Full citation: MacDonell, S.G., & Benwell, G.L. (1996) Effort estimation for the development of spatial information systems, in Proceedings of the Eighth Annual Colloquium of the Spatial Information Research Centre (SIRC'96). Dunedin, New Zealand, University of Otago, pp.149-155.

Effort Estimation for the Development of Spatial Information Systems

Stephen G. MacDonell and George L. Benwell

Computer and Information Science
University of Otago, Dunedin, New Zealand
stevemac@commerce.otago.ac.nz, gbenwell@commerce.otago.ac.nz

Abstract

The management and control of software processes has assumed increasing importance in recent times. The ability to obtain accurate and consistent indications of, for example, system quality, developer productivity and schedule projections is an essential component of effective project management. This paper focuses on these 'traditional' software engineering issues in relation to the development of spatial systems. In particular, techniques for development effort estimation are considered and a case study illustrating the application of one specific estimation method (Mark II function point analysis) is presented. Given its original basis in business information systems, the method is adjusted in order to account for (some of) the differentiating characteristics of spatial systems. The method is then retrospectively applied to a recently developed hazards analysis system. The effort estimate obtained is sufficiently close to the actual effort used in development to illustrate the potential of such a technique for project management in the spatial systems domain.

1. INTRODUCTION

For a spatial information system to be of any use, it must obviously satisfy user needs. Accordingly, the data must be structured in an appropriate form, both usable and maintainable; this form is likely to be centred around a database. A useful system also demands appropriate coded algorithms for data entry, manipulation, presentation and maintenance. Recent experience (Glassey *et al.* 1994) suggests that the effort required to develop this code may be considerably under-estimated in most development projects. Rather than contradicting research findings that suggest that a significant percentage of effort in systems development can be attributed to data-related collection and analysis (Mackaness 1989), this observation reinforces these findings; the purpose of the work here is to subdivide this effort still further. The particular interest is in the effort requirements associated with tasks that may be best described as software modelling and codification.

When designing data-centred 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). Such a design is enabled using well-established analysis and design tools such as entity relationship models (ERMs) and data flow diagrams (DFDs) (Chen 1976, DeMarco 1978). It has been further suggested (Benwell *et al.* 1991) that other tools, such as Petri nets (Benwell 1991) may be useful dynamic and concurrent information processing requirements are to be depicted.

These tools assist developers to abstract from reality necessary and sufficient information about reality, so that an effective information system can be designed. It is then necessary to codify rules and algorithms as well as to encode database interactions in order to provide the requisite functionality.

Using the products of the data-centred specification methods mentioned previously, it is possible to empirically estimate the effort required for code development using techniques from the discipline of software engineering (MacDonell 1993). In the case of the design and implementation of a spatial information system it would therefore be possible to;

1. determine the user requirements and define the scope of the system
2. model reality in terms of an ERM and DFDs
3. measure aspects of the ERM and DFDs
4. use the above metrics to indicate system *size*
5. determine from the metrics the effort required for codification
6. encode these models into a database and produce the 'lines of code'
7. implement and use the system

The aim of the current work is to determine the amount of effort involved in encoding (phase 6 above) based on data models (phase 2 above). As previously stated, it is contended that this effort is considerably under-estimated - 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).

These statements reflect the difficulty encountered in collecting and structuring spatial data in a database and in the encoding of procedures to enable and control access. Some early indications of system scope, and an assessment of the probable impact on development effort, would undoubtedly be useful in terms of effective project management.

2. SOFTWARE METRICS FOR EFFORT ESTIMATION

The discipline of software engineering is concerned with the cost-effective development of information systems to a commonly acknowledged and agreed 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.

These issues are clearly not important solely in the domain of spatial information systems. They are, however, 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.

Accurate estimation of development effort has been a long-time goal of those concerned with software project management. Research and practice have progressed from historically generated lines-of-code based estimates, through estimates derived from design documents, to present-day methods, with which effort can be estimated from functional models. It is this final class of estimation techniques that is the focus of this paper.

