Can Architectural Design Learn from Software Engineering ?

Tony Pollard, *University of Western Sydney, Nepean*
Jim Plume, *University of NSW*

Software engineering is the structured approach to the development of computer software. It centres upon the concepts of system theory, and as such, many of its concepts, philosophies and techniques can be applied to the development of systems outside the field of computer software.

The main area covered by this paper is the use of process modelling techniques as an aide in architectural design. As buildings are systems, this paper discusses process modelling, a techniques used in computer system development, and how such a concept and technique can be adopted in architectural design as an aid in graphic simulation of the construction problem.

The modelling techniques are detailed in an example describing how such models can be simply applied to simulate a particular building problem, and provide a greater understanding of particular aspects of an architectural problem and as a means of documenting design decisions.

Introduction

The proposition of this paper, is that tools and techniques used in different design disciplines may have application within another. In particular, it describes process modelling, a modelling and simulation technique used in software engineering. Process modelling has possible wider use in the field of design, particularly in relation to complex project orientated design problems such as in the field of architecture. The paper examines the parallel features of both the disciplines of architectural and software design, and uses an example to describe the technique in operation. The advantages of the technique applied to architectural design are considered, and suggestions as to how such a tool may be used in practice.

The Authors can be contacted at

Mr A Pollard, Senior Lecturer
Department of Computing
UWS Nepean
PO Box 10, Kingswood, NSW 2747
email : t.pollard@nepean.uws.edu.au
tel : (047) 360 610
fax : (047) 360 713

Mr J Plume, Senior Lecturer
Faculty of Architecture
University of NSW
PO Box 1, Kensington, NSW 2033
email : j.plume@unsw.edu.au
tel : (02) 697 4816
fax : (02) 662 1378

Foundation of Proposition

Let us define the terms which we are going to discuss, firstly *Simulation* is defined by the Macquarie Dictionary (1991) as the use of an analogue in order to study the properties of a system. The important part of this definition is the inclusion of the concepts of systems.

Secondly, the term *Software Engineering* is associated with the design and development of computer software. Brooksheare defines it as the software development process, noting that it centres around the topics of problem solving and system design (Brooksheare J G, 1985). Again the word system gains prominence, but also the area of problem solving.

It is not so simple to define the third of the terms, *Architectural Design.* Like architecture and art, this term has been defined in many ways with various biases, but as we are not here to enter into a detail discussion as to what Architecture is or is not, this

paper adopts a more catholic approach, such as that of Heimsath who noted that Architecture by definition, is built for people. Architecture is the enclosure in which people live their lives (Heimsath C,1977) . We need to consider, however, within this broad context, a number of detailed items. Kingsland brings out another commanding point, that the design process is inclusive of all of the work of the architect, not just the creative parts (Kingsland A, 1990). Time indicates the longevity of the next additional factor, when both Vitruvius and Lang presents the view that Architecture is a many faceted discipline, involving both the arts and sciences (Lang J, 1990) (Vitruvius). Both Broadbent (Broadbent G, 1970) and Cowdroy (Cowdroy R, 1990) further develop this theme to include the integration of a variety of areas of knowledge in the major professions, including architecture, such that professional judgement includes a wider range of knowledge and comes from a range of common techniques, rather than a narrow professional approach. Included in this understanding of architecture is the concept that an understanding of systems is an important part. Considering the three definitions therefore, simulation relating to the analogues of systems, software engineering relating to the creation of software using system theory, and architecture involving a required understanding of systems, the obvious common thread through these definitions is the use of systems and systems concepts. If we examine the common working methodologies of the two areas of architectural design and software engineering, in the light of the common feature of system, the study may isolate common factors that may be of interest and relevance to the architectural profession.

Architectural design is, by its very nature, a project orientated process. As such, a common method of describing the architectural design process is to classify it into the project milestones of Brief, Brief Development, Implementation and Project Evaluation.

The Briefing phase within the architectural design process generally includes the definition of the function of the building, including the required space, services, materials and finishes per function. The brief examines the feasibility of the project, in technical, financial and social terms, and produces initial models to explain the options available to the client to enable reasonable analysis. It becomes a document which is submitted to the client to :

* confirm the understanding of the designer as to the problem, this should include a description of the clients criteria for the project, to a level of detail that completely describes the requirements.

* explain the feasibility and options of the project in order that a decision may be made as to continuation of the project.

The major problem associated with architectural briefing is the identification of who actually is the client, the person who is paying the bills, society in general, or anyone else in between. This lack of clarity leads to complexity in the development of the details of the requirements and the feasibility, options consideration and the brief development phases of the design process, due to the decisions involving trade off considerations between the different ends of the client spectrum.

