A System Dynamics Model of the Integration of New Technologies for Ship Systems

by

Pushkar Damle

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Industrial and Systems Engineering

APPROVED:

K. P. Triantis, Ph.D., Chairman

H. A. Kurstedt, Ph.D. **C. P. Koelling, Ph.D.**

September 16th 2003

Blacksburg, VA

Key words: System dynamics modeling, technology integration, cost overruns, dynamic behavior, simulation, affordability

Abstract

A System Dynamic Model for the Integration of New Technologies for Ship Systems

Pushkar H Damle

System dynamics has been used to better understand the dynamics within complex natural and social systems. This understanding enables us to make decisions and define strategies that help to resolve the problematic behaviors associated within these systems. For example within an operating environment such as the US Navy, decisions taken today can have long lasting impact on system performance. The Navy has experienced large cost overruns during the new technology implementation process on ship systems that can also have an impact on total life cycle performance. The integration phase of the implementation process represents most of the cost overruns experienced in the overall new technology life cycle (development, integration, and operation/support/disposal). We have observed a general concern that there is a lack of understanding for the dynamic behavior of those processes which comprise the integration phase, among ship-builders and planners. One of the goals of our research effort has been to better understand the dynamic behavior of the new technology integration processes, using a dynamic modeling technique known as System Dynamics.

Our approach has also been to provide a comprehensive knowledge elicitation process in which members from the shipbuilding industry, the US Navy, and the Virginia Tech System Performance Laboratory take part in group model building exercises. The system dynamics model that is developed in this manner is based on data obtained from the experts. An investigation of these dynamics yields a dominant cost behavior that characterizes the technology integration processes. This behavior is S-shaped growth.

The following two dynamic hypotheses relative to lifecycle cost and performance of the inserted new technology were confirmed: (1) For the current structure of the model we observe the more the complexity of the new technology, the less affordable a technology

becomes; (2) Integration of immature (less developed) technologies is associated with higher costs. Another interesting insight is that cost is very sensitive to the material procurement.

Future research can be addressed to a more detailed level of abstraction for various activities included in the technology integration phase, such as testing and evaluation, cost of rework and risks associated with inadequate testing etc. This will add to our evolving understanding of the behavior of individual activities in the technology integration process.

Acknowledgements

I would like to express my utmost gratitude for my advisor, Dr. Konstantinos P. Triantis, for his faith in me during my entire research. In addition to his continuous motivation and support, he always supported encouraged innovative ideas that led to a very satisfying research experience.

I would also like to express my sincere thanks to Dr. Harold Kurstedt and Dr. Charles P Koelling who offered useful insight and advice as my research committee members.

I would like to thank Robert Schatzel and Kirt Kerr (Northrop Grumman Newport News), Irv Chewning, Jimmy Moy, Nicole Ray, Anne Hasson (Naval Sea Command Cost Estimating Group) and George Keller (Anteon Corp.) for their invaluable contributions in the modeling process. I would also like to thank Katherine Drew and the Office of Naval Research for their funding support of this research.

I would like to thank my research colleagues Dr. Betsy Tretola, Wassana Siangdung, John Scott and Pavinder Monga who helped me with the research through brainstorming discussions.

My family and friends have been exteremly supportive through these many years. Their love and support has seen me through both bad times and good times and they contribute hugely in whatever I have accomplished till today.

Pushkar Damle

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Technology, which is an integral part of our day-to-day life, always keeps changing. New technologies are introduced in organizations and in systems for the purpose of making our life better and easier, or in other words to achieve better performance. But not all changes are always welcome. Changes in technology are mainly resisted by its users. This is partly because of the learning that has to be accomplished to use the new technology and also because of the concern with respect to effectiveness of the new technology.

Keeping aside the mental resistance when introducing a new technology, it is very important to know the real "value" or the effectiveness[1] of the new technology that is being introduced. If the technology change is sufficiently large and spread over a long span of time, one needs to evaluate cost and performance trade-offs.

Introducing new technology has some underlying issues that relate to the impact on required efforts, training and the additional downtime incurred primarily due to unaccounted risk and the complexity of the new technology. The lack of training, risk and complexity lead an increase in the life cycle cost associated with the introduction of a new technology. This increase in cost is usually tried to balance with the reduction in performance. Furthermore, the lack of upfront planning in the systems engineering process that is responsible for the introduction of new technologies also results typically in both cost overrun and performance degradation. This defeats the fundamental purpose of introduction of new technologies into existing systems, i.e., to achieve cost reductions and performance improvements.

The process of introducing/implementing new technologies has been introduced by Vaneman and Triantis (1999). This includes three core activities viz. development,

[1] Effectiveness is the measure of usefulness of the new technology with respect to its cost and benefits.

integration and operations support and disposal. Monga and Triantis (2002) describe the technology development process. Scott (2002) presents the operations, support and disposal process. This research is concerned with the technology integration process.

The integration process of new technologies on ship systems is a process spanning over the period of two years and involves large amounts of funds and man-effort. Among the various core activities of the new technology integration lifecycle, the integration process receives largest percentage of funding. Most of the cost overruns are observed during this phase. This phase is linked with the technology development phase, which occurs before technology integration and the operations, support and disposal phase, which occurs after technology integration.

1.2 Research Objective

Technology integration process contains various activities or phases that are interlinked with each other. Interaction among these activities generates dynamic behavior within the technology integration process. The dynamic behavior of the process can be modeled using the system dynamics approach. The system dynamics framework describes the structure that is present within the system that leads to the observed behavior.

1.3 Problem Definition

The integration of new technologies in existing platforms leads to cost overruns and performance degradation the impact of which needs to be understood at the outset before any critical decisions are made. The system dynamics model developed in this research addresses this problem throughout the research.

In order to address the above problem, the primary objectives of the research are as follows:

- To study and analyze the dynamic behavior that characterizes the technology integration process
- Observe how it pertains to cost estimation and affordability science

- To build a system dynamics cost and performance framework model for the technology integration process

The primary concern of this research is to track both cost and performance of the process. These are the key performance issues that are the focus of the system dynamics modeling process in this research. In traditional model building process, in general, is formalized as Problem – Plan – Action. But in the system dynamics modeling approach, action and problem are interlinked with a feedback forming a closed loop. Thus, the modeling process becomes a continuous process. Also, the learning or the feedback from the results leads the problem is being reformulated and a new plan is implemented to improve the results. Thus, in System Dynamics modeling there is a continuous learning process.

Since the primary concern of this research is to understand cost overrun and performance degradation behavior, the system dynamics framework is intended to identify the drivers for cost overrun and performance degradation. The objective of the model building process is to identify the factors in the system that cause unwanted and unforeseen behavior.

The modeling framework would also throw light on other factors that cause the behavior of the system. It could bring up some issues that one would never think of. So one of the outcomes of this research is to discover unintentional consequences that determine the behavior of the system.

1.4 Research Motivation

The primary motivation for this research is to better understand the technology integration process within the scope of introducing a new technology, from a system-engineering point of view. It should be noted that in the literature not much work has been done in the area of system dynamics modeling of the technology integration process.

The other motivating factor is the opportunity for real world implementation and application of the model developed during this research. US Navy is the end user of the outputs of this research and the model is going to be a helpful tool that will assist them to make decisions about the affordability and value of the new technologies that are introduced.

The Office of Naval Research (ONR) has funded this research. According to experts from the Office of Naval Research, the shipbuilder and the government, the Navy has repeatedly experienced cost and schedule overruns in large new technology introductions on ship systems. It is very crucial for the Navy to keep upgrading obsolete technologies and maintain highest level of Mission Preparedness and Condition Readiness (See Glossary – Appendix A). In an attempt to better manage the technology introduction process and to reduce cost and schedule overruns, the system dynamics modeling approach should help to find factors that drive cost and performance. So the system engineering process can be re-engineered.

1.5 Overview of Methodology

System dynamics modeling approach is used in the development of the modeling framework for this thesis. Sterman (2000) provides the guidelines and steps used during the modeling process. The approach for this thesis is based on these guidelines. The following description briefly explains the methodology followed:

1.5.1 Problem Articulation

This is the first and important step of modeling process that describes the problem exactly. Various important variables that need to be considered are identified. The time horizon during which the dynamic behavior of system manifests itself is determined. Reference modes – graphs of behavior over time – for key variables are determined and system boundaries are also defined.

1.5.2 Formulation of Dynamic Hypothesis

A theory called a dynamic hypothesis is defined that describes the problematic behavior of the system. During this stage the system is mapped using various diagrams such as model boundary diagrams, subsystem diagrams, causal loop diagrams and stock and flow diagrams.

1.5.3 Formulation of Simulation Model

The simulation model is built based on the stock and flow structure developed during the second phase. For this research, computer software, which is specifically developed for building system dynamics model, is used to build the model. The software used for building the model and running the simulation is VENSIM, which is developed by Ventana Systems specifically for system dynamics model building.

1.5.4 Testing, Verification and Validation

The model is tested for its robustness in extreme conditions. It is checked as to whether it can replicate the real world behavior. Sensitivity analysis is also carried out during this stage.

1.5.5 Policy Design and Evaluation

From the results and learning from the model, the cost and performance driving factor are identified. Decisions and policies are made based on these results that will control these factors and lead to reduction in cost and improvement in performance.

1.6 Organization of Thesis

The thesis is organized in five chapters with the first chapter as in introduction to the research. Chapter 2 is a literature review that presents theories in system dynamics and new technology implementation process. Chapter 3 explains the actual system dynamics model that has been developed. The results obtained from model, model testing and the lessons learned from the model are organized into Chapter 4. Chapter 5 concludes the document with an overview of the results and suggested policy changes of the technology integration process and future research issues are discussed.

Chapter 2: Literature Review

2.1 Introduction

> *"If someone were to analyze current notions and fashionable catchwords, he would find "systems" high on the list. The concept has pervaded all fields of science and penetrated into popular thinking, jargon and mass media."*
> *(Bertalanffy, 1973, pp. 3)*

The economic boom and industrial development after World War II led to the necessity of "Systems Thinking" and "Systems Approach". The new technologies that were being developed involved interaction between heterogeneous technologies such as mechanical, electrical, chemical and physical. In addition to these interactions, the relationship between man and machine has continued to play an important role in the development of new technologies. Various financial, economical, social and political problems started dominating this development.

All of these factors, due to their interaction with each other, created a complex set, or a system. This led to an approach where the system is seen as whole rather than just considering its unit components. This approach is a "systems approach". The systems approach aims to integrate various natural and social sciences within its frameworks. This systems thinking and systems approach are the by products of General Systems Theory. Bertalanffy (1973) describes the origins of the general systems theory and then how it was evolved and applied to various engineering and economic disciplines.

2.2 General System Theory

The term "General Systems Theory" was introduced by Ludwig von Bertalanffy (1973) when he was carrying out his biological research on living organisms. He felt a necessity to study the organisms not only as isolated parts but also to account for their interaction when studied within the whole. He proposed that this “organism-as-a-whole” (Bertalanffy, 1973) approach can be used in various other fields including engineering, social, physical and science fields.

2.3 <u>Industrial Dynamics</u>

The concepts of general systems theory were applied to a number of fields. The approach of systems thinking was seen widely applicable in industrial engineering and management. Industrial Dynamics was proposed by Jay W. Forrester (1958) when he was studying management problems in corporate settings. He was especially concerned with the problem of fluctuations in factory production that was inconsistent with corporate growth and declining market share.

He developed his theory of Industrial Dynamics. In his paper, “Industrial Dynamics: A Major Breakthrough for Decision Makers” (Forrester, 1958), he states that the success of an industrial company depends on the interaction between the flows of information, materials, money, manpower and capital equipment. The interactions among these factors where one amplifies the behavior of the other and causes fluctuations, forms a basis for understanding the system structure and for anticipating the outcome of decisions, policies and investment choices.

In his book Industrial Dynamics (Forrester 1961), he describes the following industrial dynamics approach so as to design more effective industrial and economic systems.

2. Identify a problem.
3. Isolate the factors that appear to interact and to create the observed symptoms.
4. Trace the cause-and-effect information feedback loops that link decisions to actions that result in information changes and to new decisions.
5. Formulate acceptable formal decision policies that describe how decisions result from the available information streams.

6. Construct a mathematical model of the decision policies, information sources, and interactions among the system components.
7. Generate the behavior through time of the system as described by the model and compare results against all pertinent available knowledge about the actual system.
8. Revise the model until it is acceptable as a representation of the actual system.
9. Redesign, within the model, the organizational relationships and policies that can be altered in the actual system to find the changes that potentially will improve system behavior.
10. Alter the real system in the directions that the model experimentation has shown so as to lead to improved performance.

2.4 System Dynamics

Industrial dynamics was used to study and solve problems related with management in industry. But then it was used to address problems in various disciplines such as urban planning, economics, traffic engineering and medicine. The term Industrial Dynamics was transformed into a new more general term entitled "System Dynamics".

2.4.1 Fundamentals of System Dynamics

> *"System Dynamics is a methodology for understanding certain kinds of complex problems." (Richardson and Pugh, 1981, pp. 1)*

By saying "certain kinds of complex problems", Richardson and Pugh refer to complex problems as in feedback-rich environment. This environment, or a system, incorporates various components within itself that interact with each other and generate feedbacks that make it complex. In addition to that there are some factors outside the system that also affect the system in some way.

System dynamics is a policy-based methodology that evaluates the effect of policy changes on a system. For any system, the decisions that we make affect the behavior of the system. System dynamics tries to find out the factors that cause the characteristic behavior of that system. Then how the system reacts to the changes associated with these factors is observed. Based on these reactions, the changes in policy are then suggested. Thus the main purpose of system dynamics is to better understand the complex and dynamic systems and suggest the changes in the decision-making rules so as to improve the performance. System dynamics is fundamentally used to understand policy decisions and feedbacks.

Policy Decisions

Policy is a set of ideas that express the strategies that are based on some information. A policy decision is the part of decision-making process in which the decisions are based on the ideas conveyed by the policies. Forrester (1958) defines policy as a "rule that states how the day-by-day operating decisions are made". Policy decision is an important aspect of system dynamics. System Dynamics is a policy assessment methodology that re-designs policies that will aim to remove existing problem in the system.

Feedback

As the name says, there is something that is fed back. This "something" is nothing but the information. Consider one of the elements of the system that has some effect on the other element of that system. The effect meaning the change in first element causes some sort of change in the other element. Now suppose that this change in second element effects back on the first element, then that constitutes "feedback".

This feedback generates disequilibrium and hence dynamics in the system. The disequilibrium is caused because of the change in one element causes change in the other element, which, in turn, leads to the change in first element that again causes the change in second element. The feedbacks can be of two types, viz. positive or self-reinforcing and negative or self-correcting.

In the former, the changes are amplified in the system. While in latter the feedback tries to bring the system into equilibrium by opposing the change taking place in the system. It is little easier to predict the behavior of the system with a two-element feedback structure. But for the system with more elements interacting with each other, it is not possible to determine the system behavior with analytical methods. This is due to the fact that there is a combination of positive and negative feedbacks, in addition to the fact that each loop can dominate at different points in time, which adds to the system complexity.

Simulation

The conceptual modes of the system are usually too complex to analyze analytically. Also conceptual models are usually tested with feedback from the real world, which occurs very slowly. However, simulation provides the decision-makers with a tool by which they can imitate the system and can observe and analyze the effects of decisions on the system. The major advantage of simulation is that the "learning" occurs very fast. The non-linearity present in the real system adds to the complexity of the simulation models. Therefore, computers are used carry out the necessary mathematical calculations and facilitate the design of enhanced decision support systems.

2.4.2 Causal Loop Diagrams

This is an important system dynamics tool that captures the feedback structure of the system. A Causal Loop Diagram (CLD) is used to map the cause-effect relationship between different variables within the system. The two variables are linked with an arrow with one of the two states of polarity, positive (+) or negative (-) (Figure 2.1).

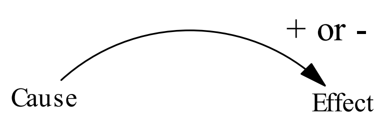


Figure 2-1: Cause and Effect Relationship

The arrow starts from the "cause" variable and goes into the "effect" variable. The positive polarity of the linkage denotes that the increase (decrease) in cause variable will lead to the increase (decrease) in effect variable, all else being equal. Thus both variables move in the same direction. On the other hand, the negative polarity designates that the increase (decrease) in cause variable will lead to decrease (increase) in effect variable, all else being equal (Sterman 2000). Thus, both variables move in the opposite direction.

Positive Causal Linkage

Figure 2-2: Positive Causal Linkage

Figure 2.2 shows that all else being equal, if X increases (decreases) then Y increases (decreases) above (below) what it would have been.

Negative Causal Linkage

Figure 2-3: Negative Casual Linkage

Figure 2.3 shows that all else being equal, if X increases (decreases) then Y decreases (increases) below (above) what it would have been. The causal link polarity is, sometimes, represented by “S” and “O”, where S stands for “Same” and O stands for “Opposite”. But it has the same meaning as + and – linkages respectively. Also, the positive linkages may be represented with blue arrows and negative with red arrows.

The main advantage of the causal loop diagram is that it quickly captures the hypotheses concerning the causes of dynamics. The whole system is represented as a series of linkages and feedback loops. This is the first stage of the modeling approach where the whole system is qualitatively represented. It forms the basis for the next modeling step, which is the quantitative description of the model provided by the stock and flow diagram.

2.4.3 System Dynamic Behaviors

Fundamental Modes of Behaviors

The feedback structure in a system gives rise to three different fundamental behaviors. They are generated because of the positive and negative feedbacks within the system and also due to the delays.

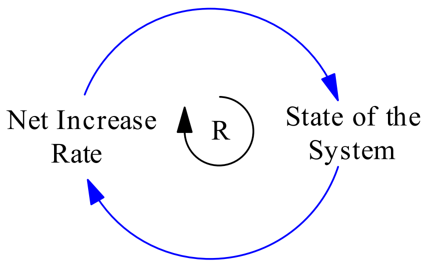
Exponential Growth: This behavior occurs due to positive, self-reinforcing feedback. The change in one quantity within the system causes positive change in the other quantity. The change in the other quantity feeds back and causes, again, a positive change in the first quantity. Thus the positive effect is reinforced. As represented in Figure 2.4 (a), the reinforcing loop is represented by "R". "Net Increase Rate" raises the "State of the System" and Increase in the "State of the System" increases "Net Increase Rate". The example of exponential growth is the feedback loop between the "Net Birth Rate" and the "Population". As the net birth rate increases, population increases and as population increases, the net birth rate increases, leading the feedback loop to exponential growth.

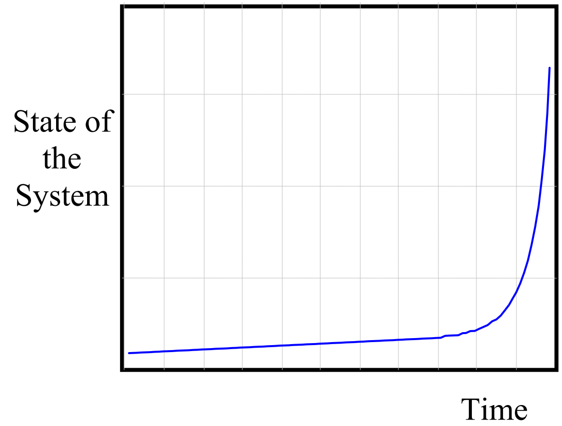
Goal Seeking: This behavior occurs due to a negative, self-balancing loop represented in Figure 2.4 (b) by the letter "B". State of the System is compared with the goal or the desired state of the system. Depending on the discrepancy, corrective action is taken. Corrective action is more if the discrepancy is more. Corrective action takes the state of the system towards the desired state. Again the state of the system is compared with the desired state and depending upon discrepancy, a corrective action is taken. Thus, the structure tries to take the system towards desired state. An example of goal seeking behavior is cooling of coffee to the room temperature. The state of the system is the current temperature of coffee. The goal or the desired state is the room temperature. Depending upon the temperature difference, heat flows out of coffee reducing its temperature. Finally, it reaches the temperature of the room.

Oscillation: This behavior is observed when there is a delay in the negative feedback loop. It is represented in figure 2.4 (c). The loop is similar to the goal seeking loop except the delay in the responses. The negative feedback loop tends to take the state of the system towards the "goal". But because of the delay, the system does not reach that goal instantaneously. Therefore, the negative feedback still tries to move the system in the same direction. This causes the state of the system to overshoot the goal. Again, the negative loop tries to bring the state towards the goal, but again the delay plays its role and causes undershooting.

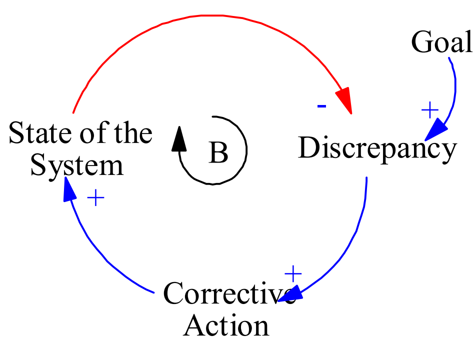
Figure 2-4: Fundamental Modes of Behavior

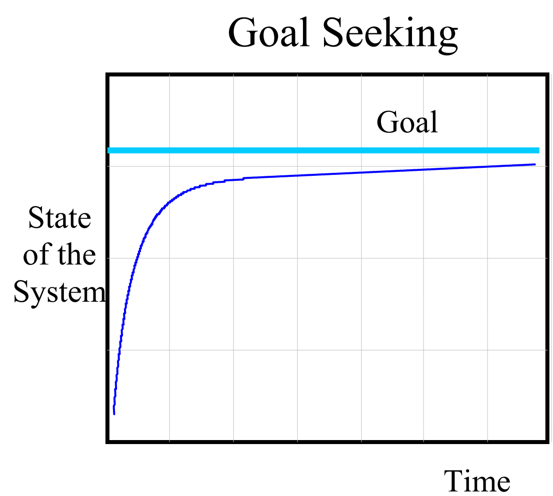
2.4 (a) Exponential Growth



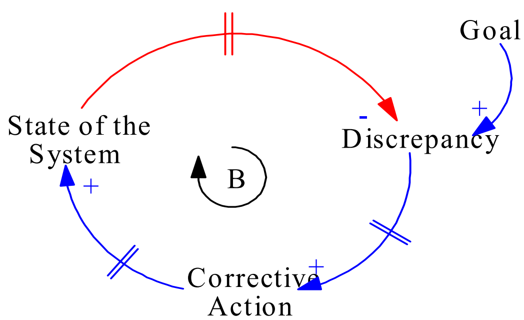


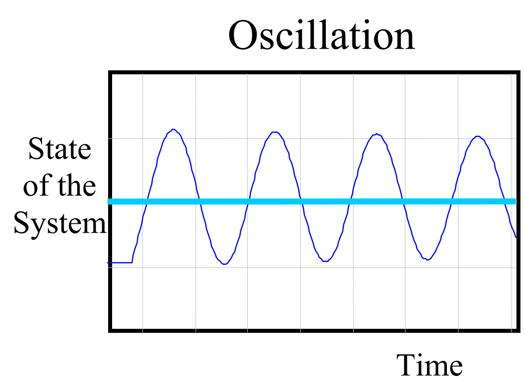
2.4 (b) Goal Seeking





2.4 (c) Oscillation





Derived Modes of Behavior

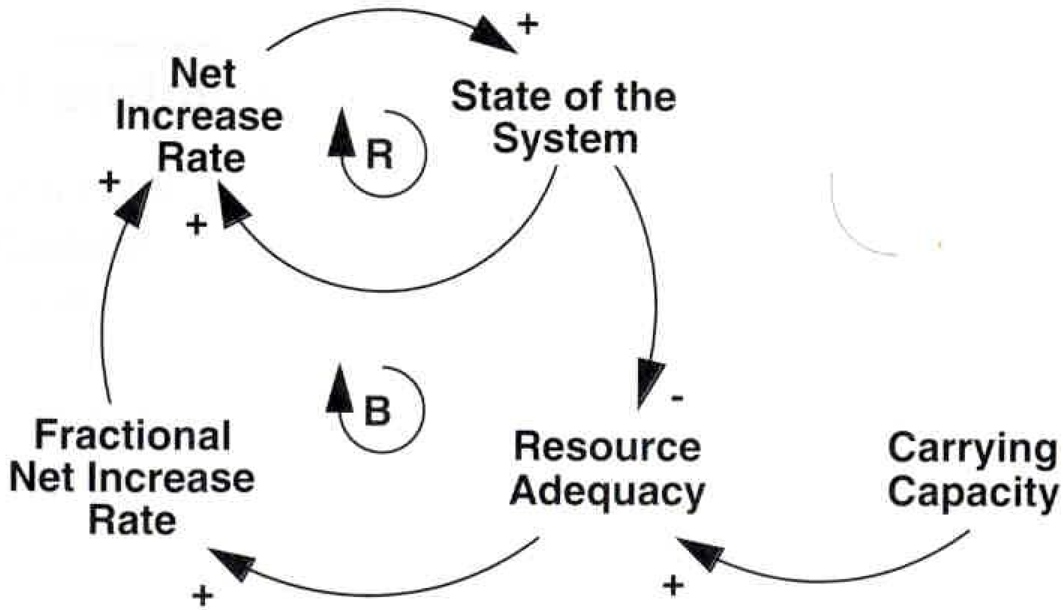
The fundamental modes are very basic. Interactions among the fundamental behaviors give rise to derived modes of behavior. The most common such behaviors are S-Shaped growth, Overshoot and Overshoot and Collapse and are represented by Figure 2.5.

S-Shaped Growth: This type of behavior is observed when the system has a positive and a negative feedback loops interacting with each other. The positive loop is stronger at first, so this causes exponential type increase. But no system can grow forever. The carrying capacity, the strength of the system to sustain itself, acts as a limiting factor to continuous growth. Due to the carrying capacity of the system, the negative loop then becomes stronger and it reduces the rate of increase of the state of the system. Thus, the behavior looks like an S-shape curve as shown in figure 2.5 (a).

S-Shaped growth with Overshoot: This is a derived behavior from S-shaped growth. This behavior is observed whenever delays are present in the negative loop of the S-shaped growth. The loop and the curve are shown in figure 2.5(b). At first, when the positive loop is stronger, the system shows exponential growth, but when the negative loops starts dominating the system, the delays in the negative loop cause system to fluctuate above and below the goal of the system. The system then slowly reaches the goal with oscillations.

Overshoot and Collapse: This type of behavior is also shown by the S-shaped curve where the carrying capacity of the system is not fixed and is consumed or eroded by the state of the system itself. The state of the system rapidly increases due to dominance of positive loop but it goes beyond the sustainability of the environment resulting in a rapid negative feedback, decreasing the state of the system very fast. The loop structure and the curve are shown in figure 2.5 (c) for this type of behavior.

Figure 2-5: Derived Modes of Behavior



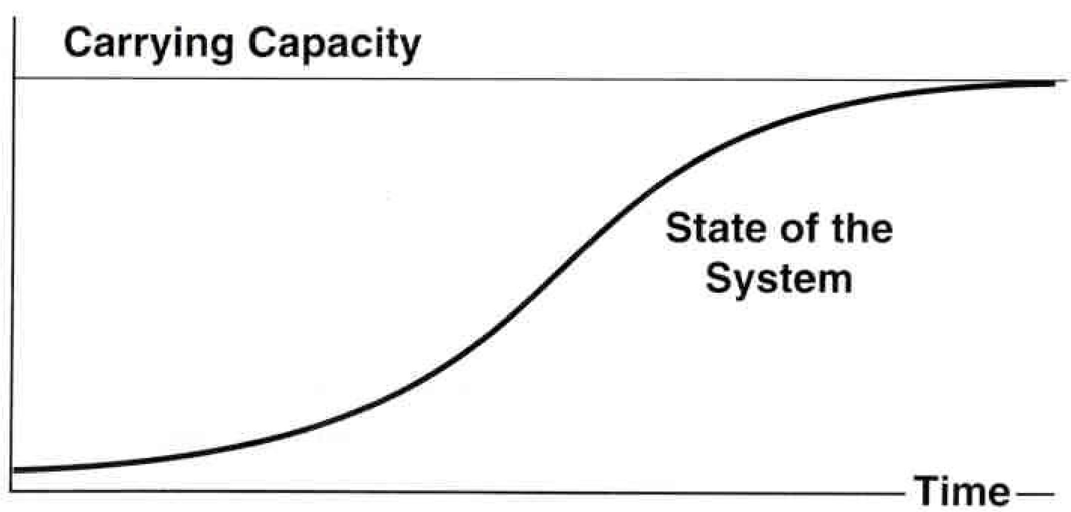


Figure 2.5 (a) S-Shaped Growth

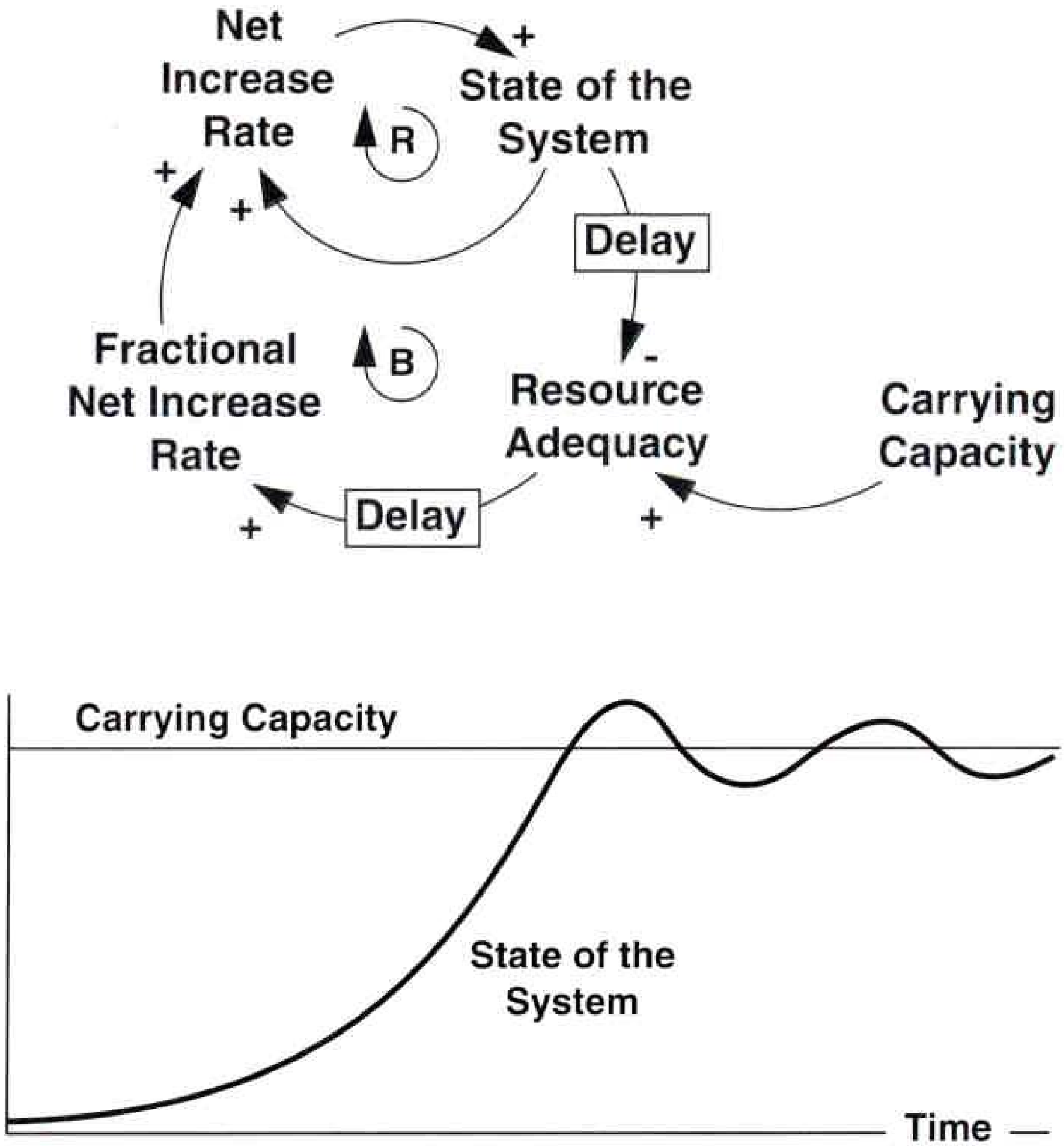


Figure 2.5 (b) S-Shaped Growth with Overshoot

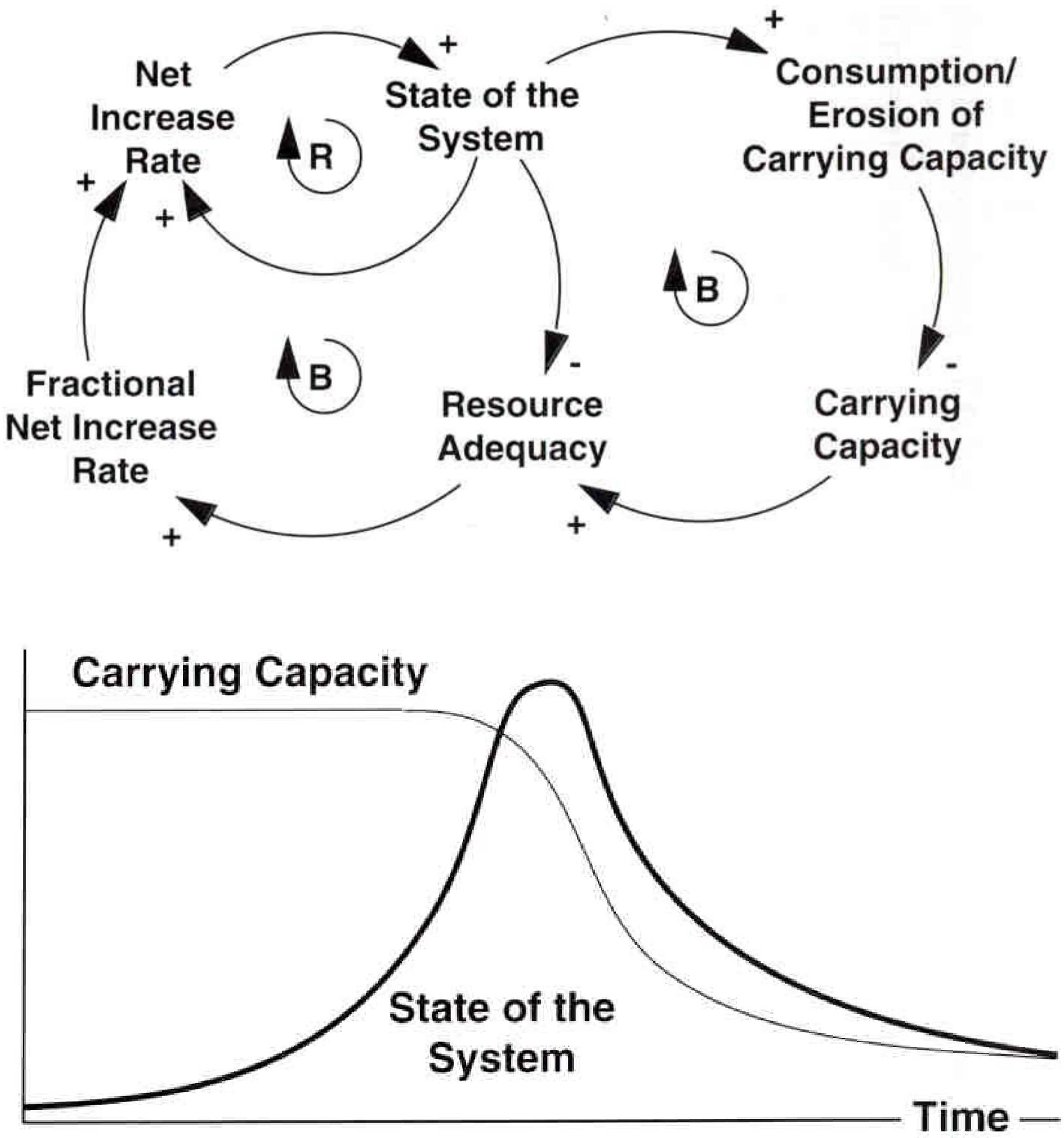


Figure 2.5 (c) Overshoot and Collapse

Other Modes of Behavior

System Dynamics models show some other types of behaviors. **Statis** or **equilibrium** behavior is the one where the state of the system remains almost constant. This is observed when the system has extremely long time horizons and great delays that dampen the disturbances in the system. When very strong goal seeking loops are present, the effect of disturbances in the system is very negligible and the negative loop brings the system to its goal immediately. **Random** type behavior can be observed in a complex system where the behavior of the system is unpredictable. The behavior may not be random actually, but due to lack of our ability of understanding it, it could look like random behavior. **Chaotic** behavior is a type of oscillatory behavior where the system fluctuates irregularly, never exactly repeating its motion. But this is different that a random behavior in a sense that its motion is completely deterministic.

2.4.4 Stock and Flows

Causal loop diagrams describe the system structure qualitatively. They present how various variables in the system affect each other. They also portray whether the relationships are positive or negative. But they do not provide information about the quantitative relationships between the variables. Stock and Flow structures capture the quantitative aspects of the system. Stock and Flow structures are developed from the causal loop diagrams by adding levels and rates variables and system delays. They provide the information about the values of various variables and the rate at which those values that are associated with them. Stocks are accumulations due to differences between inflow and outflow rates. Stocks are responsible for delays in the system and they give the system inertia and memory.

Notation for Stock and Flows

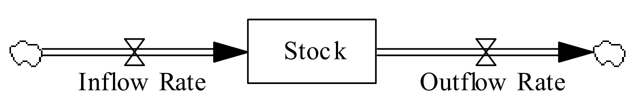


Figure 2-6: Stock and Flow Structure

Stocks are represented by rectangles while the rates are represented by valves that control the rate of filling and depleting the stock. The inflow (or outflow) of a stock can come from (go into) another stock or from (into) a cloud, which is a source (sink) outside the model boundary.

Mathematical representation of Stock and Flows

Stocks are nothing but the differential equations that are typically represented in a pictorial format. The stock and flow structure is analogous to the bathtub with water flowing in and out of the tub.

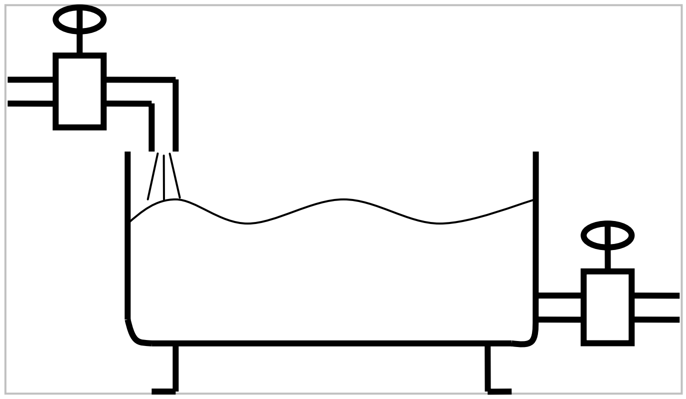


Figure 2-7: Pictorial Representation of Stock and Flow Structure

The mathematical formulation of the structure is:

Integral Equation

$$\text{Stock(t)} = \text{Stock}(\text{t}_0) + \int_{\text{t}_0}^{\text{t}} \left[\text{Inflow(t) - Outflow(t)}\right] \cdot \text{dt}$$

Thus, the value of stock at time t is the sum of the value of stock at time t_0 and the integral of difference between inflow and outflow rates from t_0 to t.