One approach to function-based estimation is now considered. The purpose of the following discussion and case study is to increase the awareness of project managers in the spatial systems domain as to the potential of such methods, with the hope that similar techniques might then be considered in the management of their projects.

2.1. Mark II Function Point Analysis

Function point analysis (FPA) 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 the current research project 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.

Within the business systems domain, industry standard calculation of the information processing size of a system (generally based on transactions manipulating attributes in data models) in unadjusted function points (UFP) is:

$$UFP = (0.58 * N_I) + (1.66 * N_E) + (0.26 * N_O)$$

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 (The weightings 0.58, 1.66 and 0.26 have been derived from analyses of around 100 business systems.)

This equation is said to account for the size of a system in terms of the required that are needed to cope with the formatting and validating of 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):

$$TCA = 0.65 + (0.005 * \sum F_i) \quad i = 1...20, 0 \leq F_i \leq 5$$

The final equation in determining overall functionality is therefore:

$$MkIIFP = UFP * TCA$$

In order to obtain the most useful estimates, specification and effort data should be routinely collected and analysed in each distinct software development environment, so that the weightings derived are appropriate. This collection and analysis 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.

3. CASE STUDY

A hazards information system has been developed (Aldridge *et al.* 1993) by the Institute of Geological and Nuclear Sciences (IGNS) through funding from the Foundation for Research, Science and Technology. The hazard register is a computer-based spatial 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 *hazards register* sub-system supports local authority routine operations: the processing of applications for building permits, resource consents, 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.

Figure 1 represents the ERM for the hazards system which, along with the functional models, can be subjected to Mark II function point analysis to determine system size and (retrospectively) development effort.

The information processing size of the hazard register system is determined from the system's logical transactions. Two main functions 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:

