Conclusions

"There is no absolute knowledge. And those who claim it, whether they are scientists or dogmatists, open the door to tragedy. All information is imperfect. We have to treat it with humility....That is the human condition".

Jacob Bronowski, The Ascent of Man, Knowledge or Certainty.

6.0. Software Engineering, Systems Theory and Connectionism

The abstract and intangible software domain is gradually coming within our control as the software engineering "revolution" attempts to find broad based solutions in an attempt to come to grips with the software crisis. Over the past two decades numerous paradigms and management techniques have been proposed but none of them alone is the magical panacea or the silver bullet that researchers have been looking for. It seems more evident now that a combination of appropriate options from different paradigms together would provide an acceptable framework for planning, managing and controlling a software project effectively.

Although, in any emerging discipline, solutions to problems emerge and evolve from ideas generated through intra-disciplinary research, it seems rather natural that ideas and techniques borrowed from well established disciplines may help solve problems effectively in
relatively new and growing areas: an inter-disciplinary research approach.

From within the discipline of software engineering, one of the fundamentally new paradigms that has emerged during the past decade suggests an entirely new way of thinking about the development of software systems - in terms of their component objects rather than in terms of their functions. Researchers propose that object-oriented techniques might help us move more rapidly towards the achievement of the fundamental goals of software engineering: reuse, reliability, economy and user satisfaction.

Systems engineering concepts have been sufficiently well established and have been applied in the past to diverse problems such as hardware systems, economic analysis and transportation planning. It appears to be plausible therefore to attempt to apply such techniques to software engineering problems which prove difficult to solve otherwise.

From the realms of artificial intelligence, connectionism has emerged as the new hope of the discipline for the solution of problems which have hitherto seemed intractable. Neural networks are extremely efficient in handling fuzzy variables and providing decision support. Their strength lies in their ability to perform associative recall, generalization based on incomplete information, and in their graceful degradation.

This thesis has attempted to integrate some of the ideas from system theory and connectionism to provide solutions and deeper insight into front end software engineering problems such as requirements analysis and design (based on the object-oriented approach) and software project resource estimation and control.
Explicit models for circuit/system simulation by system dynamics methodology, which could be used to form part of an integrated software engineering model database for computer-aided-design of systems (e.g. the Hopfield neural network and its VLSI circuit realization), have also been researched.

6.1. Object-oriented Analysis and Design using ISM

It is now well established that requirements analysis should form a major part of the software development effort, but eliciting information from users still proves difficult in practice. Interpretive Structural Modelling is one such tool which has been shown to prove effective in eliciting relationships such as dependencies amongst modules in object-oriented systems design.

Chapter 2 specifically addresses the question of object-oriented analysis and design for a numeric-control software system. A minimal edge digraph, which is the equivalent of the object-oriented Booch diagram, is derived from the preliminary software scope to bring out a "first cut" structured chart representation of dependencies amongst modules that would be components of the software system. Each module is then described in greater detail and sub-module inter-dependencies derived, thereby exemplifying the iterative approach to design through subsequent refinement.

Such a procedure may then be applied iteratively through subsequent steps of refinement on each object module to gradually introduce design details in an attempt to reduce abstraction until no further refinement is required. Appendix III discusses essential features of an ISM based software modelling system which has been developed to demonstrate the possibility of automating the aforesaid procedures.
ISM has the advantage of yielding a minimal edge digraph representation of the software structure (as opposed to Booch diagrams where transitive relationships are explicit) while providing a structured means to approach a graphical representation of the system which till now has been done normally through group consensus. This minimal structure provides and facilitates the consideration of a "unique" auditable track from specifications to solution and back. Apart from this, the iterative refinement procedure allows design details to be gradually and systematically introduced into the representation thereby supporting the concept of information hiding in a top-down design approach.

The automated refinement and structuring of elicited information is a contribution towards the development of tools for "intelligent" support of the project development life-cycle and has potential applications in the current fourth generation life-cycle paradigms.

6.2. Software Project Planning, System Dynamics and Neural Networks

Software project resource estimation, planning and control has also been a widely researched area of software engineering. This thesis demonstrates that system dynamics can provide deep insight into the dynamics of the resource estimation and planning process. The development of a comprehensive model based on the Putnam resource estimation technique while employing an embedded feed-forward neural network to provide critical decision support to the management where estimation of variables remains "fuzzy", puts the resource estimation and planning process into the "intelligent" systems arena.