Or, in other words,

Differential Equation

$$\frac{\text{d(Stock)}}{\text{dt}} = \left[\text{Net Change in Stock}\right] = \left[\text{Inflow(t) - Outflow(t)}\right]$$

Rate of change of stock is equal to the difference between inflow and outflow at any instance.

<u>Auxiliary variables</u>

While developing stock and flow structure from a causal loop diagram, various variables are added to the stock and flow diagrams. They are the auxiliary variables and are primarily added for computational convenience. Since "rate" is nothing but the change of some quantity per time, it represents the "quantity" as well as "time". In addition to that, in the case when the relationship between two variables in non-linear, an additional variable that defines the relationship between the two variables is added. For example, consider a relationship between two variables A and B which is non-linear. An additional function variable, say X is defined that represents B as a function of A. Thus inputting value of A to the function X gives the value of B. To incorporate this type of mathematical computations auxiliary variables are needed.

2.5 System Dynamics Modeling

2.5.1 Approach

The system dynamics modeling approach is a continuous process with feedback. Feedback feeds back into the problem definition forming system dynamics modeling a closed loop decision-making process.

Figure 2.8 depicts an open ended decision making approach.

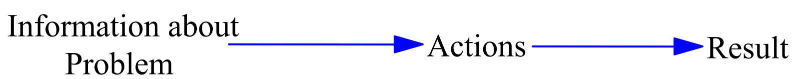


Figure 2-8: Open Ended Approach

In this case, from the information about the problem, actions are taken that produce the results. But the results may not necessarily be as expected. There is no evaluation of the result to ascertain whether they solving the problem.

Figure 2.9 shows the closed loop approach to decision-making where the results are fed back to the problem so that the problem can be re-defined and re-solved.

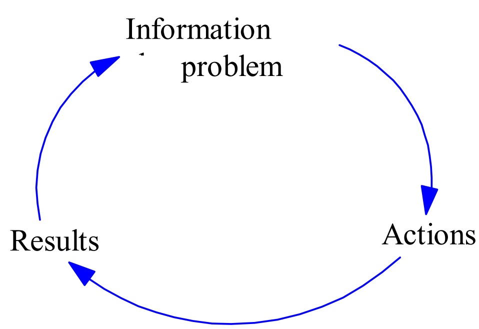


Figure 2-9: Closed Loop Approach

In this case, based on results, the problem is re-defined and new actions are taken to get new results. Thus the process is continuous and improving.

2.5.2 Steps of System Dynamics modeling

Richardson and Anderson (1980) describe six steps for system dynamics model building. The first three, they refer to as the "conceptual" steps and the last four as the "technical" steps with a considerable overlap between them. They are as follows:

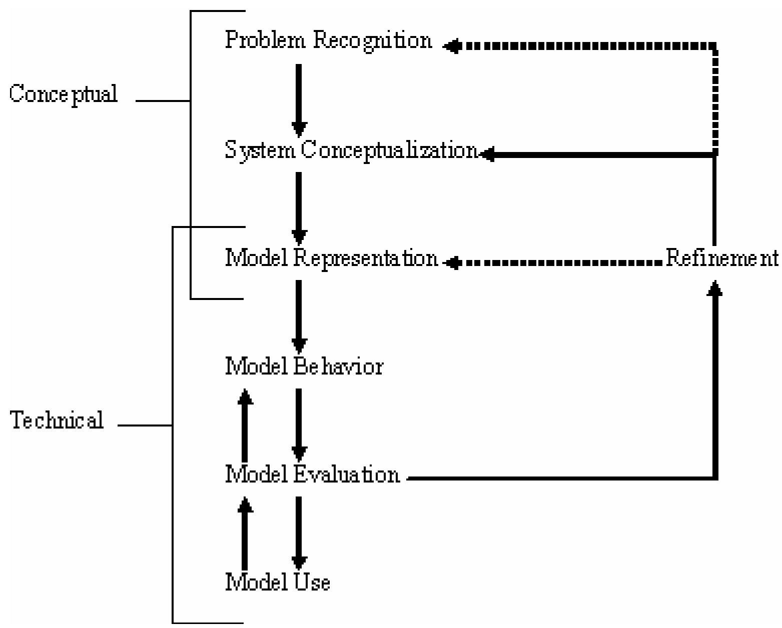


Figure 2-10: Steps of System Dynamics Modeling

2.5.3 Group Modeling and Equation Elicitation

The approach of system dynamics modeling is very much formalized as discussed in above topics. During the process of model building, the process of building the stock and flow structure is a crucial one. Stock and flow structure is built by converting the conceptual causal loop diagram into the mathematical model. It is very important that the correct mathematical relationships between the variables are represented in the model. Even if the causal loop structure is capturing the correct mental model, incorrect relationship in stock and flow diagram would result in wrong quantitative results.

The relationship between various variables included in the causal loop diagram is clear in the minds of experts who are the users of that system. But this knowledge is not explicit. It is not usually documented anywhere. Literature mentions about how this knowledge can be elicited from the minds of experts.

The knowledge is tacit, so it is difficult to describe, examine and use (Ford, Sterman 1997). It poses a need to develop a systematic approach that will help to elicit the knowledge. Burchill and Fine (1997) have introduced a methodology called "Inductive System Diagrams" for building a theory of product concept development from intensively gathered field data. They try to capture the tacit information in the mental models of the decision makers. They do so by using Inductive System Diagrams that combine Grounded Theory Methods with System Dynamics. With the Grounded Theory approach, the variables developed are intimately tied to the data. The relationships among these variables are mapped using causal loop diagramming techniques in system dynamics. Thus they present a qualitative way of mapping the product concept development to the causal loop diagram. But they do not address the issue of how to quantitatively represent these relationships.

Sterman and Ford (1997) propose a systematic methodology, with an example, to elicit data from the mental models of experts. They provide the following procedure that is divided into three phases.

Position Phase

The first phase is the position phase as referred by Sterman and Ford (1997). The experts in the team are given information that develops the context for the required information. The definitions of the variables explicitly and clearly mentioned and one relationship is considered at a time, which avoids confusion. Thus the members of the team are positioned to develop a mathematical relationship as discussed in the next section.

Description phase

In the second phase of description, the authors say that everyone in the group should be asked to think and describe the process visually, verbally and then in a written format. This helps the knowledge to flow from the minds of the experts. The two important parts of this phase are textual and graphical representations of their ideas. As an example, consider two variables “demand” and “price”. Increase in demand causes increase in price. The relationship between these two variables could be linear or non-linear. Description phase aims to describe the relationship quantitatively and identifies “how” this change takes place. To start with, the experts are asked to first think of anchor points. For example they are asked to find out the effect of demand on price for extreme values of demand i.e., with zero demand and maximum demand. When demand is zero, it has a corresponding value of price that is plotted on a graph of price versus demand. When demand is at maximum, it has increased effect on price. This point is again plotted on the graph. Thus, there are two data points that begin to establish the relationship between two variables. If the relationship is linear, then the two points are sufficient to draw the corresponding curve. Otherwise, some extra points are needed to draw the relationship curve. This can be achieved in two ways. The experts can either be asked for values of price for various values of demand such as 25%, 50%, 75% or they can be asked directly to draw a curve between the two anchor points. Sometimes it is easier for a person to think graphically rather than to think about values. They have a visual sense of the shape of the relationship rather than providing the values. Nevertheless, these two techniques are really helpful tools to extract data.

Discussion phase

The last phase in the elicitation process is the Discussion phase. The description of the relationship stated by each member of the team is written down and all these descriptions are compared. There could be difference between each person’s interpretations of the relationship. Comparing these descriptions help modelrs to come up with relationship that finally everyone agrees on.

Thus the process of converting the causal loop diagram into a stock and flow diagram is carried out.

2.6 <u>New Technology Implementation</u>

Integrating new technologies in an organization keeps it up-to-date in today's fast paced market. In order to maintain a competitive market position, organizations will continue to invest in new technology (Goodman and Griffith, 1991). Adaptation of new technology aims at improving organizational performance by increasing profits and thus, is responsible for the survival of organization. But the implementation process of new technology has some problems associated with it. They are discussed in the following section.

2.6.1 Problems Associated with the New Technology Implementation Process

In spite of its importance, the new technology implementation process is not addressed adequately in the literature. Griffith, Zammutoo and Aiman-Smith (1999) discuss the reasons why new technologies fail and often perform below expected levels. According to them the major cause for failure is the inability of organizations to develop an implementation process. Usually the technology design phase is considered as challenging and creative while the implementation phase is considered dull and boring. According to Griffith *et al*. (1999), the implementation process could be improved with active support from management, clear implementation goals, user involvement and training.

In their paper, Jassawalla and Sashittal (1998) discuss how the managerial behavior affects the organizational adaptations and technology transfer. They propose that a low-trust environment and organizational fear – paranoia – resulting from high technology based organizational adaptations decelerate the technology transfer. On the other hand, manager's contribution towards environmental constituents, employees, team members and organization's ability to change – managerial *pronoia* – accelerates the transfer of technology. The constituents of their environment are the other functional groups, customers, suppliers and competitors. Thus, the manager carries an important responsibility to create a high-trust environment among the team for successful adaptation of new technologies.

In addition to the trust in other people, the inclination or tendency of top management towards implementing a new technology plays an important role in the success of technology transfers. Due to the risk involved, not all managers are willing to adopt technological innovations. Tabak and Barr (1999) have proposed hypotheses related to top managers' inclination or intention to adopt radical organizational innovations as discussed in the next section.

2.6.2 Propensity to adopt technological innovations

Managers play an important role in adopting new technologies. The characteristics of managers that are responsible or positively related for technology innovations as stated by Tabak and Barr (1999) are as follows:

1. Risk propensity – The tendency of decision maker either to take or avoid risk.
2. Self-efficacy – It is decision maker's ability to accomplish a task.
3. Cognitive Complexity – It is described by the number of constructs used to evaluate alternatives in decisions.
4. Successful education level – Higher formal education brings more focus and receptivity to innovations making the decision makers more flexible and open to new ideas.
5. Past experience – Decision makers past innovations experience determines their tendency for adopting new innovations. Successful past experience will promote their self-efficacy.
6. Strategy – Defensive organizational strategies lead to stable but very narrow market. Offensive organizational, however, lead to diverse market and it is positively related with manager's inclination towards innovations.
7. Information processing capacity – It is the extent of interaction and subsequent information processing during strategic decision making.

8. Resource availability – It refers to the ability of organization to invest in innovations due to availability of resources.
9. Organizational specialization - The existence of a wide range of specialties indicates a broader in-depth knowledge base, resulting in faster diffusion and implementation of innovations.

According to them, only top manager's age is negatively correlated to their inclination to adapt to new technology. Younger executives have more current knowledge base and are more willing to take risks.

2.6.3 Process Approach to the Implementation of New Technology

Goodman and Griffith (1991) have developed a process oriented theoretical approach for understanding the implementation of new technology. The five processes they have discussed are as follows:

Socialization – It is defined as a process by which individuals acquire knowledge and skills about the new technology. The skill can be something like computer skills needed to understand and use the new technology. Socialization can be achieved by formal class-room training or hands-on training or informal coaching. Socialization is necessary because it determines how the individual behaves in respect to the new technology.

Commitment – It is defined as the binding of the individual towards the new technology. Commitment to a technological innovation increase a probability of performing behaviors consistent with the utilization of that technology.

Reward allocation – It refers to the allocation of different types of rewards relevant to the implementation of new technology. The technology itself is one source of reward.

Feedback and Redesign – It is a process by which data are collected about a new technology and redesign activities are initiated to enhance the operation of the new technology. Feedback and redesign is an important process and it directly affects the success of implementation process.

Diffusion – It is a process by which the technology is extended to other parts of the organization.

The literature states that for an organization, the new technology implementation process is very important in order to survive in the market. The implementation process needs the involvement from everyone who is involved in the implementation process. But not enough consideration is given to the involvement from every level and hence many of the new technologies fail.

2.6.4 New Technology Implementation on Ship Systems

Naval ship production and the introduction of new technologies on ships is a long and complex process. This process often experiences cost and schedule overruns due to various reasons. Cooper (1980) discusses these reasons in detail. Ingalls Shipbuilding was awarded contract by Navy to build nice amphibious assault ships. Because of the cost and schedule overruns, Ingalls shipbuilding claimed $500 million against US Navy. The computer simulation model developed by the author helped to quantify the costs of disruption for Navy responsible delays and design changes. The Navy agreed out of court to pay $447 million to settle the claim. Ingalls shipbuilding then extended the model and used it for strategic decision making in its shipyard operations.

As analyzed and explained by Cooper (1980), cost overruns are often blamed on contractor mismanagement and government interfaces. But there are other plausible reasons that could lead to cost overruns. The main cause is the inaccurate estimate of the work content. The shipbuilding process, as well as the introduction of new technologies on a ship system is a “first-of-a-kind” system and hence there are various ways where the cost estimate could go wrong. In addition to that, there are pressures on Navy from Congress to maintain the cost of defense systems in the palatable range.

Thus the implementation of new technology on ship system is a very complex process that has not been studied in detail yet. The “system” view of the process can lead to the better understanding of it. This study enables analysts and decision makers to get an in side view of the shipbuilding process.

2.7 <u>Risk</u>

The trim "risk" is often used in terms of implementation of new technologies. Not only in terms of new technology implementation, but the risk is always associated with any activity in our day to day life. The only important point is its probability and consequences differ from activity to activity.

Byrd and Cothern (2000) define risk as "the probability of a future loss". Though this definition is not universally accepted, it is commonly used. Experts have not been able to agree upon a unique definition of risk. The debate is over considering risk as a nature of probability or utility. Warren Hope (1998) defines risk as an uncertainty about whether a loss will occur. The experts from the Department of Defense define the risk as the likelihood that an unwanted outcome will occur. Thus, risk deals with the chances of suffering from some sort of loss in an activity.

As proposed by Byrd and Cothern (2000), risk analysis into three components as follows:

1. Risk Assessment: It is process which estimates the probability of risk under the specific conditions.
2. Risk Management: It is a process of controlling risk. It involves taking decisions that will reduce the risk and its effects.
3. Risk Communication: It is the process of explaining risk.

Risk is assessed in terms of likelihood and consequences in the US Navy. The likelihood is the chance that the risk event will occur. Consequence is the outcome if an risk event occurs. Based on these two parameters risk is assessed and necessary modifications to the program are made.

Chapter 3: The Model

The model development is a continuous and complex process. The effective model creation cannot only be done by the modelers but it also involves participation from the people who are going to use the model for decision making. The modeling process involves variety of inputs from the people from different levels. Every person who participates in the modeling process imparts his (or her) own ideas or "mental" model on the model that is being developed. For the model to be effective, it needs be developed in a systematic and standardized procedure. The procedure used for the development of the Technology Integration process follows the Steps of Modeling Process suggested in Business Dynamics by Sterman (2000). The following chapter explains in detail the steps taken towards building the model for technology integration subsystem.

3.1 Steps of the Modeling Process

Following are the steps discussed by Sterman (2000).

1. Problem Articulation
2. Formulation of Dynamic Hypothesis
3. Formulation of a Simulation Model
4. Testing and
5. Policy Design and Evaluation

This chapter deals with the first three steps of the model building. Steps 4 and 5 are discussed in chapters 4 and 5.

3.2 Problem Articulation

Problem articulation involves defining the problem. Problem articulation is the most important step in modeling (Sterman, 2000). It is very important to have a clear purpose of the model. Since system dynamics modeling is usually a group effort, a clear purpose ensures that everyone in the group will think and stay focused on a single problem and that prevents the modeling process from going off the track. Also, the model should address a limited number of issues rather than representing the whole system. The reason being, that the comprehensive model of the system will be as complex as the system and it will defeat the purpose of modeling.

3.2.1 The Problem

Cost overrun, schedule overruns and performance degradation are typically observed during the integration process of large technologies that involve large number of people over a long time period. The problem faced by our clients, the shipbuilder and the Government, was heavy cost overruns during the new technology implementation process on ship systems. In spite of the reasonable efforts, the cost overruns and frequent requests for additional funding has been experienced. There is no mean to predict the magnitude of the cost overrun and the effectiveness of the integration process of a new technology on existing ship systems.

3.2.2 Key Variables

After defining the problem, the variables that are representative of key concepts were determined. The following three variables were identified as key variables and the whole model revolves around these variables.

Risk: Risk is defined as a probability of a future loss (Byrd and Cothern (2000)). Risk associated with the introduction of new technologies is stated as the likelihood that a specific unwanted outcome would occur. Outcomes usually refer to cost, schedule and performance. Risk is a function of maturity and complexity (Monga (2001)). Increased technology maturity implies reduced risk, because the technology is better defined. Whereas, increased complexity implies higher risk. The technology integration risk is a function of the technology development risk. It is a dimensionless variable measured using a relative scale (varying from 1 to 10). There is risk involved in various stages of the technology integration process viz. engineering/design work, assembly and installation, testing and material procurement. The risk has considerable impact on these activities but it is generally not taken into account. Risk increases the effort of the technology integration activities, which leads to cost and schedule overruns.

Cost: This represents the total cost involved in the technology integration process. It is typically measured in dollars. Since the length of the integration process is long enough, inflation needs to be taken into consideration.

Performance: It is a variable that states the degree to which the system meets the expected operational characteristics. It is the variable measured on a scale from 0 to 1, 0 being the worst or totally unexpected performance and 1 being the best or exactly expected performance. The performance measurement is typically done by either considering a single critical variable or by considering various variables, each weighted according to its importance. Considering the former case, suppose the critical variable is the speed of the carrier. If the current speed that can one can get is 30 nautical miles and if 60 nautical miles is the target speed, then the target performance will be 1 corresponding to 60 and the current performance will be 0.5, corresponding to 30.

3.2.3 Reference Modes

The initial characterization of the problem is done through the charts, graphs and data that describe the modes of behavior of the problem over the time period. These graphs are called as reference modes. They clarify the problem to both modelers as well as clients, in a pictorial format. Following are the examples of reference modes for the problem being discussed. The graph shows how the cost rises over the two-year horizon of technology integration process.

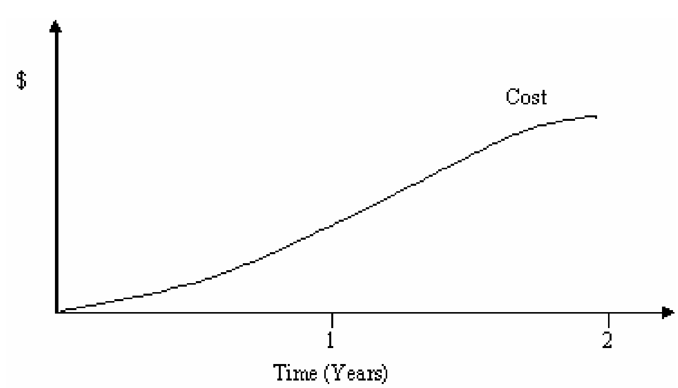


Figure 3-1: Reference Mode for Cost

The following graph shows how the performance increases over the two-year period of the technology integration process. The performance variable considered in this case is the speed of the aircraft carrier. The target speed, 60 nautical miles, corresponds to the performance of 1 (dimensionless).

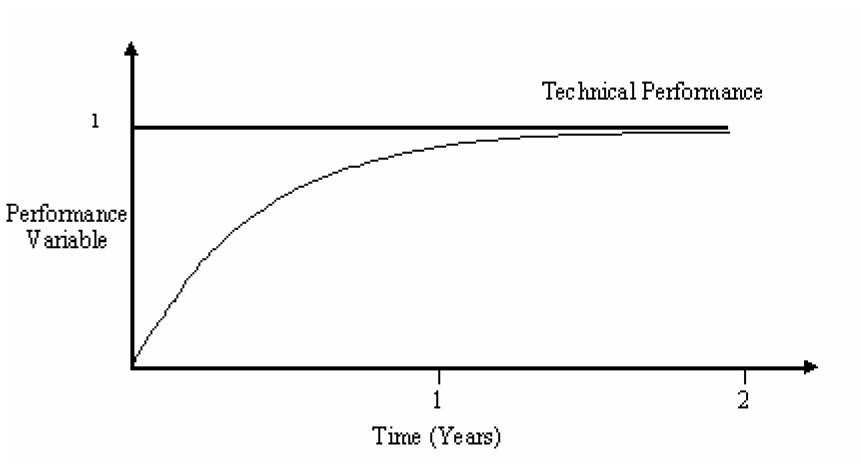


Figure 3-2: Reference Mode for Technical Performance

3.2.4 Time Horizon

It is very important to determine the time horizon for the problem. Usually we tend to think of cause and effect as local and immediate (Sterman 2000). This usually results in short time frame. But because of the feedbacks and delays involved in the large systems, the time horizon should be long enough to take care of these delays and the impact of the feedback structures. But one cannot make the time horizon extremely long either. Very long time horizons could affect the perception of the results and wrong conclusions could be drawn. It was concluded in our group modeling meetings that the time required for the technology integration process is around 2 years (104 weeks) which is sufficient enough to take into account all the activities involved in it.

3.3 Formulation of Dynamic Hypotheses

This is the second step in the modeling process. After the problem definition and the initial characterization, the modeler then starts to develop a theory about the problem.

This theory or hypothesis is called “dynamic hypothesis”. The hypothesis is referred to as “dynamic” because it characterizes the dynamics involved in the system over the given time horizon. It takes into account the feedback mechanisms and the delays involved in the system. This step involves, basically, getting inputs and views from the participants of the modeling team.

The following questions were identified as research hypothesis, in general:

1. Why are the complex technologies less affordable in terms of technology integration process?
2. Why are the less developed (immature) technologies less affordable in terms of technology integration process?

Based on the discussions in the modeling sessions, following specific dynamic hypothesis were identified:

1. Increase in complexity of new technology increases total cost of the technology integration process.
2. Increase in technology maturity reduces total cost of the technology integration process.

3.3.1 Endogenous Explanations

The system dynamics models involve various variables or theories that are interlinked and exhibit interactions within the system. They describe various feedbacks that affect the overall system performance. These theories are referred to as Endogenous (See Glossary – Appendix A) Explanations, “arising from within” as quoted by Sterman (Sterman 2000, pp. 95). In contrast, Exogenous (See Glossary – Appendix A) Explanations are the ones those come from outside the boundary of the system. They are the external factors that affect the behavior of the system. But the system, by itself, does not have any effect on them. The main focus in the system dynamics modeling is on the endogenous variables and it is recommended to have a small number of exogenous variables as compared with the endogenous variables.

3.3.2 Mapping System Structure

There are different tools in system dynamics that facilitate the modeler and the clients to build the actual model of the system being studied. These include model boundary charts, subsystem diagrams and causal loop diagrams. These diagrams were developed and are discussed below.

Model Boundary Chart

Table 3.1 provides a definition of the boundary of the model. It presents the variables that are going to be inside the system (endogenous), outside the system boundary (exogenous) and the excluded ones. The decision of under which column these variables are placed in the Table is based on the purpose of the model and the defined problem.

Table 3.1: Endogenous, Exogenous and Excluded Variables

Endogenous	**Exogenous**	**Excluded**
Cost	Risk	Environmental Factors
Various Integration activities	Requirements	
Performance	Labor and Material Costs	
Rework		
Completion Rates		
Additional Funding Requests		

Subsystem Diagram

A subsystem diagram shows the high level overall structure of the system. It highlights the flow of material, money and information. The following subsystem diagram has been developed for the technology integration system. The central block of Technology Integration includes various integration activities that are interconnected by the information flow shown by arrows. Factors such as requirements and funding have an effect on the technology integration process.

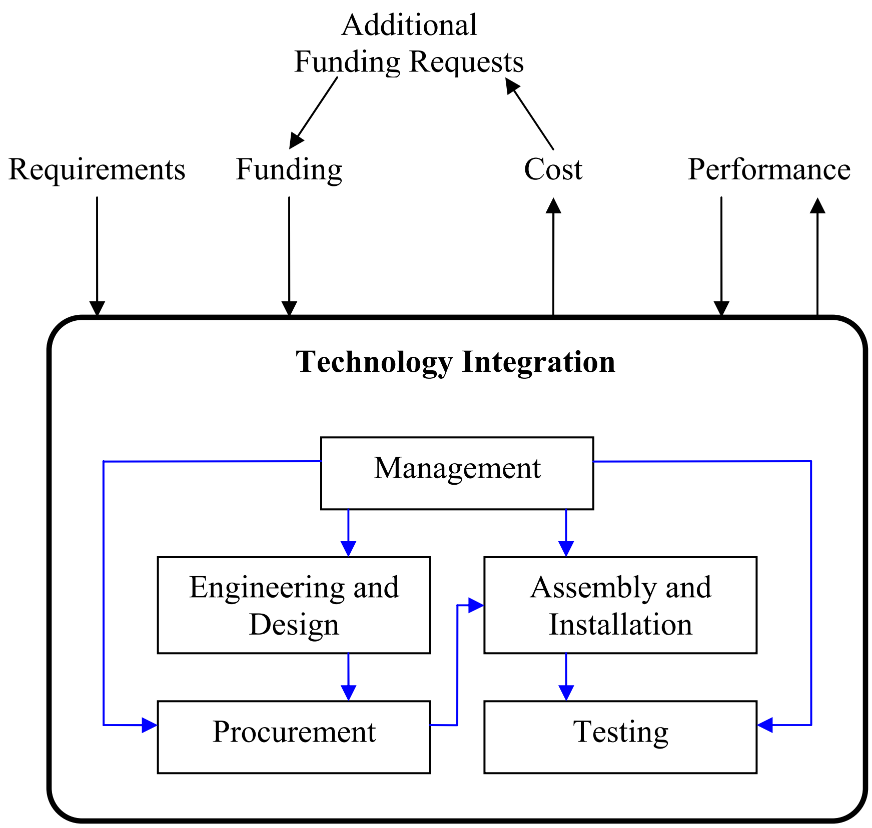


Figure 3-3: Subsystem Diagram

3.4 Formulation of the Simulation Model

Model boundary diagram and subsystem diagram gives a rough idea about linkages and feedbacks within the system. In a step towards actual building of the model, two important stages are: (1) the development of the causal loop and (2) the development of the stock and flow diagrams. During these stages the modeler and the clients define how every identified variable in the system relates to the other variables in the system. Various feedback loops are revealed during this process. The following sections explain the causal loop and stock and flow diagrams for technology integration.

3.4.1 Definitions of the Variables

1. Funding – It is the amount of money allocated initially to carry out the technology integration activities. It is measured in current dollars.

2. Planned Spending Rate – This is the planned spending rate, decided by the program manager, over the life span of technology integration process based on the initial funding. This rate wholly depends on the available funding and it is measured in current dollars/week.
3. Cost Overrun – It is defined as the amount by which the cost exceeds the funding. It is measured in current dollars.
4. Engineering and Design Effort – This is defined as the effort put into the engineering and design activities as a part of the technology integration process. It is measured in man-hours/week.
5. Planned Engineering and Design Work – It is defined as the amount of the engineering and design work, which is planned over the time horizon associated with the technology integration process. This work pattern is decided by the program manager at the outset of the technology integration process and is measured in man-hours/week.
6. Assembly, Installation and Integration Effort – This is defined as the effort put into the assembly, installation and integration activities as a part of the technology integration process. It is measured in man-hours/week.
7. Planned Assembly, Installation and Integration Work – It is defined as the amount of the assembly, installation and integration work, which is planned over the time horizon associated with the technology integration process. This work pattern is decided at the outset of the technology integration process by the program manager and is measured in man-hours/week.
8. Planned Material Procurement – It is the number of technology units planned to be purchased over the time horizon associated with the technology integration process. This purchase pattern is decided by the program manager at the outset of the technology integration process and is measured in units/week.
9. Testing Effort – This is defined as the effort associated with the testing activities as a part of the technology integration process. It is measured in man-hours/week.
10. Planned Testing Work – It is defined as the amount of the effort in testing activities that is planned over the time horizon of the technology integration process. This

work pattern is decided by the program manager at the outset of the technology integration process and is measured in man-hours/week.

11. Management Effort – This is defined as the effort associated with the management activities as a part of the technology integration process. It is measured in man-hours/week.
12. Planned Management Effort – It is defined as the amount of effort in management activities, which is planned over the time horizon of the technology integration process. This work pattern is decided by the program manager at the outset of the technology integration process and is measured in man-hours/week.
13. Integration Risk – It is defined as a measure of risk involved in the technology integration process. This risk depends on the technical performance of the technology development subsystem. It is a dimensionless variable measured on a scale from 1 to 10.
14. Integration Performance – This variable gives the idea about the overall performance of the technology integration process. It is affected by the cost overrun, integration risk and the amount of rework. It is a dimensionless variable measured on a scale of 0 to 1. The performance measurement is typically done by either considering a single critical variable or by considering various variables, each weighted according to its importance. Considering the former case, suppose the critical variable is the speed of the aircraft carrier. If the current speed that can one can get is 30 nautical miles and if 60 nautical miles is the target speed, then the target performance will be 1 corresponding to 60 and the current performance will be 0.5, corresponding to 30.
15. Target Performance – It is defined as the desirable system performance from the technology integration subsystem. It is a dimensionless variable based on a scale from 0 to 1. The maximum or best performance is 1. When the performance is measured based on a value of one variable, the targeted maximum value of that variable (e.g. the targeted speed of aircraft carrier is 60 nautical miles) corresponds to the performance of 1. The actual performance is then compared with this target performance.
16. Integration Performance Discrepancy – It is defined as the difference between the integration performance and the target performance. Being the difference between

the two dimensionless variables, integration performance discrepancy is also a dimensionless variable.

17. Rework – It is defined as the amount of technology integration work that needs to be redone because of the performance discrepancy. Rework is driven by the Integration Performance Discrepancy. It is assumed, for the purpose of simplification, that the rework is same as the integration performance discrepancy. Hence rework is a dimensionless variable based on a scale from 0 to 1.
18. Cost Realization Rate – It is defined as the rate at which the funds are actually spent for the various integration activities. It shows how fast the cost of the technology integration process builds up. It is measured in current dollars/week.
19. Cost – It is the amount of money spent for various integration activities. It is measured in current dollars.
20. Additional Funding Request – It is the amount of excess funding requested because of the discrepancy between the planned spending rate and the cost realization rate so as to avoid any further cost overrun. It is measured in current dollars.

3.4.2 Assumptions of the Model

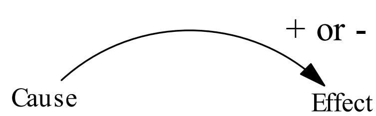
The following assumptions were made during the group modeling process. The assumptions were made to avoid unnecessary complexity in the model and they were obtained and agreed through discussions in the modeling sessions.

1. Funding allocation is done right on the first day of the project. All the funds budgeted for the technology integration process are available on the first day of the project. The project manager plans various activities in the subsystem based on these funds.
2. Integration risk is dependant on the performance of the technology development subsystem. Integration risk affects various integration activities. The higher the risk the more effort is required in the integration activities. Integration risk also negatively affects the integration performance.

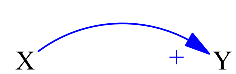
3. Cost overrun has a negative impact on all the integration activities and performance. It adversely affects the integration activities, which is based on the relationship developed by the modeling team.
4. Request for additional funding is based upon a concept of Control Spending. Control spending is spending which is assumed to be higher than the planned spending with a buffer for overspending. The request for additional funding is made when the costs exceeds the control spending. The amount asked is equal to the difference between the cost and the planned spending.
5. Additional request for funding is not satisfied immediately after the request is made. The additional funding is provided only at the end of each quarter after the first day of project. Also, no additional funding requests are satisfied during last 6 months of the project.
6. Of the additional funding requested, not all is received. Only some percentage of that request is satisfied.
7. There is a delay associated in receiving the additional requested funding.

3.5 Causal Loop Diagram

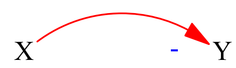
Causal loop diagram portrays the feedback structures of the system through the positive and negative linkages between the variables. It gives the qualitative description of the system.



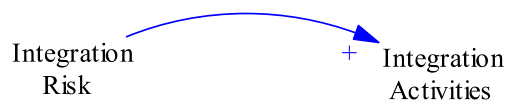
To understand the linkage polarity, consider two variables X and Y such that change in X is causing some change in Y. If the increase (decrease) in value of X causes increase (decrease) in value of Y, then they are connected by the positive linkage, all other things being equal.



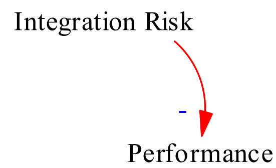
On the other hand, if increase (decrease) in X causes a decrease (increase) in Y then they are connected by the negative linkage, all other things being equal.



To be more specific, considering the actual variables in the model, Integration Risk and the Integration Activities are linked by the positive linkage.



Increase (decrease) in the integration risk increases (decreases) the integration activities. On the other hand, Integration Risk is linked to the Performance by a negative link. Risk has an adverse effect on the performance.



The more (less) the Integration Risk, the worse (better) is the integration performance. Below is a generalized causal loop diagram for the technology integration subsystem.

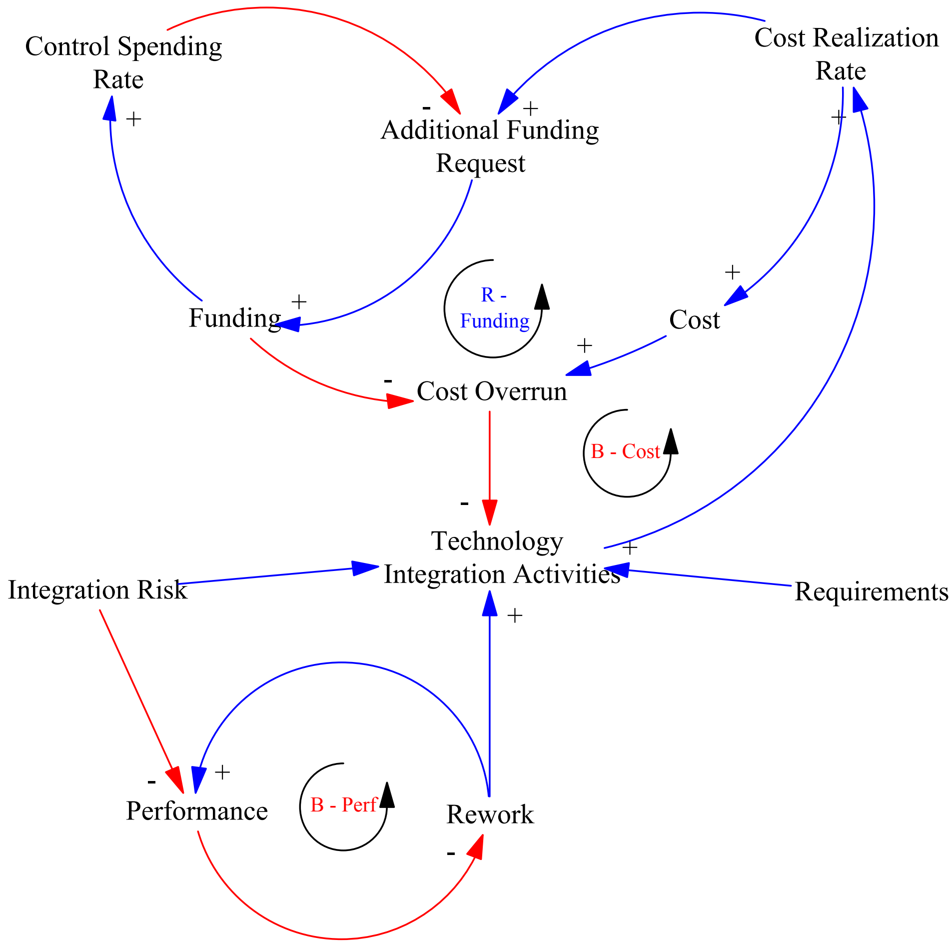


Figure 3-4: Generalized Causal Loop Diagram

At the center of the diagram is the *Technology Integration Activities* variable. The technology integration activities are primarily driven by the requirements. The cost of these activities is continuously checked with the funding to see if there are any cost overruns. Cost overruns have a negative impact on the integration activities. Control spending rate is the pre-determined rate at which the funding is supposed to spend for the project. It is compared with the cost realization rate - the rate at which cost increases - to decide for any requests for additional funding. The rework and performance loop is at the bottom. Rework increases the effort put into the integration activities. The integration risk has an impact on both integration activities and performance. Risk increases the effort required for the activities and it reduces performance. Following is the explanation for three major loops in the diagram.

1. B-Cost: It is a balancing feedback loop that keeps the cost in line by controlling the integration activities. Increase in integration activities increases the cost of the project. This leads to increase in the cost overruns. But increase in cost overruns reduces the integration activities, thus reducing the cost, in turn. Thus, this loop does not let the cost go beyond a certain limit.
2. R-Funding: It is a reinforcing feedback loop that attracts the additional funding in the project. Increase in the integration activities increases the rate at which cost rises. This increasing rate causes the project manager to go and ask for additional funding since he foresees a cost overrun in the future. Additional funding requests cause more funding to flow into the project. This causes reduction in the cost overruns and increase in the integration activities.
3. B-Perf: It is a performance-balancing loop. The lower the performance, the more rework is necessary. Increases in rework leads to increases in performance.

