DEVELOPMENT OF A PROJECT MANAGEMENT SIMULATOR APPLICATION

by

Thomas Smolko

Project and Report submitted to the Faculty of the

Virginia Polytechnic Institute and State University

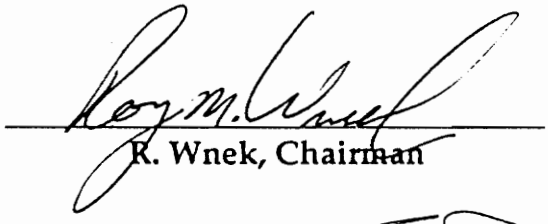
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

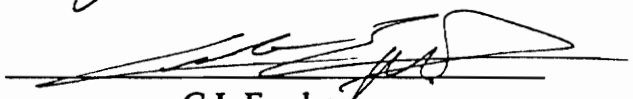
in

Computer Science

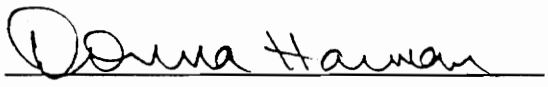
APPROVED:



R. Wnek, Chairman



C.J. Egyhazy



D. Harmon

January, 1994
Blacksburg, Virginia

Table of Contents

List of Tables

List of Figures

Introduction

Executive Summary

This masters project involved the design and development of an integrated pair of Windows-based PC applications that provides a realistic business project simulation environment for use in project management training.

The *PS Admin Workstation* is used by the instructor to configure the project simulation environment. The *Project Simulator* interface is used by the student to make critical management decisions that direct the project simulation.

Both applications integrate with *Project Scheduler 6 for Windows*, developed by the Scitor Corporation, to track the completion of the simulated project.

About this Document

This paper records all stages of the simulation life cycle used in the development of the *Project Simulator*. The simulation life cycle extends from the proposal of the project through the definition phase to the implementation and walkthrough of the simulation prototype.

The sections of this paper include:

- Introduction
- Review of Literature
- Materials and Methods
- Project Results
- Discussion
- Conclusion
- Appendix

The **Introduction** section describes the masters project requirements for the Virginia Polytechnic Institute and details the learning objectives for this masters project. This section also describes the client that requested and reviewed the simulation application.

The **Review of Literature** section describes the research performed in the development of this masters project.

The **Materials and Methods** section details the methodology used to complete the simulation application. This section also includes a discussion of simulation orientation chosen for the *Project Simulator* application.

The **Results** section details the decisions made in defining, developing and completing the Project Simulator application. The project results include the following deliverables:

The **Discussion** section describes the outstanding issues involved with the completion of the fully-functional Project Simulator application.

The **Conclusion** section summarizes the results and implications of the masters project.

The **Bibliography** section contains a list of all publications consulted in the completion of this project.

The **Appendices** contain screen captures from the two Project Simulator applications and the project tracking software integrated with the applications.

Project Objectives

The *Project Simulator* was designed and prototyped to satisfy the requirements for a masters project in Computer Science at the Virginia Polytechnic Institute (VPI) located in Blacksburg, VA.

The requirements for successful completion of a masters project at VPI include:

- ❑ Determining an appropriate advanced computer science topic to investigate
- ❑ Formulating a thesis or identifying a development project to pursue
- ❑ Performing research on the computer science topic
- ❑ Writing a project paper describing the research, methodology and results of the masters project

Learning Objectives

The main learning objective of this masters project is to deepen my knowledge of computer-based simulations by researching, designing and implementing a project management simulation prototype.

I intend to gain an appreciation of the life cycle for developing simulations, specifically:

- Interviewing clients to determine simulation scope and project objectives
- Investigating various simulation world views and orientations and selecting the best approach
- Designing a general purpose project management simulation application
- Prototyping and testing user interface designs for both instructor and student functionality
- Determining the appropriate language and tools with which to implement the simulation application
- Implementing a prototype of the project management simulation application
- Planning tests to validate and verify the full simulation application results
- Presenting the simulation prototype to the client

Each of the steps in the simulation development life cycle are important to a successful implementation.

Client Interviews

Interviewing the client for the simulation is the most critical step in developing an effective simulation. The objectives of those requesting the simulation will dictate the specification, design and implementation of the system.

World Views

Choosing an appropriate world view, or simulation orientation, will drive the design and the tools needed to develop the simulation.

Simulator Design

Documented requirements and functional specifications are needed to reach agreement with the client on the specific objectives of the simulation and to provide a foundation for the development of the data structures and processes of the application.

User Interface Prototypes

An effective, user-friendly front-end design is a key element of a successful simulation environment. The users should spend as little time as possible learning the program interface. Walk-through with the end-users of the system will provide useful feedback.

Software Tools

Selecting the best software tools to implement the simulation will determine the development time, the richness of system features and the ease of future maintenance.

Implementation

Implementation of the simulation will require following software development standards and developing appropriate internal documentation as the system is being built.

Validation & Verification

Appropriate tests are needed to validate and verify the simulation results. Validation will test whether the model accurately reflects the project management environment. Verification will compare the simulation results to real-world measurements.

Presentation

A final presentation to the client is needed to gain feedback on the prototype and provide the appropriate transfer of software and documentation.

Client Description

The *Project Simulator* application has been requested by the training organization of a large electronic commerce services provider headquartered in Rockville, MD.

The *Project Simulator* application is to be designed and prototyped for use in specific project management training courses. A representative from the training organization reviewed the requirements and the final prototype of the application.

Review of Literature

Simulation Literature

The rapid advancement of computer technology has made computer-based simulation models an effective tool for evaluating computer systems and complex real-world environments.

While volumes of research have been performed on queuing theory and the simulations of computer-based systems, relatively little of the research has been focused on the area of project management simulation.

Project management simulation involves the modeling of the real-world characteristics that influence the completion of a series of specific, interrelated tasks that must be performed or directed by people.

Projects that involve people resources are highly dependent on the often immeasurable characteristics of the human beings. Accurately predicting the actions of people is impossible, which precludes the development of theoretical solutions in the area of project management.

Project Management Literature

In my search for appropriate literature, I came across many articles that discussed the techniques of project management. The entire October, 1993 issue of *Communications of the ACM* was devoted to technical project management. Articles included:

- *"Work Organization: Paradigms for Project Management and Organization,"* by Larry L. Constantine,
- *"A Project Planning and Development Process for Small Teams,"* by Marc Rettig and Gary Simons, and
- *"Design by Walking Around,"* by Richard A. Zahniser.

While these articles provided insight into methodologies for managing people projects, none of them discussed simulating the project environment. In fact, through all my research I found no applicable articles or books that covered project management simulation.

In the development of my masters project, I relied on the principles of developing computer-based simulations found in several textbooks and publications, including:

- *The Art of Computer Systems Performance Analysis*, by Raj Jain, and
- *"AI: What Simulationists Need to Know"* by Miller, Firby, Fishwick, Franke and Rothenberg.

Simulation Modeling

As Jain emphasizes, the most important decision in the development of a simulation model is the choice of **world view** and **orientation**. These choices dictate the software tools needed, the specifics of how the simulation application will be built and, ultimately, the success of the project.

The world view of a simulation is a philosophical description of how the elements in the simulation will be represented in the model. The orientation of a simulation model reflects the world view and describes the rules or mechanisms that will drive the simulation.

World Views and Orientations

The four main approaches used to model the interaction of entities over time include:

- State Change or Next Event orientation
- Material Flow or Transaction Flow orientation
- Intercommunicating Entities or Object orientation
- Functional or Activity Scanning orientation

Next Event orientation uses events, a point in time where the state of the system changes, to drive the simulation over time. The system clock is advanced by pulling the next event off of an ordered event queue and processing its attributes to change the state of the system.

Transaction Flow orientation traces an entity through the system from its arrival to the system to its departure. The entity may flow between a number of objects or states in the system.

Object orientation maintains information on the construction and interaction between entities in the system. Object orientation is also known as the Producer-Consumer model.

Activity Scanning orientation periodically reviews all activities in the simulation to determine which can start or have been completed after each discrete event occurs.

Simulation Purpose

Another choice that needs to be made by the simulation model developer is the **purpose** of the application. Simulation models can have a limited purpose, i.e., designed to simulate a specific configuration of a system, or can be general purpose models that can be configured to run different scenarios.

To complete this masters project, I integrated these fundamental simulation principles with my experience with other computer-based project management simulation applications and my own creativity.

A complete list of publications that were consulted throughout this project can be found in the **Bibliography** section.

Materials and Methods

Project Materials

The physical materials needed to complete the *Project Simulator* applications included both an IBM PC-compatible personal computer and appropriate software development tools.

Application development was performed on an AST Premium 386C personal computer with 4 megabytes of Random Access Memory (RAM) with the capability to run the Microsoft Windows 3.1 graphical environment.

The software tools included Microsoft Windows 3.1 and appropriate development tools to create the simulation applications. The constraints of the system, as described in the **Results** section, required a flexible and familiar set of software tools.

Visual Basic 3.0

To develop the application user interfaces, I selected one of the quickest and most effective Windows applications development environments, *Visual Basic 3.0* from Microsoft.

Visual Basic 3.0 allows the programmer to easily design the look and feel of their application using familiar click and drag actions. The programmer can then write code behind the graphic objects to react to user events, providing a fully graphical, event-driven Windows application.

Visual Basic 3.0 also provides a strong database interface that allows the programmer to access and manipulate data within a number of standard database formats, including dBase III.

Microsoft Access 1.1

To develop the additional system databases, I selected a companion product to *Visual Basic 3.0 - Microsoft Access 1.1*. Microsoft developed the *Visual Basic* and *Access* applications using the same underlying code so that the packages integrate very smoothly.

The *Microsoft Access 1.1* front-end application also provides the ability to open and modify dBase III-formatted databases. This capability allows the programmers to view or modify the database imported and exported from *Project Scheduler 6*.

Simulation Decisions

The choice of world view and orientation for the *Project Simulator* application will depend on both the description of the system being modeled and the requirement constraints imposed by the client and the availability of software tools.

Selecting the Purpose

Based on the initial interview with the client, the *Project Simulator* application must be a general purpose simulation model. The client wants to be able to configure the entities in the model to teach the students various project management techniques.

Selecting the Orientation

The selection of a world view and orientation to the simulation model is a more difficult decision. There are a number of possible ways to model the entities and interactions in a complex technical project.

The client wants to integrate commercial project tracking software with the simulation, so the world view of that application must be considered first. The *Project Scheduler 6* application uses the following entities and model:

- A **project** as a collection of inter-dependent tasks called jobs.
- Each **job** has a definitive start and finish time which may dependent on the completion of other jobs in the project.
- Each **job** is completed by assigning specific resources to that job for a duration of time.
- A **resource** is a specific worker or physical entity that has limited availability and associated costs for use.

Each job in this model represents an activity in the simulation model terminology. The start and finish times for each job represent an event. Each resource represents an entity of the system.

In addition to tracking the assignment of resources to jobs, the *Project Simulator* application must allow for staff development activities, such as meetings, reviews, surveys or worker training. These activities can be represented as activities in the simulation model, with their start and finish times as discrete events.

