A System Development Methodology for Geomatics as Derived from Informatics

Dr. George L. Benwell[1]
Department of Information Science
University of Otago

February 1994

Abstract

This paper describes the creation of a system development methodology suitable for spatial information systems. The concept is substantiated on the fact that spatial systems are similar to information systems in general. The subtle difference being the fact that spatial systems are not yet readily supported by large digital data bases. This fact has diverted attention away from system development to data collection. A spatial system development methodology is derived, based on a historical review of information systems methodologies and the coupling of same with a data collection and integration methodology for the spatially referenced digital data.

---

[1] Address correspondence to: Dr G.L. Benwell, Senior Lecturer, 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 system development methodology for a spatial information system (SIS) is required for the efficient development of such systems. It is not apparent in the literature, or in practise, that such methodologies have been readily adopted. This is despite the extensive applications of methodological approaches commonly used in informatics. In an attempt to predicate the same approach for SIS development the history of systems development is presented as well as an investigation of the literature concerning methodological approaches for SIS. It is shown that SIS are similar to any information systems and therefore like development procedures can be adapted. Nonetheless, even if there were differences, there is empirical evidence to show that the adoption of system development methodologies for SIS have significant financial advantages. Finally, a methodology is presented which is the coupling of a contemporary method with an integrated data collection methodology.

2 Background to The System Development Life Cycle

Implementation of a computer application is so complex that managers need an intellectual model to plan and control the project. A system development life cycle (SDLC) methodology is an explicit breakdown of the work that is required to implement a new or modified information system. Although acknowledged not to be a strictly sequential set of activities, the methodology as shown in Figure 1, provides a framework in which to consider, in general terms, the events which occur during the development of information systems.

For the most part, these events are considered as phases of development and say more about the organisational or production process of the work in progress than the techniques which are actually applied. Each step in the cycle should be rigorous, logically connected to its neighbouring steps and have a definable and measurable output.

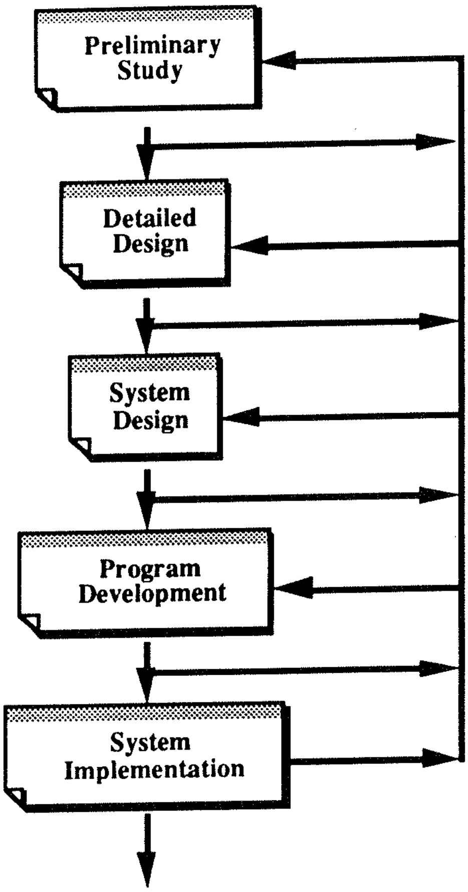

Figure 1: The System Development Life Cycle (adapted from [Benwell et. al., 1991])

There are three kinds of system development cycles [Martin and McClure 1985, p11];

- classical approach,
- reusable code, and
- information engineering/prototyping.

Briefly, the classical approach relies heavily on extensive paper documentation and approval of both requirements and design before program development begins. In essence it involves the following steps, concept identification, requirement specification, design specification, development, qualification and finally deployment. It is a simple framework within which to manage a well structured project, but lacks rigour.

With the reusable code methodology the objective is to implement the new application with as much existing software as possible. The method is similar to the classical approach but with the distinct advantage of producing more effective applications in a more efficient manner. As the use of modular software increases via the constraints of structured program languages (such as object oriented development) and 4th generation tools, so too will the acceptance of the reusable code methodology.

The information engineering - prototyping methodology is preferred. This system development model abstracts, from reality, a model which is made explicit in a computer application. Within the context of this paper the application will be confined to an information system, in particular, a spatial information system (SIS). Reality is usually modelled using the conventional functional decomposition [Blank and Krijger, 1983] which is followed by entity-relationship models. The understanding of these models may be enhanced by data flow diagrams and logical access maps.

3 Introduction to the System Development Life Cycle

The development of a computer based system from conception to implementation can be defined by a sequence of well ordered and interactive phases. A system developer may employ a defined methodology for determining user specifications, for designing system and data structures, for implementing a system and finally for evaluating and modifying system performance. These components constitute the phases of a system development methodology. A methodology typically consists of a set of interrelated tools and techniques (though these are, even now, somewhat discrete). Contemporary techniques combine to form a methodology which may be described as 'a waterfall model' and is frequently and intentionally iterative and recursive. Modern methodologies are data oriented; what data are present, how are the data used, in what form are the data, what are the data relationships, what are the data storage structures, in what output form are the data required? These are typical data oriented questions that are addressed within modern system development methodologies.

It is important to realise that the history discussion here, relates to the SDLC of (non real time) information systems (IS) in general. In that regard, it does not specifically relate to SIS. Given that SIS fall within the umbrella of IS, then of course the history and discussion are relevant. However, the discussion dates back to the 1970s whereas the current development of SIS, in Australia and New Zealand, can only be traced back to the early 1980s.