The detailed causal loop diagram is as shown in the following Figure 3.5

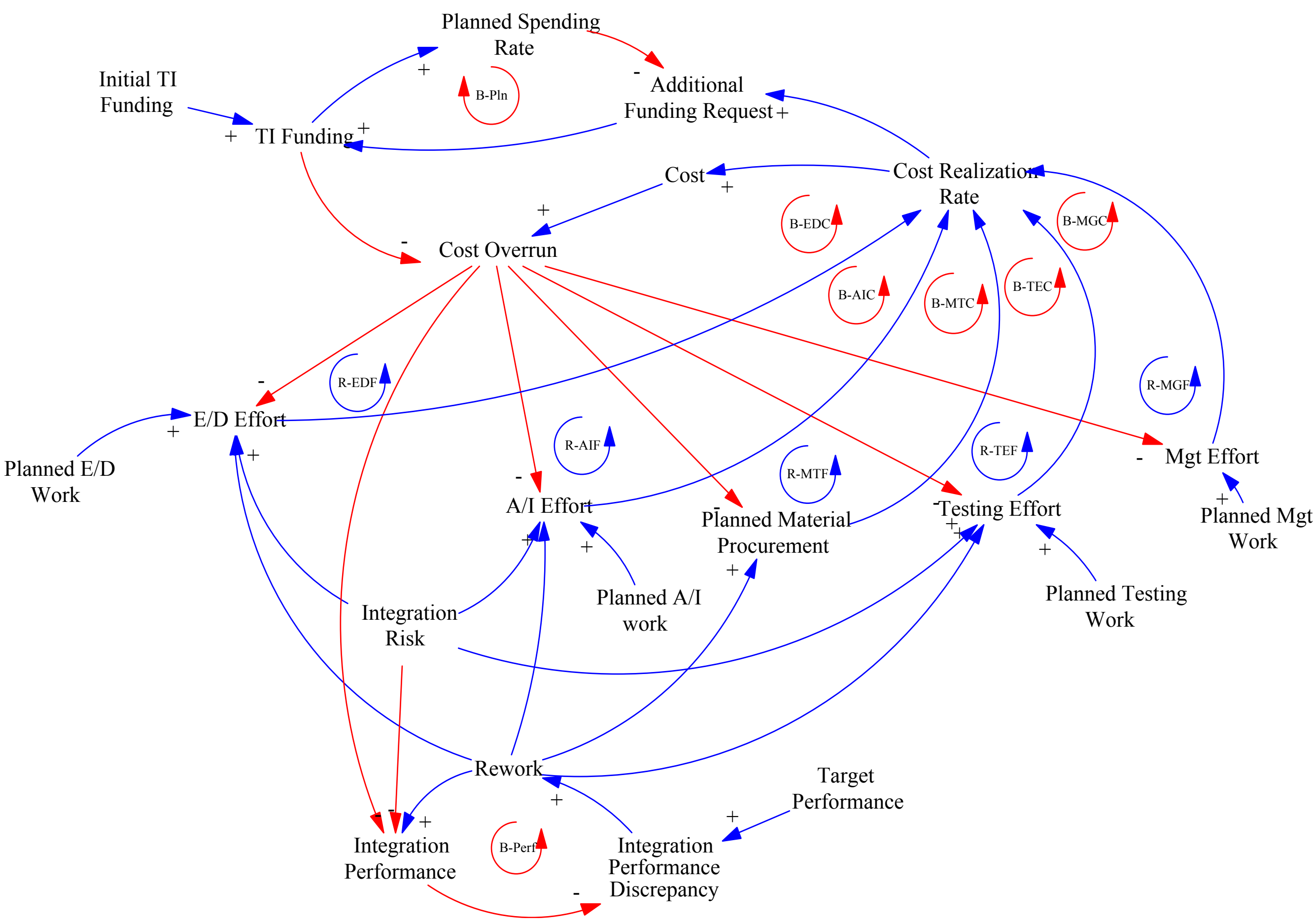


Figure 3-5: Causal Loop Diagram for Technology Integration

Above diagram is the expanded or elaborated diagram of the generalized diagram.

The technology integration process is divided into following activities:

1. Engineering and Design Effort
2. Assembly, Installation and Integration Effort
3. Testing Effort
4. Material Procurement
5. Management Effort

The basic interactions remain the same except that each activity is separated. The requirements, now, can be customized separately and behavior can be seen distinctly.

3.6 Technology Integration Stock and Flow Diagram

Causal loop diagram displays the qualitative description of the system. It provides an overall system representation where the interrelationships between variables, feedback loops, and significant delays are highlighted. But it does not provide the quantitative description of the system. The causal loop diagram provides information about the fact that one variable affects the other in either positive or negative manner, but it does not provide a description of "how" this is accomplished. The answer to "how" is found in the stock and flow diagram.

This stage of the modeling requires a close and continuous interaction between the modelers and the clients. Most of the relations between the variables have never been formulated before and they are hidden as a "tacit" knowledge in the clients' mind. It is an important and time-consuming task to elicit this tacit knowledge. The methodology suggested by Ford and Sterman (1997) ("Expert Knowledge Elicitation for Improving Mental and Formal Models") was used to develop the formulations.

3.6.1 Stock and Flow Notation

Causal loop diagrams capture the mental models and the feedback structure in the system. However they lack the ability to provide the stock and flow structure or demonstrate the real dynamics of the system. Stocks represent accumulation. They characterize the state of the system and generate the information upon which decisions and actions are based. Stocks give system inertia and provide them with memory (Sterman 2000). Thus stocks are responsible for delays and disequilibrium in the system.

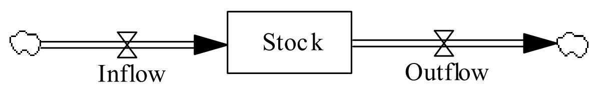


Figure 3-6: Basic Stock and Flow Structure

All stock and flow structures are composed of stocks, represented by rectangles and contain either inflows or outflows or both, represented by valves and arrows. The flows either come from or go into other stocks or they can come from or go into infinite sources and sinks, represented by clouds. Mathematically, they are represented as follows:

$$\text{Stock}(t) = \text{Stock}(t_0) + \int_{t_0}^{t} \left[\text{Inflow}(t) - \text{Outflow}(t)\right] \cdot dt \quad \textbf{Or} \qquad \textbf{... (3.1)}$$

$$\frac{d(\text{Stock})}{dt} = \left[\text{Net Change in Stock}\right] = \left[\text{Inflow}(t) - \text{Outflow}(t)\right] \qquad \textbf{... (3.2)}$$

3.6.2 The Technology Integration Stock and Flow Model

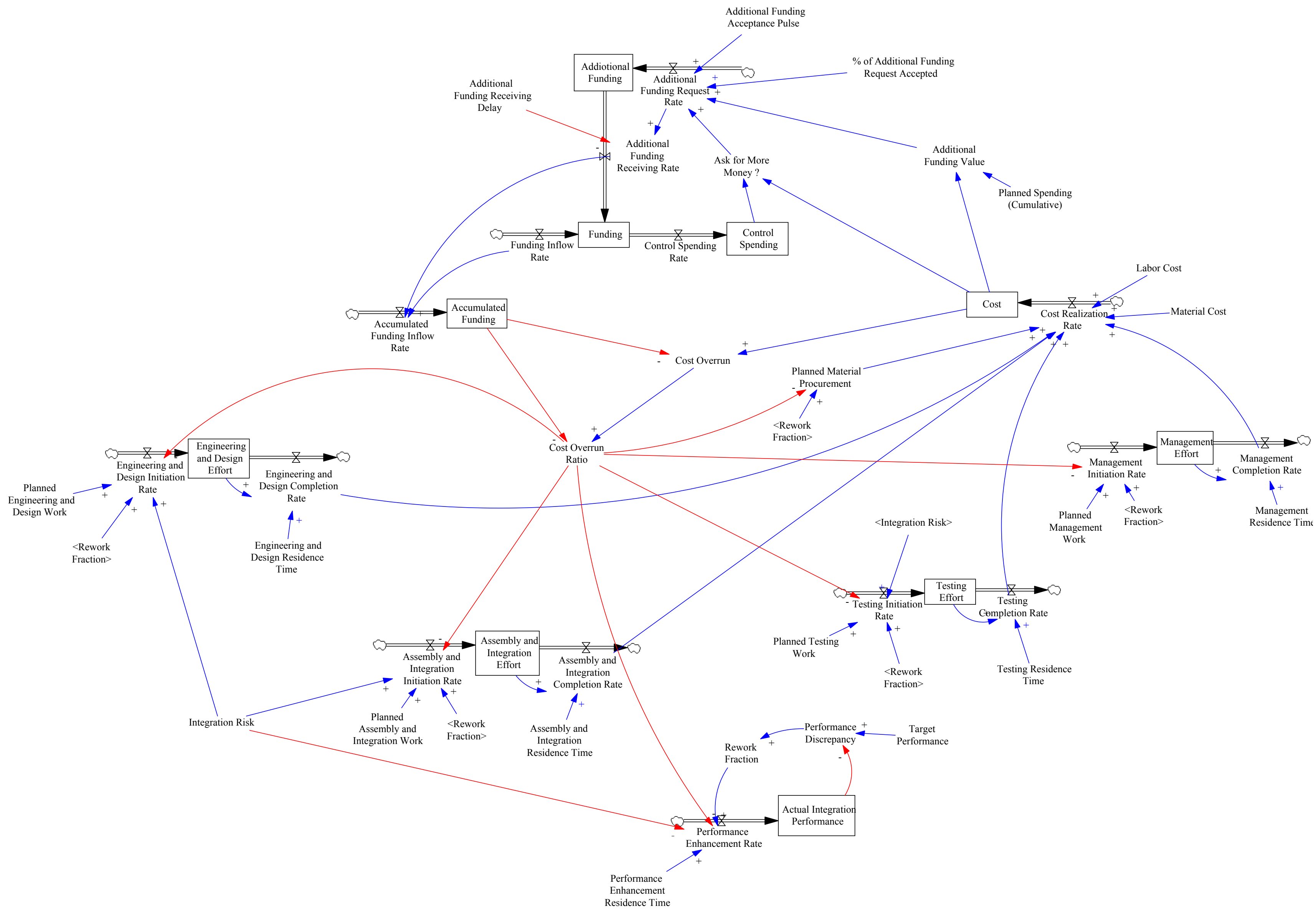


Figure 3-7: Stock and Flow Diagram for Technology Integration

3.7 The Stock and Flow Structures

3.7.1 The Engineering and Design Effort

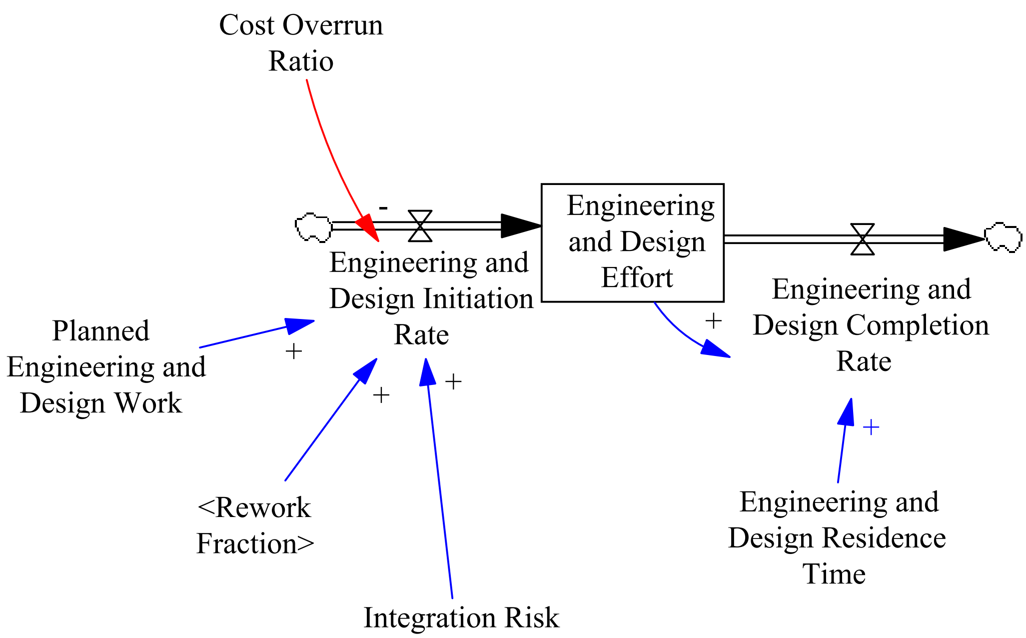


Figure 3-8: Engineering and Design Effort Structure

Above diagram shows the stock and flow structure for the Engineering and Design effort. The stock is fed in by Engineering and Design Initiation Rate and is depleted by Engineering and Design Completion Rate, both measured in man-hours/ week.

Thus Engineering and Design Effort is given by:

$$\text{Engineering and Design Effort (t)} = \text{Engineering and Design Effort (0)} + \int \begin{bmatrix} \text{Engineering and Design Initiation Rate} \\ \text{- Engineering and Design Completion Rate} \end{bmatrix} \cdot dt$$

$$\text{Engineering and Design Effort (0)} = 0$$

... (3.3)

Engineering and Design Initiation Rate is primarily driven by the requirements, i.e., the Planned Engineering and Design Work (man-hours / week). It is also affected by the Rework Fraction, Integration Risk, and the Cost Overrun Ratio. The equation for Engineering and Design Initiation Rate, developed during the modeling session, is as follows:

$$\begin{aligned}&\text{Engineering and Design Initiation Rate}\\ &= \text{Planned Engineering and Design Work}\\ &\quad * \text{IntRiskEffectOnAct (Integration Risk)}\\ &\quad * \text{COEffectOnAct (Cost Overrun Ratio)}\\ &\quad * (1 + \text{Rework Fraction})\end{aligned} \qquad \textbf{... (3.4)}$$

Where IntRiskEffectOnAct (Integration Risk) and COActivityEffect (Cost Overrun Fraction Effect) are the lookup functions that show the effect of Integration Risk and Cost Overrun on the Engineering and Design activity. Both lookup functions are defined later in this Chapter (section 3.7.12).

Engineering and Design Completion Rate is dependant on the Engineering and Design effort and the Engineering and Design Residence Time (weeks). Engineering and Design Residence Time is the average time between initiation and completion of any activity. Thus, residence time represents the average delay for the engineering and design activity. It is given by the following relationship.

$$\begin{aligned}&\text{Engineering and Design Completion Rate}\\ &= \frac{\text{Engineering and Design Effort}}{\text{Engineering and Design Residence Time}}\end{aligned} \qquad \textbf{... (3.5)}$$

3.7.2 The Assembly, Installation and Integration Effort

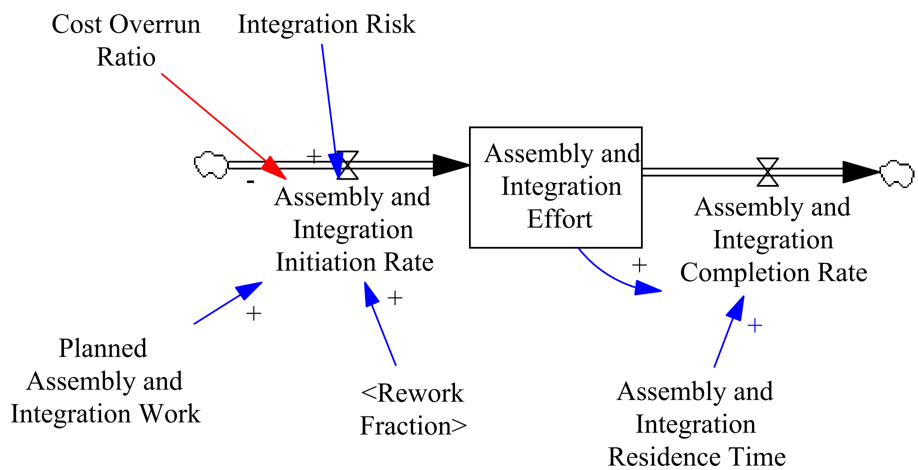


Figure 3-9: Assembly, Installation and Integration Structure

Assembly, Installation and Integration stock is increased by the Assembly and Integration Initiation Rate and is depleted by Assembly and Integration Completion Rate. The Equation for Assembly and Integration Effort is as follows.

$$\begin{aligned}&\text{Assembly and Integration Effort }(t) = \text{A/I Effort }(0)\\&+\int[\text{Assembly and Integration Initiation Rate}\\&\quad - \text{Assembly and Integration Completion Rate}]\cdot dt\\&\text{Assembly and Integration Effort }(0) = 0\end{aligned} \qquad \textbf{... (3.6)}$$

Assembly and Integration Initiation Rate is driven by the Assembly and Integration requirement, i.e., Planned Assembly and Integration Work and it is affected by risk, rework and cost overrun. Its equation was derived as follows

$$\begin{aligned}&\text{Assembly and Integration Initiation Rate}\\&= \text{Planned Assembly and Integration Work}\\&\quad * \text{IntRiskEffectOnActt (Integration Risk)}\\&\quad * \text{COEffectOnAct (Cost Overrun Ratio)}\\&\quad * (1 + \text{Rework Fraction})\end{aligned} \qquad \textbf{... (3.7)}$$

Where IntRiskEffectOnAct (Integration Risk) and COActivityEffect (Cost Overrun Fraction Effect) are the lookup functions that show the effect of Integration Risk and Cost Overrun on the Assembly and Integration activity. Both lookup functions are defined later in this Chapter (section 3.7.12).

Assembly and Integration completion rate is the ratio of the Assembly and Integration Effort and the Assembly and Integration Residence Time. Assembly and Installation Residence Time is the average time between initiation and completion of any activity. Thus, residence time represents the average delay for the Assembly and Installation activity. It is given by the following relationship.

$$\text{Assembly and Integration Completion Rate} = \frac{\text{Assembly and Integration Effort}}{\text{Assembly and Integration Residence Time}}$$

... (3.8)

3.7.3 The Material Procurement

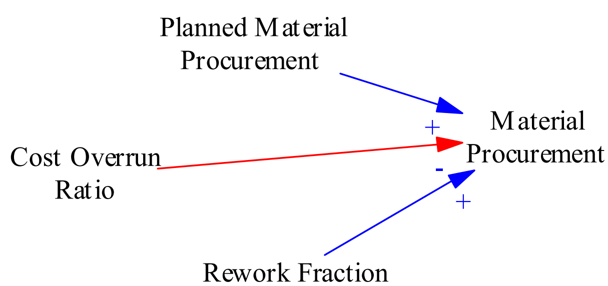


Figure 3-10: Material Procurement Structure

Planned material procurement primarily drives the actual material that is being procured. Material procurement is also affected by the rework fraction and cost overrun ratio. More the rework more is the material being procured. Also, if the cost overrun is more then it reduces the material that is procured.

The equation that is developed for material procurement is as follows:

Material Procurement
= Planned Material Procurement
* COActivityEffect (Cost Overrun Ratio)
* (1 + Rework Fraction) **... (3.9)**

Where COActivityEffect (Cost Overrun Effect on Activity) is the lookup functions that shows the effect of Cost Overrun on the material procurement activity. This lookup function is defined later in this Chapter (section 3.7.12).

3.7.4 The Testing Effort

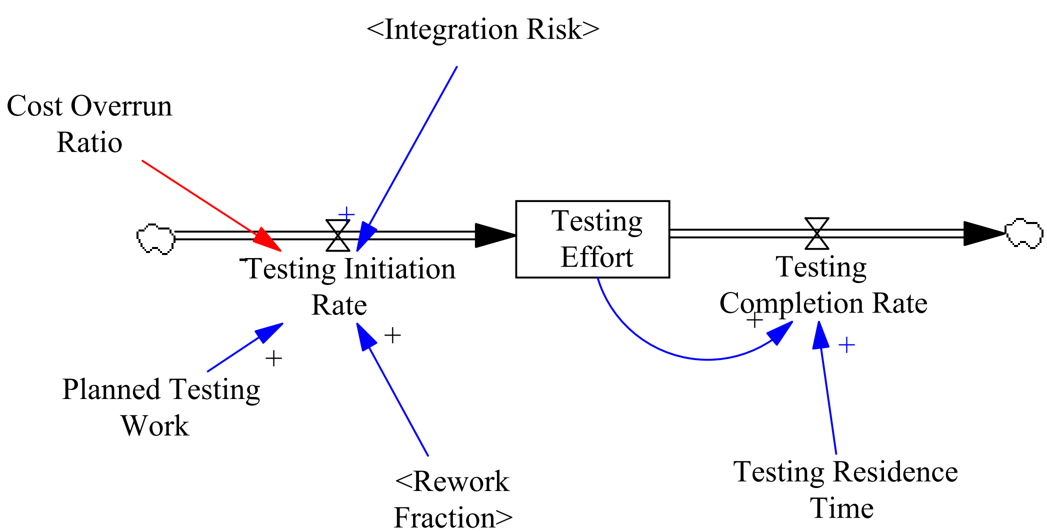


Figure 3-11: Testing Effort Structure

Testing Effort stock is increased by the Testing Initiation Rate and is depleted by Testing Completion Rate. The equation for Testing Effort is as follows.

$$\text{Testing Effort}(t) = \text{Testing Effort}(0) + \int [\text{Testing Initiation Rate} - \text{Testing Completion Rate}] \times dt$$

$$\text{Testing Effort}(0) = 0$$

... **(3.10)**

Testing Initiation Rate is driven by the testing requirement, i.e., Planned Testing Work and it is affected by risk, rework and cost overrun. Its equation was derived as follows

Testing Initiation Rate
= Planned Testing Work
 * IIntRiskEffectOnAct(Integration Risk) ... **(3.11)**
 * COEffectOnAct(Cost Overrun Ratio)
 * (1 + Rework Fraction)

Where IntRiskEffectOnAct (Integration Risk) and COActivityEffect (Cost Overrun Fraction Effect) are the lookup functions that show the effect of Integration Risk and Cost Overrun on the testing activity. Both lookup functions are defined later in this Chapter (section 3.7.12).

Testing completion rate is the ratio of the Testing Effort and the Testing Residence Time. Testing Residence Time is the average time between initiation and completion of any activity. Thus, residence time represents the average delay for the testing activity. It is given by the following relationship.

Testing Completion Rate

$$= \frac{\text{Testing Effort}}{\text{Testing Residence Time}} \qquad \text{... } \mathbf{(3.12)}$$

3.7.5 The Management Effort

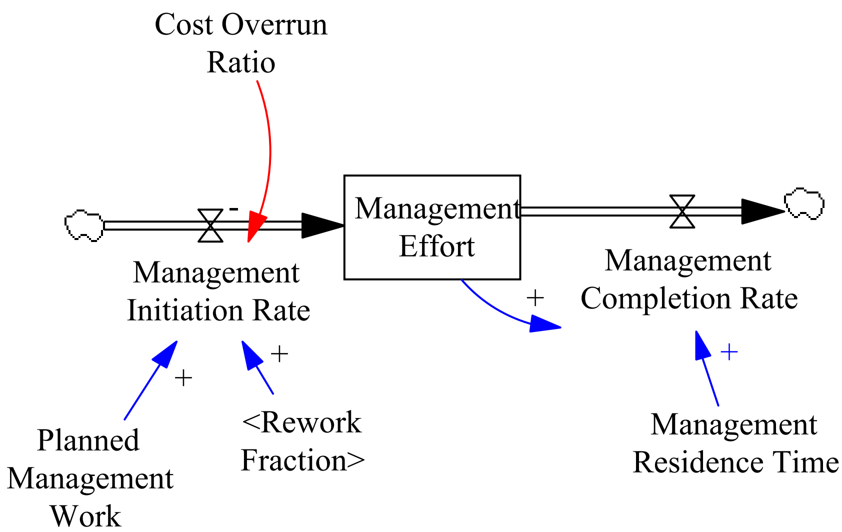


Figure 3-12: Management Effort Structure

Management Effort stock is increased by the Management Initiation Rate and is depleted by Management Completion Rate. Equation for Management Effort is as follows.

$$\begin{aligned}
&\text{Management Effort (t)} = \\
&\qquad \text{Management Effort (0)} \\
&\quad + \int [\text{ Management Initiation Rate - Management Completion Rate}] \cdot dt \\
&\text{Management Effort (0)} = 0
\end{aligned}$$

... (3.13)

Management Initiation Rate is driven by the management requirement, i.e., Planned Management Work and it is affected by rework and cost overrun. Its equation was derived as follows

$$\begin{aligned}
&\text{Management Initiation Rate} \\
&= \text{ Planned Management Work} \\
&\quad * \text{ COEffectOnAct (Cost Overrun Ratio)} \\
&\quad * (1 + \text{Rework Fraction})
\end{aligned}$$

... (3.14)

Where COActivityEffect (Cost Overrun Fraction Effect) is the lookup functions that shows the effect of Cost Overrun on the management activity. Both lookup functions are defined later in this Chapter (section 3.7.12).

Management completion rate is the ratio of the Management Effort and the Management Residence Time. Management Residence Time is the average time between initiation and completion of any activity. Thus, residence time represents the average delay for the testing activity. It is given by the following relationship.

$$\text{Management Completion Rate} = \frac{\text{Management Effort}}{\text{Management Residence Time}} \quad \text{... (3.15)}$$

3.7.6 The Cost Realization Rate

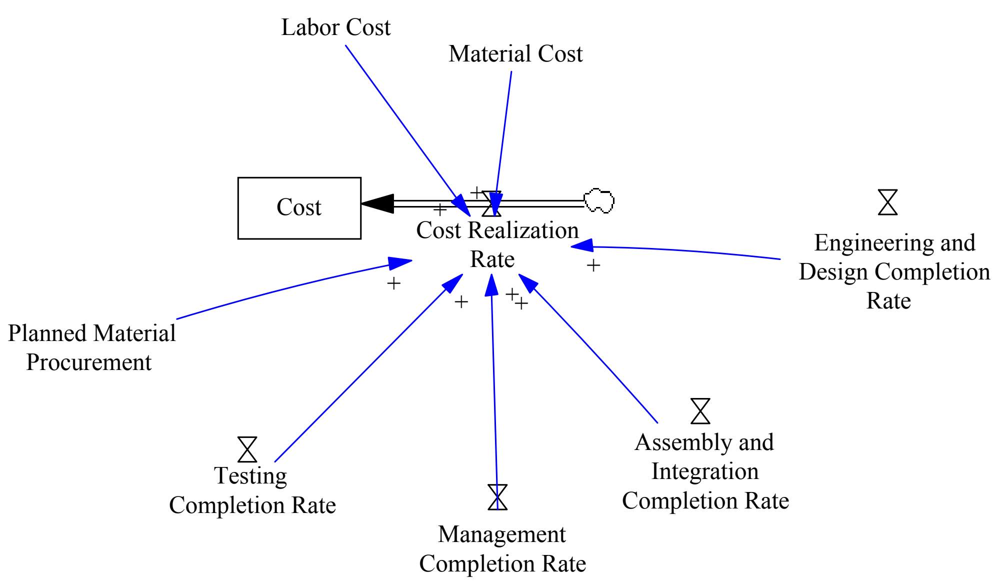


Figure 3-13: Cost Structure

The cost of stock is increased by the Cost Realization Rate. Its equation is as follows.

$$\text{Cost }(t) = \text{Cost }(0) + \int [\text{ Cost Realization Rate}] \times dt \quad \text{... (3.16)}$$
$$\text{Cost }(0) = 0$$

Cost Realization rate shows how the costs of the various activities are accumulated. All the integration activities are feeding into the cost realization rate. In addition to that, labor cost and material cost also provides information input. It converts the man-hours into dollars. The equation is as follows.

$$\text{Cost Realization Rate} = \begin{bmatrix} \text{Engineering and Design Completion Rate} \\ + \text{ Assembly and Integration Completion Rate} \\ + \text{ Testing Completion Rate} \\ + \text{ Management Completion Rate} \end{bmatrix} * \text{ Labor Cost} + [\text{Planned Material Procurement}] * \text{ Material Cost}$$

... (3.17)

3.7.7 The Performance Loop

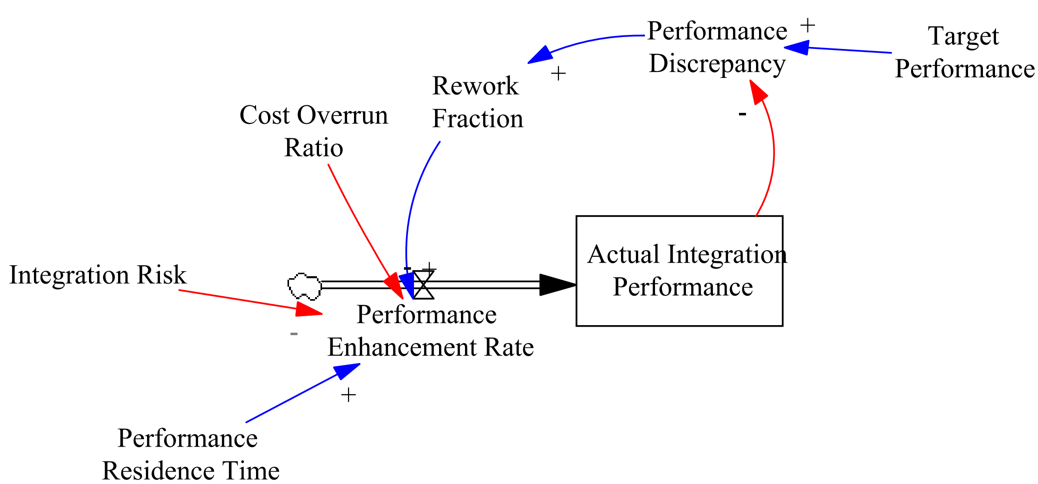


Figure 3-14: The Performance Loop

The above structure shows the balancing structure for the performance loop. The Integration Performance stock is increased by the Performance Enhancement Rate. The Performance Enhancement Rate is primarily driven by the Rework Fraction amount. It is also affected by the Cost overrun ratio and the Integration Risk and is dependant upon

Performance Residence Time. The equation for performance enhancement rate is developed as follows:

$$\text{Performance Enhancement Rate} = \left(\frac{\text{Rework Fraction}}{\text{Performance Residence Time}}\right) * \text{COEffectOnPerf(Cost Overrun Ratio)} * \text{IntRiskEffect On Perf(Integration Risk)} \quad \textbf{... (3.18)}$$

COEffectOnPerf and IntRiskEffectOnPerf are the lookup functions that portray the effect of Cost Overrun Ratio and Integration Risk, viz., on the performance enhancement rate. Thus performance enhancement rate is basically amount of rework done per unit of time. The more the risk, the less is the performance enhancement and the more the cost overrun, the less is the performance enhancement rate. Thus, both of these variables have negative impact on the performance enhancement rate.

The actual integration performance is compared with the Target performance to determine the Performance Discrepancy. Thus, the equation for performance discrepancy is:

$$\text{Performance Discrepancy} = \text{Target Performance} - \text{Actual Integration Performance}$$

Rework depends on the performance discrepancy. More the discrepancy more is the rework done to achieve target performance. It is assumed that this relationship is linear. And moreover, for simplification, it is assumed that rework equals the discrepancy. Discrepancy is measured as a dimensionless variable. So it is safe to make the above assumption that rework equals discrepancy.

One can observe from the loop that more the discrepancy more rework is done. This causes performance to improve at faster rate. This causes the discrepancy to reduce quickly. Now less the discrepancy is the rework. Hence slower is the performance enhancement rate. Thus the loop shows exponential type of behavior. The rate of performance enhancement is very high at the beginning and then as the performance reaches the target performance, the rate reduces.

3.7.8 Asking for Additional Money

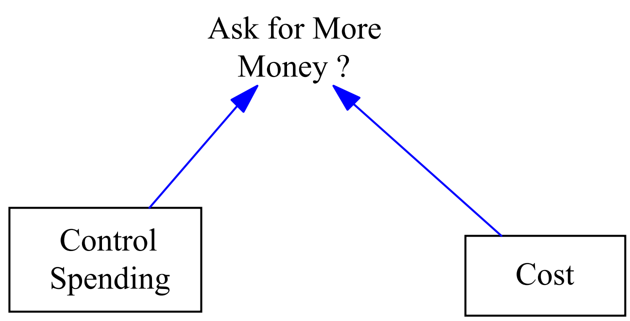


Figure 3-15: Asking for Additional Money

Asking for more money is a very crucial decision during the project. There were many discussions and various thoughts given during the group modeling session to address the question concerning point in time during which the model should ask for additional funding predicting that the cost-overrun will occur sometime in future. The following diagram shows the solution to the problem.

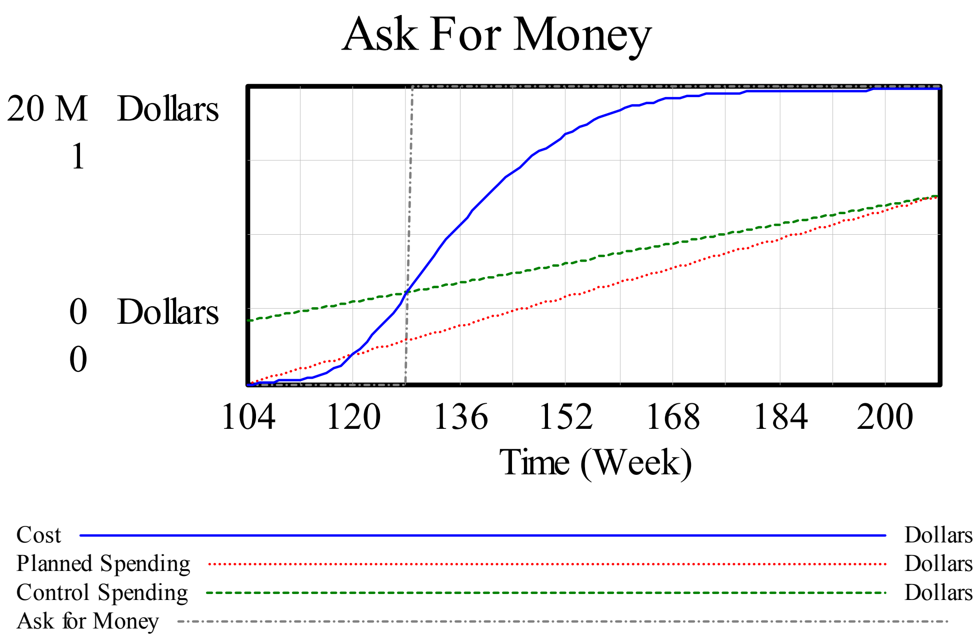


Figure 3-16: Ask for More Money – A Scenario

Figure 3.15 is above is the output from one of the runs of the model. The total funding for the integration is 12.6M. Dividing this funding over period of 2 years, some uniform spending rate called "Planned Spending Rate" was determined. If the money is spent with this rate, then the amount spent (cumulative) will be the one shown by green (dashed) line in the above graph. Thus the green line starts at zero and ends at the 12.6M value. The blue (solid) line is the actual cumulative cost of the project when one runs the model. It can be seen from the graph that there is a cost overrun since the blue line is reaching at 16M value at the end. Also it can be seen that at first few weeks, the rate of increase of cost is less than the planned spending rate. But after that, the rate of increase in cost increases and the blue line crosses the green one at around 20th week. At this point of time, there is no cost overrun in the system. But given the behavior till 20th week, one can foresee, looking at the two lines, that the project will incur cost overruns some time in future. This "realization of future cost overrun" had to be captured in the model. A need for mechanism was realized which could determine when to ask for more money to take care of the future cost overrun. The solution was the red (dotted) line, which corresponds to what called “Control Spending”. This line does not start at value 0 but starts little above 0. But it ends at the value, which is 12.6M. This red line is used as a base line to take decisions for requests for new funding requests. The moment blue line crosses the red line; model sets the trigger and asks for additional money. Thus the model waits for a while and sees if the cost is really increasing after it crosses the green line. This also takes care of the fluctuations in the cost by providing a buffer zone.

There was a problem faced how to decide on the criteria for asking more money. One of the members of the modeling team introduced the idea of the Control Spending Limit, shown by the red line in the diagram. It is a hypothetical line that starts at some positive value of dollars in contrast to the green line starting at zero value. For the current model, this value was decided as 1/3rd of the budget of the project. This line is also cumulative, and finally ends up at 12.6M value. Now, this line is used as a reference to define the criteria for asking more money. Asking for more money is now driven by the position of the blue line as compared with the red line. The moment the blue line crosses the red line, the model decides that it should trigger the request for more money. Thus the model does not ask for money at 18th week, but it waits for a while and asks at around 27th week. The amount of money that the model should ask and how much time it takes to receive new money is explained in the sections 3.7.8 and 3.7.9.

3.7.9 Additional Money Value

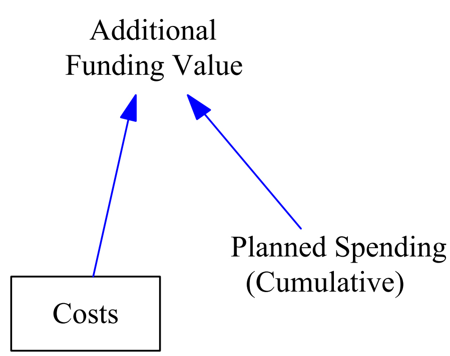


Figure 3-17: Additional Funding Value

One more important decision was to decide how much additional money should be asked for when the model decides to ask for more money. Again looking at Figure 3.16 in section 3.6.7, the model decides at around 27th week that system needs more money. The amount of money it asks at any moment thereafter is the difference between the cost and the planned spending value at that moment. Thus, say at 32nd week when it is going to ask for more money, the amount it will ask for is the difference between the values of blue line and green line at the 32nd week. The logic behind this value is straightforward. The decision maker sees that the amount cost is exceeding the planned spending and asks for the difference between the two. The time when the money actually arrives is explained in the following section.

3.7.10 Additional Money Stock

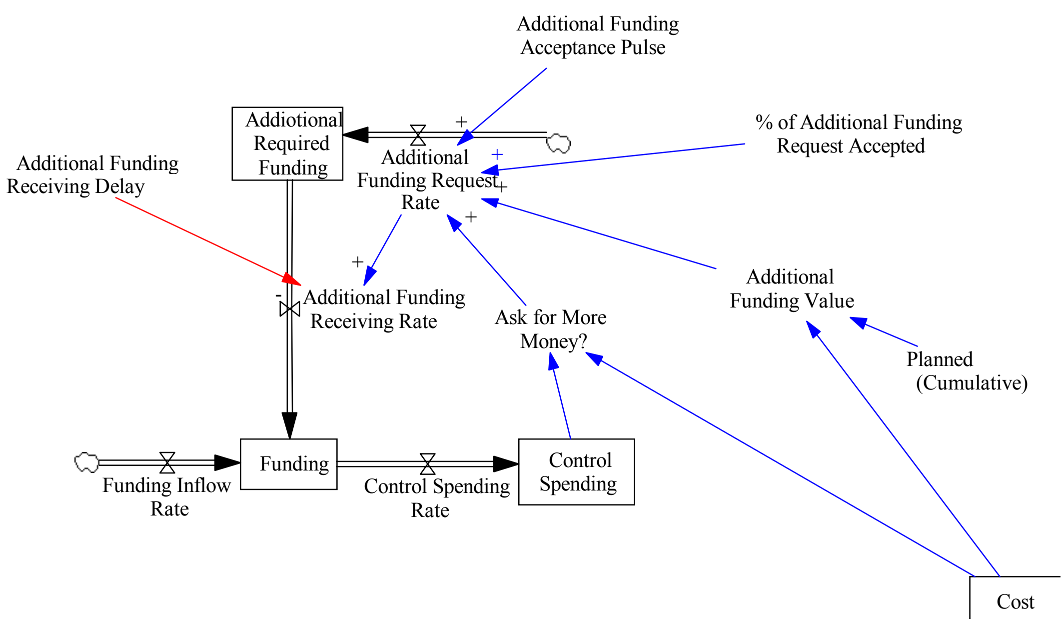


Figure 3-18: Additional Money Stock Structure

Figure 3.17 shows the mechanism of the funding flow, request for additional funding and the delay associated with the additional funding. The main funding stock holds the total initial funding of 12.6M at the start of the project. This stock is depleted by the control-spending rate, as explained in Section 3.6.7. Control spending, Cost and Planned Spending decide when to ask for more money and what amount of money to ask for.