Having been given approval to continue with one of the alternative options, the Brief Development phase aims to produce models that develop and explain the design of the chosen option to the client, enable contract agreements to be negotiated and construction carried out. Normally this is in the form of plans, elevations, sections and perspectives, supplemented with written documentation such as specifications of materials and bills of quantities, all of which describe the required building in great detail.

Implementation involves the construction of the project. The factor in building construction that is of most interest is the amount of resources in time, expertise, specific construction plant and materials that are required to translate the models produced in the brief development phase into reality.

The Project Evaluation phase involves the appraisal of building, to ensure its compliance with the initial brief, and to isolate any areas that require further work. A common problem noted during this phase is the complexity of the evaluation due to lack of initial finite definition during the briefing period, particularly the problems of client definition.

Using the Systems Development Life Cycle (Yourdon E, 1989) software development follows equivalent development phases, due to the similar project orientated nature of the discipline. Like architectural design, the milestone phases of the design process can be considered as Brief, Brief Development, Implementation and Project Evaluation.

The collection of data to become part of the brief, and the consideration of the feasibility and options for development are generally known in software engineering as the analysis phase. During this stage a lot more effort is put into the definition of the problem than in architectural design, normally attempting to define the problem in approximately a page of writing. This problem definition is further refined by the consideration of additional data collected and finally formulated as a series of models in the feasibility analysis representing the available options, referred to as the logical models. The models often used to describe the options are process or data flow models. This form of

modelling will be explained in greater detail later in the paper.

Brief Development, like its architectural design cousin, involves the further refinement of the initial models. Normally these are augmented with other forms of modelling including data, object and state transition, which will not be covered in detail here. The conversion of the logical models, or what the client wants, into physical models, which are what is physically able to be produced involves the inclusion into the models of the detailed knowledge of the systems analyst in relation to how computer systems operate.

The problems are completely different to architectural design in the Implementation phases of software engineering. Large expensive construction plant, multi specialist tradesmen and professionals are not required. Whereas in some cases the cost of the software may be equivalent to a building, the programmer is the main professional involved and the little additional plant is required. The product of the phase however remains the same, the conversion of the models of the product created and refined in the earlier phases are converted into reality.

Project Evaluation also has a different approach. As the end product of software engineering is the programs to operate as part of a computer system, the main point of measurement, the primary point of assessment is as to whether the computer operates as the brief required. Computers being digital, the answer is generally either yes or no. The need to ensure a satisfactory product in such an unforgiving environment has led to the development of specific tools and procedures to reduce the possibility of error.

The relationship is then confirmed, as both architecture and software engineering are problem solving, project structured, system orientated product development procedures. The development processes are analogous. The question is as to whether the better tools and techniques of one discipline can be used to advantage in the other.

Disadvantages of the architectural design methodology can be summarised as :

Problem Definition - this is difficult to describe in a finite manner, in the briefing phases of the design process due to the normal unstructured manner of data collection.

Architecture as Art - Some architectural design is performed in an environment labelled as creative art, without the advantages that a structured scientific approach can bring.

Resource Implications - Implementation is so expensive, mistakes or errors are not necessarily solved with a simple rectification due to the cost of the original incorrect construction and the additional cost of the rectification.

Evaluation - Product evaluation can only be as valid as the details initially defined.

On the other hand, the advantage of the software engineering approach can be described as :

Structure - The structured nature of the method provides a framework for the development process that reduces the opportunity for error.

Models - The development of the logical and physical models provide a method for ensuring all portions of the brief are included and the relationships between areas, functions and individual sections of the design can be readily defined.

An Example of Process Modelling

Tom Demarco developed the tools of the process modelling (Demarco T, 1978), which have now become standard tools for process analysis, (the analysis of the activities occurring within a system). Process modelling is a top down procedure, commencing with the general and moving to the particular. The modelling technique is modular, hierarchical and iterative, using diagrams to define the processes that are performed within a system. This form of modelling has become an integral part of the development and operation of of the system development life cycle.

The details of the system development life cycle, process modelling including the use of data flow diagrams and system dictionaries are included in many text books, particularly Yourdon (Yourdon E, 1989). The texts describe in detail the components of the set of data flow diagrams generally being :

* A context diagram, describing the relationship of the system to its environment. Figure 1 provides an example of a context diagram for a residence.