Based on this system description, the best simulation orientations are Next Event orientation and Activity Scanning. Both of these orientations focus on system states or activities affected by the occurrence of discrete events.

The Transaction Flow and Object orientations focus more on process interactions and the flow of material from one entity to another. While jobs are dependent on each other, the specific system entities that are passed between them (phases of project development) are not modeled in the system description.

Requirements Constraints

A major requirement of the system is that the application be as simple and maintainable as possible. Maintenance of a software application implies that the system be built with software tools familiar to the developers that must maintain the applications.

This constraint, along with the lack of funds available to purchase simulation languages, requires the selection of a simulation orientation that can be easily modeled in a general purpose language.

Of the two selected orientations, the Next Event orientation is most easily built using general purpose programming languages. The *Project Simulator* will use Next Event orientation to drive the simulation.

Simulation Rules

Once the simulation model purpose, world view and orientations are chosen, the interactions between entities in the model must be investigated and measured. These entity relationships will drive the development of simulation rules that create and manage the discrete events in the system.

The study of the management of complex technical projects is a field unto itself. Bookstores and libraries are filled with an array of project management books that describe the ever-changing roles and responsibilities of project and business leaders.

Managing projects requires managing both the physical and the people resources needed to complete a series of tasks. Accurately predicting the behavior of people in a number of possible scenarios is difficult if not impossible.

Developing a project management simulation application would then seem an impossible task as well. Luckily, observations of human behavior over time have revealed that how people perform assigned tasks is a result of two main factors - their skills and their motivation.

People Skills

A person's skills include technical skills, such as computer programming proficiency, as well as interpersonal skills. Interpersonal skills are how a person interacts and relates to other people with whom they must work.

These skills can be observed and measured over a period of time and can be rated according to defined criteria. While interpersonal skills are more subjective, the effects of these skills can be observed over time based on the productivity of the individual worker and the team.

Motivation

A worker's motivation is much more difficult to measure and may vary widely over time. A person's motivation may be affected by the creativity of the tasks that they are assigned, the people with whom they must work, and a host of unrelated personal factors.

While motivation is difficult to measure, there are a number of techniques that help increase worker motivation, such as praising or rotating work assignments. There are also management techniques that decrease worker motivation, such as extended overtime and lack of recognition.

Implementation

The *Project Simulation* application must be a general purpose project management simulator, so that these aggregate behaviors of people can be used to model the completion of tasks in the simulated project.

The simulation applications must also be simple. This requires that the mechanisms used to calculate the effects of worker skills and motivations must be easy to understand and change.

The rules used to guide the simulation involve the setting of skill and motivation attributes for people resources in the simulation. These attributes change over time based on the events that occur during the simulation. These attributes are then used to determine the amount of time needed to complete the assigned tasks.

A more detailed description of these simulation rules are found in the Functional Specification of the Results section.

Development Methodology

For implementation of the simulation application prototypes, I choose a top-down methodology which focused on building the overall system and interface structure before developing any specific functionality.

The choice of development methodologies for the fully-functional simulation applications will depend on the structure and experience of the development team.

Project Results

Introduction

The Results section records the design and development of the Project Simulator applications. This section is divided into the following major subsections which follow standard software and simulation life cycle methodologies:

- Requirements Definition
- Requirements Review
- Functional Specification
- User Interface Design
- Implementation
- Validation & Verification

Requirements Definition

This subsection defines the functional and non-functional requirements of the *Project Simulator* application. The purpose of this section is to outline all project requirements and to prioritize each requirement to guide development of the system.

The Requirements Definition is organized into the following major topics:

- Product Description
- User Characteristics
- Functional Requirements
- Non-functional Requirements

The following priorities used in the requirements definition reflect the importance of each requirement in the development of a working Project Simulator application:

ESSENTIAL	Absolutely necessary to provide a project management simulation; will be developed first.
HIGH	Required to provide a project simulation with realistic decision criteria and project environment; will be designed for and developed with essential priority requirements.

MEDIUM	**Provides additional complexity and features; will be designed for and developed after completion of essential and high priority requirements.**
LOW	**Future enhancements which will be designed for but not implemented until after validation and verification of the system.**

Product Description

The scope of this masters project is to define, design and build an interactive project management simulation prototype. The *Project Simulator* will be a user-friendly Windows-based application to be used as an interactive learning tool in project management training courses.

The simulator application will allow an instructor to configure the system with specific project information. The students using the simulator application will guide the execution of the project over a specified time period by making a variety of management decisions.

User Characteristics

There will be two categories of users for the *Project Simulator* application, instructors and students. Each category of users will have different backgrounds and familiarities with both project management techniques and the *Project Simulator* application.

Instructors

Instructors will be qualified project management trainers, knowledgeable of project management principles. They will also be familiar with the techniques necessary to teach adults effectively and may have experience with other project management simulation software.

Instructors will be trained in the use of the *Project Simulator* application and have access to the product documentation. Inside of the application, they will have permission to view and change various pieces of the application set-up.

Students

The students who will use the *Project Simulator* application will be using the application as part of a project management course. They may or may not be familiar with management techniques.

Students will have a wide variety of technical backgrounds and may not have used a Windows-based application. Their first interaction with the application will occur during the project management class.

Functional Requirements

The functional requirements of the *Project Simulator* application describe the data to be maintained by the system, the actions that can be performed by both instructors and students and the internal rules used to produce the project simulation.

To simulate a complex technical project, five critical factors must be considered and defined before a project management plan can be created and executed. They include:

- **Client Description**
  - Who is the client for the project? What are the objectives for the project in terms of time, money, quality and final deliverables? What level of interaction do they require with the project management team? How often will they change requirements during project execution?
- **Project Tasks**
  - What tasks are needed to complete the project? In what order must they be completed? Can tasks be completed in parallel? What resources does each task require? What skills does each task require? How long is each task expected to take?
- **Necessary Skills**
  - What technical and interpersonal skills are needed to complete each task in the project? How are these skills measured? Can they be taught? What is the learning curve for mastering these skills?
- **Available Workers**
  - What staff are available to work on project tasks? When are they available? What technical skills do they have? Do they have vacation scheduled? Can contractors be hired? How much lead time is required to hire contractors?
- **Available Resources**
  - What physical resources are needed to complete each task? What options are there for obtaining these resources? Are there substitutes/trade-offs for using different physical resources?

Each of these major factors will have a set of characteristics (data) associated with them and a group of actions which can be performed on that data by both categories of users.

Functional requirements for each of the five major decision factors are detailed in the following sub-sections and identified by requirement type (data or action), user and priority.

Client Requirements

The client in the *Project Simulator* application is the person, group or company for which the simulated project is to be completed. The client has a number of characteristics which can be defined and modified by the instructor.

Requirement CL1

Type: *Data*
User: *Both*
Priority: *Essential*

The system should maintain the name, description and objectives of the client for the project. This information should be stored with the application and should be available to the students before beginning the project simulation.

Requirement CL2

Type: *Action*
User: *Both*
Priority: *Essential*

Instructors should have read/write permission and students should have read permission on the client description. Instructors should be able to view and modify this description before and during a project simulation. Students should be able to view this information before and during the simulation.

Requirement CL3

Type: *Data*
User: *Instructor*
Priority: *Low*

The system should maintain and use client characteristics to influence the project simulation. A number of client characteristics should be defined that relate to real-world client interactions with the project. These characteristics may include:

- Frequency of changing requirements for the project
- Level of interaction and information expected
- Upper management support for the project
- Reaction to information requests

Requirement CL4

Type: *Action*
User: *Both*
Priority: *Low*

Instructors should have read/write permission and students should have no permission on the client characteristics. Instructors should be able to view and modify these client characteristics before and during a project simulation. Students should not be able to view or modify this information before or during the simulation.

> Note: It is the instructor's discretion whether to notify the students about the possible effects of client characteristics.

Task Requirements

The tasks in the *Project Simulator* application are the specific ordered work blocks which need to be completed during the simulated project.

The entire set of tasks can be defined and modified by the instructor before the simulation begins. Each task has a number of characteristics which influence the amount of time and resources that are needed to complete it.

Requirement TA1

Type: *Data*
User: *Both*
Priority: *Essential*

The system should maintain the name, description and characteristics of each task in the project. The task characteristics should include:

- ❑ Task number for identification
- ❑ Description of the task
- ❑ Projected man hours for completion of the task
- ❑ Expected number of persons assigned to the task
- ❑ Set of skills necessary for completion of the task
- ❑ Predecessors, or tasks which must be completed before this task can begin

Requirement TA2

Type: *Action*
User: *Both*
Priority: *Essential*

Instructors should have read/write permission and students should have read permission on the tasks descriptions. Instructors should be able to view and modify this description before, but not during a project simulation. Students should be able to view this information before and during the simulation.

Requirement TA3

Type: *Action*
User: *Instructor*
Priority: *Low*

Tasks can be contracted out to vendors. Specific project tasks can be identified as "outsourced" tasks to be completed by resources outside of the workers listed in the available worker table. Instructors will be able to define criteria for the outsourced tasks and define vendor characteristics such as:

- ❑ Level of interaction and information expected
- ❑ Propensity for completing tasks on time
- ❑ Level of quality

Skills Requirements

Skills are the specific talents and abilities that are needed to complete every task of the project. These skills are defined at the beginning of the project and are associated with each project task for which they are needed. The same set of skills are measured for every available worker.

Requirement SK1

Type: *Data*
User: *Both*
Priority: *Essential*

The system should maintain the name, and description for all skills needed to complete every task in the project. This information should be stored with the application and should be available to the students before and during the project simulation.

Requirement SK2

Type: *Action*
User: *Both*
Priority: *Essential*

Instructors should have read/write permission and students should have read permission on the skills descriptions. Instructors should be able to view and modify this description before, but not during a project simulation. Students should be able to view this information before and during the simulation

Worker Requirements

Workers are the people available to be assigned to the tasks of the project. Workers can be internal employees or outside contractors. Each worker has a set of measured skills and personality characteristics that influence the rate at which they accomplish assigned tasks.

Requirement WO1

Type: *Data*
User: *Both*
Priority: *Essential*

The system should maintain the name, description, skills and a personality description for each available worker in the project simulation. This information should be stored with the application and should be available to the students before and during the project simulation.

Worker information should include:

- ❑ Worker name for identification
- ❑ Time available to work during the week
- ❑ Cost per week (salary, benefits, etc.)
- ❑ Years of experience
- ❑ Scheduled vacation weeks
- ❑ Ratings for all needed project skills (rated 0 to 10)
- ❑ Personality description detailing work habits, likes and dislikes, personnel history

Requirement WO2

Type: *Action*
User: *Both*
Priority: *Essential*

Instructors should have read/write permission and students should have read permission on the worker descriptions. Instructors should be able to view and modify this description before, but not during a project simulation. Students should be able to view this information before and during the simulation.