These two variables feed into the Additional Funding Request Rate. Not all the money asked for is accepted, but only some part of it is accepted. This is designated by the term % of Additional Funding Request Accepted. This variable has a maximum value of 1 when all the requested funding is accepted. Typically the value is less than 1.

The request for new money is not immediately accepted, but it is accepted only at specific time periods. After the project begins, it is accepted only at the end of every quarter and it is not accepted during the last two quarters of the project. Thus, for the current system with two-year time horizon, the pulse goes on every quarter until 1.5 years. All these variables finally decide the Additional Funding Request Rate. The equation is as follows:

Additional Funding Request Rate =
Additional Funding Value
*Additional Funding Acceptance Pulse
* % of Additional Funding Request Accepted **... (3.19)**

Additional Funding Request Rate fills the stock of Additional funding. Accepted funding takes some time to receive. This is a "pipeline" delay and it is achieved through the structure above. New funding receiving delay sets the time that is required for money to be "really" available. The new funding inflow rate then feeds into the main funding, thus completing the loop.

3.7.11 Cumulative Funding and Cost Overrun

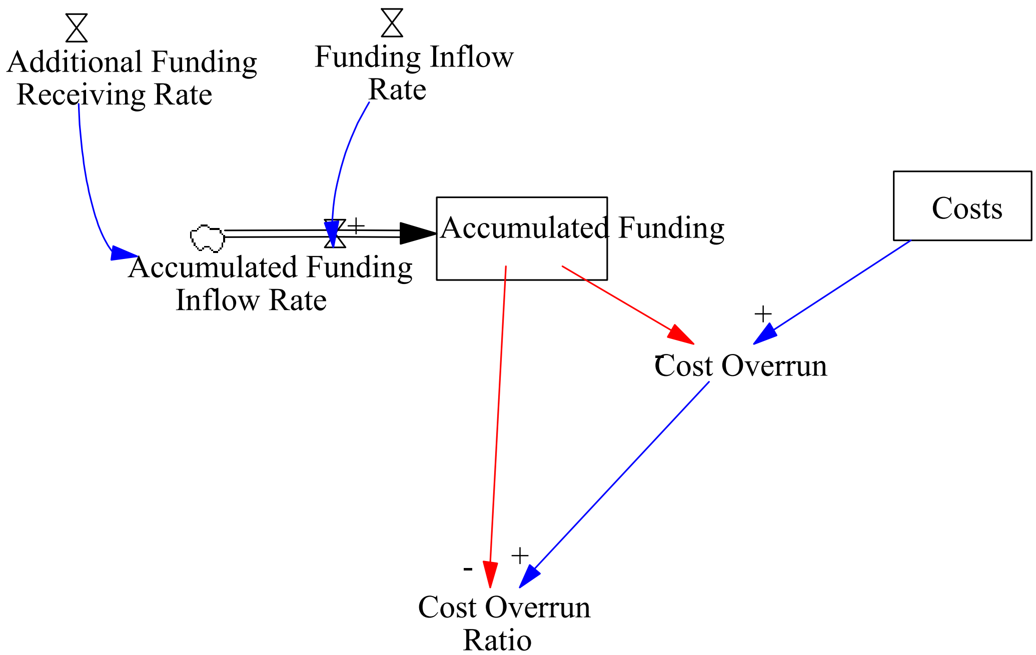


Figure 3-19: Cumulative Funding and Cost Overruns

Cost Overrun is the difference between the cost and the Accumulated funding. The Accumulated funding stock stores the value of total funding received at each point in the simulation. The funding stock cannot be used to determine the cost overrun because it is depleted by the control spending rate and hence it does not store the actual total value of money received. That is the reason that both the Funding Inflow Rate and the New Funding Inflow Rates are feeding into the cumulative Funding Inflow Rate variable that fills the cumulative Funding stock. This stock is not depleted by any flow and thus memorizes the amount of money received. Cost overrun is thus difference between the Cost and the Accumulated Funding. The cost overrun ratio term gives an idea about the percentage of the cost overrun. Cost overrun only gives the value of actual amount of overrun but it does not provide information about the percentage overrun as compared with the cumulative funding. The cost overrun ratio is defined as follows:

$$\text{Cost Overrun Ratio} = \frac{\text{Cost Overrun}}{\text{Accumulated Funding}} \qquad \textbf{... (3.20)}$$

3.7.12 Lookup Functions

Various lookup functions are defined in the model. Lookup functions are typically used to develop the relationship between two variables when the input variable has a varying effect on the output variable for different values of input variable. For example, consider the engineering and design effort. Integration risk affects this effort in some fashion. For the current model, at risk value of 2, it increases the engineering and design effort by 2%, at risk value 6, this factor is 13% while at value 10 it is 75%. Lookup function stores this information and uses it to find out the effect of risk values between 1 and 10. One more advantage of lookup function is that it assumes a linear relationship between the two specified points. E.g. the lookup function for risk is defined at the values of 0, 2, 4 and so on. To calculate the effect at value 1, it considers the values at 0 and 2, assumes a linear relationship between these two values and then determines the value at 1. Thus, instead of specifying some complex curvilinear graph and finding out its equation, it is easier and safe enough to approximate the linear relationship between a few specified points. It has one more advantage in terms of the equation elicitation. The modeler does not need to work hard to extract the tacit information from the mental models of the user to decide the shape of the curve. But he can just ask the user to specify the values at a few discrete points.

3.7.12.1 IntRiskEffectOnAct

This lookup function specifies the effect of integration risk on the engineering/design activity and the assembly/installation and testing activity. It is defined as follows in the model

[(0,0)-(10,2)], (0,1),(2,1.02),(4,1.035),(6,1.13),(8,1.36),(10,1.75)

This function is depicted by the following Figure.

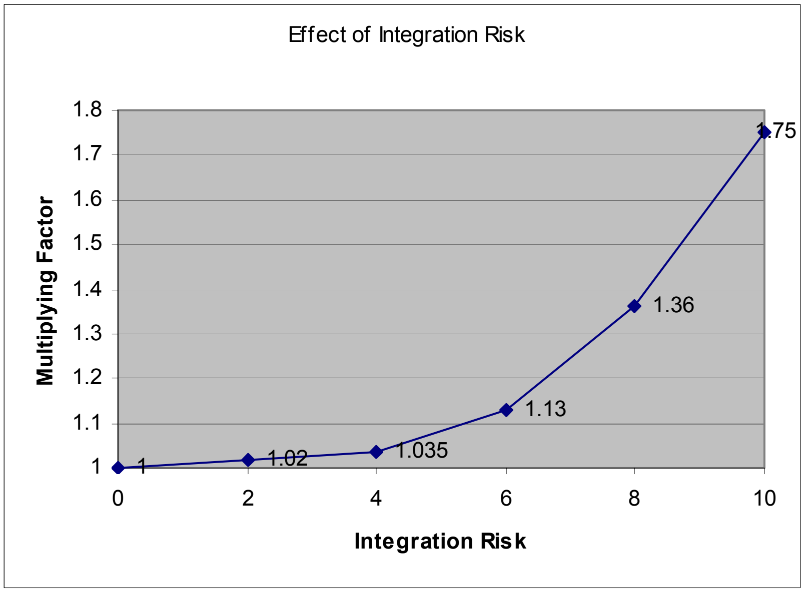


Figure 3-20: Effect of Integration Risk on Integration Efforts

At risk value of 0, it has no impact on the activity. At risk value of 2, the effort is increased by 2% hence the multiplying factor is 1.02. At risk value of 10, the effort is increased by 75% hence the multiplying factor is 1.75.

3.7.12.2 COActivityEffect

This lookup function shows the relationship of cost overrun on the effort rate variables. It is defined as follows in the model:

[(0,0)-(1,1)],(0,1),(0.5,0.8),(1,0)

This function is depicted by the following Figure. It is a modeling assumption

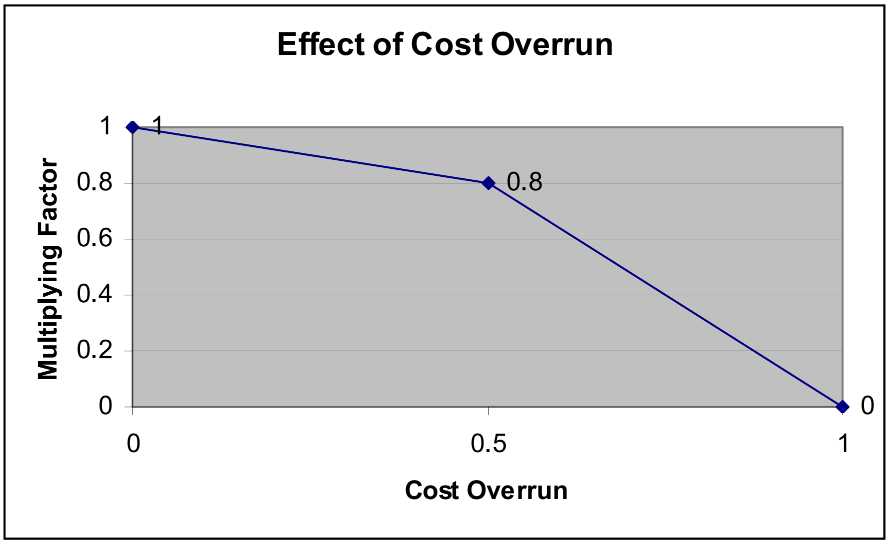


Figure 3-21: Effect of Cost Overrun on Integration Efforts

3.7.12.3 IntRiskEffectOnPerf

This lookup function defines the effect of Integration risk on the performance enhancement rate. Risk has an adverse on the performance enhancement rate . The function defined as follows in the model:

[(0,0)-(10,1)],(0,1),(1,1),(2,1),(3,1),(4,0.6),(5,0.6),(6,0.6),(7,0.25),(8,0.25),(9,0.25),(10,0.25)

This function is depicted by the following Figure.

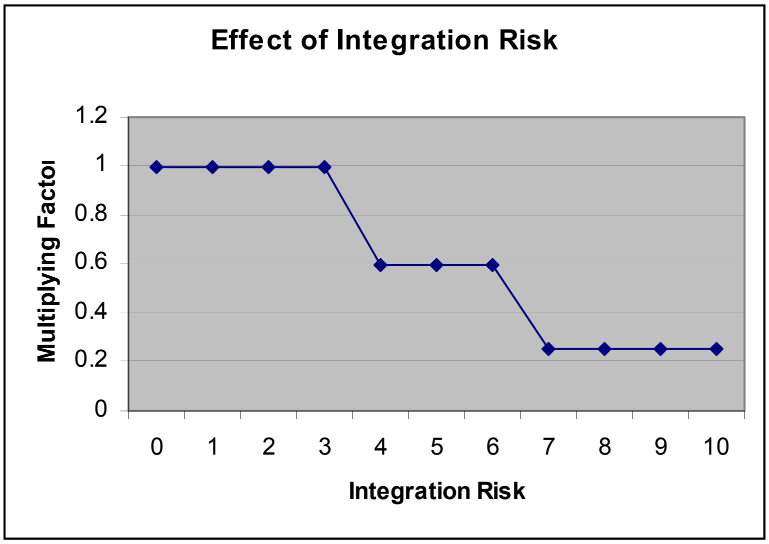


Figure 3-22: Effect of Integration Risk on Performance

3.7.12.4 COEffect

This lookup function defines the effect of cost overrun on the performance enhancement rate. Cost overrun has adverse effect on the performance enhancement. The function is defined as follows in the model:

[(0,0)-(1,1)],(0,1),(0.25,0.75),(0.5,0.5),(0.75,0.25),(1,0)

This function is depicted by the following Figure.

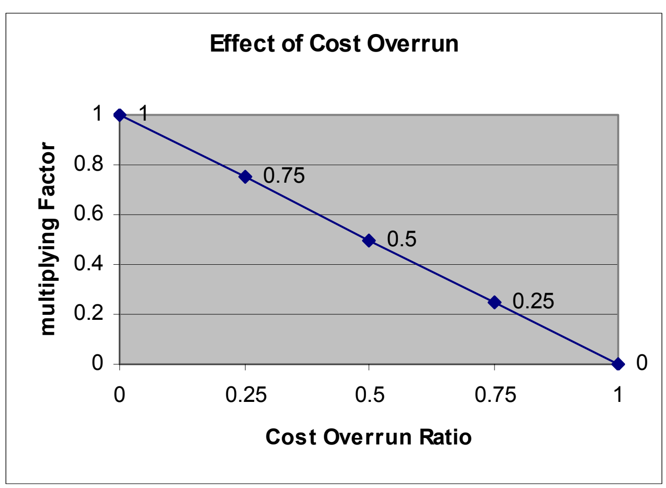


Figure 3-23: Effect of Cost Overrun on Performance

3.8 Inflation

An additional minor version of the above model was developed that addresses inflation in it. Technology integration phase runs for 2 years, which may not be a significant period to address inflation. But the whole life-cycle model runs for 54 years. This period is significant enough to consider inflation. To consider this fact, additional structure was added to the model.

Inflation rate is usually determined from the CPI (Consumer Price Index) over last few years. This inflation rate is an input to the model. Thus the decision maker receives value of inflation rate from outside sources, which is then fed into the model. Based on this inflation rate, the cost is inflated every year in the model. Thus, whatever the cost value is at the end of the year, it is multiplied by the inflation rate and that amount is added to the cost.

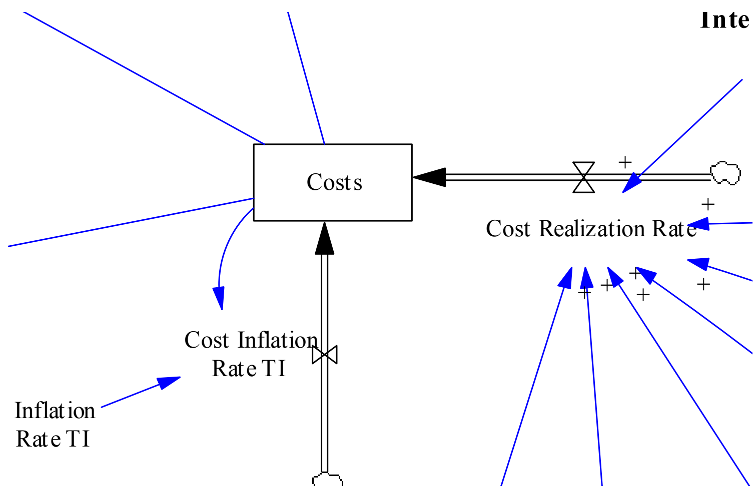


Figure 3-24: Cost Inflation Rate

Thus, in addition to the cost realization rate (described in section 3.7.6 above), an additional inflow flows into the cost stock which is Cost Inflation rate TI. This rate adds the inflated cost amount to the cost stock at the end of every year. The Inflation Rate TI is the input variable, which is the inflation rate based on the previous CPI values. The inflation rate TI variable is a pulse variable defined in VENSIM that triggers itself only once a year. The Cost Inflation is calculated as follows:

Cost Inflation Rate TI =

Costs * Inflation Rate TI

Thus, the cost inflation rate IT remains zero for the rest of the time except once a year. The results of this structure are discussed in next chapter.

Chapter 4: Results, Testing, Verification and Validation

The model discussed in Chapter 3 was programmed in VEMSIM DSS software version 5.0c. The simulation, its results, and various tests conducted on the model are presented in this chapter. The model was run for a specific technology and the input values for the model were the real world values that came from experts from the Navy and the Shipbuilding Company experts. The model output values will differ for different technologies but the behavior will remain the same.

The model used was the "integrated" model in which the other two subsystems modules (Technology Development and Operation, Support & Disposal) of the lifecycle were linked to the Technology Integration sub-module to construct an overall model. The technology development module feeds some of its variable values to the technology integration sub-module. Technology integration sub-module, then, feeds forward some variable values to the operation, support & disposal sub-module. The variable values for technology development risk and performance are passed onto Technology Integration subsystem whereas technology integration labor and material cost values are passed onto the Operation, Support & Disposal subsystem, along with integration risk.

The integrated model was run for different sets of input values of risk, technology maturity and complexity and the particular behavior of the important variables such as cost, performance was identified. In addition, sensitivity analysis was carried out to see how sensitive the model is to its input variables. Furthermore, various tests (described in detail in Section 4.5) were carried out on the model to test its validity. Verification and validation of the model is very critical to make sure that, with some assumptions, the model replicates the real world system and the outputs of the model have meaning.

In addition to the development of the model in VENSIM software, a Decision Support System was created that adds a lot of usability to the model. As seen from the client's perspective, varying the user input values from the simulation software is a complicated process and requires knowledge of the VENSIM simulation software. Working with the VENSIM software requires a considerable amount of effort that is dedicated to the input of the parameter values and detracts from observing the output and making decision analysis. To reduce the time required to work with the VENSIM simulation software, a simple and very user-friendly graphical user interface was developed using Visual Basic programming language. This application program is termed as Decision Support System Software (DSS Software). Very easy to use screens of the DSS Software accept values from the user and feed them back to the simulation software that is tied to it. Moreover, the output graphs can also be selected and observed for analysis. Some screenshots from the DSS software are presented in Appendix D along with its user's guide.

4.1 Simulation Values

The simulation was run for the following values. These values are specific for a technology and they come from government and shipbuilding industry experts.

4.1.1 Simulation Control Parameters

For the technology integration sub-module:

FINAL TIME = 208 weeks

INITIAL TIME = 104 weeks

TIME STEP = 1 week

The overall model runs through the 2808 weeks with initial time 0. The Technology Integration activities start at time 104th week. Though the final time for these technology integration activities is 208 weeks, some activities continue for a while after 208th week to satisfy the requirements. Time step of 1 week was decided to be appropriate for the simulation purpose. Validity of this time step was further checked under model testing and is discussed in section 4.5.

4.1.2 User-Defined Parameters

Listed below are the user-defined parameters. These values are specific to the technology that is being evaluated and have been obtained from the engineers at the shipyard.

Table 4.1: User-Defined Parameters

Parameter	Definition	Value
Labor Rate	The average cost of labor per man-hour.	$75/Man-Hr
Material Cost	The average cost of material (in dollars) for each unit used for Technology Integration activities.	$160K/unit
Funding	The total amount of money (in dollars) budgeted for the Technology Integration process. This is the total project funding for all technology integration activities.	$12.6M
Additional Funding Receiving Delay	The average time delay (in weeks) for receiving additional funding after the request for additional funding is made.	4 Weeks
Percentage of Additional Funding Request Accepted	The percentage of additional funding requested that is actually received.	45%
Technology Integration Initial Performance	This is the initial performance as a percent of target or expected performance.	20%
Planned Engineering and Design Work	The effort (in man-hours) anticipated for TI engineering and design activities.	17,143 Man Hours
Planned Assembly and Installation Work	The effort (in man-hours) anticipated for TI assembly and installation activities.	37,143 Man Hours
Planned Material Procurement	The number of units anticipated to procure for TI.	72 Units
Planned Testing	The planned effort (in man-hours) for TI	2,857 Man

Work	testing activities.	Hours
Planned Management Work	The planned effort (in man-hours) for TI management activities.	2,704 Man Hours
TI Engineering and Design Delay	The average delay (or elapsed time) for engineering and design activities.	1 Week
TI Assembly and Integration Delay	The average delay (or elapsed time) for assembly and integration activities.	1 Week
TI Testing Delay	The average delay (or elapsed time) for testing activities.	1 Week
TI Management Delay	The average delay (or elapsed time) for management activities.	1 Week

For the planned integration activities (such as engineering/design, testing etc.), the man-hour efforts have a certain distribution over the total period of the integration process. The distribution pattern for each of the activities was obtained from the Shipbuilding company experts. For the current model and for the technology being evaluated, the planned distribution patterns for each of the main activities associated with technology integration have been hard-coded into the model. The following graphs show the planned distribution patterns of the effort associated with each of the main integration activities. Management work is assumed to be distributed evenly over the total period.

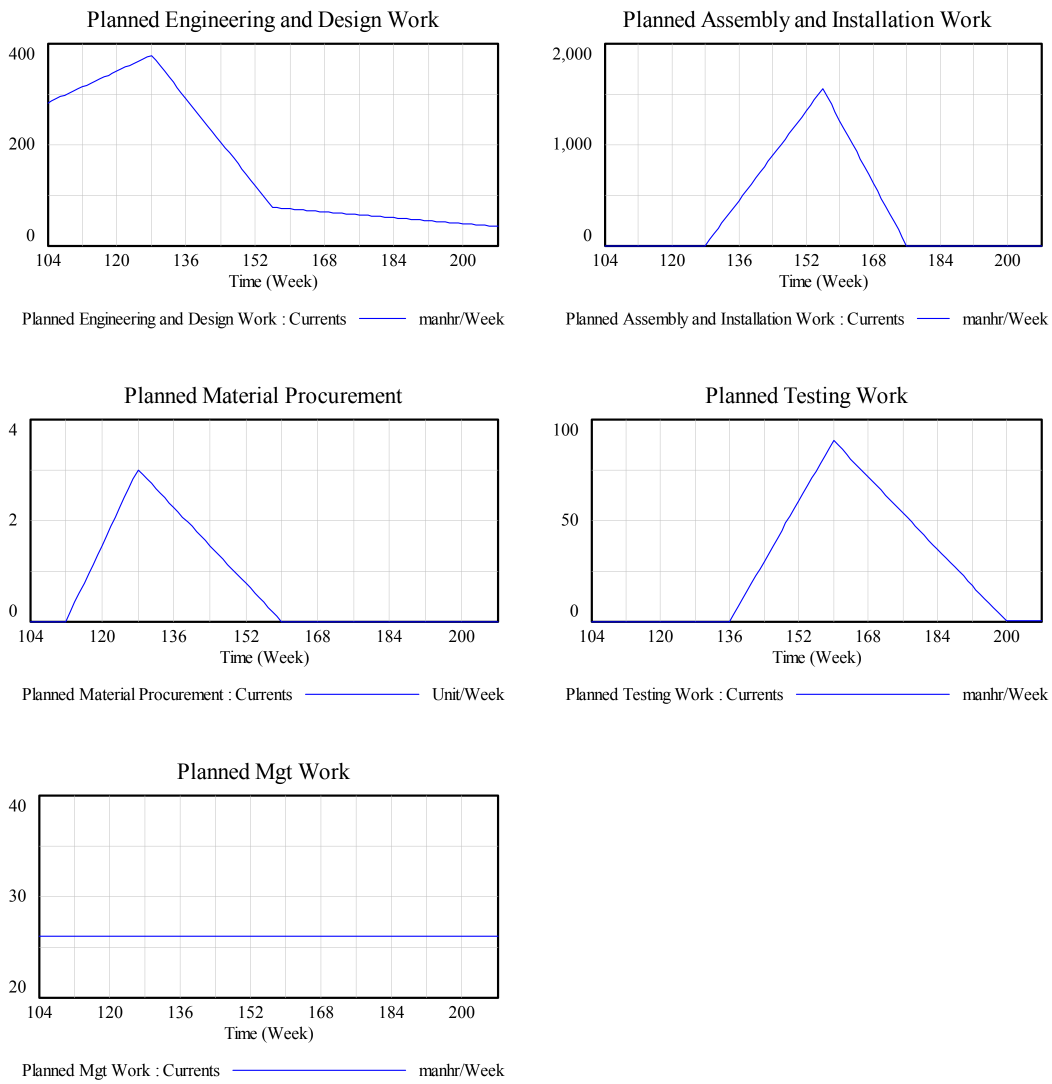


Figure 4-1: Distribution Patterns of the Planned Work for Each of the Major Technology Integration Activities

4.2 <u>Simulation Runs and Results</u>

The simulation was run for the previously defined baseline values that correspond to a specific technology. The output variables of interest are cost and performance. Particular behavior patterns were observed for these two variables. Cost showed "S-shaped growth" while performance showed "Goal Seeking" behavior. These behaviors are explained in sections 4.2.1 and 4.2.2 respectively.

The behavior of any variable is governed by the various feedback loops present in the system. There are large number of feedback loops associated with the two performance variables, i.e., cost and performance. Which loops dominate at specific time periods determine behavior of the performance variables.

4.2.1 S-Shaped Growth

This type of behavior is observed when both positive and negative feedback loops are present. The positive loop is stronger in the beginning where the state of the system grows exponentially. But no system can grow indefinitely and then the negative loop becomes dominant. This causes reduction in the rate of increase of the state of the system, causing an S-shaped behavior. The following loops were identified as those loops that play role in determining the S-shaped behavior of cost. The cost is a part of around 50 loops.

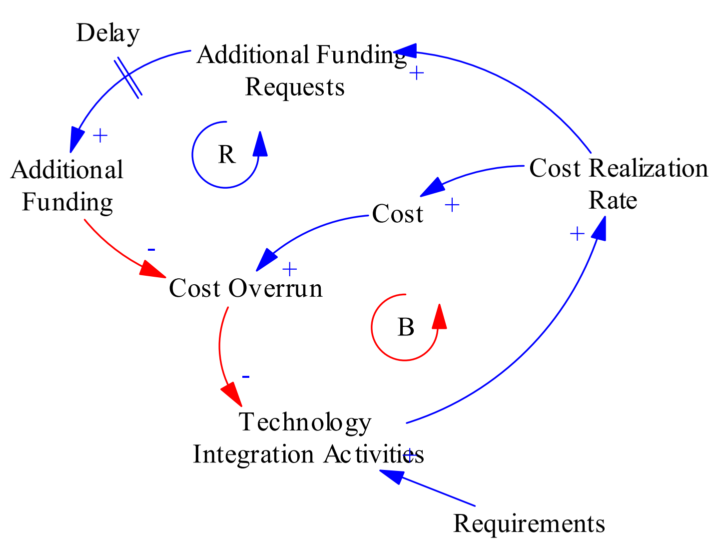


Figure 4-2: Causal Loop Diagram Explaining the Behavior of Cost

Various Technology Integration Activities are primarily driven by the requirements. Requirements, in general, are at low levels at the start of the project. Then they grow as the project advances and then again they are reduced towards as the project approaches end. The typical requirements pattern for the engineering and design effort activity is as shown below:

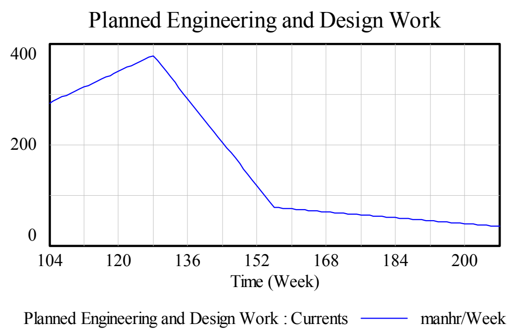


Figure 4-3: Planned Engineering and Design Work

There are various such activities in the project, with each having their own requirements pattern. This increase in the requirements in the early stages of the project increases the cost of the project at very fast rate, typically observed as exponential growth. As more and more effort is put into the integration activities, the Cost Realization Rate – the rate at which cost increases – increases. Based on the other structure in the model (the additional funding request structure, described in section 4.2.3) the model sees if this Cost Realization Rate is sufficient enough to cause future cost overruns. If this is the case then the model requests additional funding. This causes additional funding to come in. It reduces any cost overruns that might have already occurred. On the other hand, cost overrun has a negative impact on the integration activities. Thus, reduction in cost overrun causes an increase in integration activities. Though the "additional funding request" model structure "sees" or "predicts" the future cost overruns and requests additional money before the cost overrun occurs, the cost overruns occur due to the delay in receiving additional funds and because the additional funding request is only partially satisfied. The occurrence of the cost overrun reduces the integration activities. This completes the negative loop. In addition to these loops there are many more insignificant loops that affect the cost variable behavior. The combined effect of these dominant loops and the requirements causes S-shaped growth. The output graph for cost for the previously defined set of parameters (4.1.1 and 4.1.2) is shown below:

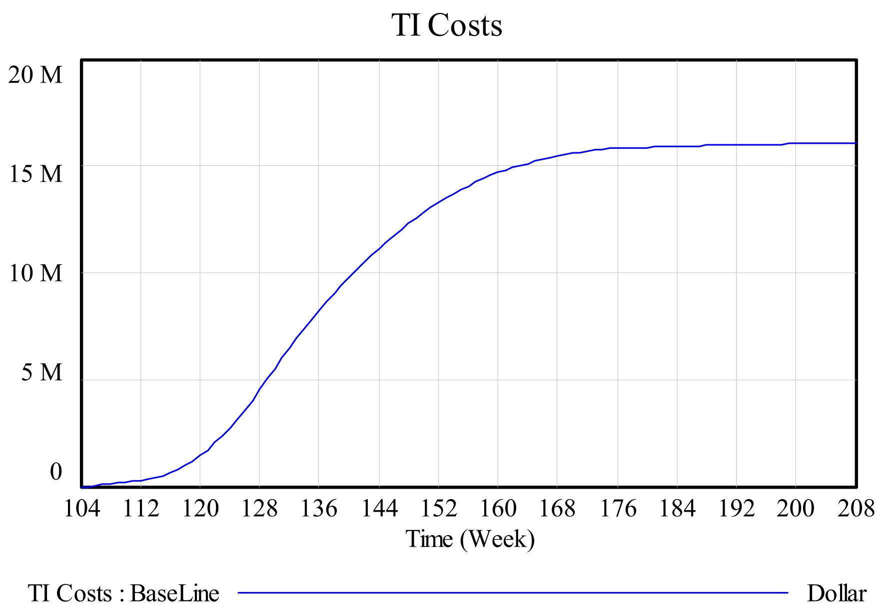


Figure 4-4: Actual Cost Behavior

(The time axis shows week from 104 through 208 because all the runs are carried out on the integrated model. The earlier subsystem of the model runs through 0 through 104 weeks.)

The technology integration cost behavior over the total lifecycle of the technology implementation process looks like follows:

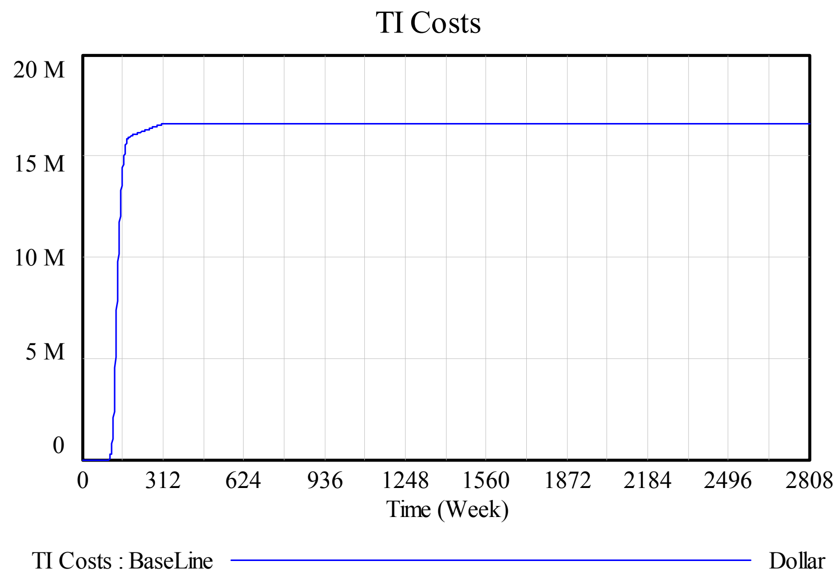


Figure 4-5: TI Cost over Life Cycle

It is interesting to see the cost behavior after 208th week (when the technology integration activities end). This looks like follows:

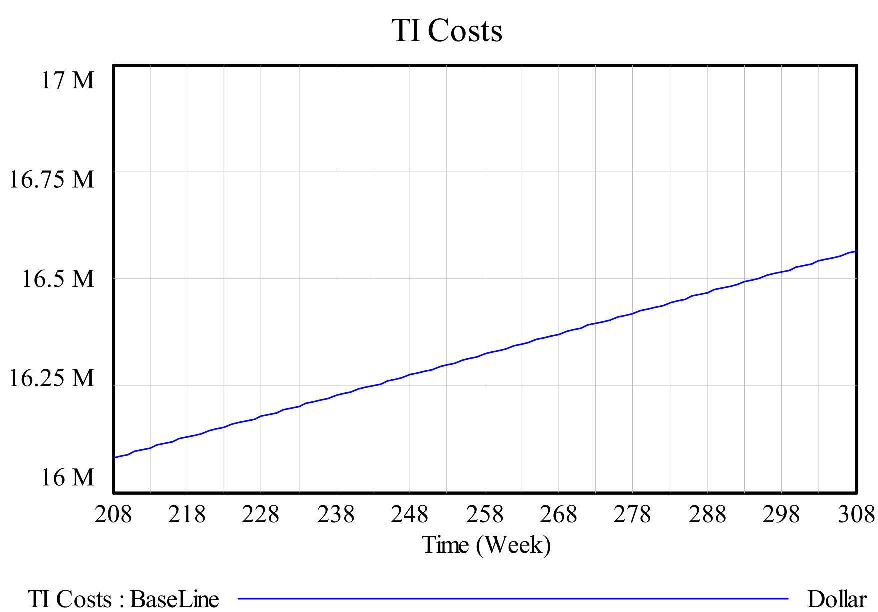


Figure 4-6: TI Cost over Technology Integration Period

The costs are still increasing after 208th week, which is because of the delays present in the system. Though the scheduled activities end at the 208th week, because of the input parameters such as risk and technology complexity and maturity, more effort is required to put into these activities. That causes delay in the system and part of the integration activities are still carried out after 208th week, which causes rise in costs.

4.2.2 Goal Seeking Behavior

This type of behavior is observed in a negative self-balancing loop. The state of the system is compared with the target state and a discrepancy is determined. Based on the discrepancy, corrective action is taken that reduces the discrepancy. The larger the discrepancy the larger is the corrective action. Due to the corrective action, the state of the system is raised towards its target state. Thus the state of the system approaches the target state. This type of behavior is observed for the performance variable. Technology Integration Performance starts with the initial performance, which is measured as percentage of the target performance[2] (level of 20% in this case). Performance Discrepancy is determined by comparing it with the Target Performance. Based on the discrepancy, Rework is carried out that enhances the Performance.

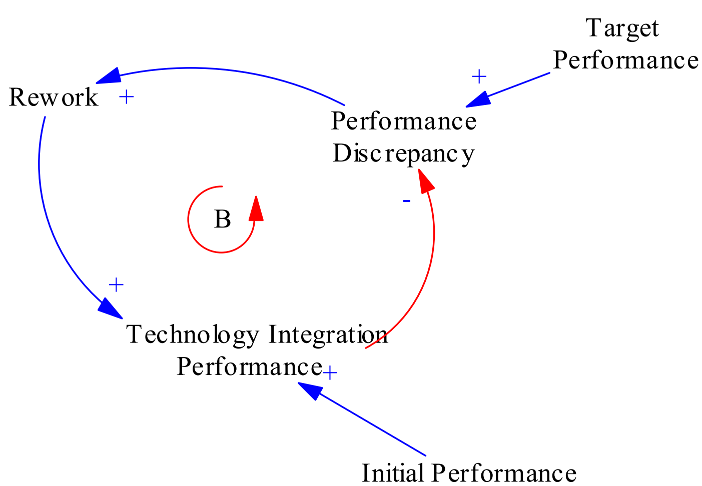


Figure 4-7: Causal Loop Diagram Explaining the Goal-Seeking Behavior of Performance

[2] The physical significance of performance is explained in chapter 3, section 3.2.2

The output graph for the performance variable (for the input set of parameters in section 4.1.1 and 4.2.1) is as follows:

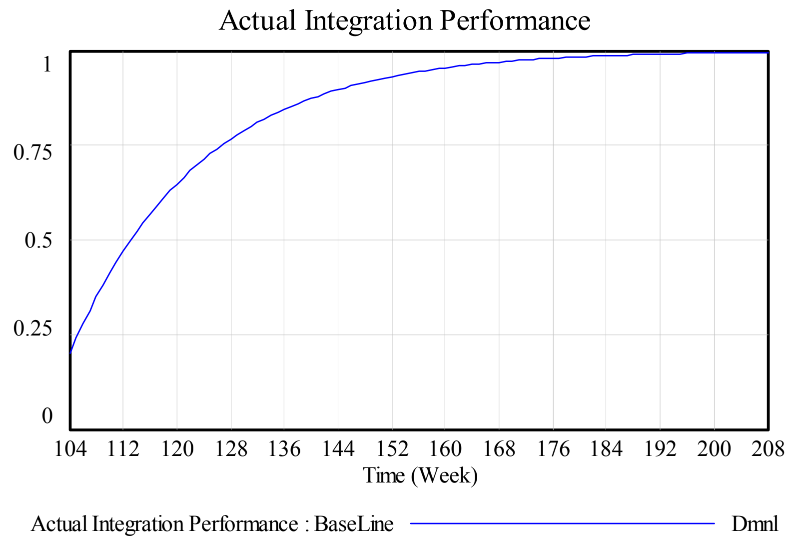


Figure 4-8: Actual Integration Performance

4.2.3 Technology Integration Activities

The graphs for various technology integration activities along with the planned efforts are shown below.