A comprehensive system dynamics simulation model of the software project planning process has been developed to include a static sub-
model for organizational productivity estimation (using historical data), software sizing (with the help of delphi polling of experts) and crucial management decisions. Embedded within the management sub-model is a neural network trained to provide support to the management on the selection of a suitable manpower buildup index for the project in question. A decision support system helps identify possible solutions to avoid violation of management-set bounds such as cost, delivery time and the risk of schedule slippage, and peak manpower. With the help of these tools a final planning point is decided upon. The dynamic sub-model simulates the time based progression of the expended development and life-cycle effort, the developed source lines of code, expected manpower loading profiles and the expected defect rate. Simulations are carried out based on Rayleigh-Norden models for project development.

A complete simulation software package has been developed, and a typical simulation run described in Appendix IV.

6.3. Circuit Simulation Models for a Software Engineering Database

However, as proposed at the commencement of the thesis, there is perhaps the need for integrated software engineering databases which would be based on AI techniques while embodying domain specific process/design simulation models to support development teams during the design phase for computer-aided analysis and design of systems of the field in question.

System dynamics has proved to be a powerful tool to provide insight into the dynamic simulation of systems, and causal loop relationships between component system variables bring increased insight into the dependencies of different variables on each other which are not obvious otherwise.
Chapter 4 brings out the detailed correspondence between the physical systems modelling approach and system dynamics simulation. Detailed system dynamics simulation models for the branch, chord, branch-chord and state model formulation approaches of physical system theory have been developed. Neural networks have been carried through the investigations as examples of the modelling procedures. Apart from this, models for non-linear systems and degenerate systems have also been developed, for system dynamics simulation.

On the one hand, the state model has been shown to provide us with a basis for the identification of rate and level variables of the system dynamics simulation model. On the other, it has been discussed that by taking a series of "test bench" measurements, one can make an empirical fit on the data and thus identify the system model as an object in its own right in terms of terminal characteristics. This model can then be used to formulate the system dynamics flow diagram and accompanying DYNAMO equations directly, thereby totally obviating the need for any matrix inversions in formulating the system dynamics model for simulation.

6.4. Sensitivity Studies of Hopfield Neural Networks

Carrying the simulation ideas a step forward, numerical simulation studies of neural networks and their sensitivity studies form the concluding platform of the thesis. The state models for voltages and sensitivities for two and three neuron Hopfield networks with inhibitory feedback have been simulated to study the time dynamics of this class of networks. A laboratory prototype was built to simulate the state model that describes the time dynamics.

The case for minimal sensitivity design of this class of networks has also been investigated. Using the steepest decent optimization
technique with a squared sensitivity function, unconstrained gradient descent has been simulated to find optimal parameter values for the amplifier input resistance and capacitance, the neuronal gain scale factor and the inhibitory feedback conductance \( T \), which would yield a minimal steady state sensitivity design. A neuronal state evolution/sensitivity state evolution simulator has been developed. An enhanced version of the software includes automated support for 2-neuron minimal sensitivity design as discussed in Appendix VI.

The importance of the minimal sensitivity design through the system dynamics simulation approach, lies in the development of VLSI implementations of such networks and, as discussed, the ideas developed can be easily extended to the \( N \)-neuron case as required, without resorting to any analytical solutions.

6.5. Future Directions

Applications of ISM to the object-oriented design process and the connectionist based system dynamics project planning model are contributions to the fourth generation techniques (4 GT's) which are becoming increasingly powerful as researchers attempt to make software systems more "intelligent". Circuit/system simulation models provide some ideas about the way in which an integrated software engineering model database for computer-aided-analysis and design of systems could be developed to lend support to the developers of CAD based simulation software and thereby indirectly support the software/hardware engineering life-cycle. Automated sensitivity analysis and design tools could also be included in the database or even be included in a library of modules which could subsequently be made part of a larger software system.
The work embodied in this thesis would be of interest to researchers in the areas of software engineering, system dynamics, and connectionism, and can be extended further in various directions.

The investigations on the use of interpretive structural modelling in OOA/OOD could be extended to demonstrate the utility of ISM to help automatically structure entity relationship diagrams (ERD's), state transition diagrams (STD's) and other structured chart representations used in software engineering. ISM could serve as a powerful knowledge acquisition tool at the front end, and be subsequently utilized to structure the elicited information.

Although the development of simulation models discussed in the preceding chapters relate to the time domain simulation of circuits and systems, the concepts may be readily extended to handle large and complex physical systems (both real and conceptual) in general, and simulation models developed.

Sensitivity studies can be further extended to address the problem of minimization of sensitivities over the entire evolution trajectory between the initial state and the steady state. Inclusion of a sensitivity term in the energy function which governs the time dynamics of the Hopfield network can also be investigated as a means to minimize the sensitivity of the states to specific parameter changes.