Requirement WO3

Type: *Data*
User: *Instructor*
Priority: *Medium*

The system should maintain and use worker personality characteristics to influence the project simulation. A number of worker characteristics should be defined that relate to real-world interpersonal behaviors. These characteristics may include:

- ❑ Overall morale
- ❑ Ability to work in teams vs. independently
- ❑ Personality conflicts with co-workers
- ❑ Rate of absenteeism
- ❑ Task preferences
- ❑ Overtime preferences
- ❑ Preferred interaction with manager

Requirement WO4

Type: *Action*
User: *Both*
Priority: *Medium*

Instructors should have read/write permission and students should have no permission on the worker personality characteristics. Instructors should be able to view and modify these worker characteristics before a project simulation. Students should not be able to view or modify this information before or during the simulation.

Resource Requirements

The resources in the *Project Simulator* application are the specific physical items which are required for the completion of project tasks. Each resource will have characteristics which reflect its performance, availability and cost.

Requirement RE1

Type: *Data*
User: *Both*
Priority: *Low*

The system should maintain the name, description and characteristics of each resource in the project. The resource characteristics should include:

- ❑ Resource name for identification
- ❑ Description of the physical resource
- ❑ Availability of the resource
- ❑ Cost for use of the resource
- ❑ Performance measurements

Requirement RE2
Type: *Action*
User: *Both*
Priority: *Low*

Instructors should have read/write permission and students should have read permission on the resource descriptions. Instructors should be able to view and modify this description before and during a project simulation. Students should be able to view this information before and during the simulation.

Instructor Configuration Requirements

Instructors will interact with the *Project Simulator* application to set up the specific project information. They will be able to alter the client descriptions, the number and sequence or tasks, the set of necessary skills, the availability and descriptions of workers, and the availability and performance of physical resources.

Requirement IC1
Type: *Action*
User: *Instructor*
Priority: *High*

Instructors should be able to alter the client descriptions. The Instructor will have access to client descriptions and client personality characteristics. Instructors are responsible for maintaining correspondence between the descriptions that the students will see and the underlying personality characteristics.

Requirement IC2
Type: *Action*
User: *Instructor*
Priority: *High*

Instructors should be able to modify the configuration of project tasks. The Instructor will have access to the task descriptions, dependencies and estimated task length. Any modifications to the task descriptions should be presented to the students before beginning the simulation. No changes to the task configuration will be allowed during the simulation, except through the internal simulation rules.

Requirement IC3
Type: *Action*
User: *Instructor*
Priority: *High*

Instructors should be able to alter the necessary project skills. The Instructor will have access to skills descriptions. They can modify, add or delete specific skills. Any skills that are required by a project task must be included in the skills database and be measured in at least one worker.

Requirement IC4
Type: *Action*
User: *Instructor*
Priority: *High*

Instructors should be able to alter the worker descriptions. The Instructor will have access to worker descriptions and can add, modify or delete workers. They can also change any worker characteristics, including skill measurements, scheduled vacations and personality factors. Care must be taken to ensure that worker descriptions match the underlying personality characteristics not seen by the students.

Requirement IC5

Type: *Action*
User: *Instructor*
Priority: *Low*

Instructors should be able to alter the physical resource descriptions. The Instructor will have access to the physical resource descriptions and can modify, add or delete resources. Care must be taken to ensure that specific resources required by tasks be available.

Student Interaction Requirements

Students will interact with the *Project Simulator* application to make a variety of project management decisions. Using the available project information, they will assign workers and resources to tasks, schedule meetings and review sessions and guide the project through to completion.

Requirement SI1

Type: *Action*
User: *Student*
Priority: *Medium*

Students should be able to select and return workers from the available worker pool Based on the availability and defined wait periods for hiring staff, students can select available workers to join the project team.

Internal employees will join the project team at the beginning of the next week. Select contractors will join the team in one or two weeks depending on the simulation set-up.

Requirement SI2

Type: *Action*
User: *Student*
Priority: *Medium*

Students should be able to schedule group meetings. Students can define the length and the day of the week for all scheduled group meetings. All workers on the project will attend the meeting, regardless of assigned tasks.

Requirement SI3
Type: *Action*
User: *Student*
Priority: *Medium*

Students should be able to implement staff development techniques. On an individual worker basis, students have the opportunity to provide educational opportunities, to meet individually with workers and to authorize extra work or vacation time.

Staff development techniques will include:

- Training an employee in a project skill
- Allowing the worker time during the week to independently study a project skill
- Counseling a worker whose productivity is not satisfactory
- Disciplining a worker
- Releasing a worker from the company
- Authorizing overtime for a worker
- Allowing extra time off for a worker
- Surveying worker morale
- Chatting with a worker

Requirement SI4
Type: *Action*
User: *Student*
Priority: *Essential*

Students should be able to assign workers to tasks for specific hours per work day. In order to complete the tasks, students must assign workers to specific tasks. Students can assign workers to more than one task per day.

Workers can also be assigned to perform enhancements on completed tasks which have quality defects. Enhancement work must also be specified in specific numbers of hours per day.

Requirement SI5
Type: *Action*
User: *Student*
Priority: *Low*

Students should be able to assign physical resources to specific tasks. In order to complete the tasks, students must assign necessary physical resources to specific tasks. Students can share resources among more than one task per day.

Requirement SI6
Type: *Action*
User: *Student*
Priority: *Low*

Students should be able to review and take actions on specific project tasks. Once a task has been completed, the student can review the task for quality defects. For vendor tasks, students can call vendor meetings and expedite completion of vendor tasks.

Requirement SI7

Type: *Action*
User: *Student*
Priority: *Essential*

Students should be able to run project reports In order to evaluate and direct progress on project tasks, students should have access to a number of detailed project management reports.

Management reports should include:

- Activity reports that monitor the progress on specific tasks
- Cost reports that detail the money spent per week on staff, training, resources, etc.
- Quality report identifying the quality defects for all completed and reviewed tasks
- Staff report detailing worker performance information
- Availability report showing the list of available workers
- Proficiency reports showing each worker's measurements for every project skill.

Requirement SI8

Type: *Action*
User: *Student*
Priority: *Essential*

Students should be able to execute the decisions for a specific time period. Once all management decisions have been made by the students, they must deliberately start the simulation. The simulation will use the decision criteria to run a time slice in the project. Once run, the simulation results cannot be reversed by the student.

Simulation Rules Requirements

The simulation will run according to a specific set of rules, influenced by the set of hidden system factors and the management decisions made by the students during the simulation.

The simulation will be run using next event simulation techniques over specific time slices for the project. Simulation criteria will be set up before the simulation begins and can be altered by the instructor during the simulation.

Requirement SR1

Type: *Action*
User: *None*
Priority: *Essential*

The system should allow only tasks that are not dependent on incomplete tasks to run. Only tasks whose predecessors have completed and are scheduled to run can be performed. Progress on tasks can only be obtained by having workers assigned to it.

Requirement SR2

Type: *Action*
User: *None*
Priority: *High*

The system should allow independent tasks to be performed in parallel. All tasks whose predecessors have completed and are scheduled to run can be performed simultaneously. The only restrictions on task performance are preceding tasks and the assignment of resources.

Requirement SR3

Type: *Action*
User: *None*
Priority: *High*

Progress on task completion is dependent on workers assigned to tasks and other environmental characteristics. The simulation will take into account the following factors in determining the completion of tasks:

- the hours per day for each worker assigned to the task
- the match of skill sets between the worker and the task for all required skills
- any personality characteristics of workers, including overall morale
- any staff development activities, such as group meetings, counseling, training, sick days, absenteeism, etc.
- any overtime authorized for a worker
- the client personality characteristics

Requirement SR4

Type: *Action*
User: *None*
Priority: *Medium*

Student management decisions will alter worker characteristics during the simulation. Depending on initial worker personality characteristics, performance of workers can be affected by counseling, discipline, chats, group meetings, work assignments, overtime and any other staff development activities.

Non-functional Requirements

This section defines the non-functional requirements of the *Project Simulator* application. Non-functional requirements include user interface design considerations and integration with existing commercial software packages.

Requirement NF1

Type: *Action*
User: *None*
Priority: *Essential*

The *Project Simulator* should be a Windows-based application. Due to the availability of IBM-compatible personal computers for project management training courses, the application should be developed for use on that platform. Windows-based applications provide greater ease-of-use because of the availability of graphics and multi-task functionality.

Requirement NF2

Type: *Action*
User: *None*
Priority: *Essential*

Wherever possible, the user interface for the *Project Simulator* should follow defined interface standards. Because of the variety of possible student user backgrounds, care must be taken to ensure that the interface provides consistent, expected access to data and functionality. The CUA Windows standards should be consulted in the user interface design and the implementation phases of the project.

Requirement NF3

Type: *Action*
User: *None*
Priority: *High*

The Project Simulator application should integrate with existing project management planning software. Part of most project management training involves the use of commercial software to develop and track project plans. The *Project Simulator* should offer a standard interface for the definition of simulation project tasks in commercial packages that offer export capabilities.

Requirements Review

A review of the Requirements Definition was held with the key decision-maker of the training organization. The client's comments and suggestions were incorporated in the design and development of the simulation application.

The client's objective for the *Project Simulator* application is to provide a realistic project management environment for students. The project simulation should exercise the following management skills:

- Planning and estimating tasks for complex technical projects
- Using commercial project management software to define and track projects
- Leveling resources using project management tools
- Assigning project tasks to existing resources
- Making staff development decisions
- Communicating with the client, the team and individual workers

If possible, the *Project Simulator* should be designed to allow for future enhancements, including:

- Sharing resources among multiple technical projects
- Managing vendor-completed tasks
- Testing and fixing quality issues

To be an effective tool in the client's project management training curriculum, the key decision-maker stated that the project should meet the following major requirements:

- A commercial project tracking application, preferably *Project Scheduler 5*, should be integrated with the simulation to emphasize the use of project tracking tools by the students.
- The *Project Simulator* application should be simply designed, well documented and built using standard development tools to allow easy maintenance after my role with the project was complete.

The decision-maker described an import/export capability available as a separate application with the *Project Scheduler 5* software. The IMEX application is run separate from the DOS-based project tracking software to import and export project tracking data from projects defined using the software.

The IMEX application can import and export data using a number of data formats, including dBase III. The decision-maker expressed hope that this capability could be used to integrate with the *Project Simulator* application.

Functional Specification

This subsection details the functional specification of the *Project Simulator* application. Using the requirements definition, this functional specification describes the overall system design, the integration of applications, the underlying data structures for the application, the flow of data through the system, the calculations used to drive the simulation and the system reports.

The functional specification is divided into the following major topics:

- System Description
- Application Integration
- Database Definitions
- System Data Flow
- Simulation Calculations
- System Reports

Project Scheduler 6 for Windows

Much of the project tasks and resource information can be maintained and displayed within a variety of commercial project management tools. The client currently uses the *Project Scheduler* application developed by Scitor Corporation to teach students the value of using project scheduling software.

Project Scheduler 6 for Windows, the newest release of the software, allows for the design and tracking of task, resource and scheduling information for a project. The application also includes an import/export utility that reads and write project information into a database file format.

