be collected. If the data is not collected, the effectiveness (or otherwise) of the models cannot be determined.
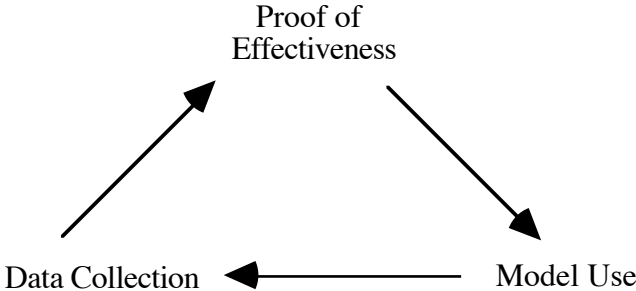

Figure 10: Cyclic obstacles to metric use

Another attempt was therefore made to obtain *intuitive* evidence for weighting values, by asking members of the list to answer the following request:

Dear Netters,

Following our totally null response from the lists, both NZGIS-L and GIS-L (except Ray Woods (in NZ) with thanks) about measuring effort we have another related request.

THIS WILL TAKE 4 MINUTES TO READ AND COMPLETE.
PLEASE SEND A RESPONSE

Use a scale of 1 to 10 (1 being extremely easy and 10 being extremely hard) to *relatively* measure the following 3 tasks for writing *code* in a GIS

1 Handling input enquiries - formatting (eg screens for enquiries, graphical or textual)
2 Accessing the data for processing (eg turning requests into a data seek and solve)
3 Controlling output - formatting (producing text or graphical output)

On what software experience is this based?

For example, if you consider that developing the code to perform task 2, accessing the data and performing processing, is very hard, you might assign it a value of 8. If developing code for presenting output, task 3, is slightly easier than this you might assign it a value of 6. And for task 1 because it's really easy, you assign the value of 1.

We are aware that your answers will be affected by software, hardware, user experience etc. The literature on *relative* measures DOES recognise these too.

Think about it and please give us your score. This has been trialed on NZGIS-L (25% response) before going to GIS-L.

Thanks for your time.

This request produced a total of twenty-seven responses. Table 1 includes summary data derived from these responses.

	INPUT	ACCESS	OUTPUT
Sum	103	107	108
Minimum	1	1	1
Maximum	9	9	10
Median	3	3	4
Average	3.81	3.96	4.00
Std Deviation	2.24	2.08	2.34
Weighting (Spatial)	0.81	0.84	0.85
Weighting (Business)	0.58	1.66	0.26

Table 1: Summary data for metric component weightings

The original (business) weightings were averaged and then relatively proportioned so as to add to 2.5. This was to enable comparisons to be made between size assessments performed with both the original and Mark II definitions of function points. A similar process was used here to determine the relative weightings of the three contributors for spatial systems. From the last two lines of Table 1 it would seem that both input and output handling for spatial systems are, in general, more difficult than for commercial systems, but in contrast data access is considered to be relatively more simple in a spatial system.

It is acknowledged that these weightings have been derived in an anecdotal manner, as opposed to their being determined empirically. However, it is considered here that weightings determined in this way from a sample of 27 respondents in the spatial domain are to be preferred over the standard business-oriented weightings otherwise available. Adopting the spatial weightings, the following equation for system size can now be computed:

$$\text{Size in UFP(Register System)} = (0.81 * 71) + (0.84 * 15) + (0.85 * 40)$$
$$= 57 + 13 + 34$$
$$= 104$$

The contributors to the Technical Complexity Adjustment were assigned degrees of influence, as presented in Table 2.