Without digressing too far, it is worth defining the word *system*. This is important as early texts have various conflicting meanings. It was generally held in texts and papers of the

1970s and earlier that *system* was confined to a program or even a single computer run. Compared with a contemporary understanding, this is a narrow definition. In a contemporary sense, *system* would be considered as including, data, a database, information, hardware and software, procedures and people [Turk, 1992]. Typically, *system* could be a suite of programs that access information stored in databases and present it in sophisticated ways on numerous types of output devices (such as screen forms, graphic screens or large format plotters). Henceforth, within these inherited contradictions, reference texts and papers will be cited and quoted verbatim with no more than an implied warning of the apparent conflict.

4 The History of the System Development Life Cycle

The history presented here has been divided into four time periods. Epoch 1 is prior to the mid 1970s, and may be labelled as *simplistic and problem oriented*; Epoch 2, from the mid 1970s to 1980, *early attempts at methodologies*; Epoch 3, from 1980 to the mid 1980s, *development of the 'waterfall' model*; and finally Epoch 4 from the mid 1980s to the present day, *CASE and integration*. These somewhat arbitrary boundaries subdivide the history of information systems development.

Epoch 1

Given that during the 1960s and 70s computers were considered to be procedure followers, it is not surprising that to develop a system meant that a solution procedure had to be created. It is therefore logical that the methodologies (and so too the SDLC) would closely reflect the episodic nature of the well tried solution methodologies for mathematical problems. This was also the method frequently advocated for software production [Tonge and Feldman, 1975, p21].

The clear emphasis here is on problem solving in a simplistic (in retrospect) episodic nature. This mind set was mirrored in the concept of a SDLC. That is to say, methodologies that were derived for conventional problem solving were applied to computer programming and in turn applied to system development. No doubt a natural progression. Concurrently, designers were plagued by the apparent lack of computing power. This manifests itself in the over emphasis of the concern for a system solution that fell within the computational ability of the computer ([Clifton, 1974, p144]).

The emerging picture is therefore one of developing a system within the constraints of a limited computer and employing a methodology transplanted from general mathematical problem solving. The methodology in these early stages was a software and hardware approach or view of the system [Couger, et. al., 1982, pp15-72]. It was certainly not a data view as mentioned earlier, a view so characteristic of present day systems development.

Epoch 2

From the middle of the 1970s there appears to have been a rationalisation in the thinking of those involved in system design and implementation. Texts (for example [Lucas, 1981, p78] and [Olle, et. al., 1982, p1]) were now devoting chapters to 'Information Systems Analysis and Design'. In fact Olle et. al. [ibid.] collated more than fifteen papers on system development methodologies. Some of these had been evolving since 1974. For example, Aschim and Mostue [1982, p15] describe an information system design methodology developed from 1974. Work commenced in 1975 to develop a computer aid for the information system development process. Possibly, for the first time, there emerges an holistic design methodology. It is not claimed here that the work by Aschim et. al. was anything more than indicative of the new approach; they may not have been first but they were innovative. Concurrent with this new approach came the fundamental work by Chen [1976]. It would appear that Chen and, Aschim and Mostue were working separately on a data oriented view of information systems development.

Integration and holism is reflected by the decomposition of functions documented by Lucas [1981, p78], where he details the steps in a SDLC. There should be no apparent dissonance between integration and holism and decomposition as the latter is used only as an aid. So, to the end of the 1970s the SDLC had evolved to the point where data oriented views were predominant, within an holistic concept of development. Various stages of the SDLC were becoming more clearly defined but the process was still somewhat episodic and without recursion or iteration. Recursion and iteration were to be recognised as important because of the difficulty users and analysts had in obtaining a shared and clear understanding of reality. Further refinements were to appear in the early 1980s.

Epoch 3

Fox [1982, p95] displays the SDLC in a recursive model with six interrelated components; requirement definition, design, program, construction, test/verify and document. This was an advancement on the Lucas approach for it showed an understanding that the process was iterative and not simply unidirectional from start to finish. Such models for the SDLC abounded in the texts of the time ([Ahituv and Neumann, 1982, p233]; [Briggs, et. al., 1980]; [Davis, 1974]; [Allan, et. al., 1978] and [Jeffery and Lawrence, 1984, pp3-9]).

Therefore, by the early 1980s the SDLC had many researchers developing it in similar ways. The constraints of inferior hardware capabilities were waning. All methodologies were recognised as iterative though the concept of obsolescence remained as a disturbing warning. For, if the process was truly iterative, evaluation and modification of the system's operation should ensure that it was never (rarely) to become obsolete. That is to say, that if the 'ground moved' then systems could be adapted efficiently to service any new demands.

What remained to be developed were the tools, the computer aided software engineering (CASE) tools, that could be utilised for the now well defined phases of the SDLC. Such tools would lead to differing methodologies. With this in mind it can be reiterated that the work by Aschim and Mostue [op. cit.] was well ahead of other developments. It had been them, as early as 1974, who had commenced work on some CASE tools, albeit natural language assistance. Their's and Chen's [1976] were both seminal works that laid the foundations for the development of CASE tools in the second half of the 1980s.

Epoch 4

Computer-aided techniques for design, analysis and documentation typified the late 1980s. The SDLC until then, had received little significant modification, notwithstanding constant refinements and the move toward improved documentation and rapid prototyping [Vlugter, 1989]. This period is characterised by the computer-aided approach being employed within the, by now, numerous methodologies.

Necco et. al. [1987, p463] describe this era as typified by the structured approach to systems design. This structured approach has continued to gain acceptance but with changing emphasis. The 'structure', derived from data flow diagrams, has remained but the terminology has given way to 'data'. In part, this reflects the inability of data flow diagramming (DFD) to adequately model all aspects of data and information. Whitten et. al. [1989, p200] conclude that while structure diagrams (such as DFDs) are easy to draw they have not ensured completeness, consistency, and accuracy in systems development. On the other hand it represents a strong move toward data and information and away from inputs and outputs. The latter are no doubt still important but are considered to be a consequence of the design - data and information are the important design variables.

Olle et. al. [1988, p41] typify this epoch as a hybrid of *data, process and behaviour oriented.* They de-emphasise any apparent distinction between the latter two terms and stress the importance of the former.