* a first level diagram, describing the individual components of the system, and their relationships, as indicated on figure 2 and

* lower level diagrams, describing the detailed processes are associated with the system, an example of one of the lower level diagrams is included in figure 3

In addition the texts include the structure and content of the system dictionary, generally being :

* Transform Dictionary, records the details relating to each of the processes described in the data flow diagrams. Details to be recorded include the actual process occuring, and an details relating to decisions etc performed in the process.

* Data Dictionary, recording the details of each data flow, and data elements used to make up data flows.

* Origin / Destination Dictionary, details of the external entities that are part of the environment of the system. Data included could involve, names, addresses, telephone numbers, policy directions and details, etc.

Demarco sumarised the process modelling tools as:

* It is graphic, made up mostly of diagrams.

* It is partitioned, not a single specification, but a network of connected "mini specifications"

* It is top down, presented in a hierarchical fashion with a smooth progression from the most abstract upper level to the most detailed bottom level.

* It is maintainable, a specification that can be updated to reflect change in requirement.

* It is a paper model of the system to be; the user can work with the model to perfect his vision of business operations as they will be with the new system in place. (Demarco T, 1978)

Process Modelling in Architectural Design

Having used the techniques described in such texts to produce a process model, how can it be usefully incorporated into the building process?. Demarco (Demarco T, 1978) described the products of the process model as:

* A communicational tool. Since user and analyst have longstanding history of failure to communicate, it is essential that their discussions be conducted over some workable negotiating instrument, something to point to as they labor to reach a common understanding.....

* A framework for the specification. The model declares the component pieces of the system, and the pieces of those pieces, all the way down to the bottom....

* A starting point for design. To the extent that the model is the most eloquent statement of requirement, it has a strong shaping influence on work of the Design Phase.

If these points are expanded with direct relevance to architectural design, the process modelling technique can be used as described below

A Communicational Tool.

There are two parts to the consideration of process modelling as a tool for communication within the architectural design process, firstly the data flow diagrams provide a complete structured simulation of a building, and if required its construction process. It can be extended to whatever level is required to assist in the communication of the design to other professionals and other designers within the team. Its graphic format is simpler to grasp than printed pages describing the same data, and avoids misunderstandings from terminology and badly written text.

Secondly, the system dictionary becomes a definitive catalogue of the information that will be handled within the design process for the building. The system dictionary provides two primary services in communication.

* Sole Reference : The system dictionary will be the sole reference for definitions within the model, and will contain a rigorous definition of each component of the system. A single point of defintion of facts within the design improves communication and reduces communication misunderstandings.

* Ease of Reference : The system dictionary will be organised for ease of reference and maintenance, with consequent improvements in communication between members of the design team. (Teague L C and Pidgeon C W, 1985).
Efficiencies will be obtained over time as many data elements would be similar from building project to another, such as the data element *people*, a design office, or design team, could easily build up libraries of commonly used data elements and processes

Enhanced communication due to the design data stored in a structured manner and displayed in graphic forms, especially within the earlier phases of the design process, will assist in improving problem definition by providing a more succinct and definitive simulation of the building. This will assist is solving the disadvantage of Problem Definition, noted earlier in this paper.

A Framework for the Specification.

Specification in the context of this section relates to the specification of the design as a whole, and would include all drawings, details, etc, not just the written words of the specification document.

The fact that the data flow diagrams are the result of a disciplined structured analysis of the building problem, the end product is an evolutionary result derived from the parameters of the problem. As

such it can become, within itself, a project customised tool for the classification of sections of the building for specifications, standardised details, window and door schedules, etc, ensuring that all details are included and confirming the relationships between the individual elements of the construction. The partition of a problem into modules allows ease of management both in terms of developing a concept and as a means of partitioning the sections of work between professionals and within the design team.

Data dictionary information is used to provide a basis for the written details in specification, details of materials, plant and equipment, and particular details of parameters for building services definition would all be included.

The structured approach would assist those architects who prefer the architectural philosophy of creative art, without taking away the artistic input. This would solve the second of the disadvantages listed, that of Architecture As Art.

The Starting point for Brief Development.

The evolutionary result of the process model ensures that the result is a simulation of the required end product. The output of the briefing phase, it naturally becomes the input of the brief development, in other words its commencement point.

Conclusions

Process modelling can be used as an aide within the architectural design process in order to provide better communication of the elements of the design, as a framework for the specification and as the starting point for the brief development phase. As it has no effect upon the actual design, merely the process of achieving the design, it could be used within any architectural design philosophy to effect.