Factor F_i	Influence value	Factor F_i	Influence value
Data communications	2	Installation ease	3
Distributed function	0	Operational ease	3
Performance	4	Multiple sites	1
Heavily used setup	2	Facilitate change	2
Transaction rate	2	Interface to systems	1
Online data entry	5	Security	2
User efficiency	3	Third Party use	3
Online update	3	Documentation	2
Complex processing	4	User training	0
Reusable code	1	Special hardware	2

Table 2: Degrees of influence for technical complexity adjustment

The sum of the degrees of influence for the twenty factors F_i is 45. This can be directly included in the TCA calculation:

$$\text{TCA} = 0.65 + (0.005 * SF_i) \quad i = 1...20 \qquad 0 \le F_i \le 5$$
$$= 0.65 + (0.005 * 45)$$
$$= 0.875$$

The final equation in determining overall functionality is therefore:

$$\text{MkIIFP(Hazard Register System)} = \text{UFP} * \text{TCA}$$
$$= 104 * 0.875$$
$$= 91$$

In isolation, this figure is of little direct use. It is unitless and provides no information about the system being measured. Its value becomes apparent, however, when data is routinely collected and analysed, and the results are fed back into the approach to improve the model through continued recalibration of the component weightings. In order to illustrate the utility of the approach, however, an industry-standard productivity measure is used here as the basis for an effort 'prediction' for the Hazard Register system. According to Symons (1991), systems development in a third-generation environment leads to an average productivity rate of 0.1MkIIFP/work-hour. If that figure is adopted here, this would result in a prediction of $91/0.1 = 910$ work-hours of effort. To carry the illustration further, actual effort records estimated for the Hazard Register System development indicate that approximately 1047 work-hours were used. This estimation therefore represents an error (under-estimation) of $(1047-910)/910 = 15\%$. Although a lesser degree of error would be desirable, this is a useful first approximation of actual effort requirements. Moreover, as estimates are derived for other projects and these are tracked against actuals, an adjustment factor may be identified. For example, it may be beneficial to always increase the final MkII-based estimate by a 10% contingency factor, given developer propensity to underestimate the difficulty of development tasks. This could then lead to effort prediction with a general confidence interval of +/- 5%.

6   Conclusion

Significant effort in the development of a spatial information system is consumed by the encoding of access rules and process control for the database. It has been determined that measuring the size and structure of data-centred specification models will provide a primary indicator of such effort. This can therefore be used to estimate the amount of effort (time/money/resources) that will be required to develop a database-oriented system. Given historical records, effort prediction can be performed well in advance of actual development.

This concept was demonstrated using a prototype development of a hazard register system. There was found to be an adequate level of agreement between the predicted and actual value.

There remains potential to improve the metrics as particular knowledge relating to the spatial information systems domain is advanced. For example there may be a need to consider a distinction between textual and graphical output. This is one of the areas that requires considerable research.

Mark II function point analysis is just one approach of several that enable effort estimates to be generated, and it does provide a reasonably quantitative and objective basis upon which projects in the spatial domain can be more effectively managed.

Acknowledgments

References

ABDEL-HAMID, T.K. and MADNICK, S.E., 1989 Lessons learned from modeling the dynamics of software development, *Communications of the ACM*, **32**, 1426-1438.

ALBRECHT, A.J., 1979 Measuring application development productivity, In *Proc. Joint SHARE/GUIDE IBM Application Development Symposium*, pp. 83-92.

ALDRIDGE, C., BENWELL, G., TURNBULL, I., HENDERSON, J., HARRIS, M. and TAY, A., 1993 Dunedin pilot hazards information system - A system analysis and proposal, In *Fifth Annual Colloquium of the Spatial Information Research Centre*, University of Otago, Dunedin, New Zealand, pp. 247-264.

BENWELL, G.L., 1991 Casting Petri Nets into the system development life cycle in the context of spatial information system, Unpublished PhD thesis, The University of Melbourne, Melbourne, Australia.

BENWELL, G.L. and DICKINSON, H., 1991 Software requirements for Petri Nets in the behavioural analysis of spatial information, *New Zealand Surveyor*, XXXIII, **278**, 64-79.

BENWELL, G.L., FIRNS, P.G. and SALLIS, P.J., 1991 Deriving semantic data models from structured process descriptions of reality, *Journal of Information Technology*, **6,** 15-25.

BROOKS, F.P., Jr, 1975 *The mythical man-month,* Addison-Wesley, Reading MA.

CALKINS, H.W., 1989   Position Paper in Initiative #4, Special meeting report and proceedings, Technical Report 89-7, *National Center for Geographic Information and Analysis,* University of California, Santa Barbara, California.