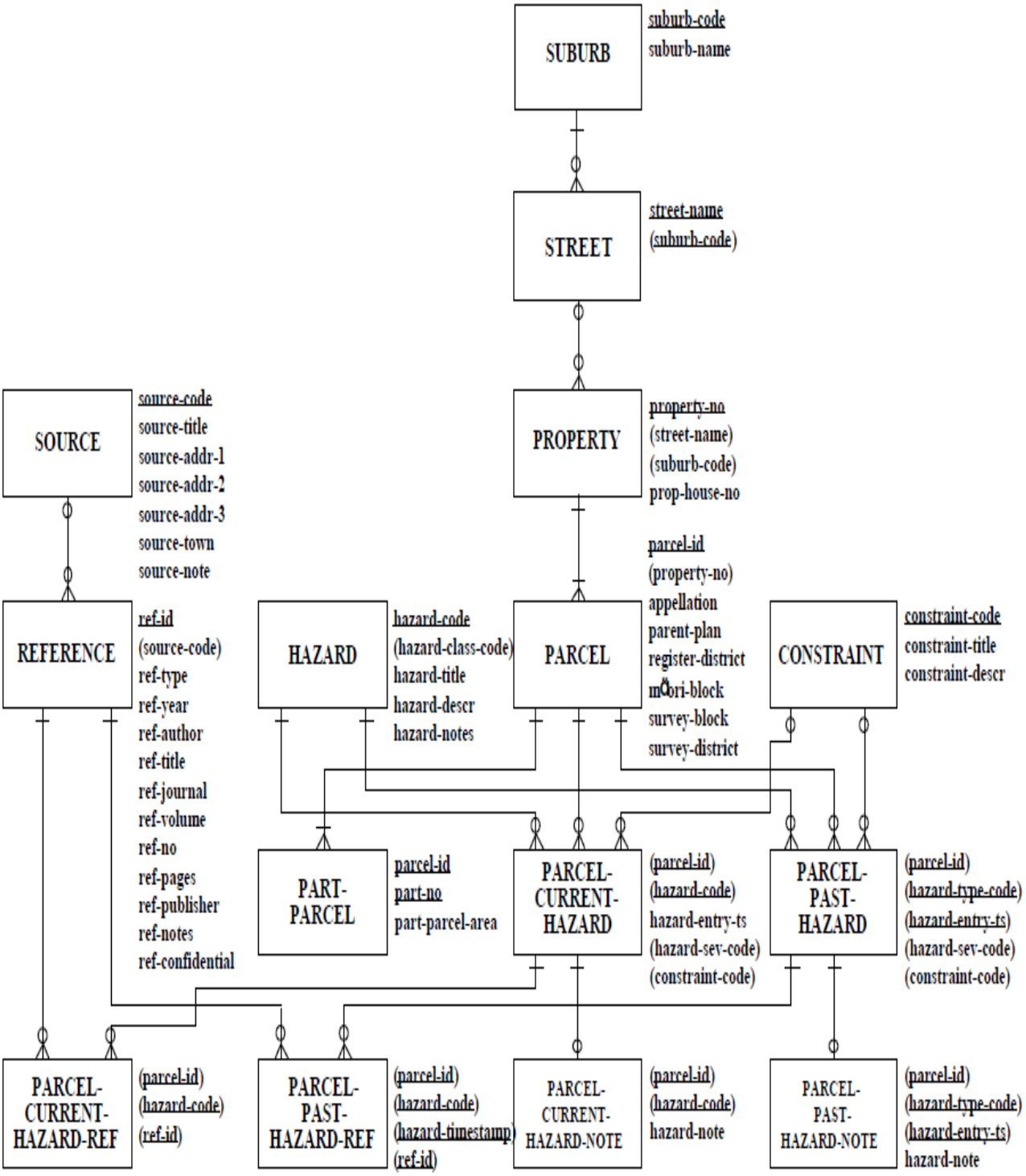

Figure 1. ERM for the pilot Hazard System

Function: *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* function 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:

$$UFP = (W_I * 71) + (W_E * 15) + (W_O * 40)$$

As mentioned in the previous section, 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 historical data available to enable the calibration. Industry-standard weightings have been supplied as a starting point (Symons 1991), but these were generated from around 100 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 order to determine more appropriate weightings, two separate approaches were made 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 developers to provide the authors with 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, effective and comprehensive data collection and analysis programmes are still rare.

A different approach was then tried. A new request to GIS-L asked developers to rate the difficulty associated with code design and development for the three components of transactions in a GIS environment. 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 component weightings

From the last two rows of Table 1 it appears that software development for performing both input and output handling for spatial systems is, 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 be computed:

$$UFP = (0.81 * 71) + (0.84 * 15) + (0.85 * 40)$$
$$= 57 + 13 + 34$$
$$= 104$$

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

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

$$TCA = 0.65 + (0.005 * \sum F_i) \quad i = 1...20,\ 0 \le F_i \le 5$$

$$= 0.65 + (0.005 * 45)$$

$$= 0.875$$

The final equation in determining overall functionality is therefore:

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

$$= 104 * 0.875$$

$$= 91$$

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, 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 have an average productivity rate of 0.1MkIIFP/work-hour. If that figure is adopted here, this results 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. The function point-based 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, adjustment factors may be identified and incorporated into the model.

4. 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 can provide a primary indicator of such effort. Given historical records, effort prediction can be performed well in advance of actual development.

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.

ACKNOWLEDGMENTS

The authors wish to acknowledge the assistance and support provided by their colleagues in the Department of Information Science; Mr Colin Aldridge for the derivation of the ERMs and DFDs for the IGNS Hazard Information System; Mr Bruce McLennan for obtaining references from obscure and far away places; and, Dr Peter Firns for constructive comments on the original manuscript. We also acknowledge IGNS, in particular Mr Phil Glassey, for information and permission to publish material relating to the hazard management system.

REFERENCES

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., 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.

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

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

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

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.

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.

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.

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.