If process modelling can be seen to be a successful addition to the toolbox of the architect, it is possible that other techniques and tools from software engineering and other disciplines can be used to improve the end products of the architectural design process by providing alternate methods of simulating the building its functions and operations. Other modelling techniques could include data modelling, object modelling, the use of state transition diagrams to describe the time dependent functions and facilities within a building

References

Broadbent G, 1978, "Systems and environmental design", in *EDRA2 1970*

Brooksheare J G, "Computer Science", Benjamin Cummins, 1985

Cowdroy R, 1990, "Architectural design teaching", in *Architecture and design in harmony*, J Plume, University of NSW

Demarco T, 1978, "Structured Analysis and System Specification", in *Classic of Software Engineering*, Yourdon E, Prentice Hall International, 1979

Heimsath C, "Behavioural architecture", McGraw Hill, 1977

Kingsland A, 1990, "Design as a total integration activity", in *Architecture and design in harmony*, J Plume, University of NSW

Lang J, 1990, "Architectural design theory and education", in *Architecture and design in harmony*, J Plume, University of NSW

Macquarie Dictionary, 2nd Revised Edition, Macquarie Library, 1991

Teague L C, Pidgeon C W, "Structured analysis methods for computer information systems", Science Research Associates, 1985

Vitruvius, "The Ten Books of Architecture", Translated by Morgan M H, Dover, 1914.

