Abstract

The discipline of software engineering has gradually matured as it comes to grips with the software crisis, and the organization, integration and management of variously proposed solutions to encountered problems has resulted in an assortment of process models or software development methodologies. The engineering of software refers to the selection, refinement and integration of these solution techniques and methods into a process model that is appropriate for the problem, application, and environment to which the methodology is applied, i.e., project characteristics help us decide how to engineer the process to achieve appropriate product characteristics. Software engineering may thus be seen to encompass three key elements - methods, tools, and procedures - that enable the manager to control the process of software development and provide the practitioner with a foundation for building high quality software in a productive manner.

Both hardware and software engineering fall under the broader umbrella of systems engineering. Systems engineering concepts have been in use for decades for the design of hardware systems and have reached a state of relative maturity. Hardware design techniques are now relatively well established, and as manufacturing methods continually improve, reliability has become a realistic expectation rather than a modest hope.

In contrast, engineering techniques for software production are less well established and have only recently gained widespread accep-
tance. Numerous ideas have been borrowed from the general systems area, but there still remains considerable scope for the application of systems engineering concepts to various problems encountered in software development.

The primary focus of this thesis is on "front-end" software engineering tools, specifically for software engineering requirements analysis and design, and for software project resource estimation, management and control. Borrowing ideas such as Interpretive Structural Modelling from systems engineering, potential applications in the area of software requirements analysis and software design are investigated. The application of Interpretive Structural Modelling (which incorporates a formal knowledge acquisition methodology) to the formalization of the process of software requirements analysis, using the object-oriented approach as a vehicle has been investigated. The potential of ISM to support an iterative top-down refinement of the object-oriented design procedure is demonstrated through an example of the preliminary design of a software system for numerical control machines.

System Dynamics is another powerful systems engineering simulation tool which gives substantial insight into the dynamics of the system at hand and the use of this tool in conjunction with an "embedded" neural network has been researched for its potential application to "intelligent" software project resource estimation and control, a study based on the Putnam Resource estimation model. The dynamics inherent in software project resource estimation, management and control are brought out in detail, and the capabilities of system dynamics and connectionist architectures to provide software project managers with comprehensive project planning support has been demonstrated.
System Dynamics also has potential applications in linear and non-linear circuit simulation, which could have considerably important implications from the software engineering point of view for the design and development of integrated CAD based tools for specific application domains. The area of system simulation for both non-degenerate and degenerate systems has been focussed on, with emphasis on how we can bring the concepts of linear physical system theory and system dynamics modelling to bear upon each other. Non-linear systems are also discussed and comprehensive system dynamics simulation models developed.

Continuing along the same lines, sensitivity studies appear to have important applications in the design of a class of networks that falls under the broader umbrella of connectionism, i.e., neural networks. The state space dynamics and time domain evolution of sensitivity state vectors of recurrent networks called Hopfield networks have been studied using system dynamics as a simulation tool. The case for minimal sensitivity design of these networks has also been investigated. This thesis considers a classic circuit simulation example in the form of the Hopfield neural network and extensive simulation results are presented demonstrating the potential of system dynamics towards developing a better intuitive understanding of the dynamics inherent in strongly fed-back networks. Sensitivity studies of network states with respect to circuit parameters have been carried out and their potential application to minimal sensitivity design through optimization, of this class of networks discussed.