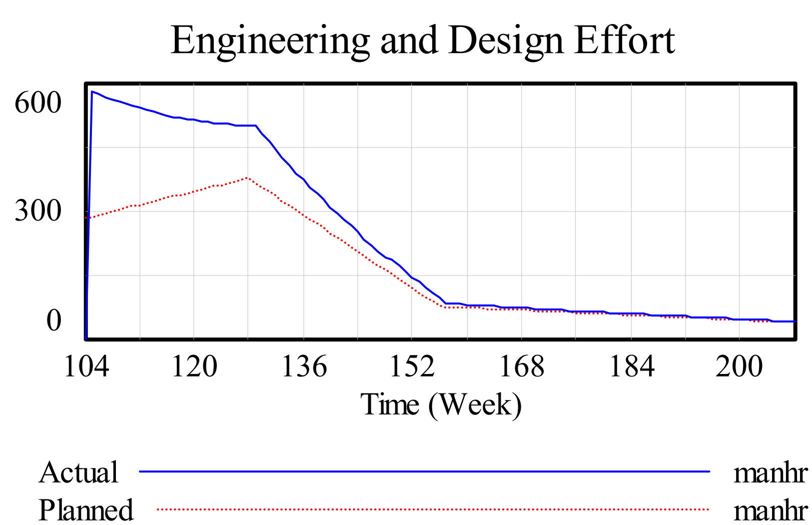


Figure 4-9: Actual and Planned Engineering Design Effort

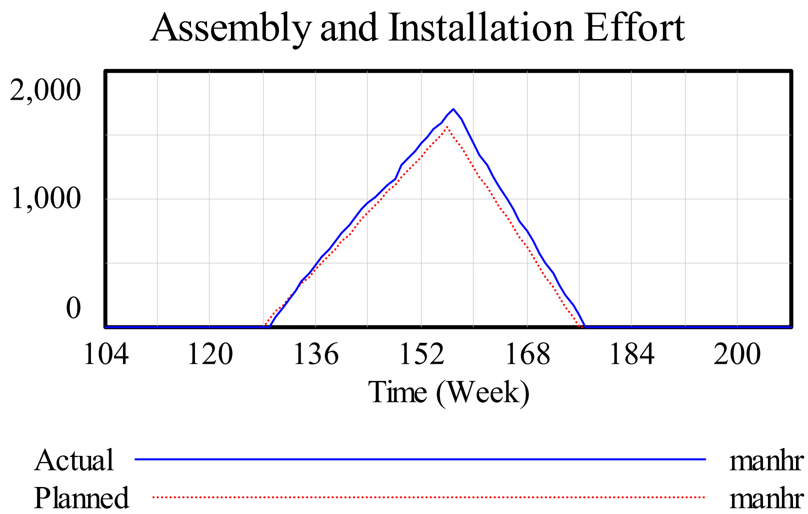


Figure 4-10: Actual and Planned Assembly and Installation Effort

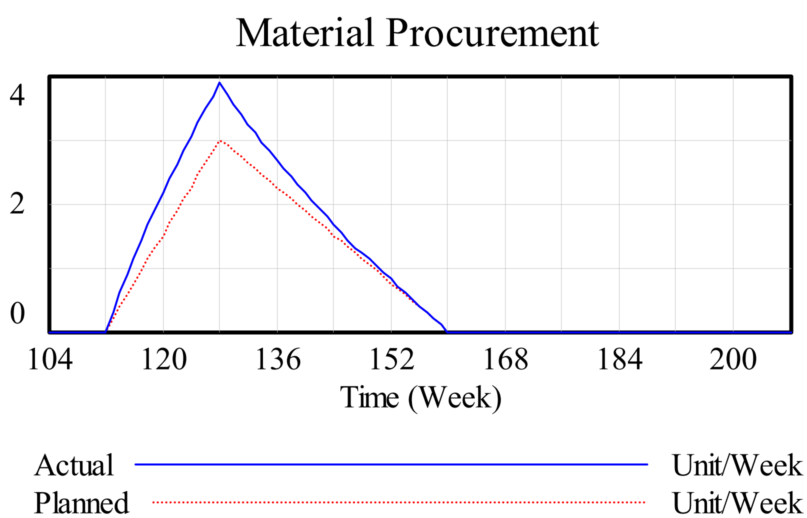


Figure 4-11: Actual and Planned Material Procurement

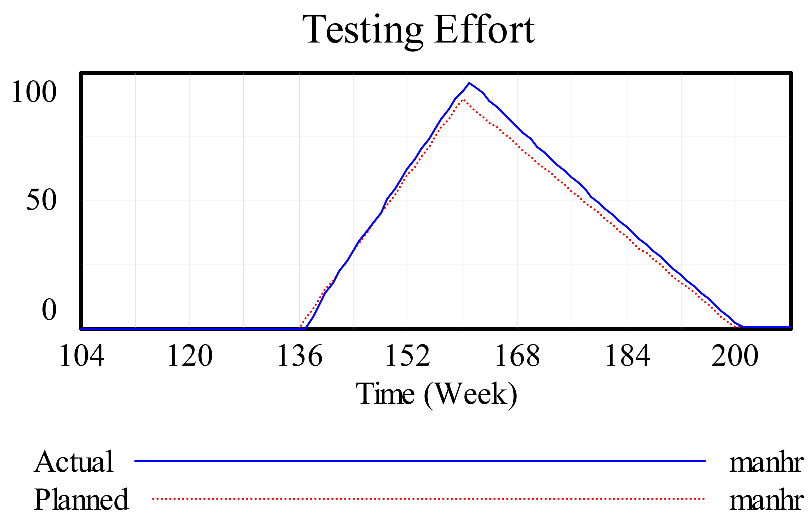


Figure 4-12: Actual and Planned Testing Effort

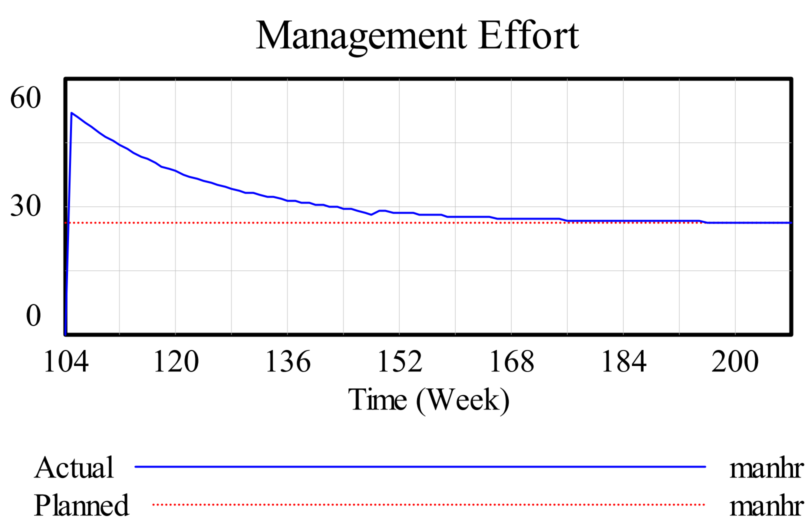


Figure 4-13: Actual and Planned Management Effort

In all cases, the actual effort is more than the planned effort. This is because of the consideration of risk and rework in the current model. The engineers did not consider these issues when they provided the initial profiles of the distribution of work for the various technology integration activities.

4.2.4 Additional Funding

One of the interesting parts of the model is the structure that captures the behavior for additional funding requests. The primary problem behind coming up with this structure was when to send the request for additional funding. The program manager will not wait until the cost overrun actually occurs, to ask for more funds. He/she has to decide, looking at the rate at which cost increases, whether there will be any future cost overruns and when will these occur.

The capability to make decisions about when to ask for more funding needed to be incorporated into the model. An idea that was discussed was to compare the cost realization rate with the planned spending rate. This fixed planned spending rate is the rate at which money will be spent uniformly over the duration of the project. The comparison would give a basis for triggering the requests for additional funding. But the flaw of this idea is that cost realization rate could go above the planned spending rate very early in the technology integration life-cycle when there are no significant signs of future cost overruns. Also, small fluctuations in these rates would trigger "false" requests for the additional funding.

This idea of using the planned spending rate was further augmented with the concept of "control spending". Control spending assumes that a certain amount of money is spent upfront (1/3rd of the total budget) and the remaining amount is spent uniformly over the duration of the project. This creates a gap between the planned spending and control spending. Now, the control spending and the cost were used as a basis for additional funding requests. This avoided early requests for the additional money. This also avoided any possibility of creating false alarms due to fluctuations in the cost realization and the planned spending rates. This can be better explained and observed in the following graph:

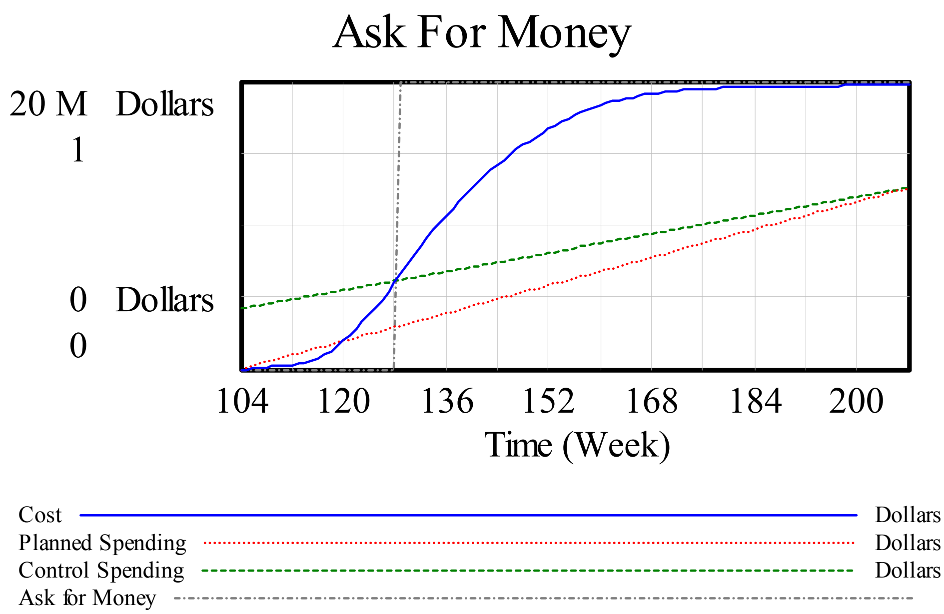


Figure 4-14: Request for More Funding

This graph is a cumulative graph. The blue (solid) line corresponds to the cost. Red (dotted) line is a planned spending line and green (dashed) line is the control spending line. The Ask for Money line (Gray, dot-dash) is triggered only when the blue line crosses the red line and not the green line. This ensures that cost realization rate has been above the planned spending rate for a while, thus avoiding false alarms due to fluctuations. Furthermore, it avoids early trigger for additional money request.

4.2.5 Cost Overruns

Along with the cost variable, cost overrun is also the variable of interest. The cost overrun graph is as follows:

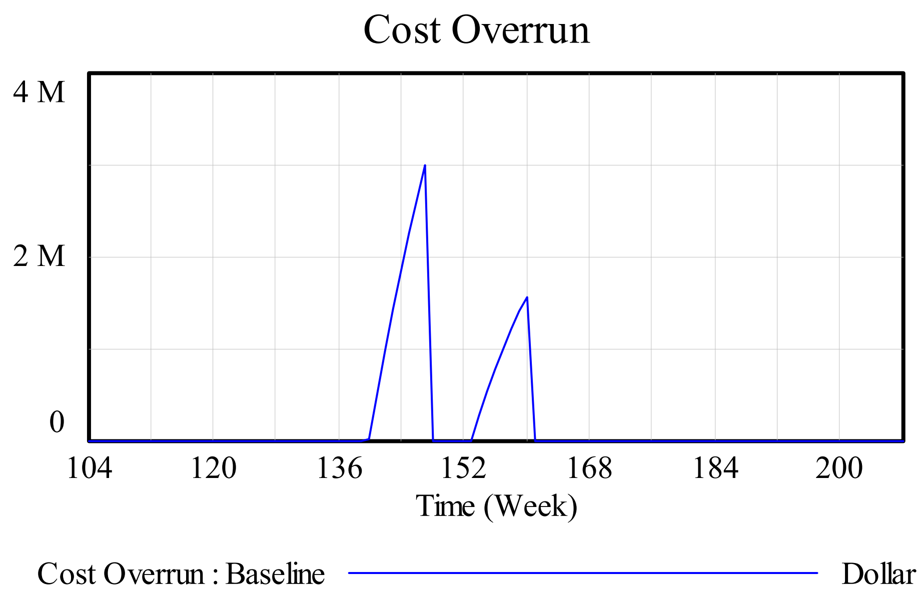


Figure 4-15: Cost Overruns

First peak starts at around 140th week and keeps rising till around 148th week. It then suddenly drops down to zero because at this moment, a pulse of additional funding comes in, which satisfies all the cost overrun. But this additional funding is not sufficient enough to avoid another cost overrun. The other cost overrun is seen starting around 153rd week, which rises till 160th week. At this point, again some additional money comes in to bring overrun down to zero.

4.2.6 Inflation

The ad-hoc inflation structure is described in section 3.8. The following figure shows the cost with and without considering inflation.

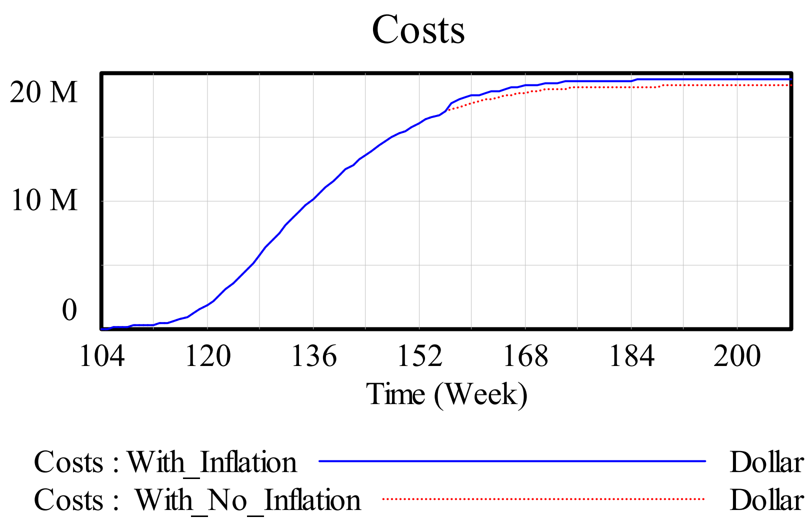


Figure 4-16: Costs due to inflation

It shows that at the end of year 1, (year 3 for the life-cycle) cost jumps up by some amount. This additional amount is because of calculating inflation at that point and adding it to the cost. Since the technology integration runs only for 2 years, inflation is seen only once in the technology integration phase.

4.3 <u>Hypothesis Testing</u>

<u>Hypothesis 1:</u> *The more complex the technology is the less affordable it becomes in terms of technology integration process*.

This hypothesis was tested for varying degrees of technology complexity. The graph that follows shows the variations of cost and performance for very little complexity (value of 1), medium complexity (value of 3) and very high complexity (value of 5). The costs for low, medium and high complexity are $20.24M, $21.95 and $23.18M respectively. Thus, there is approximately 15% increase in cost when complexity of technology becomes high as opposed to low. This increase is significant and thus it proves the hypothesis.

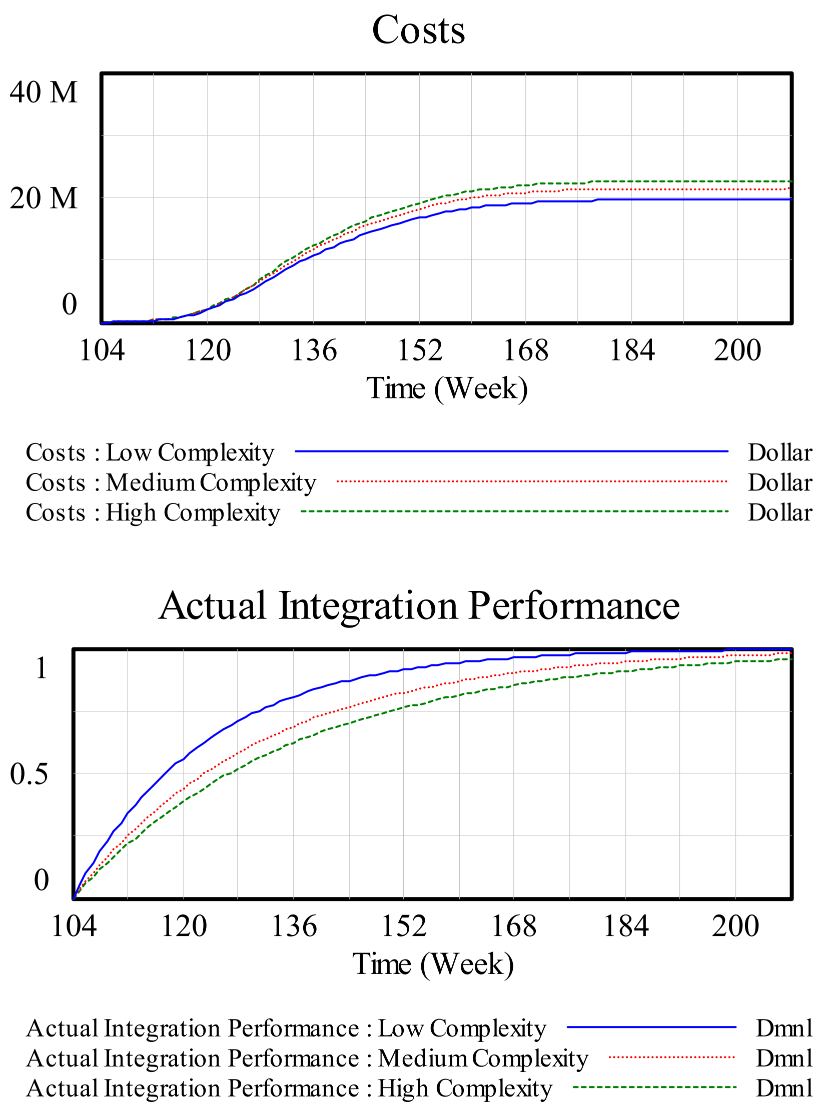
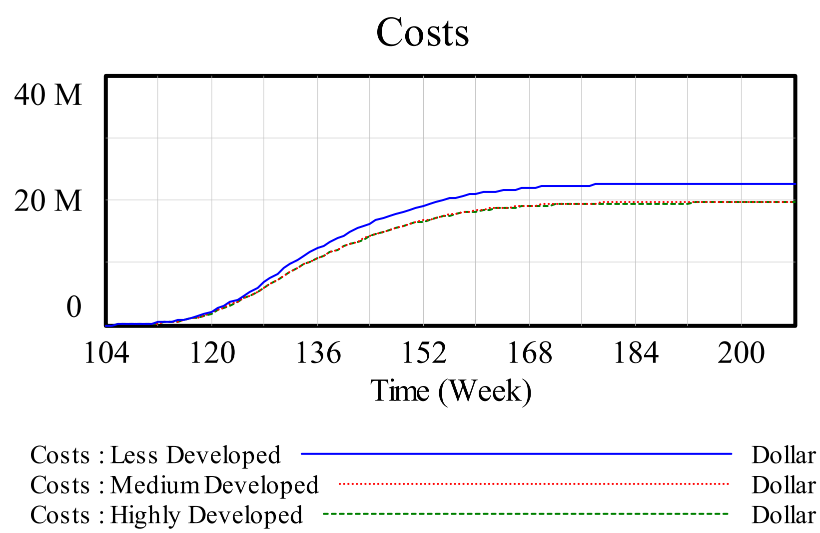


Figure 4-17: Cost and Performance for Various Degrees of Technology Complexity

Furthermore, lower the degree of complexity the more improved is the performance. It can be seen from the graph that the low level of complexity enables achieve the desired performance. Whereas, as the complexity level of technology is increased, the actual performance is below the desired performance and we do not achieve the desired goal in the period of two years.

Hypothesis 2: *Less developed or mature technologies are less affordable in terms of technology integration process.*

The graph that follows shows the cost and performance for varying degrees of technology maturity. The model was run for less mature (value of 1), medium maturity (value of 3) and highly mature (value of 5) technologies.



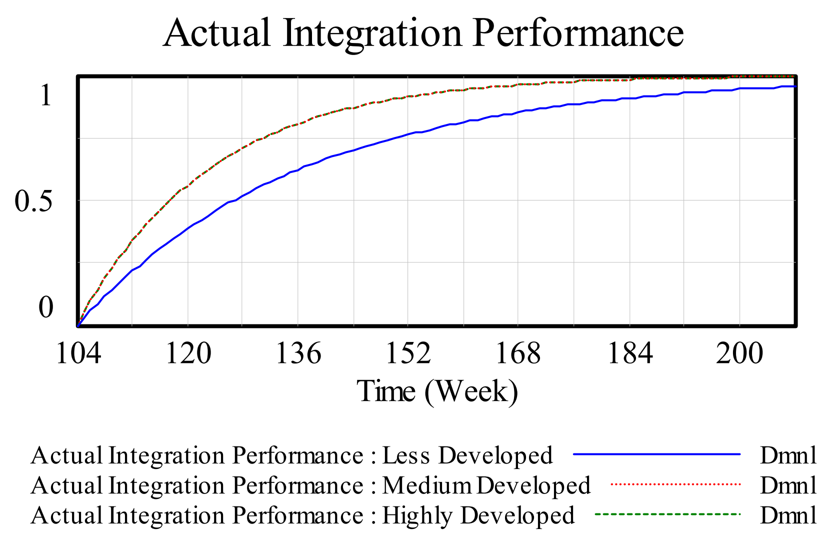


Figure 4-18: Cost and Performance for Various Degrees of Technology Maturity

The first graph of Figure 4-14 shows the relationship between technology maturity and the cost. The values of cost of less, medium and highly developed technologies are \$23.18M, \$20.24M, \$20.11M respectively. It demonstrates that there is not much difference in the cost for medium to high technology maturity (just around 0.65%). However, the increase in cost is significant when going from medium to low technology maturity (around 15%). The second graph of Figure 4-14 shows the relationship between performance and technology maturity. Performance is better for highly developed technologies as opposed to less developed ones. Also for less developed technologies, we do not reach the desired performance in two years.

4.4 <u>Sensitivity Analysis</u>

By varying the values of critical variables a sensitivity analysis was carried out. The critical variables are those variables that are supposed to have impact on the cost and performance of the technology integration process or those variables that can be altered to keep the costs low.

4.4.1 Program Risk

The program risk varies from low risk to high risk (risk values from 1 to 10). The cost and performance graphs are as follows when varying the values of risk.

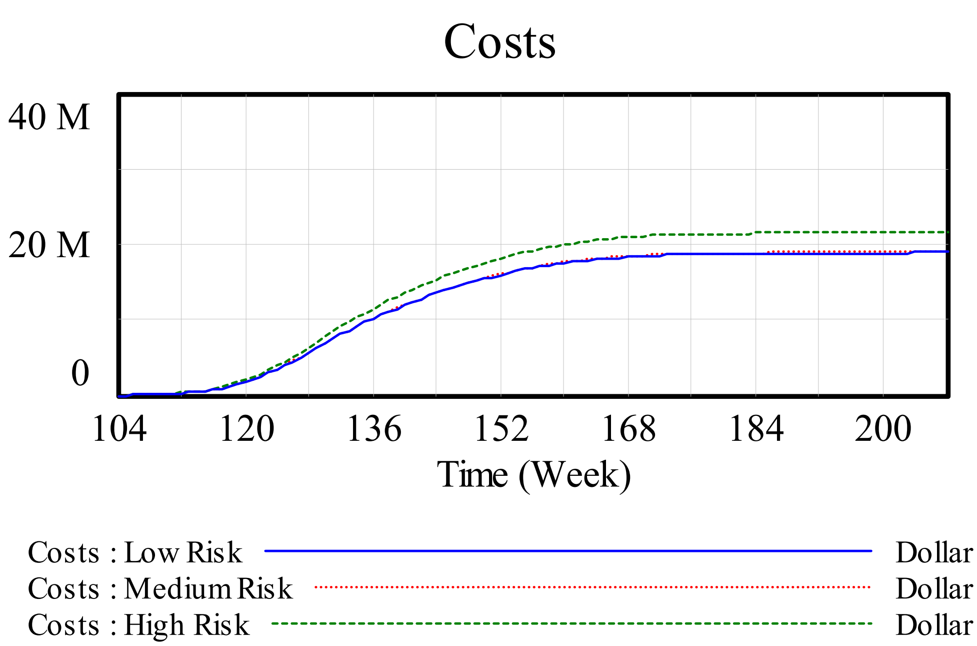


Figure 4-19: Technology Integration Risks as a Function of Risk

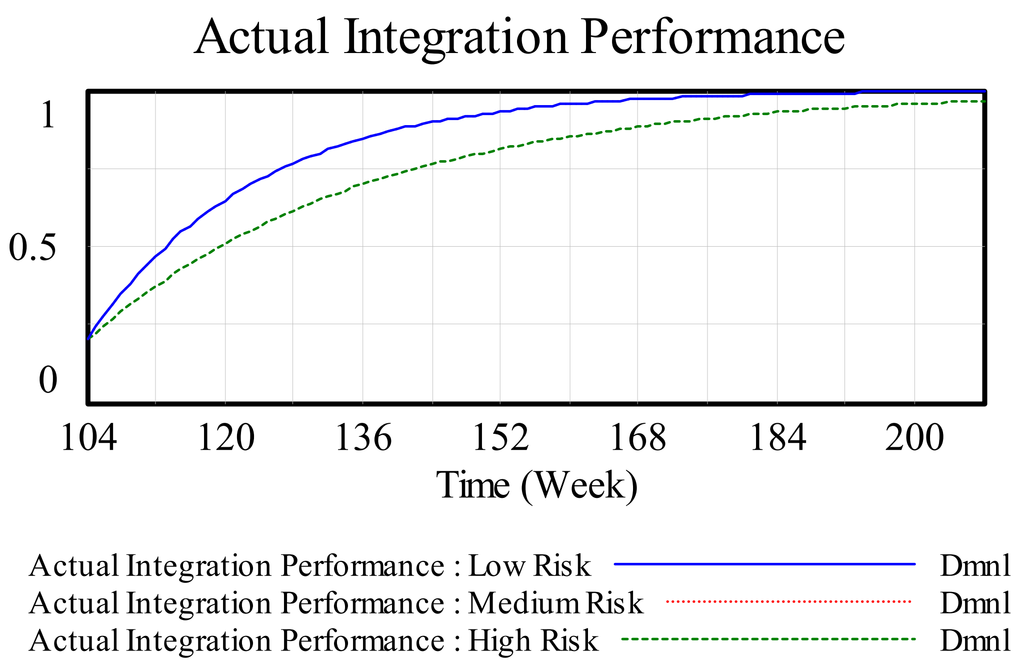


Figure 4-20: Technology Integration Performance as a Function of Risk

It is observed that the higher the risk the higher is the technology integration cost and the performance declines. Nevertheless, there is not much effect on cost and performance when the observed values for risk are between low and medium. But as risk increases, significant increases in technology integration cost and a reduction in performance are observed. Thus, while comparing the technologies, the risk may not be an important factor when the risk associated with those technologies range from low to medium level. However, it needs to be considered when it is high for any technology.

4.4.2 Labor Rate

Figure 4-17 depicts the graph for varying labor costs, \$60 per hour, and \$75 per hour and \$90 per hour. Labor cost seems to have a significant effect on the overall technology integration cost. However, labor cost has no effect on performance.

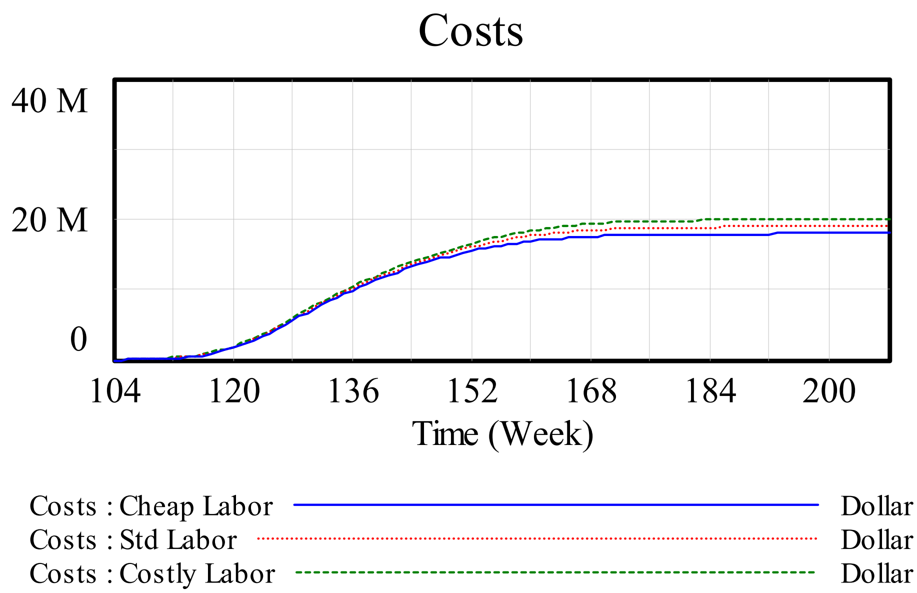


Figure 4-21: Technology Integration Costs as a Function of Varying Labor Rates

4.4.3 Material Cost

Figure 4-18 depicts the cost graph for varying values of the material cost. Simulations were run for material cost of $150K, $160K and $170K per unit. Total cost seems to be significantly sensitive to the material cost.

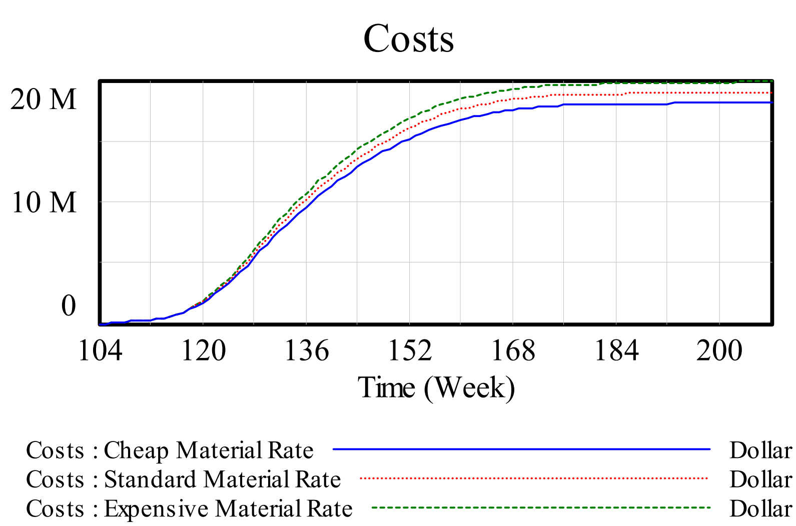


Figure 4-22: Technology Integration Costs as a Function of Varying Material Costs

4.4.4 Funding

When the funding level is changed, cost overrun is affected. This can be seen from Figure 4-19. Simulations were conducted for funding levels of 10.6M, 12.6M and 14.6M.

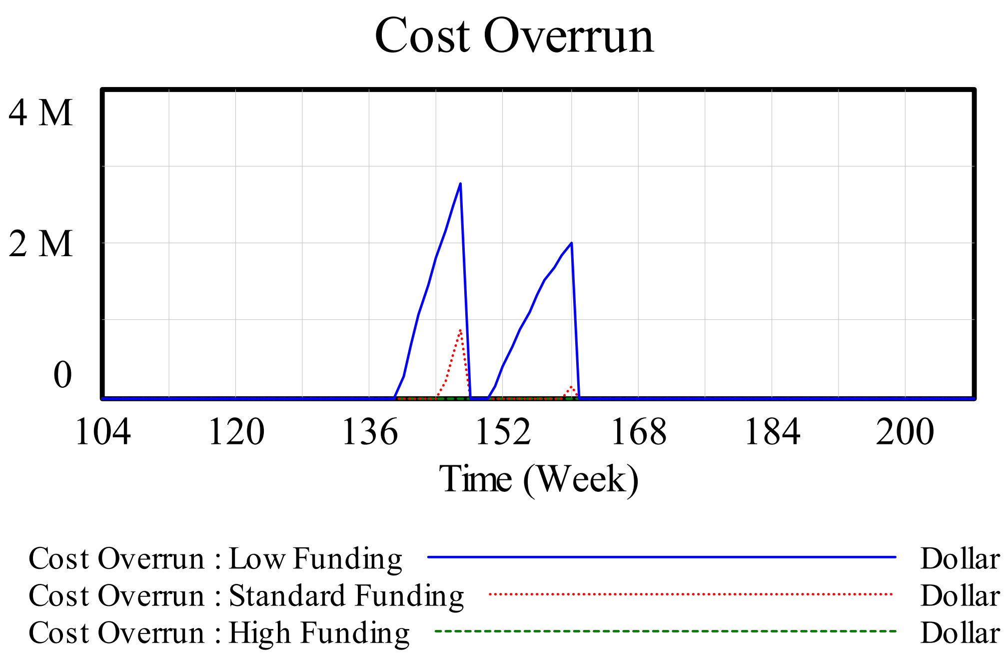


Figure 4-23: Cost Overrun as a Function of Funding Levels

With a low level of funding (10.6M), two large cost overrun peaks are observed. With standard level of funding (12.6M), two smaller peaks of cost overrun are observed. Whereas with high level funding (14.6M), no cost overrun is observed. Absence of cost overrun for the last case does not necessarily imply that the technology integration costs were below funding level of 14.6M. This is because the model structure predicts future cost overruns and consequently asks for additional money. Thus, there could be additional funding that is requested because of potential cost overruns.

4.5 Testing, Verification and Validation

By carrying out the necessary testing of the model so as to uncover errors and to find out the model's limitations is very important to build the confidence in model for both modelers and clients. Verification means checking the truth or reality of the model (Sterman 2000). Whereas validation means to ensure that the model supports the objective truth (Sterman 2000). Sterman (2000) says that with these definitions of verification and validation, no model can ever be verified and validated. The reason is because no model is ever an exact representation of reality or truth since it is based on many limiting assumptions. Thus the model can be verified and validated only based on a set of limiting assumptions. Also, though the model's validity cannot be proved, its falsity can surely be proved. Forrester (1961) substitutes the term "validity" by "significance" and states that the validity should be judged by the model's suitability for a particular purpose.

Sterman (2000) and Forrester (1961) state various fundamental tests that can be carried out. Many errors and limitations of the model can be discovered during these tests. The model can be rectified accordingly. Nevertheless, testing is not a process that is done at the end, only after the model developed. But testing is a continuous and iterative process that starts right from the start of the model building. Model testing is explicitly or implicitly carried out during every step of model development. Based on the results and feedback from the tests the model is continuously improved.

The tests that were carried out on the model as a part of the verification and validation process are described below.

4.5.1 Face Validity

Face validity is a process of ensuring the validity of the casual loop and stock & flow diagrams. These model diagrams were started from very general and high level drawings. Once when everyone on the modeling team (including experts from Navy, Government, Shipbuilders and Virginia Tech agreed on the validity of the diagram, a lower level construct was developed. The development of the causal loop diagram is explained in Appendix C.

4.5.2 Boundary Adequacy Test

This testing is particularly done at the beginning of the model development. The boundary adequacy test is to make sure that the important concepts and variables (for the purpose of the model) are within the boundary of the model. The decision of including or excluding variables in the model boundary was based on group discussions. The current boundary of the model is comprised of the variables that are important for the purpose of the model.

4.5.3 Structure Assessment Test

This test ensures that the structure of the model is consistent with the relevant knowledge about the technology integration process. The model development was based on the inputs from the experts from the Navy and Shipbuilders. Every effort was made to keep the model structure consistent with the information acquired from these experts. Moreover, the level of aggregation was sufficient enough to distinctly observe the output behavior for various critical variables in the model. Furthermore, care was taken to make sure that the model conforms to the basic laws of conservation such as stocks not going negative. This was achieved by careful development of equations for outflows of the stocks. As the stock approaches zero, its outflow also approaches zero to make sure that the stock does become negative.

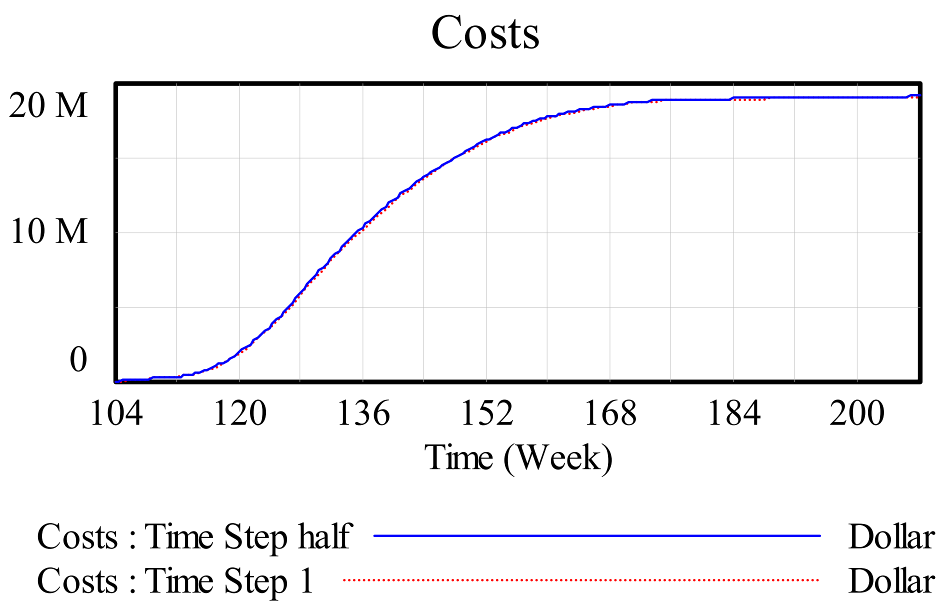
4.5.4 Dimensional Consistency

This test is concerned with the dimensional units of stocks, flows and variables in the model. Dimensional consistency can be checked through the simulation software. The lack of units for some variables and other dimensional errors were rectified to remove all of the dimensional inconsistencies.

Dimensional inconsistencies can also be spotted by errors during a simulation run and by manual inspection of the units of variables. Simulation runs did not generate any errors. Also, the manual inspection of the units did not show any units that did not have a real world meaning (e.g. $dollar^2/man\text{-}hour^3$).

4.5.5 Integration Error Test

Integration error tests have to do with the software's ability to provide inconsistent results for different time steps. The simulation software uses numerical integration methods and hence the results could differ when choosing different time steps. Time step should be such that the simulation results do not change (significantly) by cutting the time step into half. This test can be carried out by making simulation runs for different time steps. In Figure 4-20 performance and the technology integration costs for the time step of 1 week, 0.5 week and 0.25 week are depicted.