So, with an increasing importance being placed on *data*, the late 1980s and early 1990s have witnessed the emergence of *Information* and *Systems Engineering*. Information Engineering (IE) (based on the early work by Martin and Finkelstein [1981]) seeks to de-emphasise processing and outputs, shifting to an emphasis on data and structures. IE also places much greater responsibility for system development on end-users. As a consequence of the increased role of the end-user it has been found necessary to adopt a technique called *rapid prototyping*. This facilitates the creation and testing of input designs, output designs, terminal dialogue, and simple procedures. It should not be seen as prototyping a system but rather building prototypes of selected components of the system. The creation and testing are intimately involved with feedback from end-users. The analyst, with a data or an information view of the system is able to rapidly present the user with designs which are known to be, at least initially, less than complete. The speed and convenience with which improved alternatives can be produced have been accepted as beneficial to analyst, designer and user. The data view and rapid prototyping are presently working in harmony to expedite the efficient production of information systems that (should) meet the needs of users.

5 A Contemporary Methodology - Otago IE/P

Complementary to the preceding historical section is a discussion of a contemporary methodology. The chosen methodology will provide the reader with an understanding of the present status of the SDLC.

The Otago Information Engineering/Prototyping (IE/P) Methodology [Kennedy, 1991] is founded on four tenets;

- *the system should be implemented on time and within budget,*
- *to provide the basis for the implementation of a computer-based solution where appropriate,*

- *for each and every application there is an intrinsic data structure which will satisfy all needs of the system,*
- *given a working prototype a potential user will be able to confirm or deny whether all requirements have been satisfied.*

The methodology is also based on the concept of rapid prototyping and as such has some reliance on fourth generation (4GL) systems. While a particular 4GL product is used the methodology has no product specific requirements. The 4GL facilitates development of the prototypes which can be rapidly amended or enhanced after user reviews. Finally, interaction between the user and designer is vigorously encouraged. This is utilised after the production of the first prototype, which in effect becomes the basis of the user requirement specifications for subsequent prototypes. The IE/P methodology has three phases; system proposal; user requirement specification; and system implementation. These phases are shown in Figure 2; the figure is intended to indicate that;

- the arrangement of the process blocks demonstrates their episodic and concurrent nature.
- processes interact and several iterations of one or more processes may take place.
- there is steady progress toward implementation.
- client approval is sought at the end of each phase before proceeding.
- confirmation of user requirements is obtained via an evaluation of the functional prototype by the client.
- the working prototype is iteratively improved until it converges on the updated user requirements.

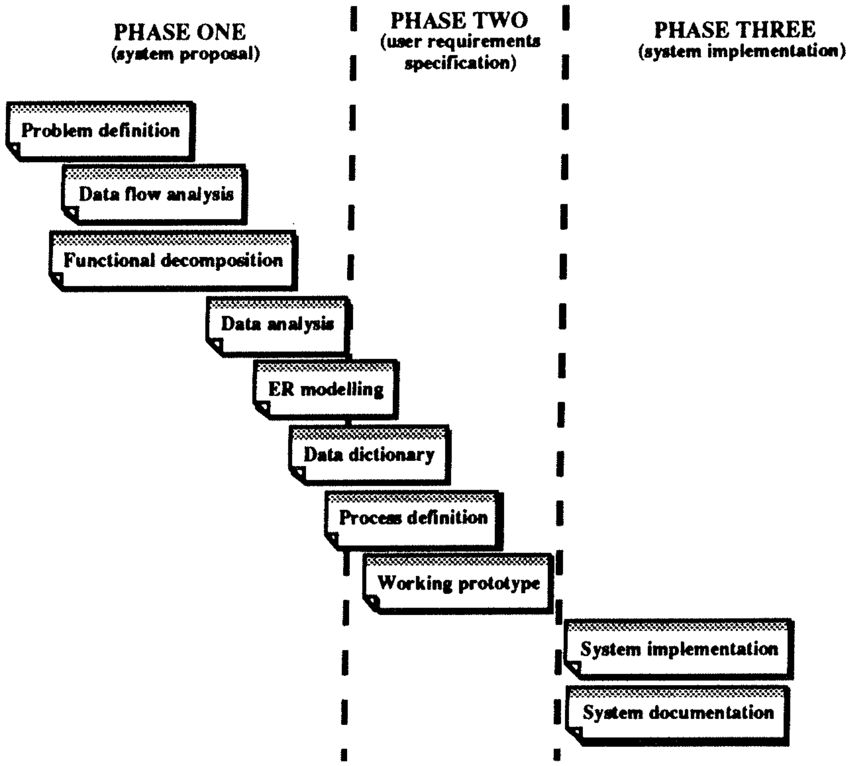

Figure 2: Phases of the IE/P Methodology (adapted from [Kennedy, 1991])

6 Introduction to the Modelling of Spatial Data

Along with the ever increasing introduction of spatial information systems (SIS) into industry there is a parallel emergence of the need to understand the holistic processes within which they are being used. It is all too easy to embrace new technology and its applications without first having a complete understanding of the processes that it purports to support. A similar phenomenon is evident in software development and systems design. Software and information engineering concepts are used extensively to increase the efficiency of software production. These concepts are not alien nor inappropriate to the introduction of SIS [Calkins and Marble, 1987] but by themselves are not complete tools. It is desirable that the understanding of process behaviour be a prerequisite to the application of SDLC methodologies to the design of SIS.