Instead of duplicating this functionality, the *Project Simulator* application will integrate with *Project Scheduler 6 for Windows* where appropriate to provide a utility to display and modify project data. The integration of the *Project Simulator* and *PS6* is detailed in the subsection, Application Integration.

Project Scheduler 6 does not maintain information regarding the client and worker characteristics that drive the simulation. This data must be kept in separate database tables for use by the *Project Simulator* application.

System Description

The *Project Simulator* application can be described in terms of the primary functions performed by both types of users, the underlying data structures that hold project and simulation information, and the applications that integrate to produce the entire simulation environment.

Their are five main functions in the *Project Simulator* application that set up and execute the simulation environment. The relationships between these functions are depicted in Figure 1.

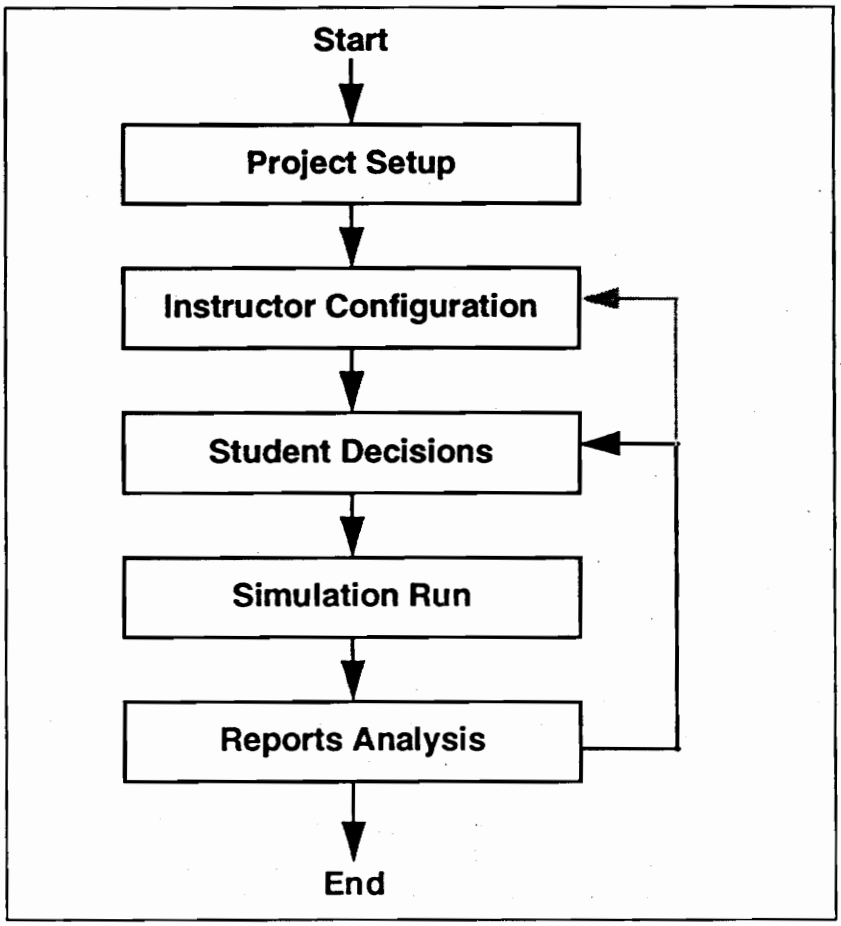


Figure 1 Primary Functions of the *Project Simulator*

First, the instructor sets up the project environment with specific task information and resource descriptions. The instructor then adds client and worker characteristics to supply realistic project management variables.

The simulation cycle begins with the student decisions function where students make all management decisions concerning project assignments and staff development. The simulation is run for a time slice and actual data is recorded for the project.

Students can analyze various project reports. Before the next simulation time slice, instructors can change client or worker characteristics and students can make additional assignment and staff development decisions. The simulation continues in this loop until the project is completed.

Project Setup

The Project Setup function is where the details of the project are input by the instructor. This information includes the following:

- specific project tasks (dependencies between tasks, estimated duration, estimated resources, start dates, finish dates, etc.)
- specific worker descriptions (names, availability, cost)
- project calendar (standard work hours, holidays)

The instructor will input the project setup information using *Project Scheduler 6 for Windows*. This application is designed specifically for defining projects and their associated tasks. A complete list of project setup information is found in the Database Definitions subsection.

Instructor Configuration

The Instructor Configuration function is where the specific characteristics of the people and environmental factors involved with the project are input by the instructor.

The people who will influence the progress of the project include the client, and any workers that may be assigned to tasks in the project. Environmental factors include any natural or unnatural phenomenon that can influence the completion of the project.

The instructor will input the client, worker and environment characteristics using the *Project Simulator* front-end application. The *Project Simulator* should provide an effective interface to allow the instructor to set all necessary project characteristics.

The specific client characteristics, worker characteristics and environmental characteristics are described in the Database Definitions subsection.

Student Decisions

The Student Decisions function is where all the project management and staff development decisions are input by the student.

The project management decisions include leveling resources, adjusting task durations and start times and assigning workers to tasks. Staff development decisions include counseling, praising, training and meeting with their workers.

Students will input their project management decisions using *Project Scheduler 6 for Windows*. This application provides all the necessary project and task management capabilities.

Students will input the staff development information using the *Project Simulator* application. The *Project Simulator* should provide an easy-to-use interface that allows students to schedule and monitor staff development activities.

The specific decision data input by the student is described in the System Data Flow subsection. The interface that allows the student to input decisions is described in the User Interface Design subsection.

Simulation Run

The Simulation Run function is where the *Project Simulator* application runs the project simulation for a specific amount of time. This function is started by the student and runs independently until all calculations are complete.

The results of the simulation run are determined by the specific project information input by both users and by the simulation rules built into the application. The simulator will run for a duration of one week.

Using the project information, the simulator will determine the completion of assigned tasks and the effect of staff development activities. The results of the information will be written to the underlying data structures and presented in summary to the student.

The simulation run is called from and executed inside of the *Project Simulator* application. The *Project Simulator* application should provide a clear mechanism to allow the student to run the next simulation time slice.

A complete description of the information used to drive the simulation run and where the results are written can be found in the System Data Flow subsection. The rules used to govern the simulation run are described in the Simulation Rules subsection.

Reports Analysis

The Reports Analysis function is where the student views and prints project management reports. The student performs reports analysis after a simulation run to prepare the management and staff development decisions for the next simulation week.

The reports describe various aspects of the project from task completion to budget graphs to worker skills. The reports are calculated from the project information maintained in the underlying data structures.

Students will view and print project management reports within *Project Scheduler 6 for Windows*. This application provides all of the project tracking reports and graphs needed to evaluate project progress.

Students will view and print staff development reports inside of the *Project Simulator* application. *Project Simulator* should allow a clear, flexible interface to allow students to view and print reports on staff development activities.

A full description of reports available within *Project Scheduler 6* and the *Project Simulator* are provided in the System Reports subsection.

Application Integration

The successful integration of the *Project Scheduler 6 for Windows* and the *Project Simulator* application is key to developing an effective project management simulation tool. The integration should be as transparent as possible to both types of users

Project Scheduler 6 for Windows

Prior to the simulation exercise, both students and instructors will be familiar with *Project Scheduler 6* or similar project management software. To remain consistent with their experiences, the use of the *Project Scheduler 6* application in the simulation exercise will be identical to the way that is familiar to them.

Instructors will use *Project Scheduler 6* to define the project environment Students will use the commercial application to plan and record project activities. No functional integration of simulation concepts will be done within *Project Scheduler 6*, i.e., no additional menu called "Simulation" will appear in the main menu for students to access simulation functionality.

Project Simulator

The *Project Simulator* application will be run as an independent application separate from *Project Scheduler 6*. Whereas *Project Scheduler 6* will be used as the familiar project tracking software, the *Project Simulator* will represent the more real-world aspects of managing a project, such as staff development and environmental factors.

Instructors and students will interact with the *Project Simulator* to perform all project management techniques aside from tracking the project with software. The *Simulator* will provide an environment to interact with the people and environment in the project.

Since both users are familiar with the project management software, the terminology and naming conventions used within *Project Scheduler 6* will be incorporated in the *Project Simulator* application.

For example, *PS6* uses the term "job" instead of "task" when describing an activity in the project. The *Project Simulator* application will incorporate this term for consistency.

Windows 3.x Environment

Since *Project Scheduler 6* and *Project Simulator* are run as separate applications, each will need to be started inside the Windows 3.x desktop environment.

The instructor or student should start the *Project Simulator* application. To ensure smoother integration, *Project Scheduler 6* should be started by the *Project Simulator* application so that the proper project files are opened automatically.

Once both applications are open, they can be toggled between using a number of Windows switch application techniques - minimizing application windows, pressing the "ALT-TAB" keys, etc.

Underlying Data Structures

Both *Project Scheduler 6* and the *Project Simulator* applications will maintain and share underlying data structures that hold project information. The manner in which the applications access and share information should be as transparent to the user as possible.

Project Scheduler 6 for Windows maintains all project tracking information in internal data structures. These data structures and the information that they contain can be exported from *PS6* to a dBase III-formatted database structure using a separate application called IMEX.

This exported database can then be accessed and written to by the *Project Simulator* application. Once changes have been written to the exported database, the database can be imported back into *Project Scheduler 6* using the IMEX application.

The *Project Simulator* application will also maintain a database that holds client, worker and environmental characteristics for the project. This database cannot be accessed within *Project Scheduler 6*. This project information will be used along with the exported/imported database to drive the simulation.

A full description of the flow of project data within the system is provided in the System Data Flow subsection.

Database Definitions

The databases maintained by both applications must contain all the project information needed to run the simulation. This section describes the structure and contents of these databases, as well as which applications and users access them.

There are three databases used to produce the project simulation environment:

- Project Tracking
- Project Characteristics
- Simulation Results

Project Tracking Database

Project Tracking database contains the data exported from *Project Scheduler 6*, accessed and written to by the *Project Simulator* and imported back into *Project Scheduler 6*. This database contains all necessary information to track and display project progress.

The information within these databases are setup by the instructor using *Project Scheduler 6* during the Project Setup function. Students also use *PS6* to modify the information during the Student Decisions function and access the data in the Reports Analysis function. The databases are accessed and written to during the Simulation Run function of the *Project Simulator*.

The Project Tracking database is separated into six tables of project management information. The tables include:

- Project Information
- Job Information
- Resource Information
- Link Information
- Assignment Information
- Calendar Information

The fields contained within each of these tables are described below, along with databases field names, data type and width.