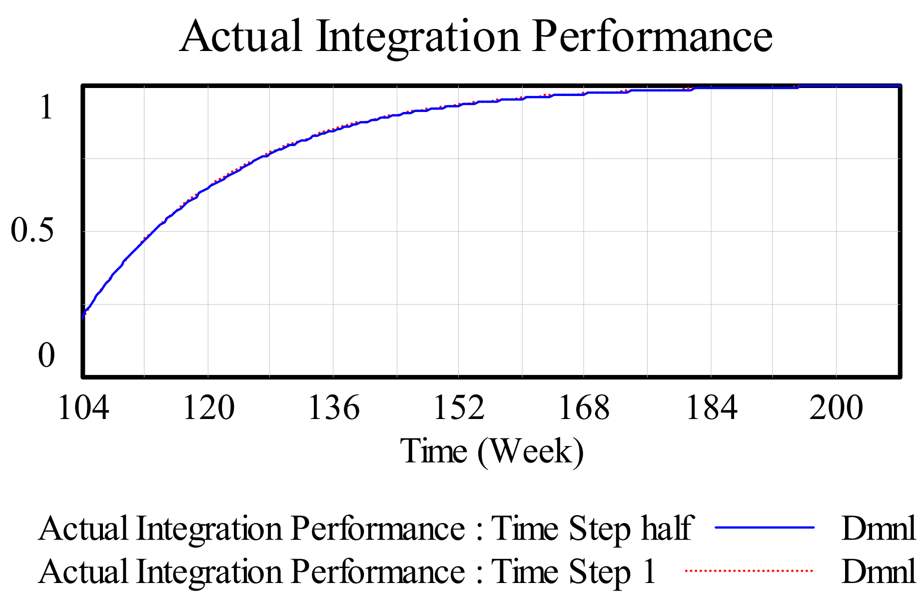
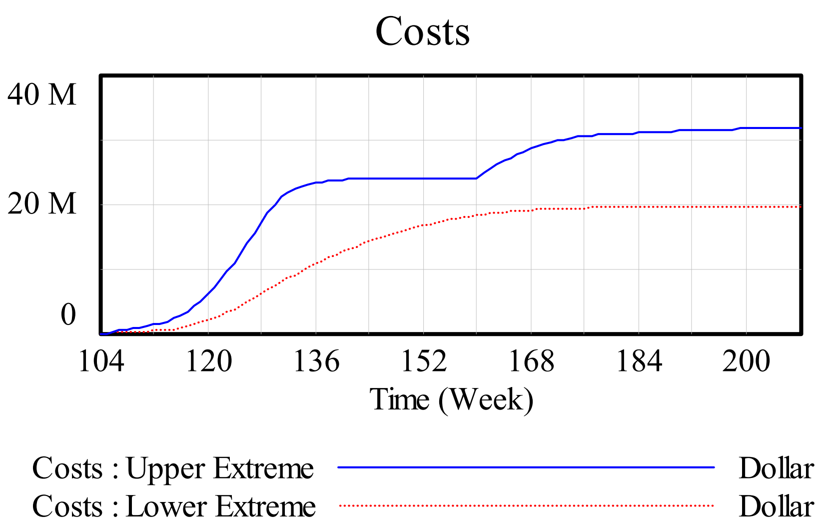


Figure 4-24: Simulation Results for Different Time Steps

The graphs show that there is very insignificant difference in values when one cuts the time step into half. The minor differences are only due to the fact that the variable values are calculated at a larger number of intervals for smaller time steps. Thus, the model has no integration errors.

4.5.6 Extreme Condition Test

Extreme condition tests are concerned with the robustness of the model. The model is subjected to the extreme set of parameter values for certain variables. For example for the current model, the model was subjected to the highly immature, very complex technology with very high risk of implementation. On the other hand, the model was also subjected to very mature, very less complex and low risk technology. The model results are shown below.



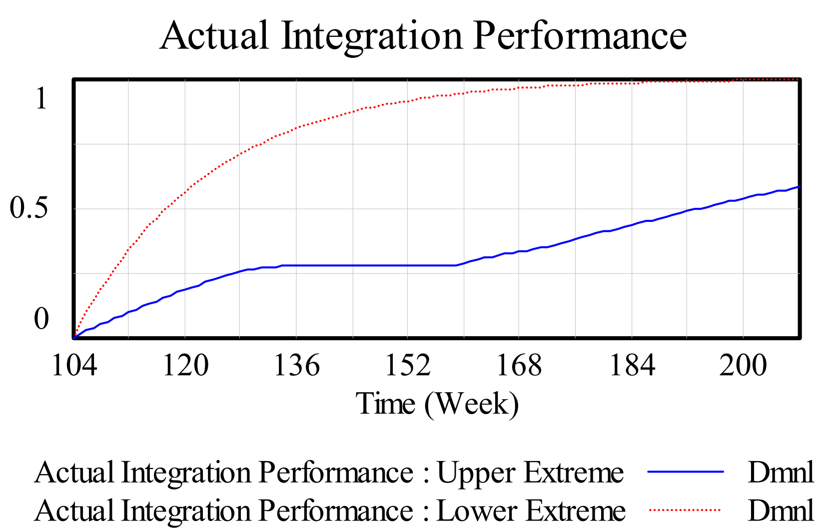


Figure 4-25: Simulation Results for Extreme Conditions

The model did not crash for either of the extreme conditions. Furthermore, the model did not show any unexpected behaviors in either of the cases. Thus, the model will run safely and will not behave unexpectedly for any values in between the two extremes.

4.5.7 Behavior Reproduction Test

This test is concerned about the ability of the model to reproduce the behavior of interest in the technology integration system, both qualitatively and quantitatively. As explained in Chapter 3, the reference modes of behavior for cost and performance of the technology integration system are depicted as follows:

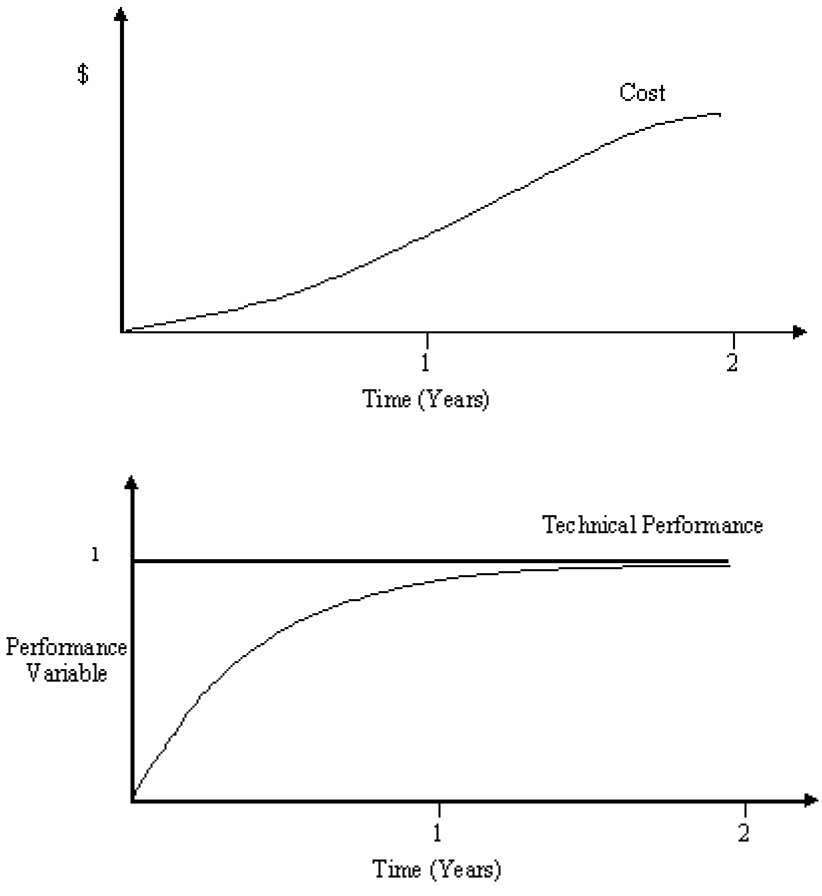


Figure 4-26: Reference Modes

The observed outputs for cost and performance are shown as follows:

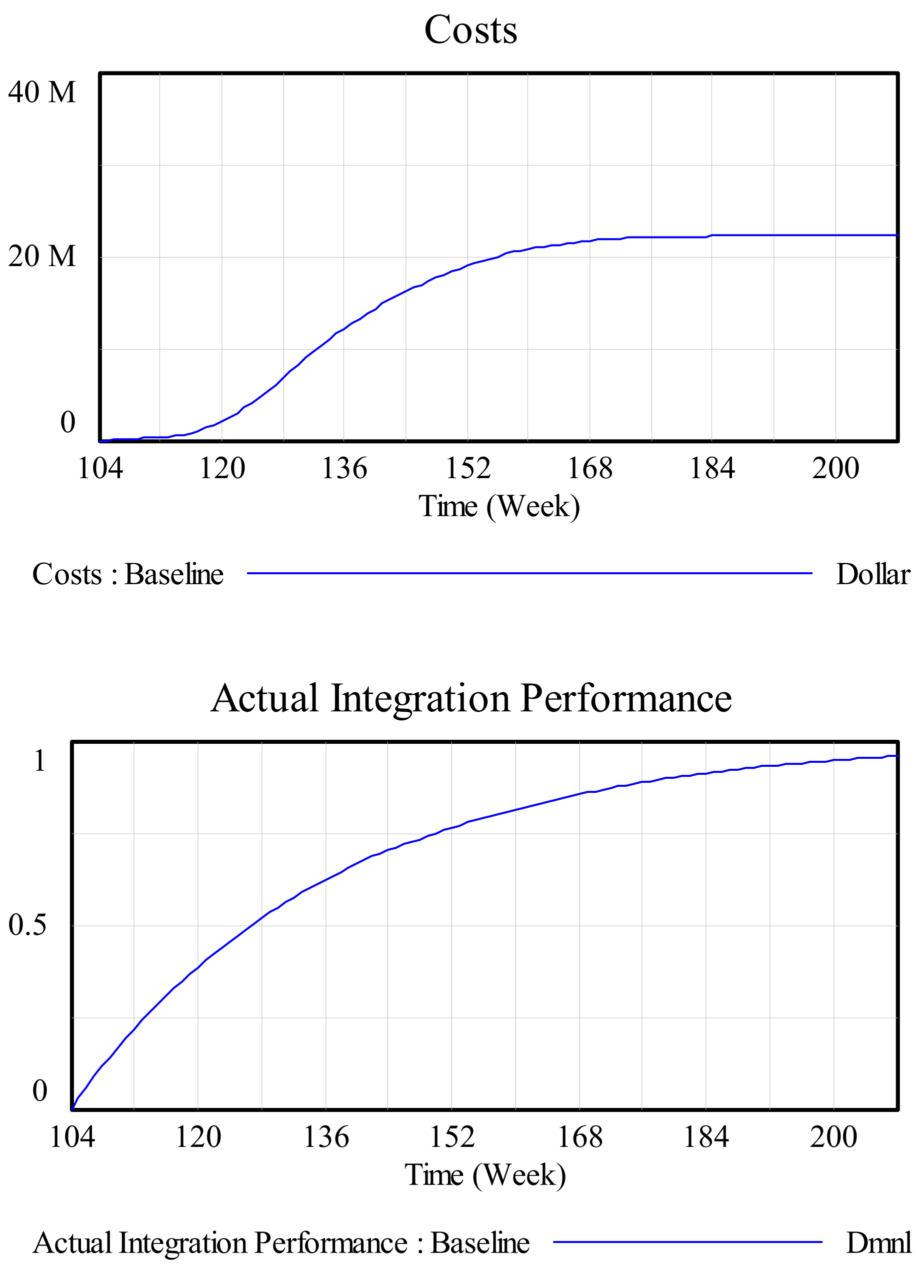


Figure 4-27: Observed Behavior

Thus the model results reproduce the anticipated behaviors, giving some additional insight with respect to those behaviors that the behavior of the cost is a distinct S-shaped growth that is observed from the simulation output.

Chapter 5: Conclusions

5.1 Overview of the Results

The system dynamics technology integration model was simulated for a specific technology with user-input parameters provided by shipbuilders and government experts. The user input parameters would have been different for a different technology. The results obtained from running the simulation model were discussed in detail in Chapter 4. The simulation was run for 2 years (104) weeks, which is the time horizon for the technology integration process and for 54 years, which is the length of the new technology introduction life cycle. Two major modes of behavior were exhibited by the simulation run.

S-Shaped Growth

An S-shaped growth behavior was shown by the cost variable. The cost of the technology integration process first increases in somewhat exponential fashion but then its rate of increase tapers down to make its behavior look like a stretched S shape curve. This behavior and the feedback loop structure that characterizes this behavior are explained in Section 4.2.1.

Goal Seeking

Goal seeking behavior is observed in a system where the state of the system seeks a specific target. The variable showing this behavior was the "Integration Performance" variable. Integration performance seeks the Target Performance and displays a goal seeking behavior. This behavior and the structure behind causing this type of behavior are explained in Section 4.2.2.

5.2 Verification of Dynamic Hypotheses

The dynamic hypotheses discussed in Chapter 3, Section 3.3 and in Chapter 4, Section 4.2 were tested using the model developed in VENSIM 5.0 by varying parameters and observing the changes in the subsequent results from the simulation.

The first hypothesis was that the complex technologies are less affordable, in terms of technology integration process. This hypothesis was tested by subjecting the model to varying values of the complexity of the technology. From these simulation results, it was observed that the cost and the cost realization rate increased with increased technology complexity. Thus, the cost of technology integration process was found to increase as the technology complexity increases. This means that complex technologies are less affordable.

The second hypothesis was that the immature (less developed) technologies are less affordable. From the simulation results, it was observed that the cost of the technology integration process and the cost realization rate are significantly increased with the decreasing levels of technology maturity. Thus, the more immature the technology is, the more is the cost associated with its integration, making it less affordable.

5.3 Research Innovations

During the course of development of the model, there were various interesting discussions involved during the modeling sessions. Those were:

1. The structure for requesting additional funding;
2. The use and the implementation of the Decision Support System (DSS) software.

The structure for requesting additional funding: During one of the structured group modeling sessions, an innovative idea for requesting additional funding was developed. In reality, the additional funding is requested when the program manager sees that the costs are increasing at high rate and when he/she senses that the current level of funding is going to be insufficient. There was a discussion on how to incorporate this feature of requesting additional funding in the model. The concept of control spending was introduced, which is discussed in Sections 3.7.8 and 4.2.3. A structure was developed in the model to incorporate this concept.

The Decision Support System Software: An innovative interface was developed for effective use of the model. The software was developed in Visual Basic and it has a very user-friendly and intuitive interface. The software is linked to the dynamic link library files of the VENSIM software using the Application Program Interface technique. The variable values can be inputted to the model through the simple interface and also the results can be observed with the help of the software. The software needs no user interaction with the model and requires no knowledge of the VENSIM simulation software. The user manual and the screenshots of the decision support system software are shown in Appendix D.

5.4 Policy Suggestions

Based upon the simulation results and the testing of the system dynamics simulation model, policy suggestions can be made to improve the technology integration process as a part of system engineering process. These suggestions come from the testing of the hypothesis testing as well as additional insights from the model results. During the new technology integration process, the main concern is the total cost and the cost overruns. The policy suggestions could help reduce both of these variables.

The model predicts the possible cost overruns for the given funding and the work requirements. As a result of it, people from both the budgeting and work-load planning team can sit together and adjust the values so as to see no cost overruns in the model and possibly in reality. But adjusting the work requirements to avoid cost overrun would have a strong effect on the performance, which will be disclosed by the model. Thus, the target performance will not be reached if the work requirements are reduced to keep costs in control. Thus, there is a trade-off between the performance and the cost overruns. There could be various discussions that can be done among these two teams as well as performance and quality control team to make policies that will not let trade the performance off with cost overruns. Thus, policies can be set that says that the performance of the process has to be above certain minimum level irrespective of the costs and cost overruns.

Technology maturity has a significant effect on the cost of the integration process. Thus, the policies (such as “mature technologies should be given preference while selecting new technologies”) that favor the more mature technologies will help keep low integration costs. Also, more effort and funds should be devoted towards the technology development efforts so that the technology becomes more mature.

Complexity of the new technology also has a considerable effect on the cost of the integration process. These integration costs can be kept low by adopting less complex technology.

Higher program risk (the probability of an unwanted outcome to occur) causes costs to rise. A policy that will avoid installing a technology above a pre-determined risk level (medium risk level) would help to keep costs within limits.

The above policies are suggested especially when comparing or benchmarking different new technologies. They are useful when one has to choose a technology among the several parallel technologies.

In addition to the characteristics of the new technology, cost and cost overruns are also sensitive to the labor and material costs. Thus, during the process of technology integration, special care should be taken while deciding the number of material units to be procured.

Thus, the more mature and less complex technologies are more affordable if the risk in the implementation is up to the medium level. As the risk of implementation becomes higher, the technologies are less affordable to implement.

5.5 Areas for Future Research

Any model and research are always based on assumptions. These assumptions help restrict the research work so as to address specific questions. These assumptions can be relaxed to do further research on additional issues. The current research assumes that the incoming funding is available on the very first day whereas the additional funding takes some delay to come in after a request has been made. Furthermore, it assumes a pre-determined pattern for the engineering, design, testing, assembly, integration and material requirements. These assumptions can be relaxed. Requirements can be incorporated in the model so as they can be customized by the decision maker.

To make a more realistic representation of the real world, some of the model variables can be made random to see their effect on the technology integration process. For example the variables such as 'additional funding receiving delay', various activities delay time and the inflation rate can be incorporated as random variables with certain kind of distributions.

The model boundary can be expanded to include more variables in the boundary of the system to make it more general. For the current research, the effect of environmental factors on the new technology integration process was excluded for the sake of simplicity. This can be given further consideration to analyze their effect on the integration process.

Future work can be done in the area of degree of abstraction of the current model. Various variables in the research can be disaggregated into more specific and detailed variables to study them separately. For example the testing activity, as a part of the integration process, can be further divided into sub-activities to observe how each sub-activity behaves.

Finally, a product of the research, which is the decision support system software, can be further modified to incorporate various additional features for analysis, such as making customized graphs, ability to run partial simulation runs, etc.

References

Bertalanffy, L. V., (1973). *General Systems Theory: Foundations, Development, Applications*, George Braziller: New York, NY.

Bühlmann, Hans., (1970). *Mathematical Methods in Risk Theory*. Springer-Verlag: New York Heidelberg Berlin.

Burchill and Fine, (1997). Time Versus Market Orientation in Product Concept Development: Empirically Based Theory Generation. *Management Science*, 43, 4. pp 465-478.

Byrd, D. M. and Cothern, C. R., (2000). *Introduction to Risk Analysis*. Government Institutes: Rockville, Maryland.

Cooper, K. G., (1980). Naval Ship Production: A Claim Settled and A Framework Built. *Interfaces*, Vol. 10, No. 6, 20-36.

Evangelos Petroutsos, (1998). *Mastering Visual Basic* 6. BPB Publications: New Delhi, India.

Ferrari, Th. J., (1978). *Elements of System-Dynamics Simulation*. John Wiley & Sons: New York – Toronto

Ford, D., and Sterman, J. D. (1997). Expert Knowledge Elicitation for Improving Mental and Formal Models. *System Dynamics Review*, Vol. 14, no. 4, pp. 309-340.

Forrester, J. W., (1961). *Industrial Dynamics*. MIT Press: Cambridge, MA.

Forrester, Jay W., (1973). *World Dynamics*. Wright-Allen Press Inc.: Cambridge, Massachusetts.

Forrester, Jay W., (1975). *Collected Papers of Jay W. Forrester*. Wright-Allen Press, Inc.: Cambridge, Massachusetts.

Goodman, P. S. and Griffith, Terri L., (1991). A process approach to the implementation of new technology. *Journal of Engineering and Technology Management*, 8 (1991) 261-285.

Griffith, T., Zammuto, L., & Aiman-Smith (1999). Why new technologies fail? *Industrial Management*, 29-34.

Hope, Warren T., (1998). *Introduction to Risk Management*. Insurance Institute of America.

Jassawalla, A. R. and Sashittal, H. C., (1998). Accelerating technology transfer: thinking about organizational pronoia. *Journal of Engineering and Technology Management*, 15 (1998) 153-177.

Monga, P. and Triantis, K., (2002). A Dynamics Model for the Development of New Technologies for Ship Systems. Submission to *System Dynamics Journal*

Pouliquen, Louis Y., (1972). *Risk Analysis in Project Appraisal*. The Johns Hopkins Press: Baltimore and London.

Randers, J., (1980). *Elements of the System Dynamics Method*. The MIT Press: Cambridge, Massachusetts, and London, England.

Richardson, G. P and Anderson, D. F. (1980). Toward a Pedagogy of System Dynamics. System Dynamics, *TIMS Studies in the Management Sciences,* vol. 14 (1980) 91-106.

Richardson, G. P. and Pugh III, A. L., (1981). *Introduction to System Dynamics Modeling*. MIT Press: Cambridge, MA.

Scott, John, (2003). *A System Dynamics Model of the Operations, Maintenance and Disposal Costs of New Technologies for Ship Systems*. Unpublished masters degree project and report. Virginia Polytechnic Institute and State University, Falls Church, VA.

Senge, P., Roberts, C., Ross, R., Smith, B. and Kleiner, A., (1994). *The Fifth Discipline Fieldbook – Strategies and Tools for Building Learning Organizations*. Doubleday Dell Publishing Group, Inc.: New York.

Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin McGraw-Hill: Boston, MA.

Tabak, F. and Barr, S., (1999). Propensity to adopt technological innovations: the impact of personal characteristics and organizational context. *Journal of Engineering and Technology Management*, 16 1999 247–270

Vaneman, W. and Triantis, K., (1999). Defining, Evaluating and Controlling the Implementation Phase of a Systems Life-Cycle. *American Society for Engineering Management National Conference Proceedings*, Old Dominion University, October 1999, 234-242.

Vensim 5.0 (Reference Manual) (2002). Ventana Systems, Inc.: Harvard, MA.

Appendix A. Glossary

Condition Readiness: The Navy has three Conditions of Readiness. A Condition I battle readiness requires all operational systems fully manned and operating and no maintenance actions except urgent repairs. A Condition II limited action requires all possible operating stations manned and ready and urgent preventive maintenance actions and administrative functions performed. A Condition III cruising requires operating stations manned only as necessary and normal underway-preventive maintenance actions and administration functions performed.

DSS (Decision Support System): It is a software system, designed in Visual Basic, which provides interface to the simulation model. This system provides a very user friendly capability that enables decision maker to conduct analysis on the model easily.

Endogenous: It is defined as "arising from within", that is, from inside the boundary of the model.

Exogenous: It is defined as "arising from without", that is, from outside the boundary of the model.

Facilitator: A facilitator conducts and manages the group session, with an aim of facilitating the expression of views of all the key players involved in the modeling exercise.

First Order Delay: A first order delay is a stock and flow dynamics characterized by the outflow rate being proportional to the stock size. Some units have lower residence time than the average delay time while others have greater residence times than the average delay time.

Gatekeeper: A gatekeeper is a contact person within the target organization who helps select the appropriate people to work within the organization with, helps plan the modeling sessions, and works closely with the modeling group.

Mapping System Structure: It is the process of mapping the boundary of the model and representing its causal structure.

Modeler: A modeler's role is to be a keen observer in the modeling exercise, listening and capturing the ideas and views of different session members and providing inputs on the technical modeling aspects as and when required.

Model Boundary Chart: It lists the key endogenous and exogenous variables of the model.

OSCAM (Operations and Support Cost Analysis Model): OSCAM is a joint UK/US program to develop a tool that can provide rapid assessments of the Operation and Support (O&S) costs of high cost capital assets and their component systems.

Process Coach: A process coach ensures that the modeling group does not stray substantially from the modeling agenda, and provides feedback to the facilitator in terms of the effectiveness of the modeling process being followed.

Recorder: A recorder takes detailed notes on the various session activities and developments.

Refresh: A technology refresh refers to an upgrade of the technology features or operational capabilities.

Appendix B. System-subsystem structure

System

The system is the system engineering process that is responsible for the development, implementation, maintenance, upgrade, and retirement of new technologies.

Subsystems

Four subsystems were identified for the overall system, namely, Technology Development (Subsystem #1), Technology Integration (Subsystem #2), Operations, Support & Disposal (Subsystem #3) and Program Management (Subsystem #4). The program management subsystem is not considered separately for the modeling purposes but it is incorporated into each other subsystem models (refer to overall system/subsystem diagram at the end of this appendix).

Subsystem #1: Technology Development

In the *Technology Development* phase**,** new technologies are developed and/or assessed. The various activities for this subsystem include *R&D technology development activities* (technology research, requirements definition, specification development, engineering, modeling and simulation, drawing development, hardware and software development, system architecture development, and project management), *Project Management activities, and Testing activities.*

Subsystem #2: Technology Integration

This subsystem is primarily responsible for the integration of the new technology into the ship operations. The various activities for this subsystem include *Engineering and Design* (feasibility evaluation of new designs, design, development and implementation of design specifications, development of drawings, and managing engineering changes), *Procurement and Fabrication* (Purchasing of parts, components, hardware, and software), *Assembly, Installation and Integration* (Integration of new technologies with existing technologies and operations, and technology installation), *Project Management* activities and *Testing* activities

Subsystem #3: Operations, Support & Disposal

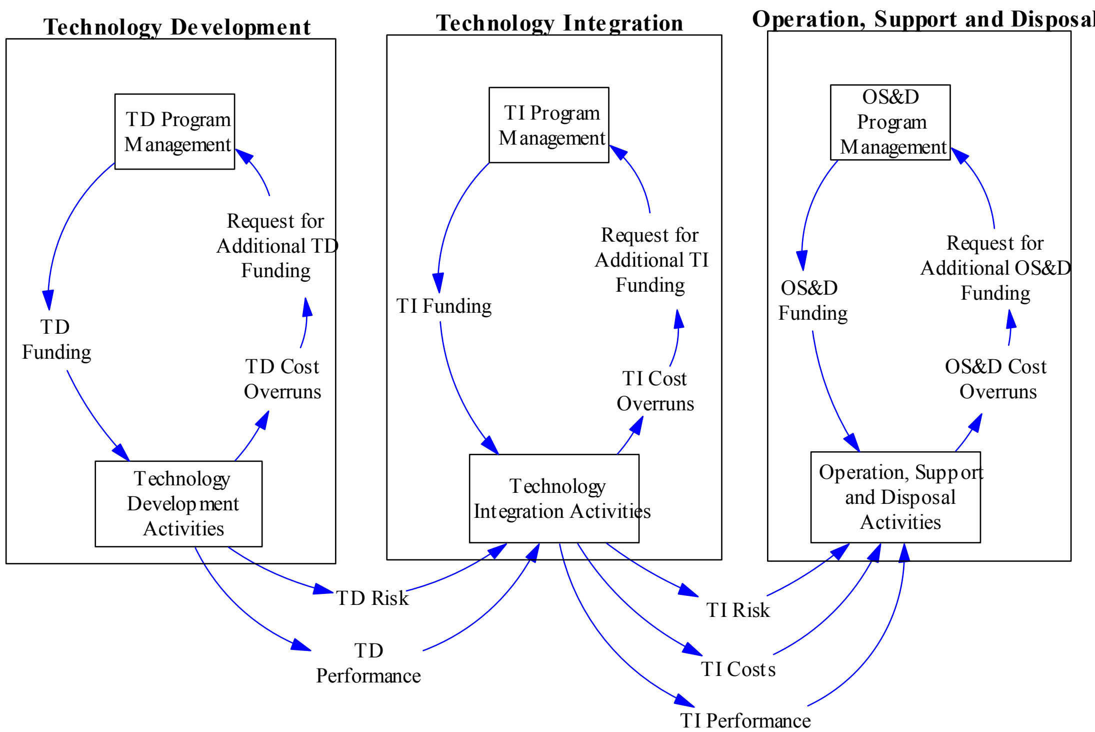
This subsystem is primarily responsible for all the activities necessary for the operations, support and disposal of the new technology. The various activities for this subsystem include *Shipboard Maintenance* (operation and maintenance of the systems onboard the ship), *Organizational and Intermediate Maintenance* (preventive and corrective actions requiring calibration, repair, and replacement of parts, components, and assemblies), *Depot Maintenance* (performing major overhauls or maintenance on a ship and associated equipment at centralized repair depots, or contractor repair facilities), *Other/Indirect/Sustaining Support* (course training for the ships crew to enable them to perform assigned maintenance and operational tasks), and *Operations and Support Disposal* (compliance with Environmental/HAZAMAT laws and regulations and with handling and disposal of hazardous material during the operating life of the ship).

Subsystem #4: Program Management

Program Management is responsible for managing the technology implementation in response to validated operational requirements. It provides the coordination and direction of activities required to develop, integrate, operate, maintain, and dispose the new technology. It oversees cost and secures funds for the technology implementation activities.

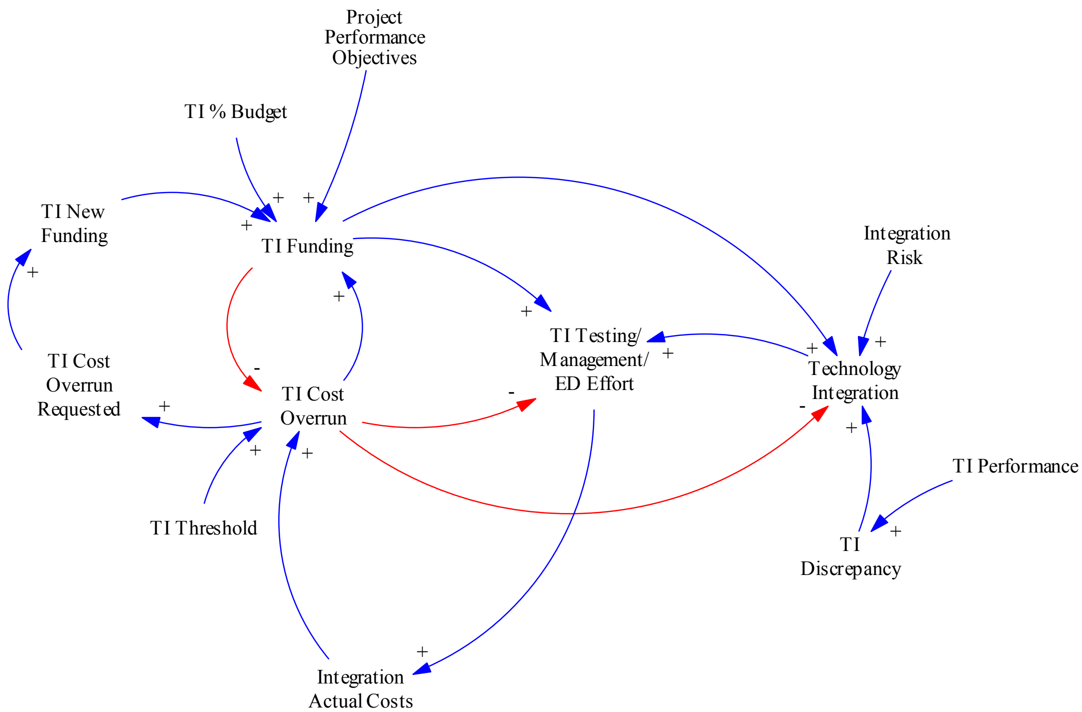
Subsystem Diagram

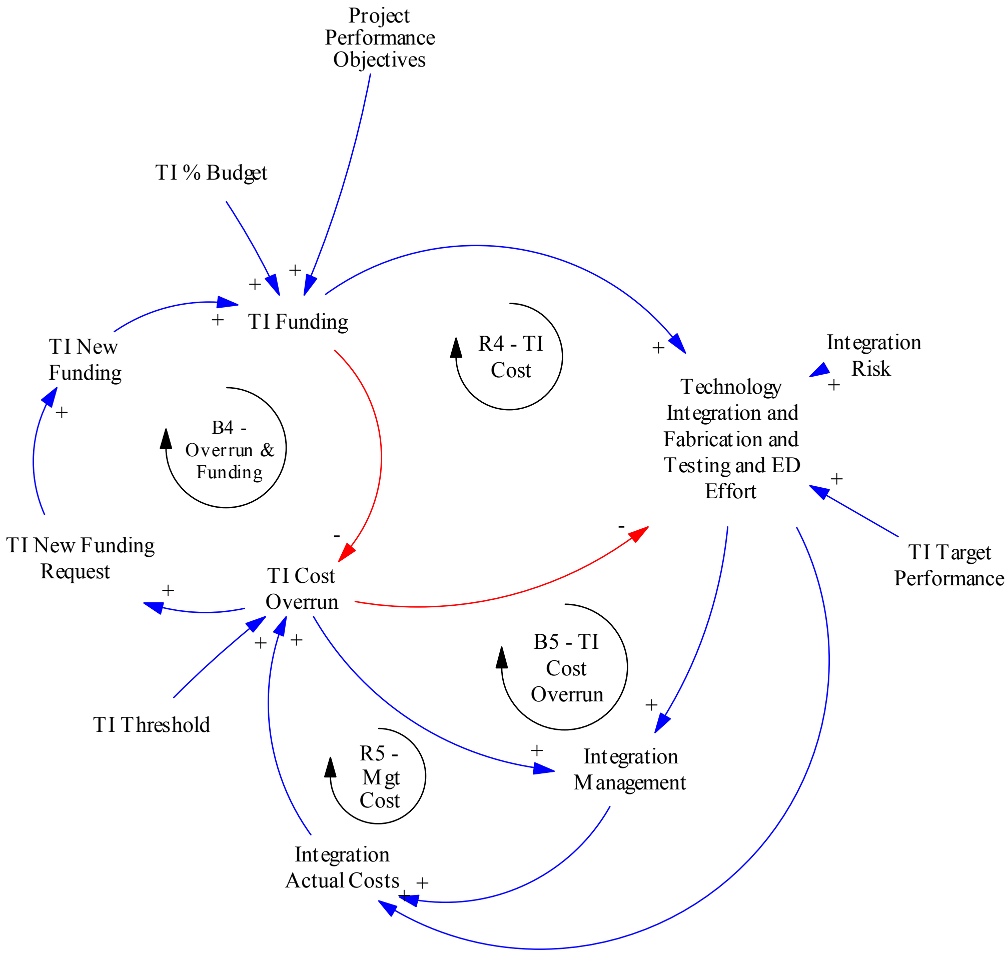
Overall System/Subsystem Diagram

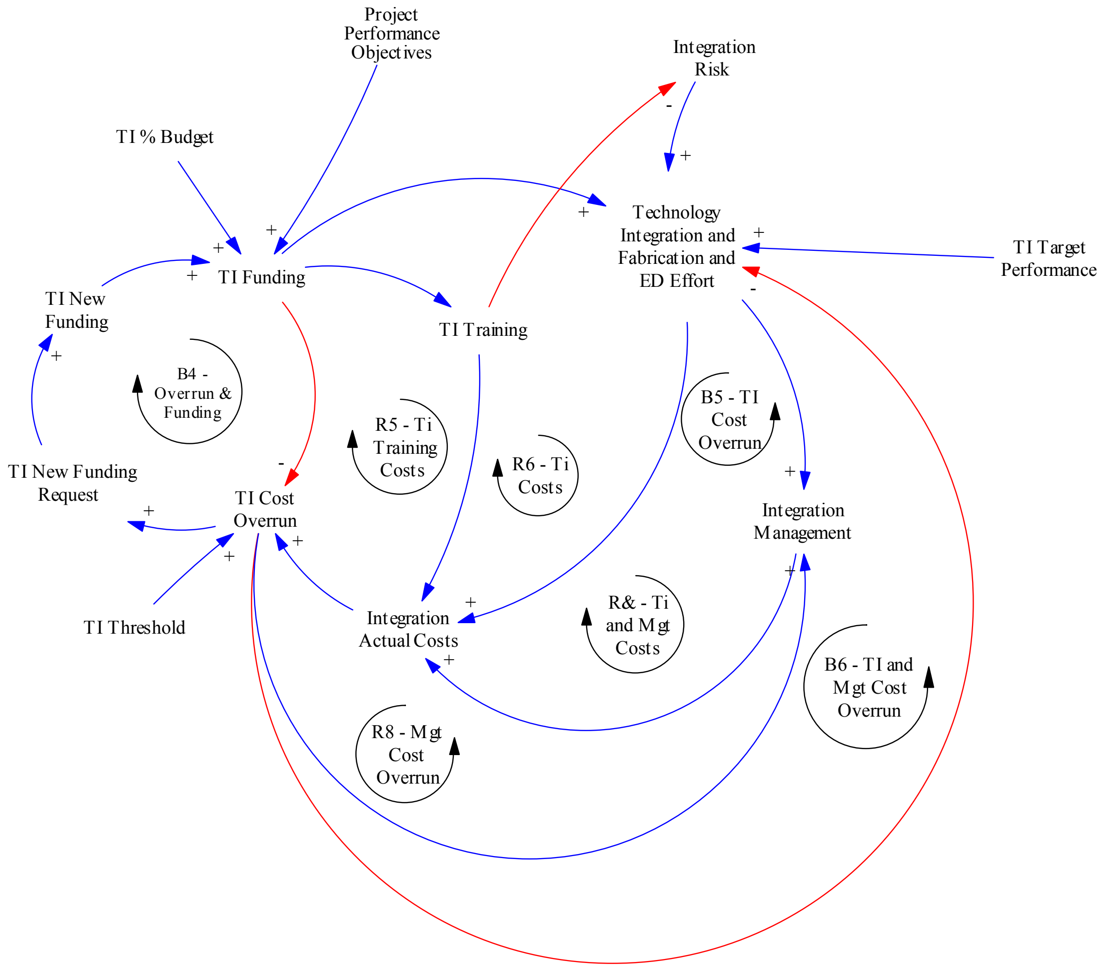


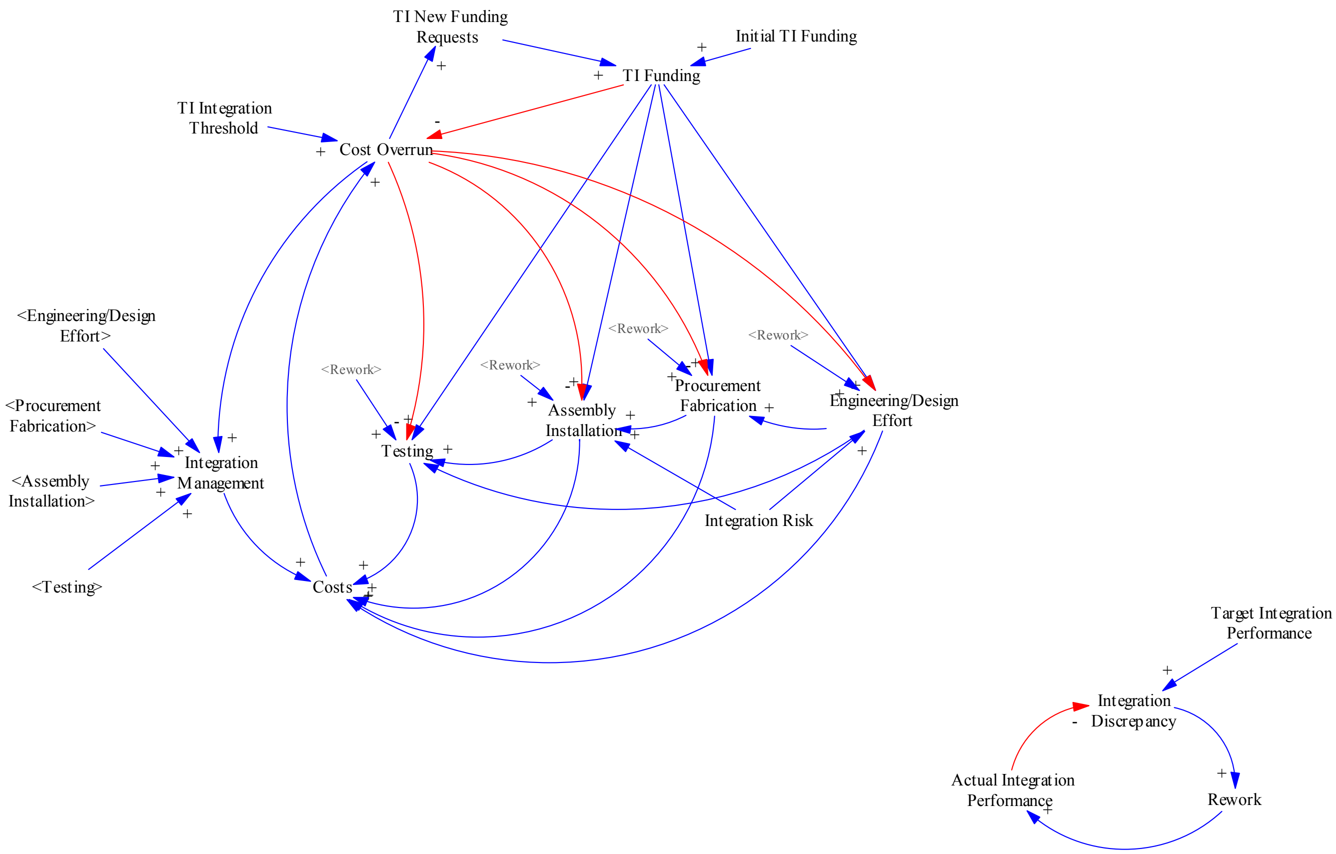
Appendix C: Evolution of Causal Loop Diagrams