Behavioral analysis explains the 'how and what' of a process. It is not concerned with the behaviour of people (therefore it excludes the social science discipline of human behaviour), but rather the behaviour of processes. How is a spatial decision made and what are the logical interactions of its components? Process behaviour is therefore defined as the interaction of the process and its affecting and effecting environment with particular emphasis on the inputs and outputs. The definition would also include a detailed and rigorous explanation of the processes' modus operandi. Zachman [1988, p91] contends that there are several models of reality that are required. He concludes that a data model is necessary and sufficient to create a data system (presumably a database). But;

> ... I do not believe that the data model alone is an adequate description on *[of]* an enterprise ... There are other descriptive models which are relevant and inextricably related including at least;
>
> - the functional model (describing "how")
> - the geographic (or, logistics) model (describing, "where")
> - the event/cycle model (describing "when")
> - the objectives model (describing "why"), and
> - the organisational model (describing "who")
>
> all in addition to the entity/relationship model (describing "what").

He concludes with the following rather prophetic statement;

> It is clear that the focus for formalizing methodologies and tools is currently shifting to the data realm and I would predict that it will shift again in the future with the logistics (or "network") model next in line.

It is important in the SDLC to include a more rigorous method to assist in the understanding of process behaviour. If successfully completed, the behavioural analysis will lead to a complete set of unambiguous process requirements that may eventually lead to a similar set of design requirements. Importantly, behavioural analysis precedes design analysis with both components forming part of a system development life cycle. [Sallis, 1988, 1989; Benwell et. al., 1991].

7 Are Spatial Information Systems Different?

It has been argued by others that SISs are <u>so</u> different to other information systems that they should be designed and developed using non-standard or unique techniques and tools. Ezigbalike et. al. [1988, p284] support this concept, and state;

... the users do not have a good understanding of the system to be able to unambiguously describe their requirements to the developer. Also, because of the novelty of the system, there may not be any existing tools with which to develop it, even with accurate specifications. LISs are large, multi-user systems whose opportunities and problems are still being studied. Applying SDLC to their development will therefore suffer from both the communication and requirements problems. *[Ezigbalike et. al. define SDLC as Structured Development Life Cycle, the more common international definition is Systems Development Life Cycle]*

Assuming that such comments can be generalised from land information systems (LIS) to SIS, there are doubts about the basic tenet. It would seem more appropriate to recognise that users may have a good understanding, but if they do it is implicit not explicit. Also, while it may be true that *LISs* are *large*, if compared to betting or airline booking databases they may be *small*. On the other-hand when large corporations amalgamate or are formed to oversee national projects this assumption may not be valid. These situations are not common when compared with the general use of SIS. Nonetheless the lack of a clear understanding of reality is quite clear and is typified by the example of the Murray-Darling Basin Commission [Leahy et. al., 1989].

Dale and McLaughlin [1988, p10] demonstrate that at least at one level of abstraction, SISs are considered to be different to other information systems (Figure 3). While their classification is valid and an important contribution it is nonetheless based on application type. The discussion here is concerned with system development not application. Senn [1989, p24] considers that there are three types of information systems; transaction processing; management information; and decision support.

Transaction processing systems (TPS) are aimed at improving the routine business activities on which all organizations depend. A transaction is any event or activity that effects the organization. ... *[op. cit., p21]*

... *management information systems* (MIS) assist managers in decision making and problem solving. They draw on data stored as a result of transaction processing, but they may also use other information. ... *[op. cit., p23]*

Decision support systems (DSS) assist managers who must make decisions that are not highly structured, often called *unstructured* or *semi-structured decisions*. *[op. cit., p25]*

It logically follows therefore, that a SIS, as an information system, must fall within this classification. Based on Senn's classification some SISs would be considered to be similar to transaction based systems or retrieval systems. This is particularly true of LIS as they have been presently implemented.

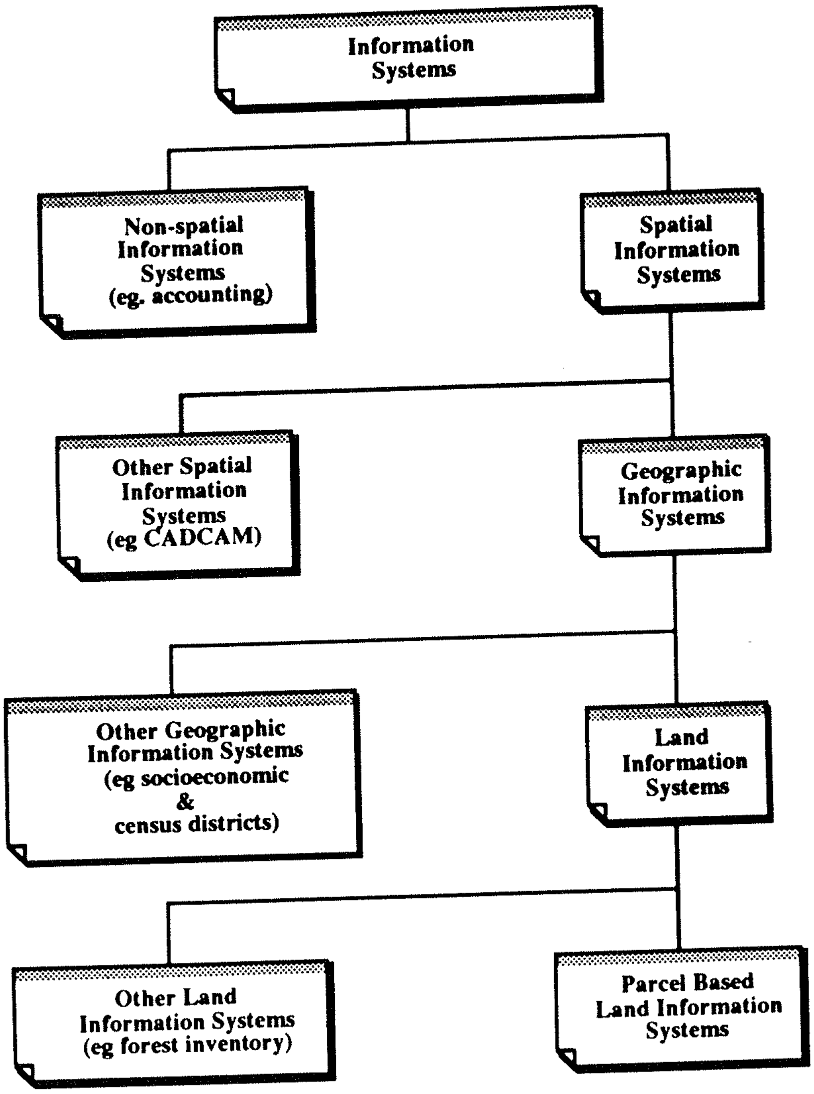

Figure 3: Classification of Information Systems (adapted from [Dale and McLaughlin, 1988])