CALKINS, H.W. and OBERMAYER, N.J., 1991 Taxonomy for surveying the use and value of geographical information, *International Journal of Geographical Information Systems*, **5**, 341-351.

CARSWELL, J.L. and NAVATHE, S.B., 1987 SA-ER: A methodology that links structured analysis and entity-relationship modeling for database design, *Entity-Relationship Approach*, Elsevier Science Pub., North-Holland, pp. 381-397.

CHEN, P.P., 1976 The entity relationship model: towards a unified view of data, *ACM Transactions on Database Systems*, **1**, 9-36,

CLARKE, A.L., 1991 GIS specification, evaluation, and implementation In Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (Eds), *Geographical information systems: principles and applications*, Longman, London, pp. 477-485.

DEMARCO, T., 1978   *Structured analysis and system specification,* Prentice-Hall, Englewood Cliffs NJ.

DEMARCO, T., 1982 *Controlling software projects*, Yourdon, New York.

DICKINSON, H.J. and CALKINS, H.W., 1988 The economic evaluation of implementing a GIS, *International Journal of Geographical Information Systems*, **2**, 307-327.

FINKELSTEIN, C., 1989   *An introduction to information engineering - from strategic planning to information systems*, Addison-Wesley, Sydney, 393 pages.

FIRNS, P.G., 1990 Entity relationship modelling in GIS design: an efficacious approach or an exercise in futility, In *Proc. 18th Australasian Conference in Urban and Regional Planning Information Systems*, Australia.

FIRNS, P.G., 1994 A spatially extended entity relationship model, Unpublished PhD thesis, University of Otago, Dunedin, New Zealand.

GLASSEY, P., FORSYTH, P., TURNBULL, I., ALDRIDGE, C., CLEMENTS, R. and BENWELL, G. 1994 Dunedin pilot hazards information system - trial by GIS, In *Sixth Annual Colloquium of the Spatial Information Research Centre*, University of Otago, Dunedin, New Zealand, pp. 105-116.

HAWRYSZKIEWYCZ, I.T., 1988 *Introduction to systems analysis and design*, Prentice Hall, New York, 373 pages.

JESSEN, M.R., 1987   Urban land use capability survey handbook, *Water and Soil Miscellaneous Publication*, Water and Soil Directorate, Ministry of Works and Development, Wellington, No. 105, 74 pages.

MACDONELL, S.G., 1993   Quantitative functional complexity analysis of commercial software systems, Unpublished PhD thesis, Cambridge University, Cambridge, England.

MACKANESS, W., 1989 Introducing GIS into organisations, In *Inaugural Colloquium of the Spatial Information Research Centre*, University of Otago, Dunedin New Zealand, November, pp. 106-119.

MAHMOOD, M.A., 1987   System development methods - A comparative investigation, *Management Information Systems Quarterly*, September, 293-311.

MOUNSEY, H.M, 1991 Multisource, multinational environmental GIS: lessons learnt from Corine, In Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (Eds), *Geographical information systems: principles and applications*, Longman, London, pp. 185-200.

PURVIS, M.K., BENWELL, G.L. and PURVIS, M.A., 1994 Dynamic modelling of the resource consent process in the Resource Management Act, will be published in the *New Zealand Surveyor*, 20 pages.

PUTNAM, L.H., 1978   A general empirical solution to the macro software sizing and estimating problem, *IEEE Transactions on Software Engineering*, **4,** 345-361.

RHIND, D.W., GOODCHILD, M.F. and MAGUIRE D.J., 1991 Epilogue, In Maguire, D.J., Goodchild, M.F. and Rhind, D.W. (Eds), *Geographical information systems: principles and applications*, Longman, London, pp. 313-327.

SMITH, D.A. and TOMLINSON, R.L., 1992 Assessing costs and benefits of geographical information systems: methodological and implementation issues *International Journal of Geographical Information Systems*, **6**, 247-256.

SYMONS, C.R., 1988   Function point analysis: difficulties and improvements, *IEEE Transactions on Software Engineering*, **14,** 2-10.

SYMONS, C.R., 1991 *Software sizing and estimating: Mk II FPA (function point analysis)*, Wiley, Chichester.