September 5, 2001

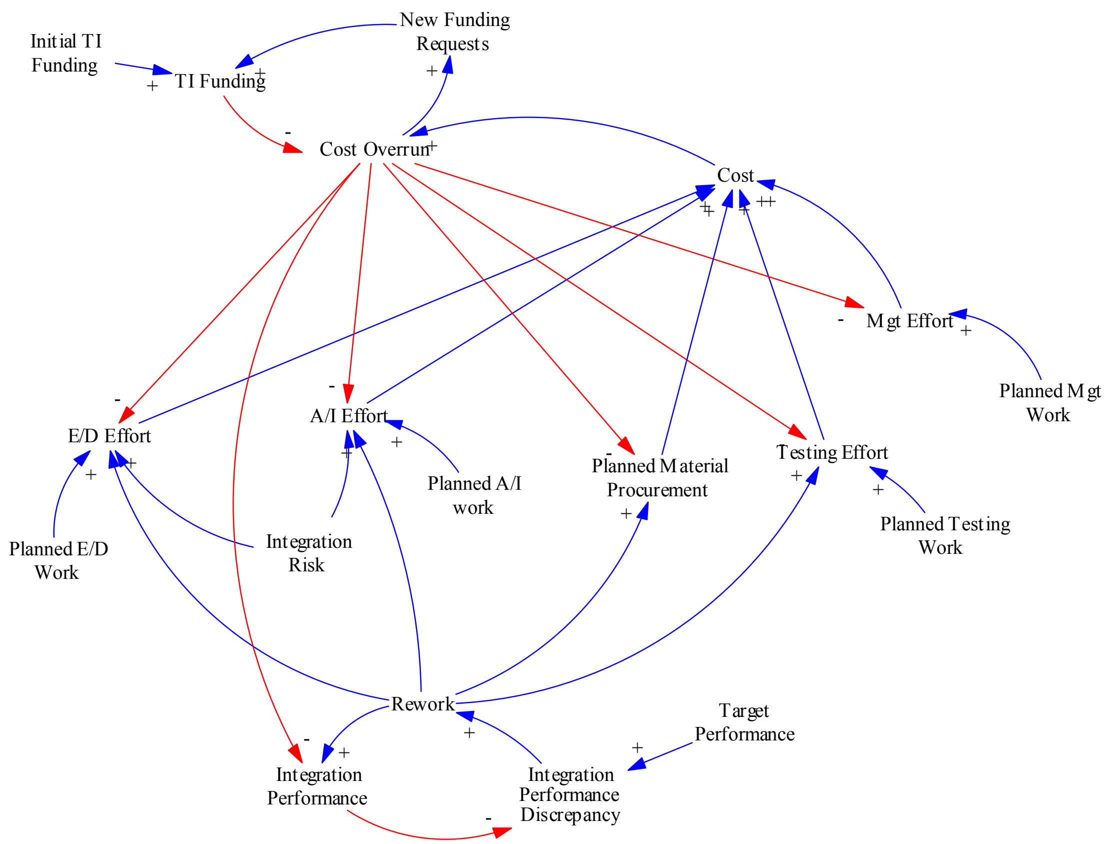



Project Performance Objectives
TI % Budget
TI Funding
TI New Funding
B4 - Overrun & Funding
R4 - TI Cost
Integration Risk
Technology Integration and Fabrication and Testing and ED Effort
TI New Funding Request
TI Cost Overrun
TI Target Performance
B5 - TI Cost Overrun
TI Threshold
R5 - Mgt Cost
Integration Management
Integration Actual Costs

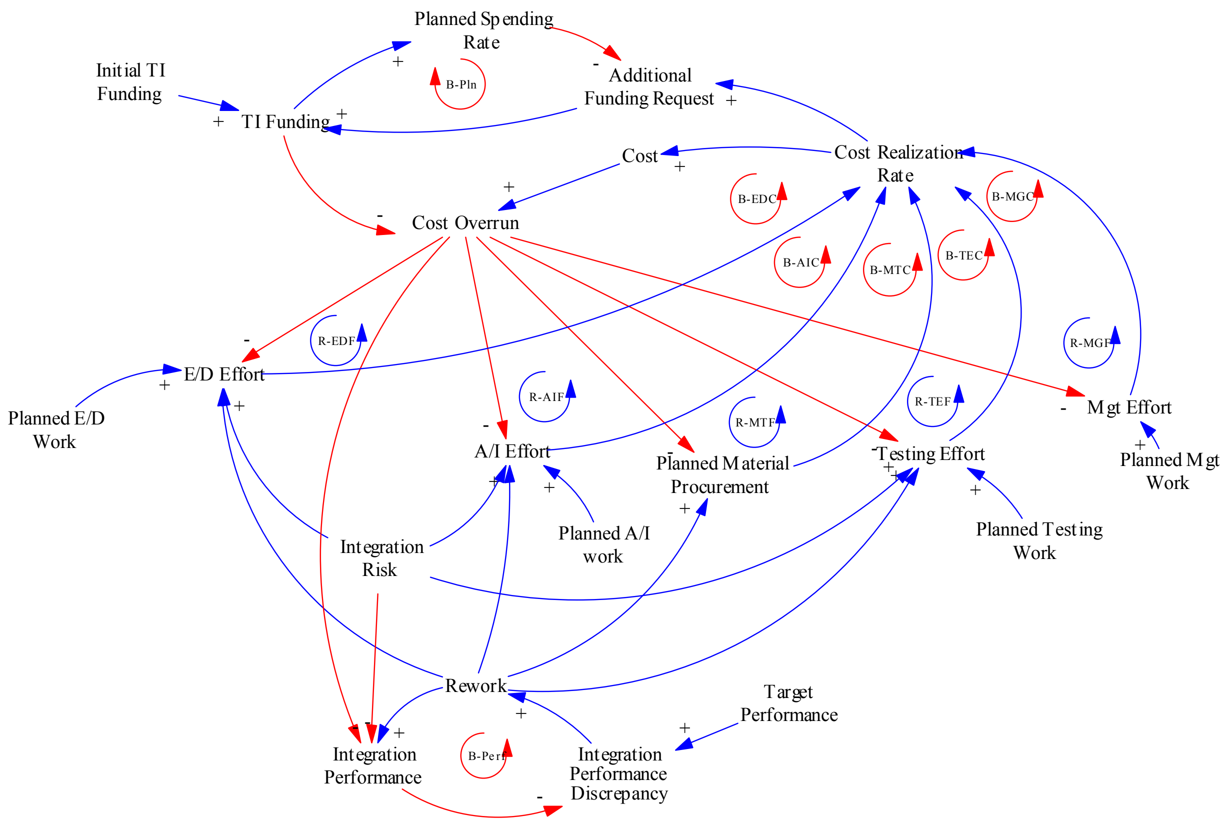

Project Performance Objectives
Integration Risk
TI % Budget
TI Funding
Technology Integration and Fabrication and ED Effort
TI Target Performance
TI New Funding
TI Training
B4 - Overrun & Funding
R5 - Ti Training Costs
R6 - Ti Costs
B5 - TI Cost Overrun
TI New Funding Request
TI Cost Overrun
Integration Management
TI Threshold
Integration Actual Costs
R& - Ti and Mgt Costs
B6 - TI and Mgt Cost Overrun
R8 - Mgt Cost Overrun

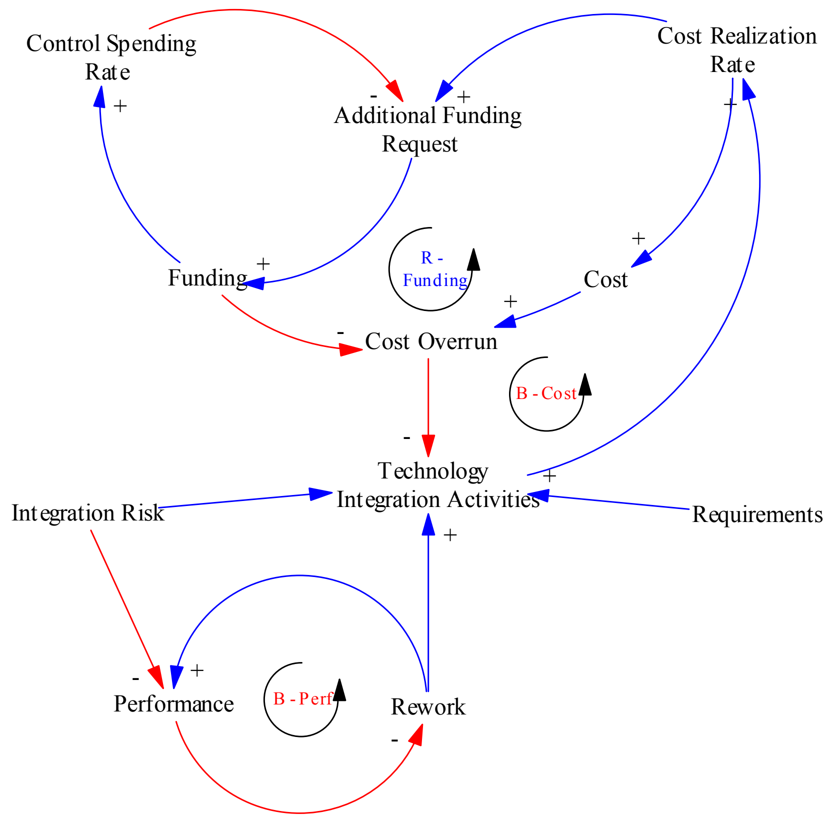

TI New Funding Requests
Initial TI Funding
TI Funding
TI Integration Threshold
Cost Overrun
<Engineering/Design Effort>
<Procurement Fabrication>
<Assembly Installation>
<Testing>
Integration Management
<Rework>
Testing
<Rework>
Assembly Installation
<Rework>
Procurement Fabrication
<Rework>
Engineering/Design Effort
Integration Risk
Costs
Target Integration Performance
Integration Discrepancy
Actual Integration Performance
Rework

March 23, 2002



April 25, 2002




Control Spending Rate
Additional Funding Request
Cost Realization Rate
R - Funding
Funding
Cost
Cost Overrun
B - Cost
Technology Integration Activities
Integration Risk
Requirements
Performance
B - Perf
Rework

Appendix D. DSS User's Manual

Getting Started

This user's guide gives information on how to use the "Decision Support System" software.

"Decision Support System" (hereafter mentioned as DSS Software) is a tool that can be used to run and analyze the "New Technology Implementation" model developed by System Performance Laboratory (SPL) at Virginia Tech, with valuable participation from the Office of Naval Research, and Northrop Grumman Newport News Shipbuilding, and NAVSEA 017. The DSS software lets the user to input values into the model and run the model for in various different scenarios. The output of the model is the behavior over time graphical displays for important model variables such as cost, system degradation, development effort, and many others. The software is divided into three major parts, i.e., The Introduction, The User Inputs and The Analysis. This user's guide also contains three main sections that correspond to each of these three parts of the DSS software.

In the section, a brief introduction to the model and about sub-parts of the model called subsystems is provided. The help on using introduction part of the software is given into Introduction section of this user's guide.

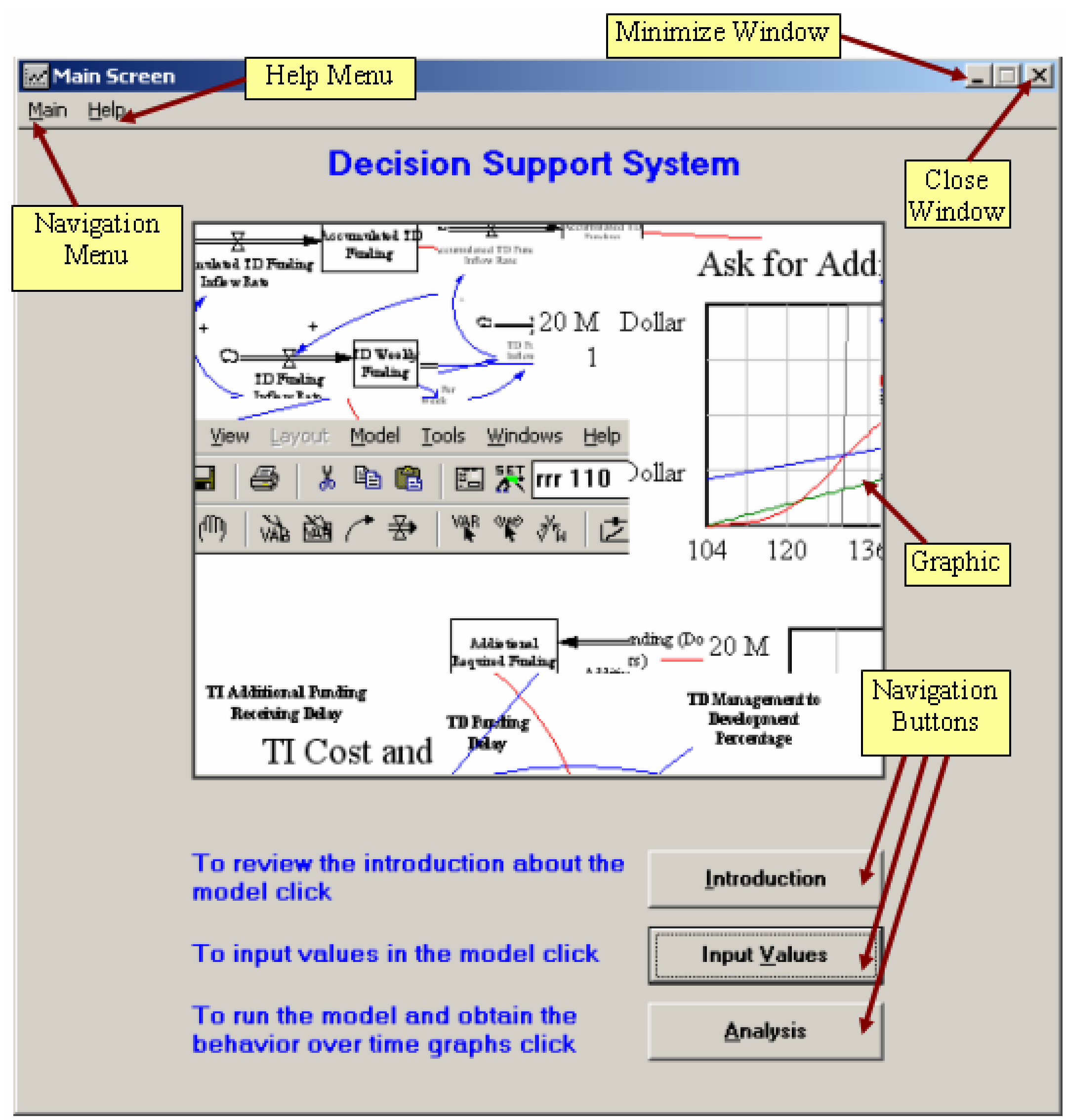
Information on how to use the DSS Software to input values to the model is provided. under User Inputs section.

Third part, the describes how to run the model and obtain the behavior over time graphs observing behavior of various model variables.

The "Getting Started" section chapter consists of the following topics:

Main Screen

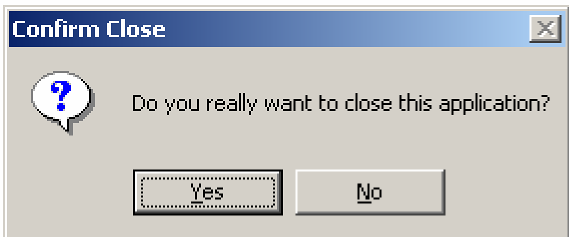
The opening screen of the DSS Software looks as follows. This screen is referred to as the "Main Screen"



This window provides access to the three parts of the DSS Software, i.e., the Introduction, the User Inputs and the Analysis.

Using the standard windows minimization button at the right top corner can minimize the window.

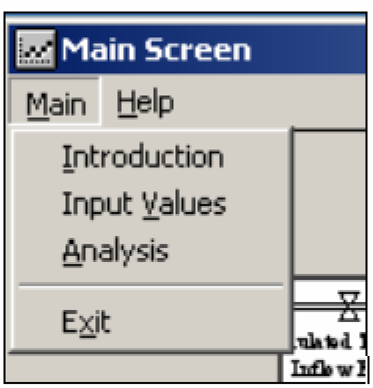
The standard windows Close button at the right top corner can close the DSS Software. A "close confirmation" window is shown as follows:



Clicking "Yes" closes the application. Clicking "No" does not close application and returns back to application.

Main Menu

The Menus look like this:

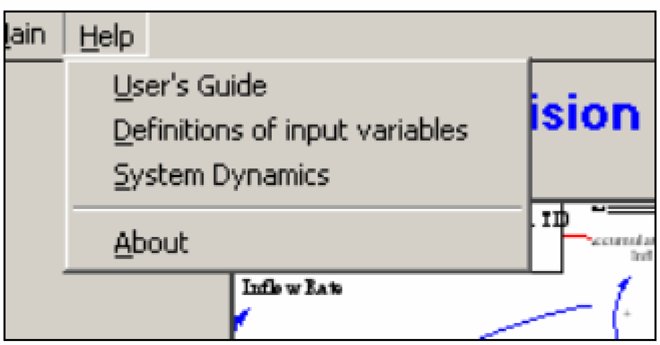


This menu is primarily used for navigation purposes. Three parts of the DSS Software can be accessed by clicking corresponding menu item. Clicking on the corresponding menu item, one can access one of the three parts of the DSS software. A menu item opens a new window and hides the main screen.

Note: Three menu items at the top perform the same function as clicking on the three buttons on the main screen.
Clicking the menu item Exit closes the application. A "Close Confirmation" window is not shown while exiting through this menu. Thus, clicking this menu will close the window without asking for confirmation.

Help Menu

The Help menu looks like this:



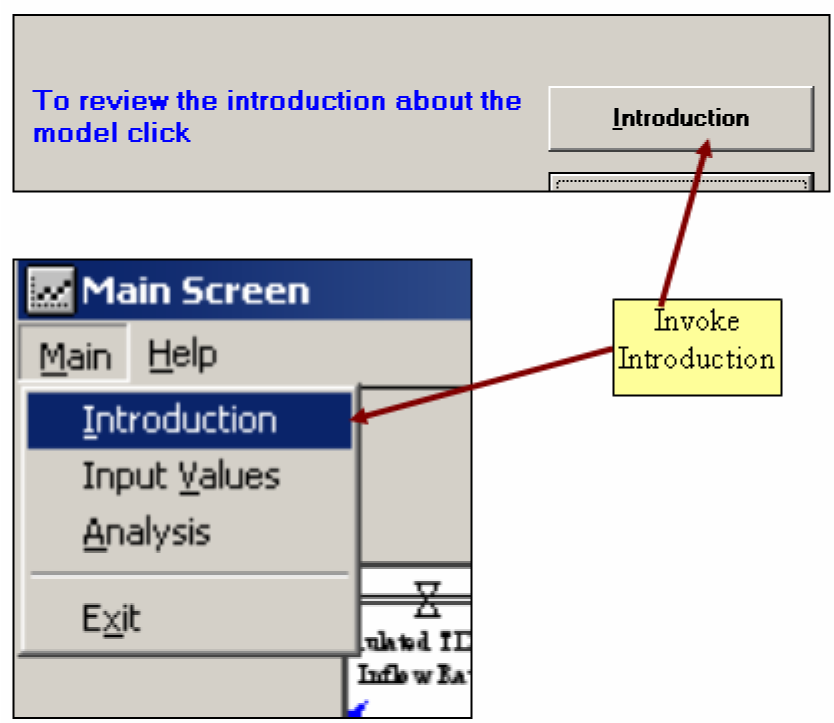
Clicking the "User's Guide" item invokes this user guide help file.
Clicking the "Definitions of input variables" item brings up a help file showing the definitions for all the input variables in the model.

Note: This definitions help file can also be invoked by clicking on the "information" button on the User Input screen.
Clicking on the "System Dynamics" item brings up a help file giving basic information on the topic of System Dynamics.

Clicking on the "About" item brings up a small window showing the brief information about the DSS Software.

Introduction

This section gives information on using the introduction part of the Decision Support System (DSS) software.



The introduction can be brought up either by clicking "Introduction" button on main screen or by clicking "Introduction" item under .

This section consists of the following topics:

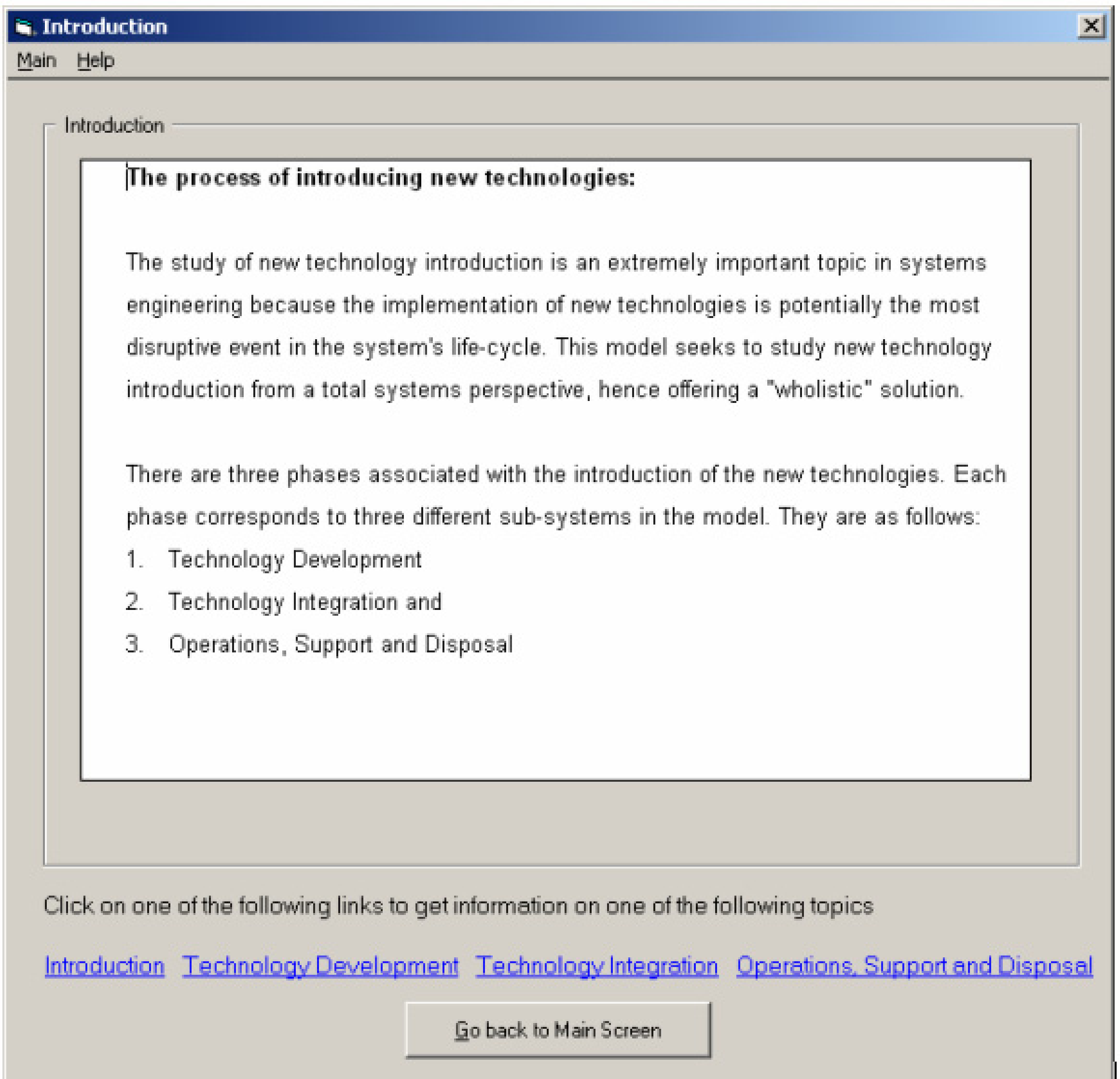
Navigating through the Introduction

Viewing the Model (Stock and Flow Diagram)

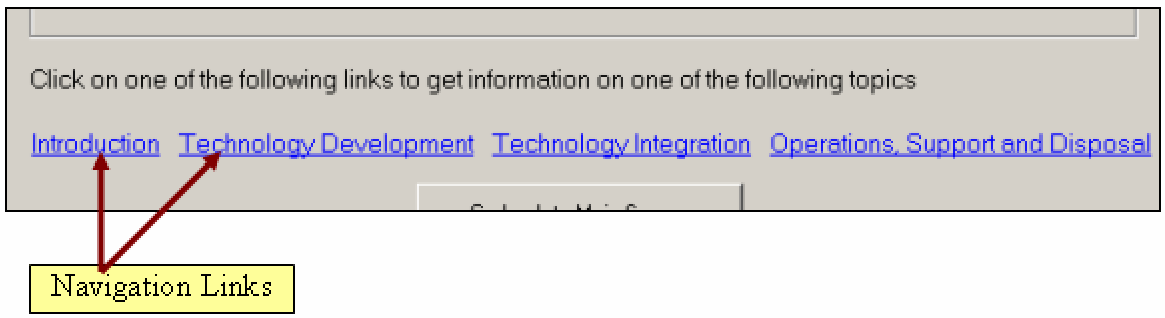
Model View Window

Navigating through the Introduction

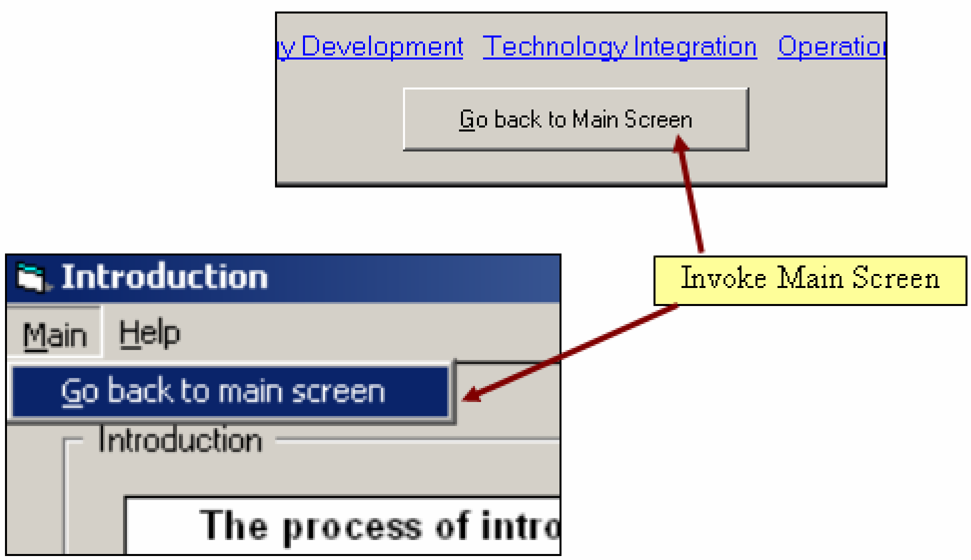
Clicking the Introduction button or the Introduction item from the brings up introduction window that looks as follows:



This window provides a link to general introduction to the model and also links to a description of the three subsystems of the model. The introduction to the subsystems can be seen by clicking the corresponding link at the bottom of the window.

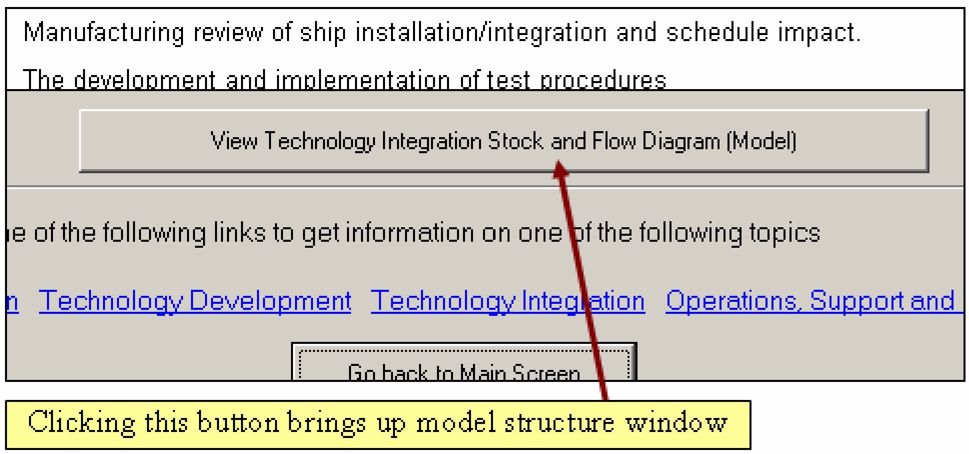


The user can go back to the Main Menu either by clicking "Go back to Main Screen" button at the bottom of the window or by clicking "Go back to Main Screen" Main menu item.



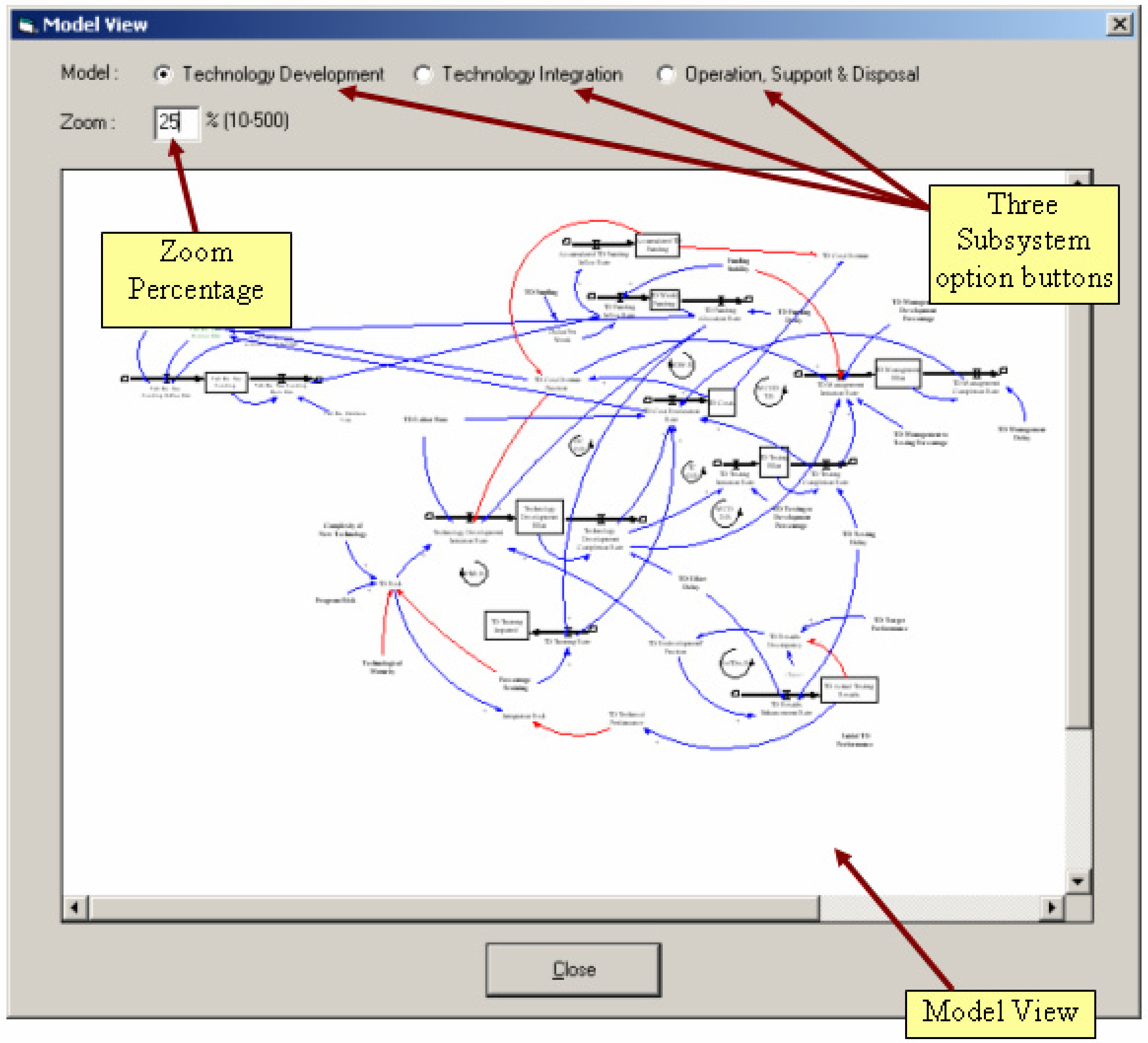
Viewing the Model (Stock and Flow Diagram)

The model structure (Stock and Flow Diagram) for any subsystem can be seen by clicking the "View subsystem diagram (model)" button at the bottom of that subsystem introduction window. Clicking this button brings up .



Model View Window

The model view window looks as follows:



The model structure (Stock and Flow diagram) is shown in the big frame at the center. The diagram for each of the subsystems can be seen by clicking on the corresponding option button. Zoom percentage can be defined (between 10-500%) by entering the appropriate value in the Zoom input box. The window is closed by clicking on the Close button at the bottom of the window.

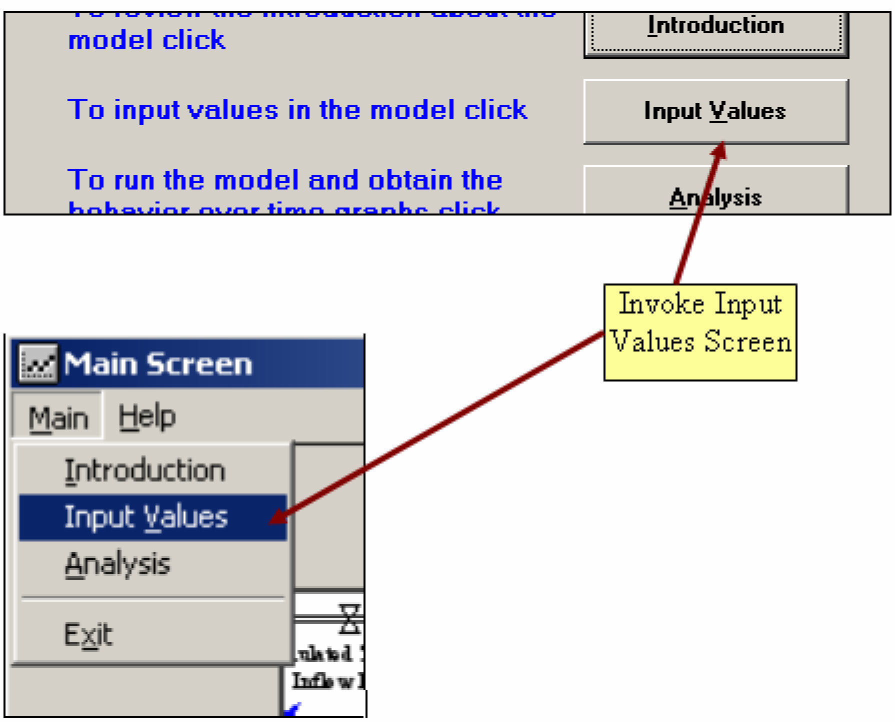
The model view is displayed just for observation. The user cannot make any changes to the model structure. The stock and flow diagram provides a pictorial representation of the structure of each subsystem.

Note: This window is a modal window. This means that the user cannot return to the software (its various screens and menus) unless this window is closed.

User Inputs

This section gives information on how to enter values into the software so that they can be used by model during its execution.

The user input screen can be brought up either by clicking the "Input Values" button on the main screen or by clicking the "Input Values" item under the .



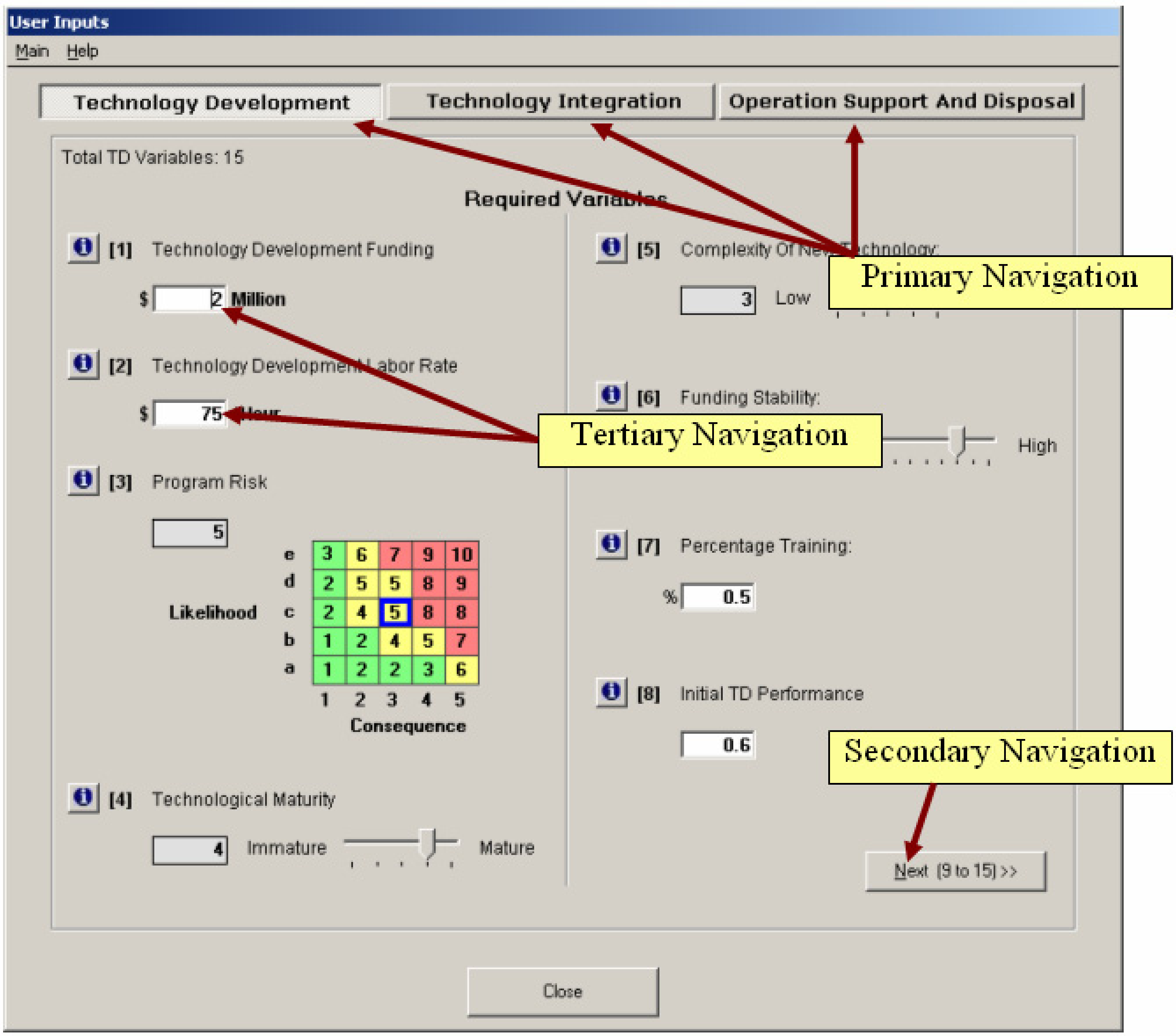
This section includes the following topics:

Navigating through the User Inputs

Entering Values in the User Inputs Screen

Navigating through the User Inputs

Clicking on the Input Values button or the Input Values item from the Main Menu brings up the Input Values window that looks like this:



There are 46 input variables for the three subsystems. The variables are divided on the screens according to the subsystem they belong to. For each subsystem, the input variables are also displayed according to their importance in the model. There are three types of navigations used, i.e.,

Primary Navigation: This navigation occurs between the three subsystems and is achieved through the three buttons at the top of each window that correspond to each subsystem.