Project Information Table

Field Name	Description	Data Type	Width
Project Name	Name of the simulated project	String	20
Project File Name	Name of the PS6 project file (".JBX")	String	30
Table Name	Name of the PS6 table file (".TBX")	String	30
Project Manager	Name of the project manager	String	15
Project Start Date	Start date for the project	Date	8
Project Start Time	Start time for the project	Time	4
Project End date	End date for the project	Date	8
Project End Time	End time for the project	Time	4
Schedule Direction	Schedule orientation (forward or reverse)	String	7
Priority	Resource leveling priority (for use with multiple project files)	Integer	1
Revision Number	Revision number of the project	Integer	5
Revision Date	Revision date for the project	Date	8
Revision Time	Revision time for the project	Time	4
PERT	Whether PERT mode is on or off	String	3

Table 1 Project Information Table

Job Information Table

Field Name	Description	Data Type	Width
Job Number	Job number for task	Integer	4
WBS Code	Work Breakdown Structure Code	String	10
OBS Code	Organizational Breakdown Structure	String	10
Job Name	Name of job	String	30
Job Notes	Any job notes	String	80
Status	Job status (Critical, Complete, Milestone, Non Critical)	String	11
Percent Complete	Percent complete	Integer	3
Term	Job termination (ASAP, ALAP, HAMM)	String	4
Priority	Job priority	Integer	1
Network Row	Row in network diagram	Integer	3
Network Column	Column in network diagram	Integer	3
Job Duration	Job duration	Integer	5
Job Scale	Scale for job duration	Time Scale	1
Most Likely Duration	Most likely duration	Integer	5
Most Likely Scale	Scale for most likely duration	Time Scale	1
Pessimistic Duration	Pessimistic duration	Integer	5
Pessimistic Scale	Scale for pessimistic duration	Time Scale	1
Completed Duration	Completed duration	Integer	5
Completed Scale	Scale for completed duration	Time Scale	1
Remaining Duration	Remaining duration	Integer	5

Table 2 Job Information Table (1 of 3)

Remaining Scale	Scale for remaining duration	Time Scale	1
Delay Duration	Delay duration	Integer	5
Delay Scale	Scale for delay duration	Time Scale	1
Free Float Duration	Free float duration	Integer	5
Free Float Scale	Scale for free float duration	Time Scale	1
Total Float Duration	Total Float duration	Integer	5
Total Float Scale	Scale for total float duration	Time Scale	1
Conflict Duration	Conflict duration	Integer	5
Conflict Scale	Scale for conflict duration	Time Scale	1
Date Dep. Date	Date Dependency date	Date	8
Date Dep. Time	Date Dependency time	Time	4
Req. Date Date	Required Date date	Date	8
Req. Date Time	Required Date time	Time	4
Actual Start Date	Actual Start date	Date	8
Actual Start Time	Actual Start time	Time	4
Actual Finish Date	Actual Finish date	Date	8
Actual Finish Time	Actual Finish time	Time	4
Base Start Date	Base Start date	Date	8
Base Start Time	Base Start time	Time	4
Base Finish Date	Base Finish date	Date	8
Base Finish Time	Base Finish time	Time	4
Early Start Date	Early Start date	Date	8
Early Start Time	Early Start time	Time	4
Early Finish Date	Early Finish date	Date	8

Table 2 Job Information Table (2 of 3)

Early Finish Time	Early Finish time	Time	4
Late Start Date	Late Start date	Date	8
Late Start Time	Late Start time	Time	4
Late Finish Date	Late Finish date	Date	8
Late Finish Time	Late Finish time	Time	4
Sched. Start Date	Scheduled Start date	Date	8
Sched. Start Time	Scheduled Start time	Time	4
Sched. Finish Date	Scheduled Finish date	Date	8
Sched. Finish Time	Scheduled Finish time	Time	4
Labor Units	Labor units	Real	15
Labor Costs	Labor costs	Real	15
Other Units	Other units	Real	15
Other Costs	Other costs	Real	15
Material Units	Material units	Real	15
Material Cost	Material cost	Real	15
Actual Units	Actual units	Real	15
Actual Costs	Actual costs	Real	15
Total Costs	Total costs	Real	15
Completed Costs	Completed costs	Real	15

Table 2 Job Information Table (3 of 3)

Resource Information Table

Field Name	Description	Data Type	Width
Resource Number	Resource identifier	Integer	3
RBS Code	Resource Breakdown Structure code	String	5
Resource Name	Name of resource	String	15
Unit Cost	Unit cost for resource	Real	15
Unit Cost Scale	Scale for unit cost	Time Scale	1
Availability	Resource availability	Real	15
Availability Scale	Scale for resource availability	Time Scale	1
Resource Type	Resource type (L, M or O for Labor, Material, Other)	String	1
Inflation Schedule	Inflation schedule (A, B, C, D or E)	String	1

Table 3 Resource Information Table

Link Information Table

Field Name	Description	Data Type	Width
Pred. Job Number	Predecessor job number	Integer	4
Pred. Job Name	Predecessor job name	String	30
Succ. Job Number	Successor job number	Integer	4
Succ. Job Name	Successor job name	String	30
Lead/Lag Duration	Lead/Lag duration	Integer	5
Lead/Lag Scale	Lead/Lag scale	Time Scale	1
Link Type	Link type (FS, FF or SS for Finish-Start, Finish-Finish or Start-Start)	String	2

Table 4 Link Information Table

Assignment Information Table

Field Name	Description	Data Type	Width
Job Number	Job number for task	Integer	4
WBS Code	Work Breakdown Structure Code	String	10
OBS Code	Organizational Breakdown Structure	String	10
Job Name	Name of job	String	30
Resource Number	Resource identifier	Integer	3
RBS Code	Resource Breakdown Structure code	String	5
Resource Name	Name of resource	String	15
Occurrence Number	Occurrence number	Integer	3
Start Value	Start value	Integer	5
Start Scale	Start scale	Time Scale	1
Duration	Assignment duration	Integer	5
Duration Scale	Scale for assignment duration	Time Scale	1
Units Per - Value	Units per value	Real	15
Units Per - Scale	Units per scale	Time Scale	1
Total Units	Total units	Real	15
Total Costs	Total costs	Real	15
Actual Units	Actual units	Real	15
Actual Cost	Actual cost	Real	15

Table 5 Assignment Information Table

Calendar Information Table

Field Name	Description	Data Type	Width
Date	Calendar date	Date	8
Kind	Date kind	Date Kind	12
Workdays	Whether the day is a workday or not (represented by 0 or 1)	Integer	1
Workhours	Whether the hour is a workhour or not (represented by 0 or 1)	Integer	1

Table 6 Calendar Information Table

Project Characteristics Database

The Project Characteristics database contains the client, worker and environmental characteristics that influence the execution of the simulation.

The information within these databases are set up by the instructor in the *Project Simulator* application during the Instructor Configuration function. Subsets of this data can be accessed and viewed by students during the Student Decisions function.

The data structures are accessed and written to during the Simulation Run function. They are also accessed and viewed by students during the Reports Analysis function.

The Project Characteristics database is separated into the following five tables containing people and environmental characteristics:

- Client Characteristics
- Worker Characteristics
- Job Characteristics
- Worker Skills
- Skills Definition

The fields contained within each of these tables are described below, along with databases field names, data type and width.

Client Characteristics Table

Field Name	Description	Data Type	Width
Client Name	Name of the client for the project	String	30
Location	Client location	String	20
HL Support	High-level management support for project (0-9)	Integer	1
IS Support	Level of Information Systems management support (0-9)	Integer	1
Finances	Willingness to expend financial resources for the project (0-9)	Integer	1
Change	Willingness to embrace change (0-9)	Integer	1
Consistency	Frequency of changing project requirements (0-9)	Integer	1
Interaction	Frequency of project team interaction expected (0-9)	Integer	1
Information Delay	Average delay in responding to information requests (0-9)	Integer	1
Other 1	Extra client characteristic that can be set by the instructor (0-9)	Integer	1
Other 2	Extra client characteristic that can be set by the instructor (0-9)	Integer	1
Other 3	Extra client characteristic that can be set by the instructor (0-9)	Integer	1

Table 7 Client Characteristics Table

Worker Characteristics Table

Field Name	Description	Data Type	Width
Resource Number	Resource number	Integer	5
Resource Name	Name of worker	String	15
Morale	Overall worker morale (0-9)	Integer	1
Teamwork	Ability to work well in teams (0-9)	Integer	1
Popularity	How well the worker is liked by fellow employees (0-9)	Integer	1
Respect	How well the worker is respected by fellow employees (0-9)	Integer	1
Absenteeism	How often the worker is absent from work (0-9)	Integer	1
Overtime	Worker's overtime preference (0-9)	Integer	1
Aptitude	Ability to learn new job skills (0-9)	Integer	1
Other 1	Extra worker characteristic that can be set by the instructor (0-9)	Integer	1
Other 2	Extra worker characteristic that can be set by the instructor (0-9)	Integer	1
Other 3	Extra worker characteristic that can be set by the instructor (0-9)	Integer	1

Table 8 Worker Characteristics Table

Job Characteristics Table

Field Name	Description	Data Type	Width
Job Number	Job number	Integer	5
Job Name	Name of job	String	15
Skill 1 Needed	First worker skill needed	String	3
Skill 1 Importance	Importance of skill to job (1-3)	Integer	1
Skill 2 Needed	Second worker skill needed	String	3
Skill 2 Importance	Importance of skill to job (1-3)	Integer	1
Skill 3 Needed	Third worker skill needed	String	3
Skill 3 Importance	Importance of skill to job (1-3)	Integer	1
Skill 4 Needed	Fourth worker skill needed	String	3
Skill 4 Importance	Importance of skill to job (1-3)	Integer	1
Skill 5 Needed	Fifth worker skill needed	String	3
Skill 5 Importance	Importance of skill to job (1-3)	Integer	1
Other 1	Extra job characteristic that can be set by the instructor (0-9)	Integer	1
Other 2	Extra job characteristic that can be set by the instructor (0-9)	Integer	1
Other 3	Extra job characteristic that can be set by the instructor (0-9)	Integer	1

Table 9 Job Characteristics Table

Worker Skills Table

Field Name	Description	Data Type	Width
Resource Number	Worker number	Integer	5
Skill 1 ID	Skill identifier	String	3
Worker Rating 1	Worker skill rating (0-9)	Integer	1
Worker Pref 1	Worker's preference to use skill (0-9)	Integer	1
Skill 2 ID	Skill identifier	String	3
Worker Rating 2	Worker skill rating (0-9)	Integer	1
Worker Pref 2	Worker's preference to use skill (0-9)	Integer	1
Skill 3 ID	Skill identifier	String	3
Worker Rating 3	Worker skill rating (0-9)	Integer	1
Worker Pref 3	Worker's preference to use skill (0-9)	Integer	1
Skill 4 ID	Skill identifier	String	3
Worker Rating 4	Worker skill rating (0-9)	Integer	1
Worker Pref 4	Worker's preference to use skill (0-9)	Integer	1
Skill 5 ID	Skill identifier	String	3
Worker Rating 5	Worker skill rating (0-9)	Integer	1
Worker Pref 5	Worker's preference to use skill (0-9)	Integer	1
Skill 6 ID	Skill identifier	String	3
Worker Rating 6	Worker skill rating (0-9)	Integer	1
Worker Pref 6	Worker's preference to use skill (0-9)	Integer	1
Skill 7 ID	Skill identifier	String	3
Worker Rating 7	Worker skill rating (0-9)	Integer	1
Worker Pref 7	Worker's preference to use skill (0-9)	Integer	1

Table 10 Worker Skills Table

Skill Definition Table

Field Name	Description	Data Type	Width
Skill ID	Skill identifier	String	3
Skill Name	Name of the worker skill	String	20
Skill Description	Description of the worker skill	String	80
Learning Curve	Learning curve associated with learning that skill (0-9)	Integer	1

Table 11 Skill Definition Table

Simulation Results Database

The Simulation Results database maintains project history information as the simulation exercise is in progress. The data within these structures represent the original project baseline plan and actual project activity measurements recorded during the simulation runs.