Consider a local government SIS [Betts and Murnane, 1990]. The system contains cadastral and fiscal information. The TPS is concerned with (inter alia) updating of rate payers and generating rate notices; the MIS may be concerned with management of overdue payees and pipe networks; the DSS is concerned with the unstructured problems that relate to what is the level of next year's rates to receive a certain financial return.

Hence, perhaps SIS should be considered as being similar to any information system and therefore tractable to the common SDLC approach.

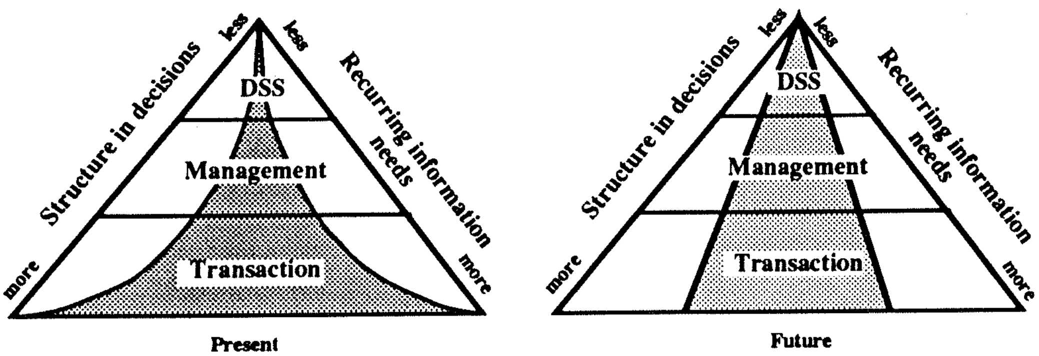

Figure 4: Classification of Information Systems [adapted from Benwell, 1991]

Figure 4 shows the three categories of information systems; transaction, management and decision support. As a reference, the present and expected future applications of SIS are hatched on the Figure. The present use of SIS is more characterised by transaction oriented information systems with little use being made of spatial information for decision support. It is expected that this situation will change as time passes and use increases. The right-hand part of Figure 4 indicates that the future use of SIS will trend away from transaction oriented toward DSS. This would indicate a maturing of the use of SIS, but there is little indication in Australia and New Zealand that will this occur within the next five years. The absence of complete digital cadastral and resource databases will confine SIS to transaction oriented until at least the mid 1990s.

8 Similarities

The taxonomy (Figure 4) adopted is relevant to information systems in general and hence to those systems being developed using SDLC methodologies. In addition SIS can be mapped onto this classification as the taxonomy has a utility that can include most if not all information systems.

There is little proof that SISs are <u>that different</u>. Ezigbalike et. al. [1988] contend that this is not the case. For that assertion to hold, they would have had to compare SIS with numerous other systems. Such a reference was neither implicit or explicit. All systems are different. An airline booking system, a doctor's office transaction system and a hydrographic survey data acquisition system are different. They are different mainly in use. The Dale and McLaughlin chart has been often misquoted to substantiate the (false) difference. It would seem to be a null argument to say that SIS are so different that they require a different approach.

A lack of adequate requirement specifications is a difficult problem that has to be addressed within all system developments. The problem is not unique to a SIS. As a consequence, the following statement by Ezigbalike et. al. [1988, p284] cannot be supported;

> Also, because of the novelty of the system, there may not be any existing tools with which to develop it, even with accurate specifications ...

Any system that is being developed for the first time will have this problem. The users, in all cases, will (may) have difficulty defining (in explicit terms) the requirements. Developers will (may) have difficulties supplying the required techniques and procedures. Ezigbalike et. al. [ibid] in attempting to show that SIS are different, quote;

> McCracken and Jackson contend that 'systems requirements cannot ever be stated fully in advance, not even in principle, because the user doesn't even *know* them in advance - not even in principle'. *[McCracken and Jackson, 1982, p31]*

This is a non qualified statement and applies equally to all information systems which would include SIS, GIS or LIS. Boutin and Bédard [undated, p1], have concluded that SIS are not sufficiently different than any other information system but added that SDLC methodologies may have to be amended or adapted;

> We conclude that a standard methodology can be used for the design of a LRIS *[Land Related Information System]* with few changes. Specific steps of the methodology have to be adjusted to the specific context of LRIS. The most important problem is about the modeling tools (data and processing modeling tools) that are not sufficient to fully represent the characteristics of land related data and processing.

The LRIS is, by any measure, a large (spatial) information system consequently the comment by Boutin et. al. is most relevant. It significantly adds weight to the argument that SIS are not different and that they can be developed using the SDLC methodologies employed by information scientists. Furthermore, LRIS is typical of many large land based information systems currently under development around the world. A statement about LRIS therefore, can logically be generalised to most spatial information systems. For examples within Australia and New Zealand, refer to Hesse and Williamson [1990].

Olson also writes [Olson, 1988, p110];

> ... the development of a geoprocessing system is no different than any other information system (from a development cycle standpoint) ...

Similar conclusions have been reached by Campbell [undated, p2].