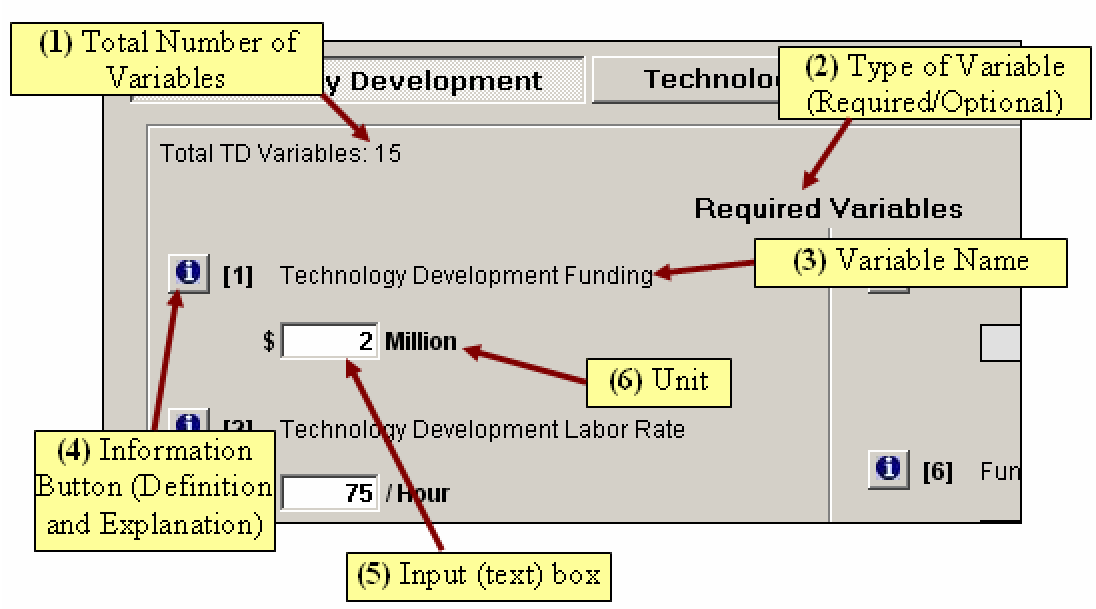
Secondary Navigation: This navigation is within each subsystem and is achieved through the buttons at the lower or right corner of the screen.

The third type of navigation is the Tertiary Navigation. This navigation is achieved by pressing the TAB keys on the keyboard. For example if the cursor is in one input box where the user just entered value, the cursor can be moved to the next variable's input box just by clicking Tab button on the keyboard. (This can also be achieved directly by clicking mouse in the desired input box.)

Entering Values in the User Inputs Screen

The user inputs screen enables the user to enter values for different variables. It also has some other features that are explained subsequently.

A section of the user inputs screen looks as follows:



(1) Total Number of Variables: This label gives information on the total number of variables in that subsystem.

(2) Type of Variable: This label gives information on the type of variable, whether it is required or optional. Required variables are the most important variables that the user MUST enter values for. Optional variables are less important variables and the values for these variables can be the default values already included in the software. The values for these variables may be changed if the user chooses to do so.

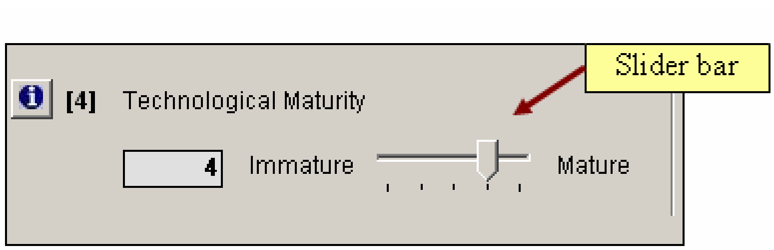
(3) Variable Name: This label shows the name of the variable.

(4) Information Button: Clicking this button, information on that variable can be invoked. A help window is opened that gives an explanation for that variable.

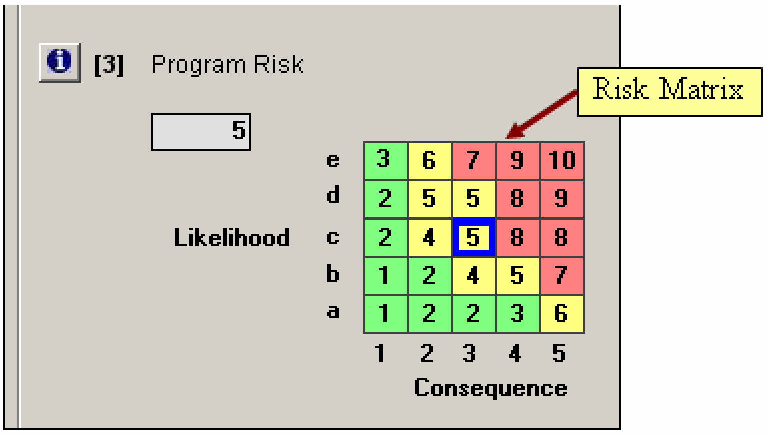
(5) Input (text) box: The value of the variable is entered in this box.

For some of the variables, there are different ways to input values for these variables.

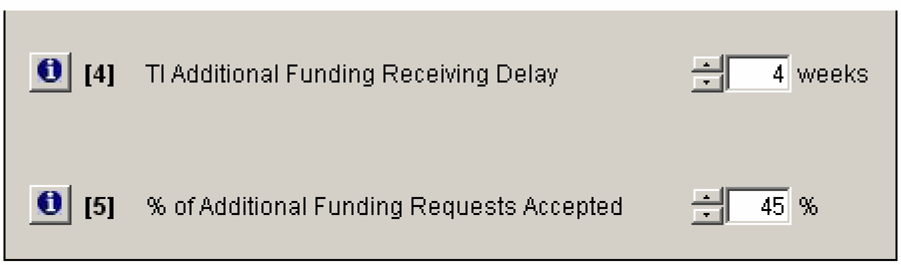
They are as follows:



For the variable Technology Maturity, the input box is grayed and the value cannot be entered in the box, but the value can be changed using the slider bar on the right side. Sliding the bar automatically changes value in the input box.



For the Program Risk variable, the value is entered with the help of Risk Matrix. Based on the likelihood of failure and its consequence, an appropriate box in the matrix is clicked. The value corresponding to the box in the matrix is reflected in the adjacent input box.



For some variables there are two small buttons adjacent to the input box. The value can either be changed by clicking the up and down buttons or by directly typing it into the input box.

Note: Use of up and down buttons will cause increment (or decrement by) 1. Where typing the value in the input box lets the user input fractional values e.g., 45.4.

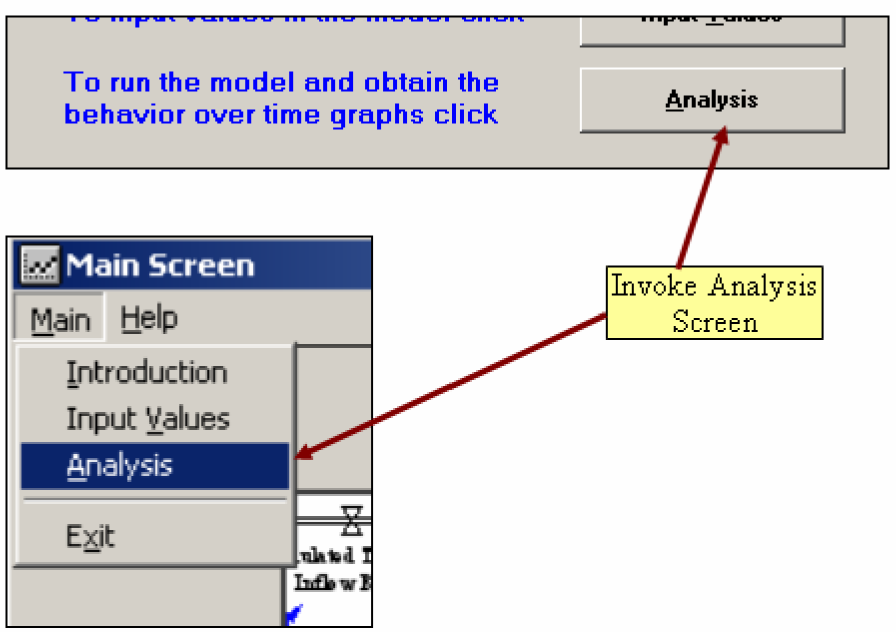
(6) Unit: This is the unit for the variable. If the variable is dimensionless, it is left blank.

Note: Entering values in the user inputs screen does not immediately change the values of the model variables but they are saved in an array. The values in the array are updated as the values are entered in the input boxes. The values are also saved in a text file (runname.txt) that is updated when the user inputs window is closed. It is not recommended that the user to make changes to that file.

Analysis

This section gives information on how to run the model based on the input values and observe behavior over time graphs of various variables.

The Analysis window can be brought up either by clicking on the "Analysis" button on or by clicking the "Analysis" item under main menu.

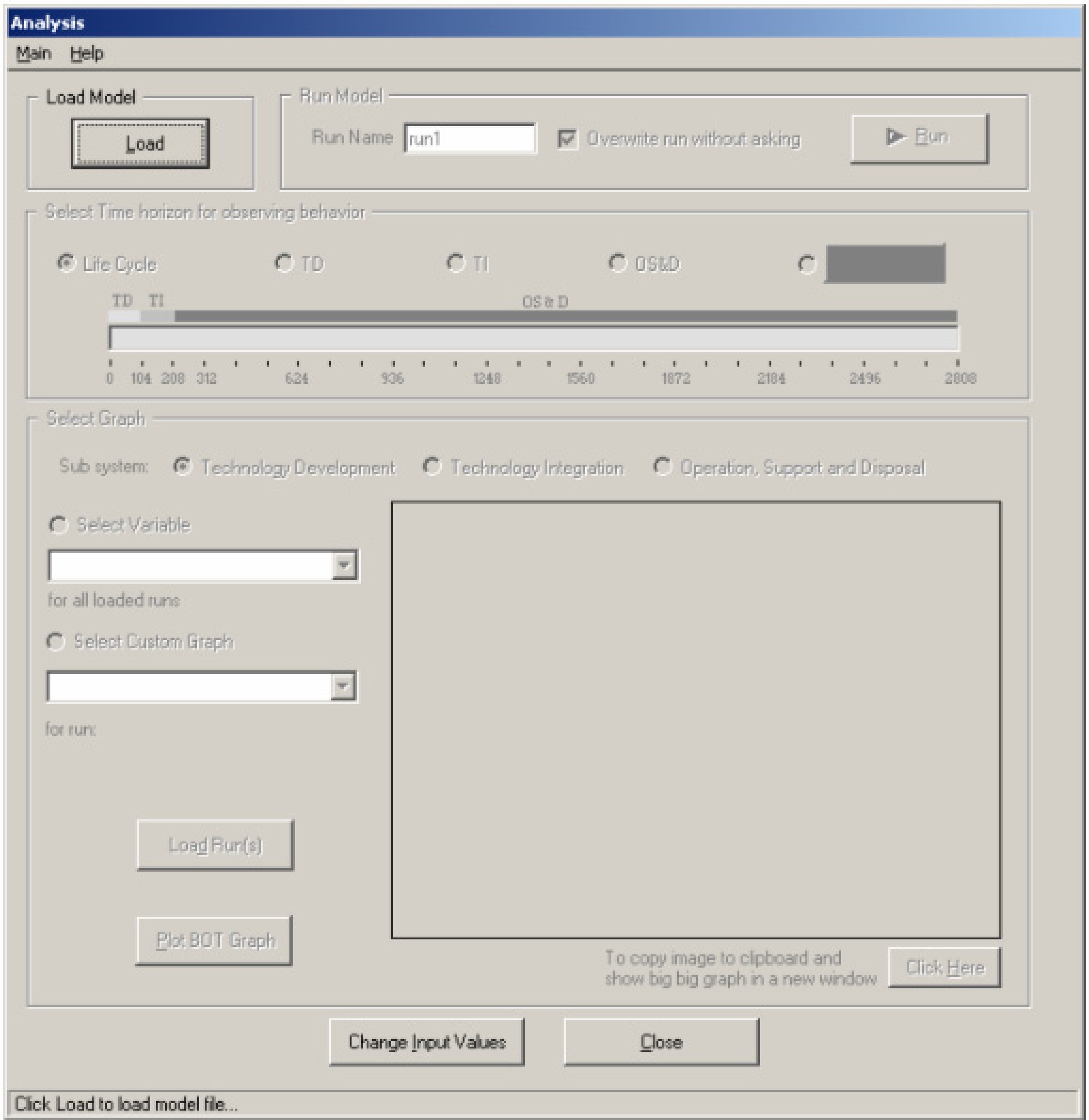


This section includes the following topics:

Selecting the Time Horizon for observing behavior
Selecting a Behavior Over Time (BOT) Graph

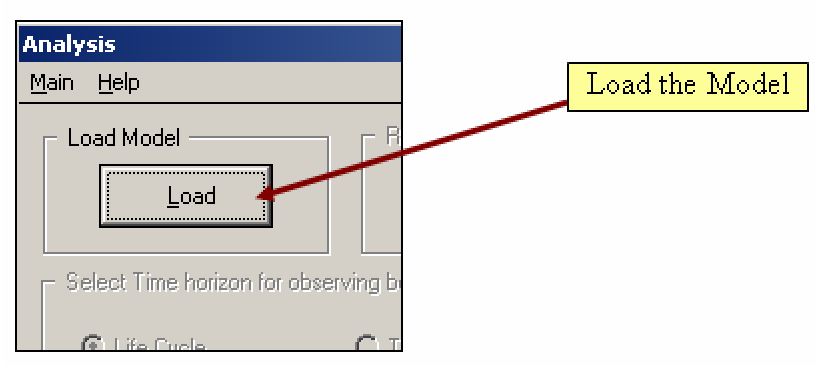
Loading the Model

Clicking the Analysis button or the Analysis item from the Main menu brings up the Analysis window that looks like follows:



Most of the window contents are grayed since these controls can be used only when model is loaded and the model run is complete.

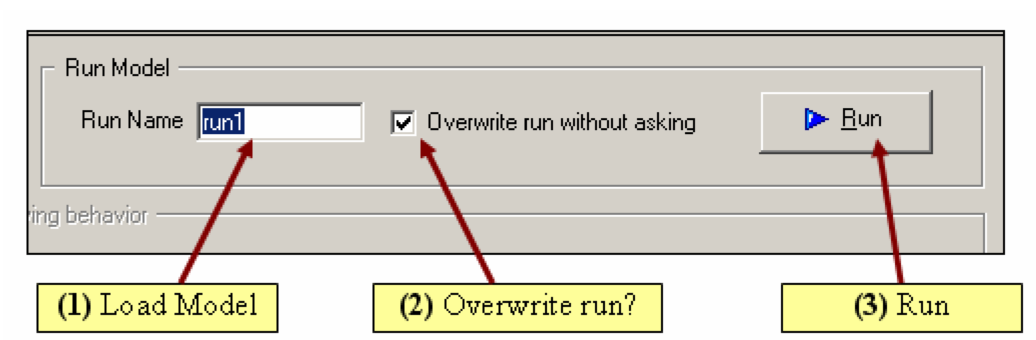
The model can be loaded by clicking the "Load" button.



The model file name and path are already set in the application. The user need not select any model file explicitly.

Running the Model

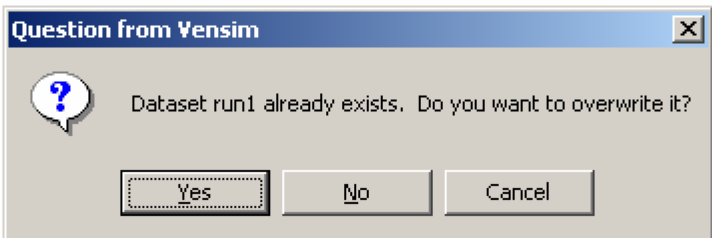
After the model is loaded, the "Run Model" part of the screen becomes enabled and the user is ready to run the model.



(1) Run Name: Each run of the model can be saved as a different run that is nothing but the execution of the model for a different set of input and output data. Thus a user can make multiple runs changing one or more input values. The user can then compare these different runs. Each run is identified by its name referred to as "run name". Suppose user makes three different runs of the model for three different values of input

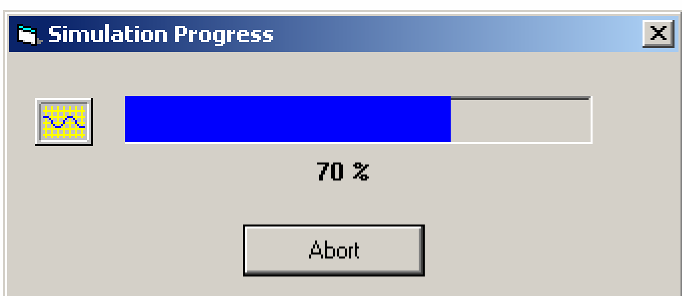
variable, say risk. He/she can name the three runs as risk_run1, risk_run2 and risk_run3. He/she can then compare the results of these three different runs as described later. The default run name is "run1".

(2) Overwrite run without asking: If this box is checked, the run is overwritten if it already exists. If this box is unchecked and if the run name already exists, a message box is displayed that confirms overwrite or asks for a different run name. The message box looks as follows:

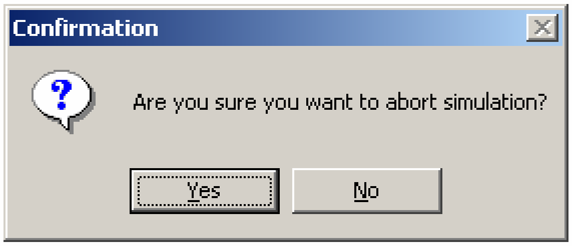


Clicking Yes overwrites the run. Clicking No asks the user for a different run name.

(3) The Run button: Clicking this button, the model run starts. Usually, it takes very short time for model to run. A window is displayed, as a model is running, showing the progress of the run.



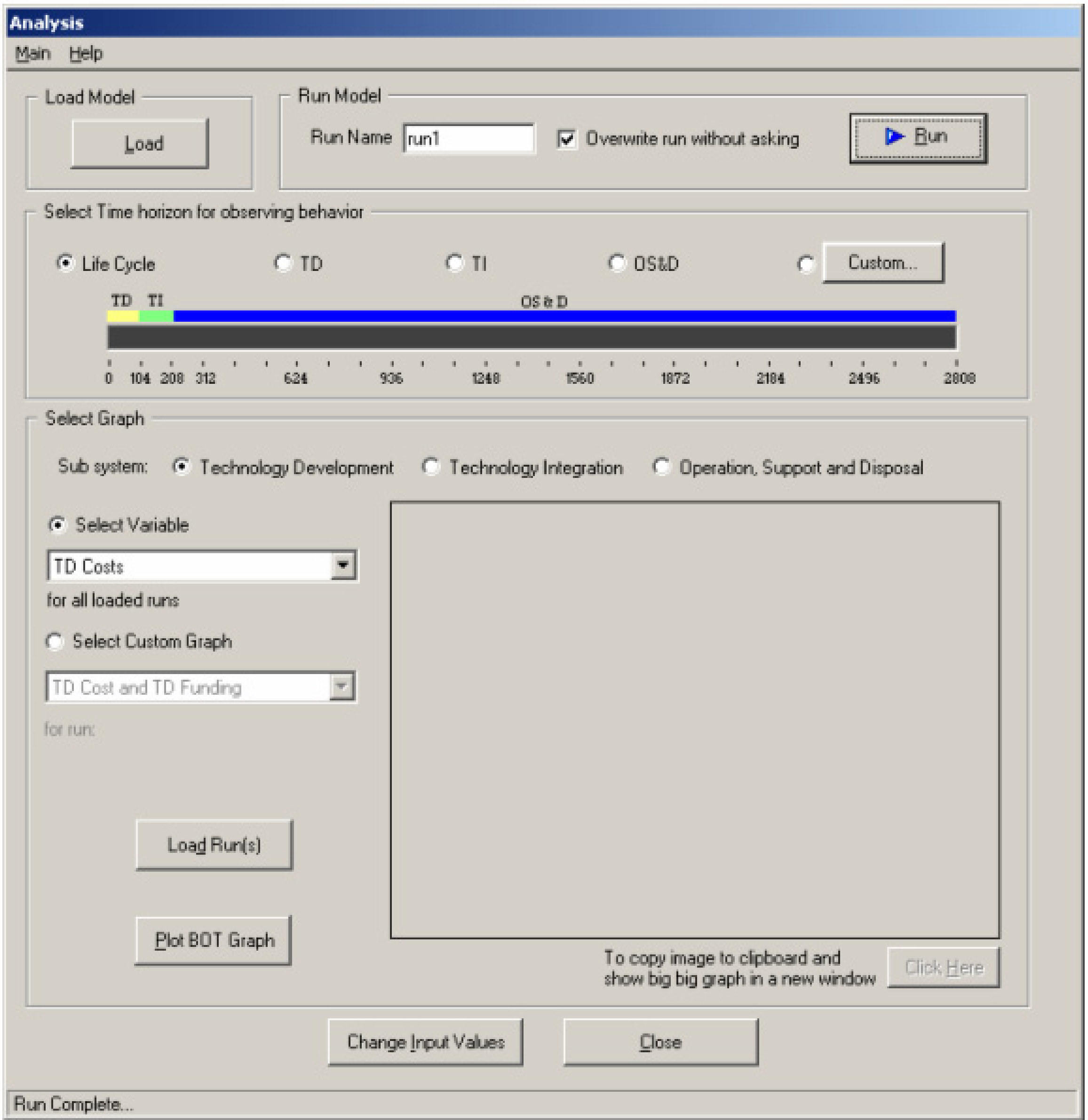
Window shows the progress of the simulation as a percentage of total time. It also has a button to abort the simulation. Clicking the Abort button brings up a confirmation window.



Clicking No resumes the run. Clicking Yes terminates the simulation. The values are saved in a run till the time the run was made.

Selecting the Time Horizon for observing behavior

After the run is complete, all the controls of the screen are enabled. The window now looks as follows:

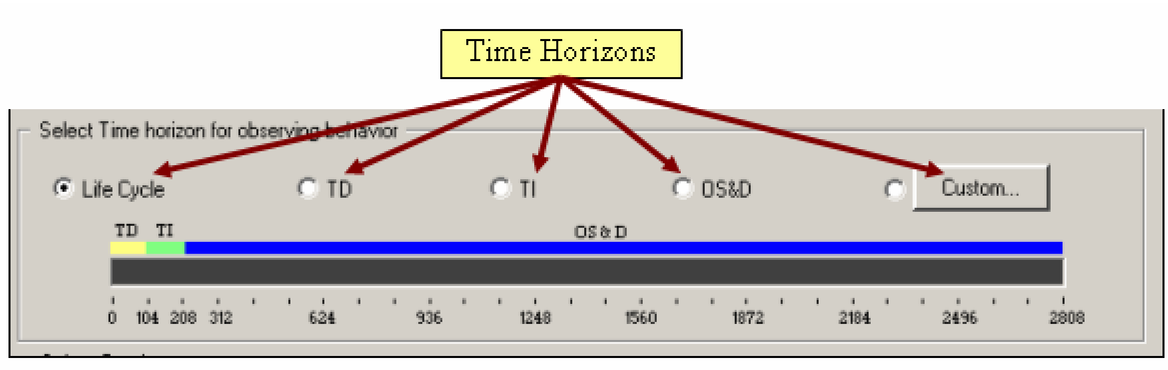


Time horizon of the model run is very long and equals 54 years. Because of such a long time horizon, while observing graphs for the whole lifecycle, some peculiar characteristics and shapes of the graphs cannot be clearly identified. The feature of selecting different time horizon enables the user to observe behavior for a specific selected time horizon, which shows a "focused time" view as compared to the total lifecycle.

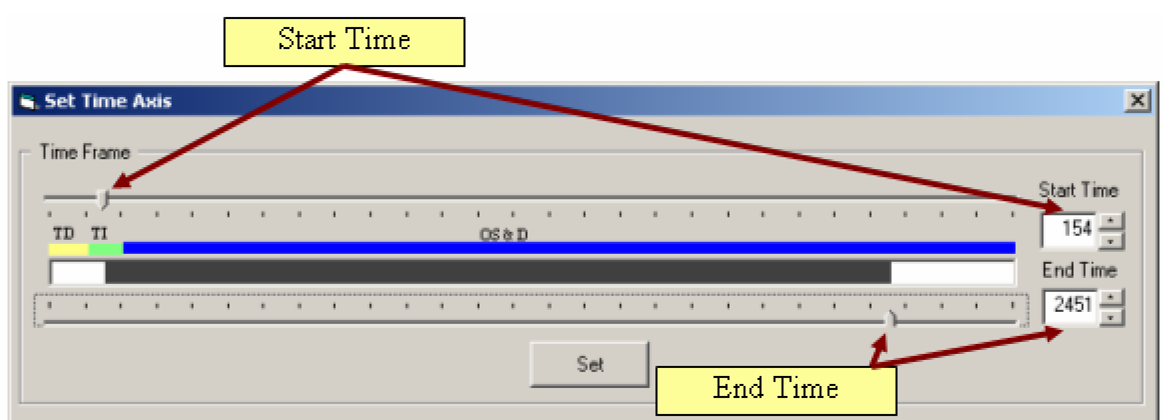
The assumed time horizons for each subsystem are as follows:

1. Technology Development (TD): 104 weeks (or 2 years)
2. Technology Integration (TI): 104 weeks (or 2 years) after TD is complete
3. Operations, Support & Disposal (OS&D): 2600 weeks (or 50 years) after TI is complete

Total lifecycle: 2808 weeks (or 54 years)



The time horizon for any of the subsystems for the total life cycle can be selected by clicking the corresponding option button. A "custom" option enables the user to select the starting and ending times. A window comes up after clicking the "custom" option that looks as follows:

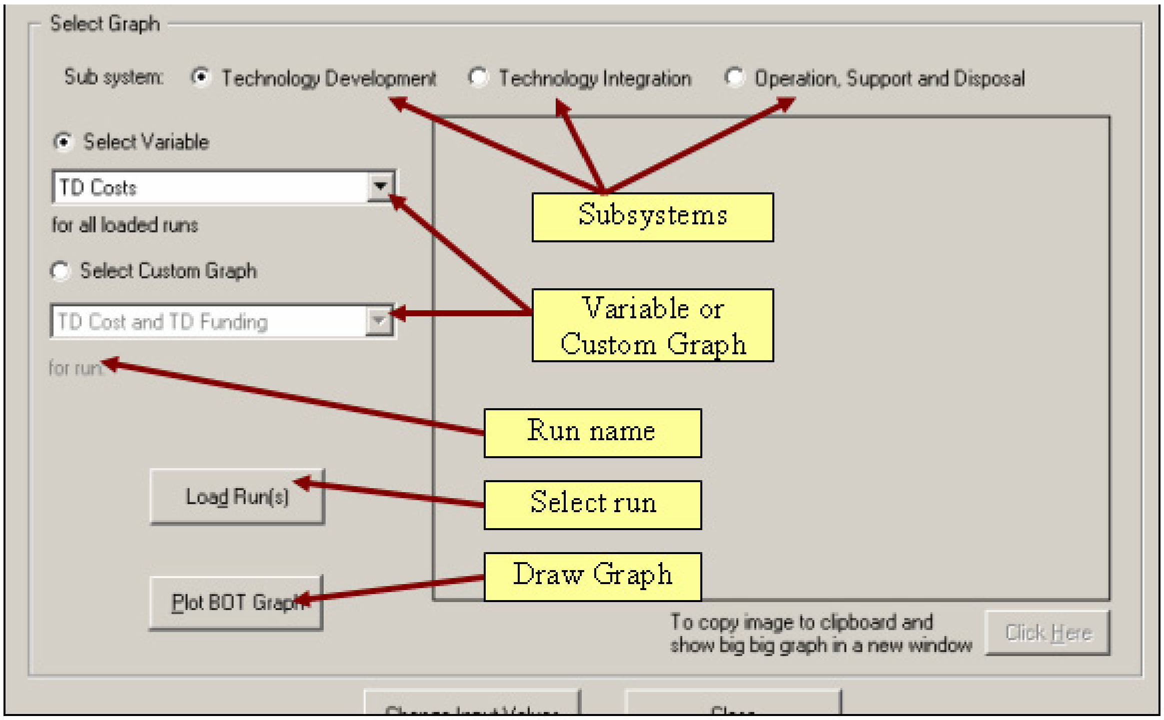


Using two sliders, the user can select the starting and ending times for time horizon. He/she may enter values in the adjacent input boxes as well.
The colored bands show the time frame for each subsystem.
Clicking the "Set" button sets the time horizon and closes the window.

Note: The time horizon selection is only for the purpose of observing variable behavior. Irrespective of the selected time horizon, the model is run for the entire lifecycle.

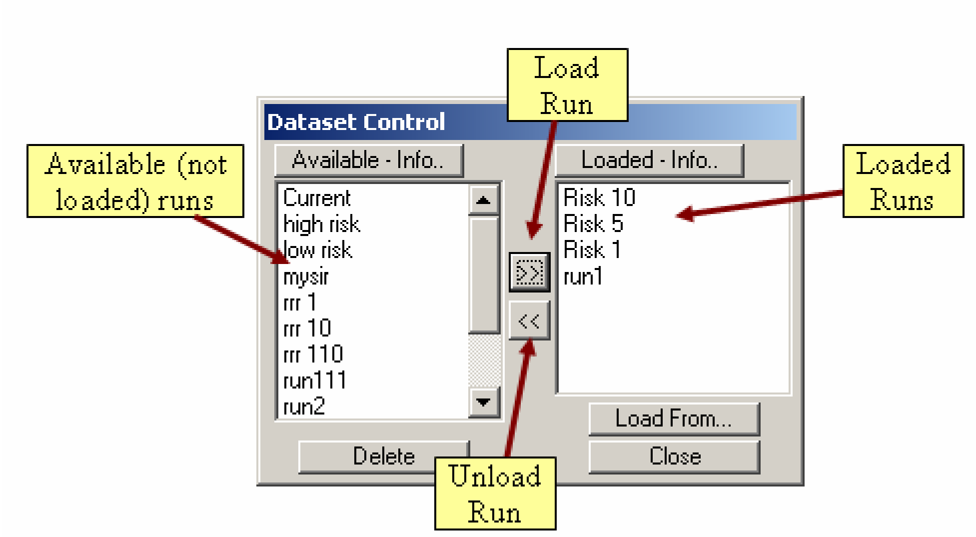
Selecting a Behavior Over Time (BOT) Graph

Once a time horizon is selected, behavior over time (BOT) graphs can be selected and observed.
The BOT graph section looks as follows:



All the variables are divided by the subsystem they belong to. After selecting the desired subsystem, a BOT graph can be viewed either for a variable in that subsystem or for a pre-stored custom BOT graph in that subsystem. A BOT graph for a variable can be seen for all loaded runs while the custom BOT graph can be seen only for the first loaded run.

Runs can be loaded by clicking the Load Runs button. A window is brought up that shows list of loaded and available runs:



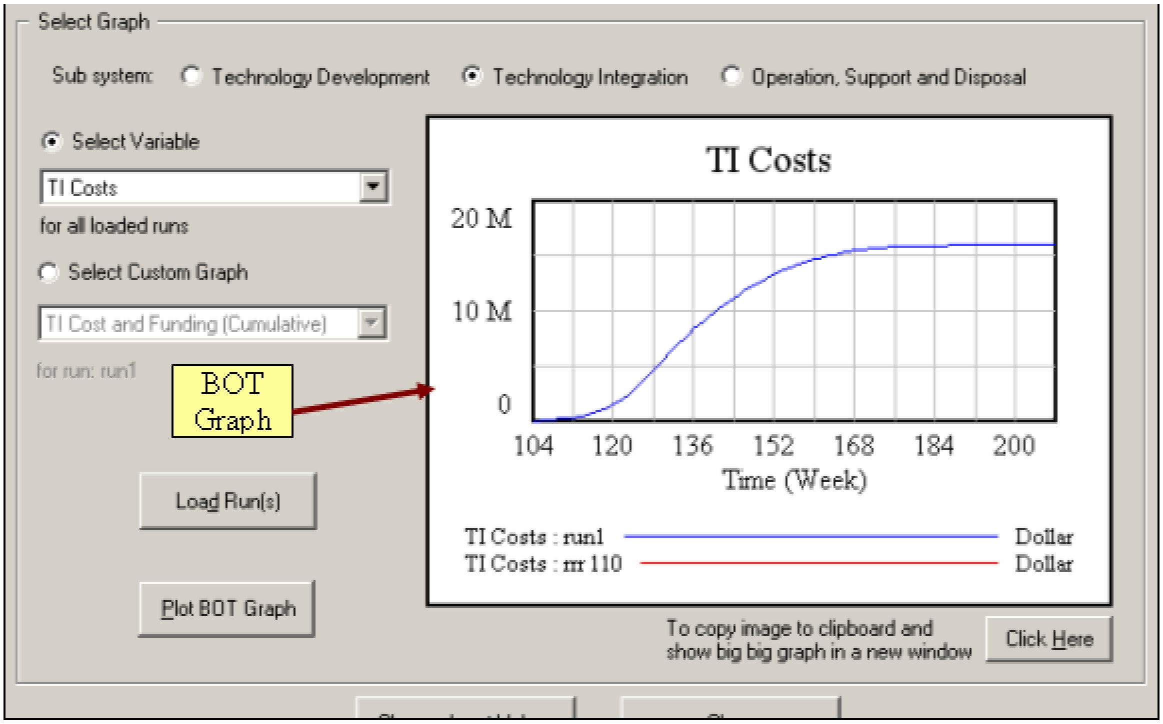
The list on the left contains all the available runs that are not loaded. They are nothing but the different run names for different runs that were executed.

The list on the right side shows loaded runs. Loaded runs are the runs that are used for displaying BOT.

Runs can be selected and then loaded or unloaded using the two buttons in between the two lists. Multiple runs can be selected using the Shift or the Ctrl key.

Clicking Close closes window.

Clicking on Plot button plots the BOT graph in the adjacent window as shown below.



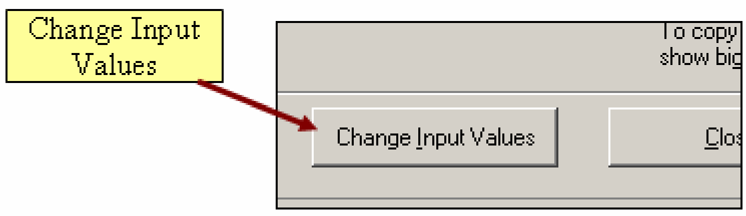
Copying a BOT Graph to the Clipboard

It is often desired to view a larger BOT graph for clarity and to identify specific behavior patterns. It is also required to use the BOT graph into some other software (e.g. MS Word, Power Point).

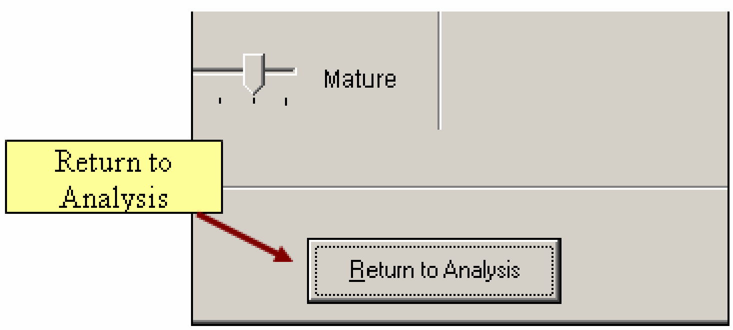
Two functions can be achieved by clicking the "Click Here" button in front of "To copy image to clipboard and show big BOT graph in a new window" label. Clicking this button opens a larger BOT graph in a new window. It also copies the BOT graph to clipboard at the same time. The BOT graph can be pasted into other software by using the "paste" command in that software.

Changing Input Values and Rerun Model

Quick changes can be made to input values by clicking the button "Change Input Values" at the bottom of screen.



This brings up the User Inputs window where the user can alter the input values. Clicking button "Return to Analysis" returns directly to the Analysis window. The user can, now, run the model with a different run name.



Vita

Pushkar H Damle

Personal History:

Born in Pune, Maharashtra State in India on November 16th 1977.

Education:

- M.S., Industrial & System Engineering (Operations Research), Virginia Polytechnic Institute and State University, August 2000 – July 2003
- B.S., Mechanical Engineering, University of Pune, India, August 1995 – May 1999

Professional Experience:

- Capital One Financial Corporation
  Recently accepted an offer to start working from Mid September 2003
- Virginia Polytechnic Institute and State University
  Graduate Research Assistant, August 2001 – June 2003
- Abweb Infotech Pvt. Ltd, Pune, India
  Software and Web Developer, July 1999 – June 2000

Professional Membership and Certifications:

Student Member, Institute of Operations Research and Management Science (INFORMS), Since August 2000

Japanese Languate Proficiency Test, Level 3, Conducted by Japan Foundation, 1999