These data structures are accessed and written to during the Simulation Run function. The data is also accessed by the students during the Student Decisions and Reports Analysis functions.

The Simulation Results database is separated into the following three tables containing student decision and job progress information:

- Worker Training
- Worker Meetings
- Team Meetings
- Client Meetings

The fields contained within each of these tables are described below, along with databases field names, data type and width.

Worker Training Table

Field Name	Description	Data Type	Width
Week Number	Week number	Integer	3
Start Date	Week start date	Date	8
Resource Number	Worker identifier	String	5
Skill Trained	Skill trained to worker	String	3
Training Hours	Hours spent training worker	Integer	2
Rating Improvement	Improvement in skill rating (0-8)	Integer	1

Table 12 Worker Training Table

Worker Meetings Table

Field Name	Description	Data Type	Width
Week Number	Week number	Integer	3
Start Date	Week start date	Date	8
Resource Number	Worker identifier	String	5
Purpose	Purpose for meeting (CHat, PRaise, COunsel or DIscipline)	String	2
Meeting Hours	Hours spent meeting with worker	Integer	2
Morale Impr	Improvement in morale rating (0-8)	Integer	1
Teamwork Impr	Improvement in teamwork (0-8)	Integer	1
Absenteeism Impr	Improvement in absenteeism (0-8)	Integer	1
Overtime Impr	Improvement in overtime pref (0-8)	Integer	1
Other 1 Impr	Improvement in other 1 rating (0-8)	Integer	1
Other 2 Impr	Improvement in other 2 rating (0-8)	Integer	1
Other 3 Impr	Improvement in other 3 rating (0-8)	Integer	1

Table 13 Worker Meetings Table

Team Meetings Table

Field Name	Description	Data Type	Width
Week Number	Week number	Integer	3
Start Date	Week start date	Date	8
Purpose	Purpose for meeting (CHat, PRaise, COunsel or DIscipline)	String	2
Meeting Hours	Hours spent meeting with worker	Integer	2
Morale Impr	Improvement in morale rating (0-8)	Integer	1
Teamwork Impr	Improvement in teamwork (0-8)	Integer	1
Absenteeism Impr	Improvement in absenteeism (0-8)	Integer	1
Overtime Impr	Improvement in overtime pref (0-8)	Integer	1
Other 1 Impr	Improvement in other 1 rating (0-8)	Integer	1
Other 2 Impr	Improvement in other 2 rating (0-8)	Integer	1
Other 3 Impr	Improvement in other 3 rating (0-8)	Integer	1

Table 14 Team Meetings Table

Client Meetings Table

Field Name	Description	Data Type	Width
Week Number	Week number	Integer	3
Start Date	Week start date	Date	8
Purpose	Purpose for meeting (CHat, PRaise, COunsel or DIscipline)	String	2
Meeting Hours	Hours spent meeting with worker	Integer	2
HL Support	Improvement in HL support (0-8)	Integer	1
IS Support	Improvement in IS support (0-8)	Integer	1
Finances	Improvement in finances (0-8)	Integer	1
Change	Improvement in change (0-8)	Integer	1
Consistency	Improvement in consistency (0-8)	Integer	1
Information Delay	Improvement in info delay (0-8)	Integer	1
Other 1 Impr	Improvement in other 1 rating (0-8)	Integer	1
Other 2 Impr	Improvement in other 2 rating (0-8)	Integer	1
Other 3 Impr	Improvement in other 3 rating (0-8)	Integer	1

Table 15 Client Meetings Table

System Data Flow

The project management simulation is driven by the data maintained in the system databases. This section describes the overall flow of data through the project management system during the five functions of the project simulation.

The Project Tracking and Project Characteristics databases are used throughout the system functions. The Simulation Results database is only used by the Simulation Run and Reports Analysis functions.

A depiction of the overall flow of data within the system showing the application, user type, system functions and databases is presented in Figure 5.2. The actual database tables accessed and written to by both sets of users during each simulation function are described in the subsections that follow.

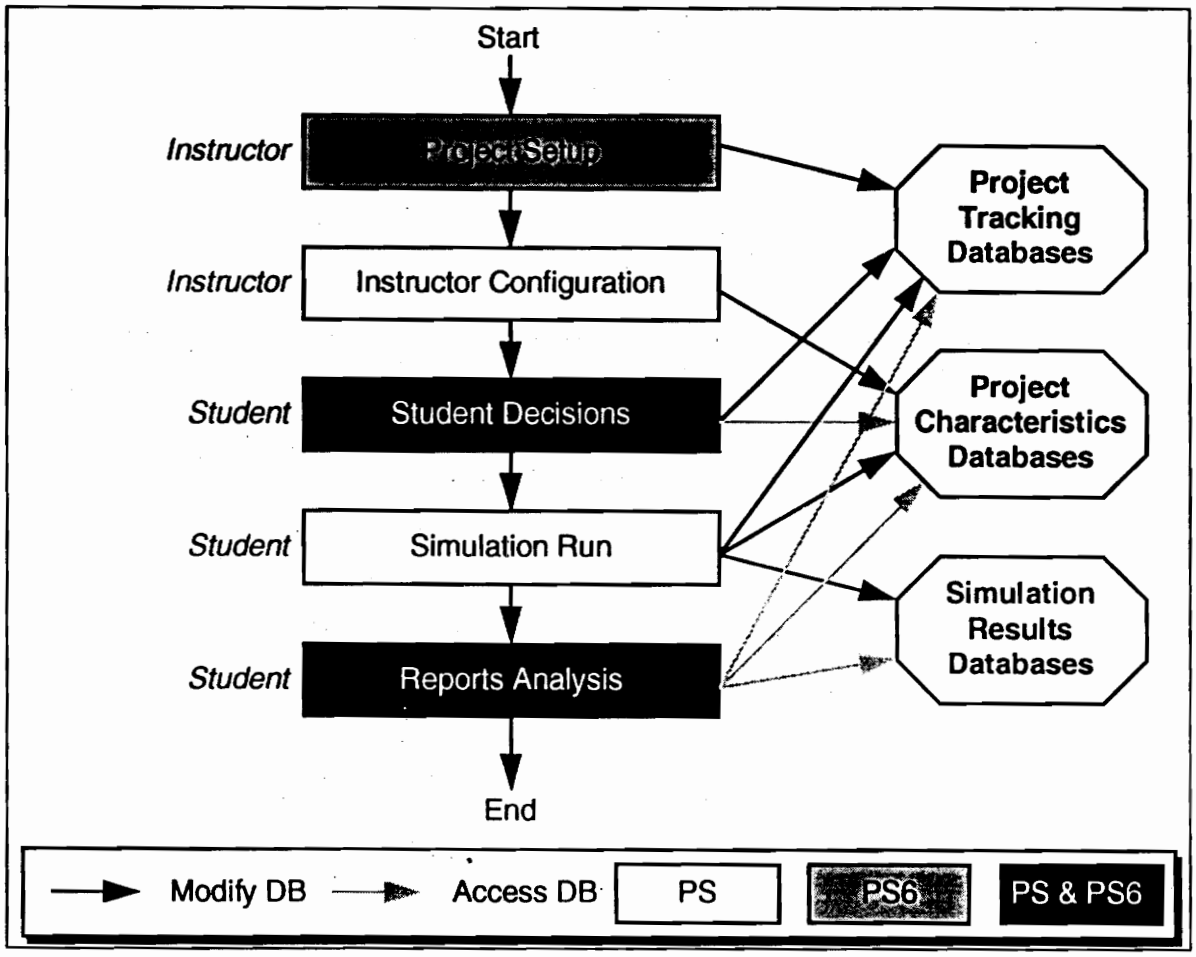


Figure 2 Overall System Data Flow

Project Setup

The Project Setup function is where the details of the project are input by the instructor using the *PS6* application. The instructor **modifies** the following databases and tables:

- ❖ Project Tracking
  - Project Information
  - Job Information
  - Resource Information
  - Link Information
  - Assignment Information
  - Calendar Information

Instructor Configuration

The Instructor Configuration function is where the specific characteristics of the people involved with the project are input by the instructor. The instructor **modifies** the following databases and tables:

- ❖ Project Characteristics
  - Skills Definition
  - Client Characteristics
  - Worker Characteristics
  - Job Characteristics
  - Worker Skills

Student Decisions

The Student Decisions function is where all the project management and staff development decisions are input by the student. The student **accesses** the following databases and tables:

- ❖ Project Characteristics
  - Skills Definition
  - Job Characteristics
  - Worker Skills

The student **modifies** the following databases and tables:

- ❖ Project Tracking
  - Project Information
  - Job Information
  - Resource Information
  - Link Information
  - Assignment Information
  - Calendar Information
- ❖ Simulation Results
  - Worker Training
  - Worker Meetings
  - Team Meetings

- Client Meetings

Simulation Run

The Simulation Run function is where the *Project Simulator* application runs the project simulation for a specific amount of time. This function **modifies** the following databases and tables:

- ❖ Project Tracking
  - Project Information
  - Job Information
  - Resource Information
  - Link Information
  - Assignment Information
  - Calendar Information
- ❖ Project Characteristics
  - Skills Definition
  - Client Characteristics
  - Worker Characteristics
  - Job Characteristics
  - Worker Skills
- ❖ Simulation Results
  - Worker Training
  - Worker Meetings
  - Team Meetings
  - Client Meetings

Reports Analysis

The Reports Analysis function is where the student views and prints project management reports. The student accesses the following databases and tables:

- ❖ Project Tracking
  - Project Information
  - Job Information
  - Resource Information
  - Link Information
  - Assignment Information
  - Calendar Information
- ❖ Project Characteristics
  - Skills Definition
  - Client Characteristics
  - Worker Characteristics
  - Job Characteristics
  - Worker Skills
- ❖ Simulation Results

- Worker Training
- Worker Meetings
- Team Meetings
- Client Meetings

Simulation Rules

The project management simulation is driven by both the data maintained in the system databases and the simulation rules built into the application. This section describes the rules and the individual data fields used to run the simulation environment.

The project management simulation is run one week at a time. The results of the simulation are based on the job assignments and staff development decisions made by the student as well as the underlying project characteristics set by the instructor.

Project Baseline

After the Project Setup and Instructor Configuration functions are completed by the instructor, the student begins the first iteration of the simulation cycle. Once the student levels resources and makes their initial job assignments, the project information is saved. This initial project schedule is called the **project baseline.**

The project baseline is used as the basis for determining the time needed to complete specific jobs. It also is used at the end of the simulation to determine how well the student has managed the completion of the project.

Simulation Run

When the student selects to run the simulation for the next week, the *Project Simulator* application accesses all three system databases, performs the simulation calculations and writes the results of the simulation run back to the system databases.