Finally, Morris emphatically states [Morris, 1991, p5] that standard information science methodologies can be applied to the development of SIS. He goes as far as to say that if such methodologies had been used, $9m ($AUD) could have been saved in a $25m project! [Morris, 1991, (personal communications at question time)]. The Brisbane City Council began the development of a SIS in 1981 with a budget estimate of $14m; this went out to an estimate for 1995 of $25m. The difference of $11m is reported to be made up of $9m wasted (due to the lack of a defined methodology) and $2m in new and unforeseen components. Morris went on to say;

> Given current knowledge, a question could be asked as to whether the Brisbane City Council would follow the same path if it were starting its LIS/GIS system again. The answer to that question is NO. With the advent of new processes such as CASE tools, a formal approach would now be adopted.
>
> The Council has recently adopted the APT methodology as a means of following an information technology project through its development life cycle. Using such an approach provides direction, cohesion and better reporting facilities as staff know where the project is and what to expect.

The problem seems to be therefore, not that SISs are difficult or different but rather that some SDLC methodologies are incapable of easily building a system. In an attempt to solve this problem contemporary methodologies use rapid prototyping.

On this point Ezigbalike would seem to agree [Ezigbalike, 1988, p184, cites [Boar, 1984, p5]]. Rapid prototyping, as a solution, begs the real problem at hand. There remains a need for a clear and shared understanding of reality by the user and designer. Rapid prototyping partly solves this, but largely circumvents the problem. The real issue may be found in a formalism of reality that provides the required clarity and mutual understanding.

9 System Development in Spatial Information Systems

The history of SIS can be traced back to the 1960s with developments in Canada and north America [Marble and Wilcox, 1991, p2]. The related history of systems development in SIS can only be traced back to the 1970s with the work by Calkins [1972]. Unfortunately, this work basically lay idle for a decade.

Such early attempts at a SIS oriented SDLC should be more correctly described as conceptual modelling as they exceed the domain of a SDLC. There should be no misunderstanding that a SDLC is definitely a sub-set of a conceptual model [Williamson, 1988, p34]. Therefore it will not cover all the areas of a conceptual model but in part will be far more oriented towards detail and implementation. Any confusion may simply lie in the differing use of words. Marble and Wilcox used 'GIS Design and Implementation' when talking about an overall conceptual model. The matter should rest with an agreement to disagree on the meaning and use of words. May be, at some point in the future, a standard will be agreed to. The idea of conceptual modelling is understandable in that SIS (as discussed below) is centred on integration - of data and organisations. It is the latter that commanded much initial attention.

The advances in SIS methodologies have been in realising that such systems, at least in one sense, are atypical; they are one of the few management information systems that consolidate data originating from a number of organisations at different levels of administration [Zwart, 1984]. Similarly there is a growing realisation that the discipline of geomatics can gain from the experiences in informatics. Love [1991] noted this in his work on SIS design and implementation methodologies. He advocated that a sound methodology was necessary and emphasised the importance of technical and humanistic perspectives.

10 A Synthesis

The preceding discussions have reviewed the historical development of system development methodologies, and reviewed matters relating to spatial information systems.

The points to note are, that conventional information systems development was occurring, at least, a decade and a half before that for SIS. This is important, more so for SIS developers, as there was a wealth of experience to refer to. While this is true, there was a perceived reluctance by SIS designers to draw on this knowledge or to take note of authors such as Calkins. There also seems to have been (and possibly still are) some misconceptions as to the difference between conceptual modelling and systems designs and methodologies.

It has been established that, as far as systems are concerned, SIS can be considered as a subset of information systems in general. This is an important finding as it follows that methodologies derived for one should be applicable to the other. Having said that, there is an important difference that needs to be addressed.

Data for general information systems has been undergoing digital conversion for the last three decades; its sources being well defined and structures relatively simple. On the other hand SIS have stormed into business with a notable lack of digital data. This has certainly delayed their advancement, but more importantly, it has concentrated research and development efforts on 'data collection' rather than systems design and development. That statement is true not only of Australia and New Zealand, but all countries embarking on the development of SIS. So it is with this as back ground that a SIS methodology has been defined.

The methodology is the meshing or interlocking of a contemporary systems methodology (in this case Otago IE/P) with what may be seen as project management methods for data collection. While that may be one view, the alternative and preferred view is, that a spatial data collection methodology has been coupled with a systems development methodology to create an appropriate régime within which to develop SISs. The data collection methodology has been developed from a concept of 'conflict resolution'. Conflict resolution is the process of determining;

1 the source(s) of non-digital data appropriate to the problem under consideration.

2 the form of the source data.

3 the form and structure of the digital data.

4 the most appropriate scale(s) and accuracy(ies) of the source data (where source data is in map form).

5 the data acquisition method(s).

Conflict must be resolved, as in the majority of cases of SIS development, source data is in one graphical form or another. It is also generally true, that the data are of less accuracy than is now required for a SIS. This creates conflicts or errors when source data is used beyond its spatially valid capabilities. These matters are of considerable current research [Hunter and Goodchild, 1994] and are particular to SIS. Data must be collected in form, structure and of known linage at the appropriate stages of the SIS development.

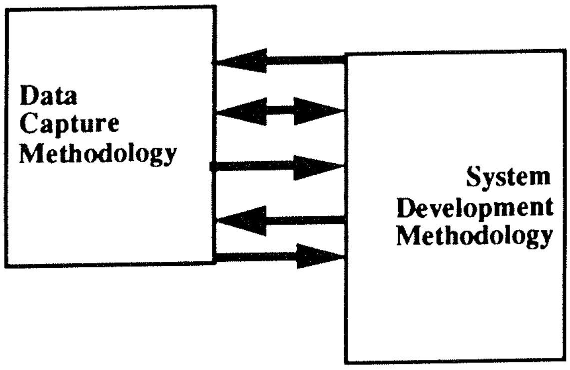

Figure 5: Conceptual Coupling of Methodologies

So, it is the close coupling of a data collection methodology (DCM) and a system development methodology (SDM) that creates a new framework within which a SIS can be built. The coupling are directed in two ways. Firstly, the SDM will place 'demands' on the DCM when data is required. It may well also indicate that a particular data type in a specified structure should be available. On the other hand, the DCM requires of the SDM similar information. There are occasions when the interaction is, at the same instant of the development, bi-directional. These concepts are shown in Figure 5 and detailed in Table 1.