Yourdon E, "Modern Structured Analysis", Prentice Hall International, 1989

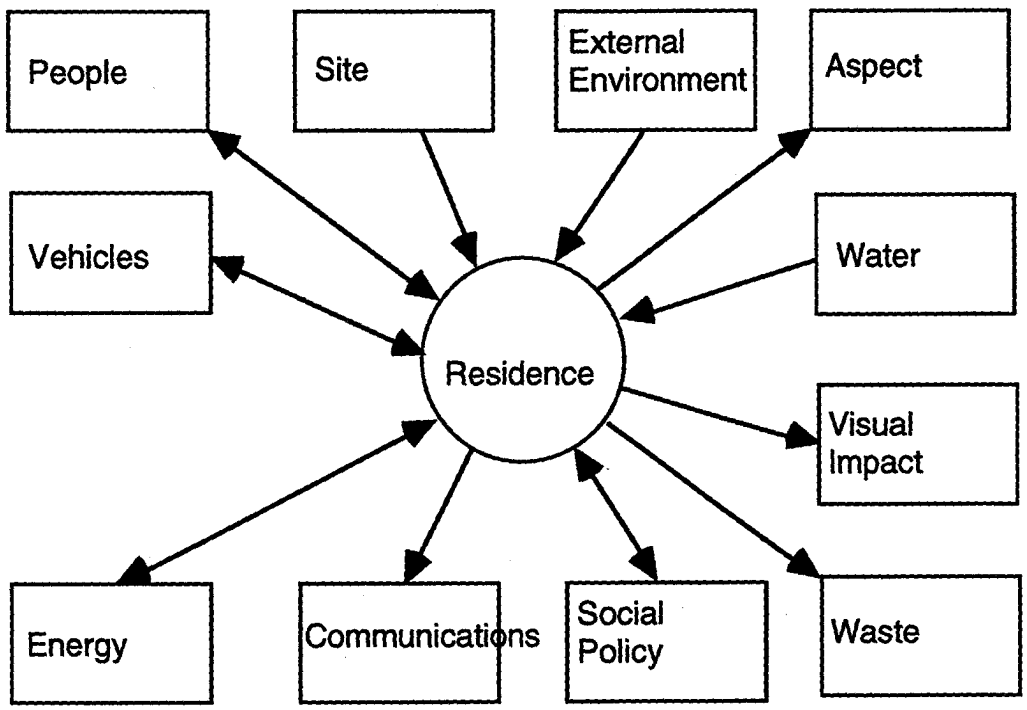

Figure 1 Context Diagram of a Residence

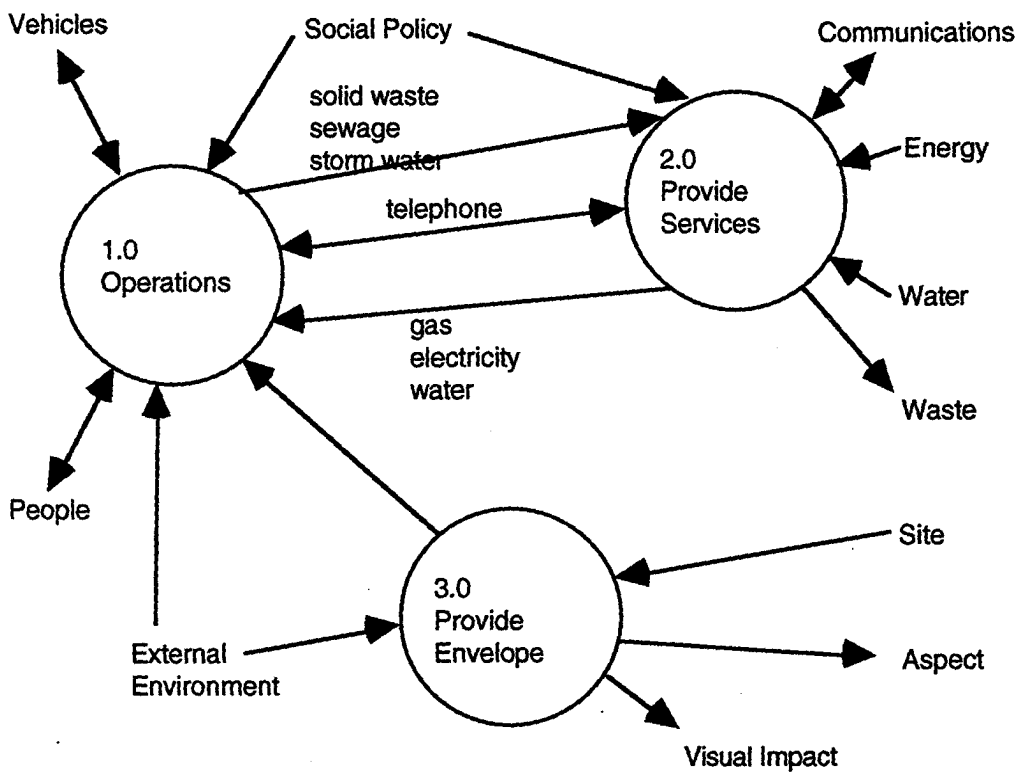

Figure 2 First Level Data Flow Diagram for a Residence

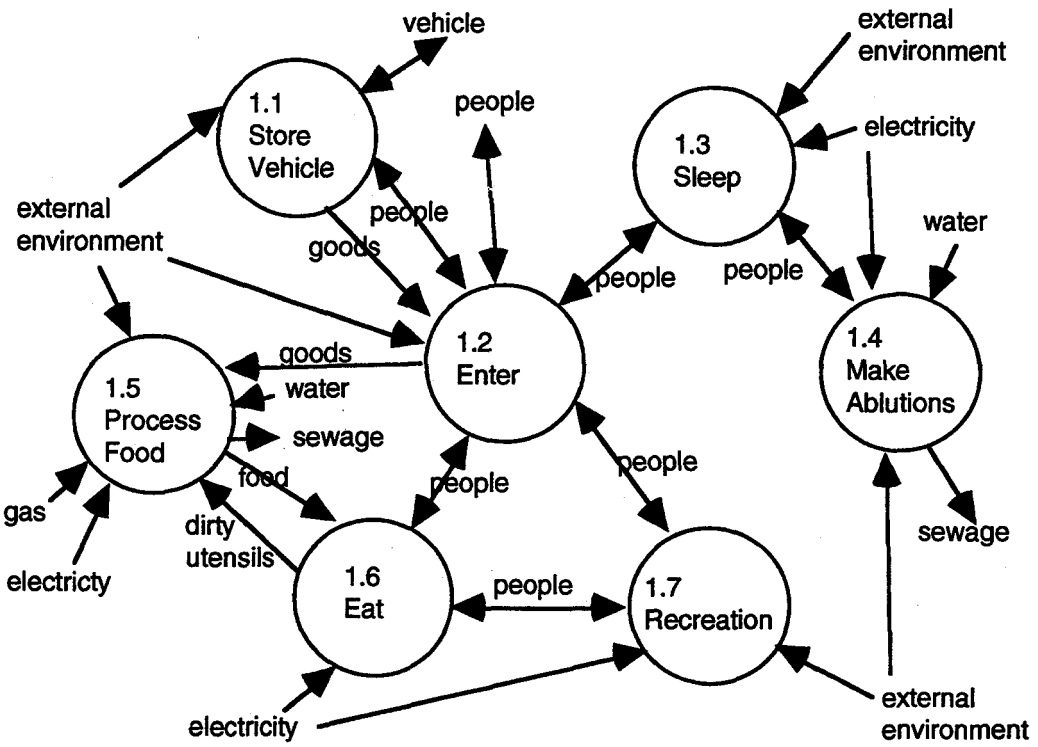

Figure 3 Level 1 Data Flow Diagram fo a Residence