For each simulation run, the *Project Simulator* application runs through the following steps:

1. **Determines the week's activities**
2. **Schedules the week's events**
3. **Calculates the remaining time for all scheduled jobs**
4. **Simulates the week's activities**
5. **Reports results of the simulation run**

Step 1
Determine Activities

The first step of the simulation run is to determine all activities that are scheduled for the week and that may occur during the week. The activities are determined by :

- Reading specific job information from the Project Tracking database
- Reading scheduled staff development activities from the simulation results database
- Calculating unscheduled activities using data from the Project Characteristics database

The Project Tracking database contains all information concerning the assignment of jobs to resources. All information for jobs that are scheduled for the next two weeks are read into memory. The extra week is read in case the scheduled tasks are completed early.

The Simulation Results database contains information regarding any scheduled staff development activities, including team and individual meetings, client meetings and worker training.

Any unplanned activities are determined by probability calculations using the information stored in the Project Characteristics database. The type and amount of unplanned activities are determined by the instructor during the Instructor Configuration function.

Step 2
Schedule Activities

Once all the activities have been determined, the *Project Simulator* application schedules the start of each staff development and unplanned activity for a specific hour of the week. A record is written for each of these activities and is kept in an activity queue ordered by time.

Step 3
Calculate Remaining Job Times

The total and remaining amount of time needed to complete each job is held in the Project Tracking database. The *Project Simulator* application uses this time plus information from the Project Characteristics database to recalculate the remaining time.

The expected time to complete the task that is saved in the project baseline is used as the basis for calculating actual job times. All "rated" project characteristics influence this calculation. A rated characteristic is any client, worker or job characteristic with a value greater than 0.

The simulator uses the following two rules as a basis for calculating actual job times:

The time needed to complete a job is equal to the number of hours specified in the project baseline if no project characteristics are rated.

The time needed to complete a job is equal to the number of hours specified in the project baseline if all rated project characteristics are equal to 5.

These base rules mean that the time needed to complete a job is dependent on all project characteristics stored in the system databases.

Each project characteristic is used in the calculation and has equal impact, except for the job skill importance. The skill ratings for each worker for each skill required in a task is factored in as many times as indicated in the importance field.

For example, if a particular skill is rated "3" in importance for a job, each skill rating for each worker assigned to that task is included 3 times in the actual job time calculations.

The following calculation is used at the beginning of the simulation run to calculate the remaining time for each jobs:

$$\text{Remaining Job Time} = \text{Baseline Rem Job Time} * \frac{1}{5} * \frac{\text{Sum of Rated Factors}}{\text{Rated Factors}}$$

The remaining times for each job scheduled in the next two weeks is saved in memory and used during the actual event simulation.

Step 4
Run Next Event Simulation

The *Project Simulator* application now begins to simulate the project hour by hour. A system clock is used to drive the simulation. The *Simulator* uses the following rules to drive the simulation.

At the beginning of each hour, the *Simulator* checks the activity queue for a scheduled activity. If an activity is found (meeting, training, etc.), the appropriate workers are assigned that activity. If no activity is found, the workers are scheduled for their assigned job tasks.

The system clock is advanced one hour and the results of that hour are recorded. Job and activity times are decremented. If a scheduled activity is completed then the results of that staff development activity is determined.

If a job is completed, the time of completion is saved and the next scheduled tasks are used as the default assignments for workers. The system continues the simulation hour by hour following these rules until the system clock reaches the end of the simulated week.

Step 5
Report Simulation Results

After the *Project Simulator* has completed the simulation of the week's activities, the results of the simulation are written back to the project databases and any unusual activities are reported to the students through the *Project Simulator* application.

The *Project Simulator* application recalculates the remaining time for jobs each time the Simulation Run function is called by the student. Once run, the simulation cannot be reversed by the student.

System Reports

Project tracking reports are used by the students to determine the progress of the project and to plan future assignment and staff development decisions.

System reports consist of standard project tracking charts and reports available through the *Project Scheduler 6* software and reports created by the *Project Simulator* application.

Project Scheduler 6 for Windows

The *Project Scheduler 6* commercial software provides a myriad of project tracking reports. These reports reflect both planned and actual progress and can be easily accessed within the *Project Scheduler 6* application.

Project Tracking reports provided include:

- ❖ Gantt Chart report
- ❖ Network Diagram Report
- ❖ Tree Structure Report
- ❖ Histogram Report
- ❖ Cost Graph Report
- ❖ Project Description Report
- ❖ Job Description Report
- ❖ Job Column Report
- ❖ Job Distribution Report
- ❖ Resource Distribution Report
- ❖ Resource Allocation Report

- ❖ Resource Table Report
- ❖ Calendar Report
- ❖ Variance Report

Examples of these system reports are found in Figures C.1 through C.4 presented in **Appendix C, Project Scheduler 6 Interface.**

Project Simulator Application

The *Project Simulator* application provides reports on the project characteristics and Simulation Results data. Students are prohibited from seeing certain project characteristics such as client and worker personalities. Descriptions of these characteristics may be provided by the instructor in non-numerical form.

Reports provided by the *Project Simulator* include:

- ❖ Skills Definition Report
- ❖ Worker Skills Report
- ❖ Worker Performance Report
- ❖ Employee Survey Report

User Interface Design

A consistent, understandable user interface is critical to developing an effective educational software application. The student using the tool should not be distracted by the application interface or be forced to learn a new environment in order to benefit from the exercises.

As a Windows-based application, the *Project Simulator* should follow the basic Windows CUA standards for interface development, where applicable. These standards have been developed by reviewing user preferences and productivity in Windows applications.

There are few project management simulation software applications similar to this proposed *Project Simulator* from which to compare interface designs. In my research, I could find several DOS, but no Windows-based simulation applications.

Proper user interface design involves presenting the functionality and the information in a familiar and accessible manner to the user.

Interface Design Decisions

A significant user interface design decision is where to give the user access to the functionality they request. In early Windows applications, nearly all user functionality was accessible only through the program menus.

These menus and their associated commands were direct descendants of the layered-menu styles found in DOS character-based applications.

Menus vs. Icons

As Windows programmers began to exploit the graphical capabilities that the environment provided, more intuitive buttons and pictures were used to allow the user to navigate through the system.

An extension of these graphics is the icon-based desktop environments where familiar graphics are used to allow users to control their system environment by minimizing and maximizing numerous windows.

The *Project Simulator* interface will be designed to give both menu and icon functionality. All user tasks can be accomplished through the main menu structure, but separate components of the system can be maintained on the desktop in windows that can be minimized and managed.

MDI Applications

Multiple Document Interface (MDI) applications are Windows-based applications that have a main parent window. This parent window creates and manages several child windows which always remain inside of the parent's borders.

MDI interfaces allow for icon-driven desktop environments inside of an application. MDI parent windows will be used for the *Project Simulator* application interfaces.

Student vs. Instructor Functionality

Students and instructors will use the *Project Simulator* for distinctly different functionality. Instructors need access to the entire system database structure while students are restricted in their access to information.

Instructors will use the application primarily to set up the simulation environment and monitor the student's progress through the simulation.

The student will use the application to view reports, to make weekly project management decisions and to run and view results of the simulation.

Due to their distinctly different needs, the *Project Simulator* application will be separated into two Windows applications. The application interface built for the instructor will be called the *PS Admin Workstation*. The application interface used by the students will be called the *Project Simulator*.

PS Admin Workstation

The *PS Admin Workstation* will be accessible through the Windows desktop environment by double-clicking a program item icon or by selecting **Run** from the Program Manager **File** menu.

The instructor will use the *PS Admin Workstation* to:

- Set up the client characteristics
- Define the skills needed for each job in the project
- Define the worker personality characteristics
- Rate each worker in each project skill
- Monitor student's progress during the simulation

All user interface screen captures for the *PS Admin Workstation* interface will be found in **Appendix A, PS Admin Workstation Prototype.**

Menu Structure

The main menu for the parent *PS Admin Workstation* window will have six menu pads. Screen captures that show the menu options for each of these main menu pads are presented in Figures A.1 through A.6 in **Appendix A.**

Each of the main menu pads and their associated commands are described below.

File Menu

The **File** menu contains the standard Windows application file commands that allow the user to manipulate the files associated with the application and to quit the application via the Exit command.

Command	*Description*
New	Open a new simulation
Open...	Open an existing simulation
Close	Close the current simulation
Save	Save changes to current simulation
Save As...	Save simulation to new file name

Print...	Print selected information
Exit	Exit the PS Admin Workstation

Edit Menu

The Edit menu contains the standard Windows application edit commands that allow the user to cut, copy, paste and delete text from the application.

Command	*Description*
Cut	Cut selected text
Copy	Copy the selected text
Paste	Paste the selected text
Delete	Delete the selected text

Configure Menu

The Configure menu allows the user to define the skills for each job in the simulation and to set the characteristics for the client and the workers..

Command	*Description*
Project Skills...	Define skills needed for project
Job Skills...	Define skills needed for each job
Worker Skills...	Define skill ratings for each worker
Client...	Define client characteristics
Workers...	Define worker characteristics, including rated job skills
Resources...	Define physical resource characteristics (not prototyped)

View Menu

The View menu allows the instructor to monitor the progress of the student's simulation exercise. This functionality depends on the Local Area Network (LAN) capabilities of the computing facilities. This functionality is not provided in the prototype.

Command	*Description*
Class...	View class progress

Student...	View individual student progress
Statistics...	View simulation statistics

Window Menu

The **Window** menu is a standards Windows application menu that allows the user to arrange the windows on the desktop and to provide access to individual windows by listing them in the menu.

Command	*Description*
Cascade	Cascade open windows
Tile	Tile open windows
Arrange	Arrange all windows
Windows	List of all MDI child windows

Help Menu

The **Help** menu is the last standard Windows application menu that allows the user access to any on-line help facilities. The help facilities are not implemented in the prototype. In the full application, the help facility would be built using the Windows Help Compiler.

Command	*Description*
Contents	Help contents
About PS Admin...	About the application

Child Windows

When a command is selected from the main menu, a child form is displayed to the user to allow them to complete the task that they have selected. If the task selected is a standard Windows application command, like Cut or Copy, the default Windows action would occur.

If the action requested is a *PS Admin Workstation* function, an MDI child window is created by the parent window with the appropriate information and functionality for the task. Once created, these windows can be minimized and closed by the user.

These windows can also be started up and minimized when the simulation application is opened by the user. In addition to showing the menu structures, Figures A.1 through A.6 also show the child window icons that are opened at runtime.

The *PS Admin Workstation* prototype has five child windows that can be created using the **Configure** menu. These windows allow the instructor to set up the simulation environment before the student begins the simulation exercise.

Configure Project Skills

The **Configure Project Skills** window allows the instructor to add, modify and delete the skills necessary to complete the project tasks. This window is presented in Figure A.7.

The instructor will input an ID, name, description and learning curve factor for each skill needed in the project.

Configure Job Skills

The **Configure Job Skills** window allows the instructor to add, modify and delete the skills necessary to complete each job in the project. This window is presented in Figure A.8.