11 Conclusion

A methodology for the creation of spatial information systems has been presented. It is founded upon experiences of such techniques in informatics. Geomatics' considerations have been acknowledged and are reflected in the coupled data collection methodology. The concept of the coupling has been developed in an academic environment but is undergoing practical testing in two commercial case studies. One is for a large regional government in southern New Zealand and the other is the establishment of an urban geological hazards information system being developed in co-operation with a Crown research institute in New Zealand.

Data Collection Methodology	Coupling	I/E P Methodology
		PHASE I
		1 Interview client
investigate available data and sources	⟷	2 Identify problem
		3 Identify functions
how data are to be collected	⟵	4 Define outputs
		5 Model data flows
		6 Generate ER model
what data are to be presented and in what form	⟷	7 Perform data analysis
availability of 'suitable' data and structure/form	⟶	8 Assess feasibility
		PHASE II
		1 Develop data dictionary
		2 Develop process definition
what data are to be used and where	⟵	3 Design test data
		1 revise data model
		2 create data structure
		3 develop input screens
collect and aggregate test data with the appropriate tool, eg scan, digitise, keybd	⟶	4 enter test data
		5 generate sample output
		6 develop menu structure
sample data co-ordinated to facilitate completeness	⟵	7 develop test cases and testing schedule
data appropriateness checking	⟶	*Confirm User Requirements*
		PHASE III
re-design/engineer data structure, type, volume, etc	⟶	1 revise/refine prototype
		2 complete details
		3 user/system documentation
gather/organise/present 'real' data sets	⟵	4 develop 'live' data sets
		5 generate sample output
		6 implement system security
		1 user up-skilling
		2 site preparation
develop procedures which define how data are to be used and controlled	⟷	3 develop clerical procedures and fault reporting
		4 develop acceptance test schedules
translate real data (via conversion tool) into system, readable form, and populate	⟷	5 plan data up-take

Table 1: The Coupled Methodologies

References

AHITUV, N. and NEUMANN, S., 1982 Principles Of Information Systems For Management, Wm. C. Brown, Dubuque, Iowa, 606 pages.

ALLEN, J. and LIENTZ, B.P., 1978 Systems in Action, Goodyear Pub. Comp., Santa Monica, cited in [AHITUV et. al., 1982, p276].

ASCHIM, F. and MOSTUE, B.M., 1982 IFIP WG 8.1 Case Solved Using SYSDOC and SYSTEMATOR, paper in [OLLE et. al., 1982], North-Holland Pub. Co., Amsterdam, pp15-40.

BENWELL, G.L., 1991 Casting Petri Nets Into the System Development Life Cycle in the Context of Spatial Information Systems, Unpublished PhD thesis, The University of Melbourne.

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, No. 6, pp15-25.

BETTS, R. and MURNANE, A.B., 1990 Land Information Systems in the Board of Works - The Future, Conference Proc., SIRC '90, Proceedings of the 2nd Annual Spatial Information Research Centre Conference, University of Otago, pp1-12.

BRIGGS, C.L., BIRKS, E.G. and ATKINS, W., 1980 Managing the System Development Process, Prentice-Hall, Englewood Cliffs, cited in [AHITUV et. al., 1982, p276].

BLANK, J. and KRIJGER, M.J. (Eds.), 1983 Software Engineering: Methods and Techniques, Wiley-Interscience, New York, 295 pages.

BOAR, B.H., 1984 cited in [EZIGBALIKE, I.F.C., 1988, p284].

BOUTIN, G. and BÉDARD, Y., undated, On The Use of a Standard "Information Systems Design Methodology" For the Development of a Land Related Information System, unpublished Masters' research extract, Centre d'Expertise en Géomatique

(CEGEO), Département des sciences géodésiques et télédétection, Faculté de Foresterie et Géodésie, Université Casault, Quebec, Canada, English facsimile copy (part of), p22.

CALKINS, H.W., 1972 An Information System and Monitoring Framework for Plan Implementation and the Continuing Planning Process, unsighted, unpublished Ph.D. thesis, Department of Civil Engineering, University of Washington.

CALKINS, H.W. and MARBLE, D.F., 1987 The Transition to Automated Production Cartography, Design of the Master Cartographic Database, The American Cartographer, April, 23 pages.

CAMPBELL, G.G., undated, ABSTRACT: Integrated Spatial Information Systems Design Through Information Engineering, Legal Surveys Division, Canada Centre for Surveying, Energy, Mines and Resources, Ottawa, Canada, facsimile copy (part of), 17 pages.

CHEN, P., 1976 The Entity-Relationship Model Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, No. 1, pp9-36.

CLIFTON, H.D., 1974 Systems Analysis for Business Data Processing, Second Edition, Business Books Limited, London, 260 pages.

COUGER, J.D., COLTER, M.A. and KNAPP, R.W., 1982 Advanced Systems Development Feasibility Techniques, John Wiley and Sons, New York, 506 pages.

DALE, P.F. and McLAUGHLIN, J.D., 1988 Land Information Management, An Introduction With Special Reference to Cadastral Problems in 3rd World Countries, Clarendon Press, Oxford, 266 pages.

DAVIS, G.B., 1974 Management Information Systems: Conceptual Foundation, Structure, and Development, McGraw-Hill, New York, cited in [AHITUV et. al., 1982, p276].

EZIGBALIKE, I.F.C., COLEMAN, D., COOPER, R.H. and McLAUGHLIN, J.D., 1988 Land Information Systems Development; Methodology, Conference Proc., the 26th

Annual Conference of the Urban and Regional Information Systems Association, pp282-298.

EZIGBALIKE, I.F.C., 1988 Land Information Systems Development; Software and Management Considerations, unpublished PhD. Thesis, The University of New Brunswick, Canada, 237 pages.

FOX, J.M., 1982 Software and Its Development, Prentice-Hall, Englewood Cliffs, New Jersey, 299 pages.

HESSE, W.J. and WILLIAMSON, I.P., 1990 A Review of Digital Cadastral Data Bases in Australia and New Zealand, The Australian Surveyor, Vol. 35, No. 4, pp351-367.

HUNTER, G.J. and GOODCHILD, M.F., 1994 Managing Uncertainty in Spatial databases, Journal of the Urban and Regional Information Systems Association, Vol 5, No. 2, pp55-61.

JEFFERY, D.R. and LAWRENCE, M.J., 1984 Systems Analysis and Design, Prentice-Hall, Sydney, 230 pages.

KENNEDY, G.J., 1991 A Methodology for the Development of Information Systems Using Rapid Prototyping, working paper, Department of Information Science, University of Otago, 10 pages.

LEAHY, F.J., TURK, A.G., HAZELTON, N.W.J. and WILLIAMSON, I.P., 1989 Proposals for Utilisation of Geographic Information System Technology by the Murray-Darling Basin Commission, Consultancy Report, The Department of Surveying and Land Information, The University of Melbourne, 92 pages.

LOVE, W.R., 1991 GIS Design and Implementationm A Successful Methodology, Conf. Proc., Proceedings of the 19th Australian Conference on Urban and Regional Information Systems, Wellington, pp474-484.

LUCAS, H.C., 1981 The Analysis, Design, and Implementation of Information Systems, Second Ed., McGraw-Hill, Auckland, 419 pages.

MARBLE, D.F. and WILCOX, D.L., 1991 Measure Twice - Cut Once: A Structured Approach to Successful GIS Design Implementation, Conference Proc., 11th Annual ESRI Users Conference, Redlands, California, 13 pages.

MARTIN, J. and FINKELSTEIN, C., 1981 Information Engineering, New York: Savant Institute, cited in [WHITTEN et. al., p270, 1989].

MARTIN, J. and McCLURE, C., 1985 Structured Techniques for Computing, Prentice-Hall, New Jersey, 388 pages.

McCRACKEN, D.D. and JACKSON, M.A., 1982 Life Cycle Concept Considered Harmful, ACM SIGSOFT, Software Engineering Notes, Vol. 7, No. 2 pp29-32.

MORRIS, K.J., 1991 Local Government GIS Under the Spotlight, A discussion paper presented to a seminar tour of New Zealand, ARISA, Wellington, New Zealand, 7 pages.

NECCO, C.R., GORDON, C.L. and TSAI, N.W., 1987 Systems Analysis and Design: Current Practices, Management Information Systems Quarterly, December, pp461-476.

OLLE, T.W., HAGELSTEIN, J., MACDONALD, I.G., ROLLAND, C., SOL, H.G., VAN ASSCHE, F.J.M. and VERRIJN-STUART, A.A., (Eds.), 1988 Information Systems Methodologies: A Framework for Understanding, Addison-Wesley, Wokingham, England, 346 pages.

OLLE, T.W., SOL, H.G. and VERIJN-STUART, A.A., (Eds.), 1982 Information Systems Design Methodologies: A Comparative Review, North-Holland Publishing Co., Amsterdam, 648 pages.

OLSON, D.A., 1988 Integration of Hierarchical and Relational Database Structures in a Contemporary Geographic Information System, Conference Proc., URISA '88, Proceedings of The 26th Annual Conference of the Urban and Regional Information Systems Association, Los Angeles, California, Vol II, pp110-118.

SALLIS, P.J., 1988 Quality Assurance and Productivity Enhancement, Conference Proc., Proceedings of the Rutherford Conference, New Zealand Computer Society Biennial Conference, New Plymouth, 12 pages.

SALLIS, P.J., 1989 Integrating Computer Aided Software Engineering Tools With a Generalised Information System Development Methodology, Conference Proc., Proceedings of the 11th New Zealand Computer Society, Wellington, pp589-596.

SENN, J.A., 1989 Analysis and Design of Information Systems, 2nd Edition, McGraw-Hill, Sydney, 853 pages.

TONGE, F.M. and FELDMAN, J., 1975 Computing: An Introduction to Procedures and Procedure-Followers, McGraw-Hill, Computer Science Series, New York, 303 pages.

TURK, A.G., 1992 GIS Cogency: Cognitive Ergonomics in Geographic Information Systems, Unpublished PhD thesis, The University of Melbourne.

VLUGTER, M., 1989 Rapid Prototyping for Computer Aided Learning Applications, New Zealand Journal of Computing, Vol. 1, No. 1, pp73-79.

WHITTEN, J.L., BENTLEY, L.D. and BARLOW, V.M., 1989 Systems Analysis and Design Methods, Irwin Pub. Co., Homeland, Illinois, 797 pages.

WILLIAMSON, I.P., 1988 Developing Land Information Systems - Some Issues, Conference Proc., FIG Sponsored Workshop on Land Information Systems in Developing Countries, Bali, Indonesia, The Institution of Surveyors, Australia, Sydney, pp32-42.

ZACHMAN, J.A., 1988 Data Modelling as a Tool for Data Systems Planning: Can Data Modelling Accurately Reflect the Enterprise?, (Abstract) Entity-Relationship Approach, Elsevier Science Pub., Amsterdam, Holland, 91 pages.

ZWART, P.R., 1984 The Rise and Decline of Land Information Systems, Conf. Proc., 12th International Cartographic Association Conference, Perth, pp123-133.