For each job in the project, the instructor will select the skills and the relative importance of each skill in completing the job.

Configure Worker Skills

The **Configure Worker Skills** window allows the instructor to rate each worker on each skill needed in the project. This window is presented in Figure A.9.

For each worker in the project, the instructor will rate their ability and desire to perform that skill.

Configure Client Characteristics

The **Configure Client Characteristics** window allows the instructor to describe the client for the project. This window is presented in Figure A.10.

The instructor will describe the client and set several characteristics that will affect the simulation.

Configure Worker Characteristics

The **Configure Worker Characteristics** window allows the instructor to describe each worker in the project. This window is presented in Figure A.11.

The instructor will describe each worker and set several characteristics that will affect how well they complete their assigned jobs in the simulation.

Project Simulator

The *Project Simulator* application will be accessible through the Windows desktop environment by double-clicking a program item icon or by selecting **Run** from the Program Manager **File** menu.

The student will use the *Project Simulator* to:

- View client, job and worker descriptions
- Schedule client, team and worker meetings
- Schedule worker training
- Review project reports
- Run the simulation

All user interface screen captures for the *Project Simulator* interface will be found in **Appendix B, Project Simulator Prototype.**

Menu Structure

The main menu for the parent *Project Simulator* window will have seven menu pads. Screen captures that show the menu options for each of these main menu pads are presented in Figures B.1 through B.7 in **Appendix B.**

Each of the main menu pads and their associated commands are described below.

File Menu

The **File** menu contains the standard Windows application file commands that allow the user to manipulate the files associated with the application and to quit the application via the Exit command.

The New and Save commands or omitted from the menu pad since all user actions will automatically be saved by the application.

Command	*Description*
Open...	Open an existing simulation
Close	Close the current simulation
Print...	Print selected information
Exit	Exit the PS Admin Workstation

Edit Menu

The Edit menu contains the standard Windows application edit commands that allow the user to cut, copy, paste and delete text from the application.

Command	*Description*
Cut	Cut selected text
Copy	Copy the selected text
Paste	Paste the selected text
Delete	Delete the selected text

Schedule Menu

The Schedule menu allows the student to schedule meetings with the client, team and workers; worker training; project and job reviews and client and worker surveys..

Command	*Description*
Meeting...	Schedule meetings with the client, team or workers
Training...	Schedule worker training
Review...	Schedule project or job review (not prototyped
Survey...	Schedule client or worker survey (not prototyped)

View Menu

The View menu allows the student to view the history of the simulation and to view descriptions of the skills, jobs, client, workers and resources.

Command	*Description*
History...	View history of simulation decisions and results (not prototyped)
Project Skills...	View project skill definitions
Job Skills...	View skills needed for each job
Client...	View client description
Workers...	View worker descriptions and skills
Resources...	View resources (not prototyped)

Run Menu

The Run menu allows the student to run the simulation for the next week with the scheduled staff development activities.

Command	*Description*
Run...	Run the simulation for the week

Window Menu

The Window menu is a standards Windows application menu that allows the user to arrange the windows on the desktop and to provide access to individual windows by listing them in the menu.

Command	*Description*
Cascade	Cascade open windows
Tile	Tile open windows
Arrange	Arrange all windows
Windows	List of all MDI child windows

Help Menu

The Help menu is the last standard Windows application menu that allows the user access to any on-line help facilities. The help facilities are not implemented in the prototype. In the full application, the help facility would be built using the Windows Help Compiler.

Command	*Description*
Contents	Help contents
About PS...	About the application

Child Windows

When a command is selected from the main menu, a child form is displayed to the user to allow them to complete the task that they have selected. If the task selected is a standard Windows application command, like Cut or Copy, the default Windows action would occur.

If the action requested is a *Project Simulator* function, an MDI child window is created by the parent window with the appropriate information and functionality for the task. Once created, these windows can be minimized and closed by the user.

These windows can also be started up and minimized when the simulation application is opened by the user. In addition to showing the menu structures, Figures B.1 through B.7 also show the child window icons that are opened at runtime.

The *Project Simulator* prototype has six child windows that can be created using the **Schedule, View** and **Run** menus. These windows allow the student to make view project information, schedule staff development activities and run the simulation exercise.

Schedule Meeting

The **Schedule Meeting** window is accessed through the Meeting command on the **Schedule** menu pad. The window allows the student to schedule meetings with the client, the team and with individual workers. This window is presented in Figure B.8.

The student selects the date of the meeting, who will attend, the purpose, the topic and the number of hours.

Schedule Training

The **Schedule Training** window is accessed through the Training command on the **Schedule** menu pad. The window allows the student to schedule skills training for individual workers. This window is presented in Figure B.9.

The student must identify the start date, the worker, the skill to be trained and the number of hours of training.

Project Skills

The **Project Skills** window is accessed through the Project Skills command on the **View** menu pad. The window is an information window that displays a list of all skills defined for the project. This window is presented in Figure B.10.

Job Skills

The **Job Skills** window is accessed through the Job Skills command on the **View** menu pad. The window is an information window that displays the skills needed for each job in the project. This window is presented in Figure B.11.

Client Description

The **Client Description** window is accessed through the Client command on the **View** menu pad. The window is an information window that displays the client description. This window is presented in Figure B.12.

Worker Description

The **Worker Descriptions** window is accessed through the Workers command on the **View** menu pad. The window is an information window that displays the descriptions for each of the workers in the project. This window is presented in Figure B.13.

Run Simulation

The **Run Simulation** window is accessed through the Simulation command on the **Run** menu pad. The window is a dialog window that displays the scheduled project activities and allows the user to run the simulation. This window is presented in Figure B.14.

Implementation

Once the requirements, functional specification and user interface design have been completed, the implementation phase can begin. The implementation phase includes the following major steps:

- Selecting the development methodology
- Selecting the appropriate software tools
- Developing the system according to specification
- Documenting the system development

Although the scope of the masters project was to implement simply a prototype of the *Project Simulator* application, the same development issues had to be resolved. The decisions made in developing the prototype are documented to direct the implementation of the fully-functional system.

Prototype Implementation

Using the functional specification and the user interface design descriptions, I developed prototypes of both the *PS Admin Workstation* and the *Project Simulator* applications using *Visual Basic 3.0*.

I began by opening new projects for each of the applications. I then created a main form which served as the MDI parent for the application. On this form, I constructed the main menu pads with their associated commands.

I then created MDI child forms for all of the windows that allowed the user to input or view information. These windows were built using both standard and professional edition *Visual Basic* tools.

After the child windows were built, I tied the application together using Basic commands to load the child forms when the main application started and when the appropriate commands were selected from the main menu pads.

> **Note: In the prototype implementation, none of the underlying database structures were built using Microsoft Access 1.1. The prototype was built to allow the client to see the proposed user interfaces for the applications.**

The screen captures for the prototype applications are found in **Appendix A, PS Admin Workstation Prototype** and **Appendix B, Project Simulator Prototype**.

Actual Implementation

The implementation of the fully-functional *Project Simulator* applications will require a more rigorous development strategy with standard coding and naming conventions.

The actual implementation of the applications can be done from the prototypes by building the underlying database structures, connecting the field elements in the child windows and writing the underlying code to manage the flow of data throughout the system.

The sparse development documentation in this masters project report serves as the only system documentation for the prototype. The full implementation of the system will require more detailed and appropriate programmer documentation to be written by the developers as the system is being built.

Validation & Verification

Validation and verification of the simulation is the final step in the simulation development life cycle. These two test allow the developer and the client to determine how closely the simulation model reflects the entities and interactions of the real-world systems as well as how the simulation results compare to actual system behavior.

Since there were no results produced by the *Project Simulator* prototype, the validation and verification phase of this project will consist of walking the client through the prototype application and the proposed system data flow.

The subsection following the walk through describe how validation and verification could be performed in the fully functional *Project Simulator* application.

Client Prototype Review

The key decision-maker from the client reviewed this document and the *Project Simulator* prototype applications. The issues that arose from this review are presented in the Discussion section.

Prototype Demonstration

I then walked the decision-maker through the prototype applications for both the *PS Admin Workstation* and the *Project Simulator*. We reviewed all of the menu designs as well as the input screens.

The client felt that the interfaces were very effective, but suggested that the fully functional simulation applications be run through a usability test with actual users.

A usability test involves having a series of "typical" users of the system walk through a set of routine tasks. Their performance and comments are recorded and used to improve the look-and-feel of the application.

Functional System Tests

Validation of the fully functional *Project Simulator* application will involve testing the model against the real-world project management environment.

The client and other experts in the field of project management should review the simulation model and the interactions that students use to direct the completion of the exercise. Their feedback will test the validity of the model both as a reflection of the system and as an effective educational tool.

Once simulation results are obtained from the functional applications, the developer and client can then verify the results. Verification will compare the results of the simulation model to actual data from real-world project management activities.

The *Project Simulator* applications can easily be verified by using actual project data from projects tracked in *Project Scheduler 6* and running the simulation with the actual decisions. If the simulation is accurate, then the results should be very close.

If the results are not within an acceptable range defined by the client, then the simulation rules will need to be rewritten to reflect the real-world behavior.

Discussion

Client Review

The client's review of the document and prototype application provided areas for improvement and refinement of the project management simulation environment. These topics are discussed below.

- The amount and structure of the information returned to the student at the end of each simulation run has not been defined. The system will update the project tracking databases with the actual information.
- The *Project Simulator* should also display and maintain simulation results information for the students to review. This capability will be available through the History... command in the View menu.
- The client would like to review the inventory of individual personality characteristics for completeness, but welcomed the availability of additional factors for rating clients and workers.

Other Topics

Due to the complex nature of project management simulation environments, a countless number of additional features and factors can be considered in the fully-functional applications. Several of these topics include:

- Unplanned events
- Catastrophic events
- Complex simulation rules
- Reversing the simulation results

Each of these features requires additional research, the construction of a prototype and review with the key decision-makers of the project.

Conclusion

Learning Objectives

The learning objective of this masters project was to gain an appreciation of the techniques used in computer-based simulations by designing and developing a project management simulation environment for use in training.

My research into project management and simulation literature deepened my knowledge of this challenging area of computer science. Putting these principles into practice in the development of a simulation application allowed me to better understand the issues faced by simulation developers.

Following standard software and simulation life cycle methodologies helped me to focus my project efforts. Also, my experiences with the client reviews and the user interface design reinforced the principle that the decision-maker for which the simulation is being done should be intimately involved throughout the development process.

Client Objectives

Aside from accomplishing the learning objectives, the masters project also resulted in a full project management simulation application design, as well as an expandable prototype of the simulation interfaces.

The client can now use the system design, the interface prototype, and a list of additional capabilities to obtain developer resources to build the complete simulation application environment for use in their project management training curriculum.

Bibliography

Project Management Literature

Constantine, Larry L., ""Work Organization: Paradigms for Project Management and Organization." Communications of the ACM (October 1993): 35-40.

Rettig, Marc and Simons, Gary, "A Project Planning and Development Process for Small Teams." Communications of the ACM (October 1993): 45-54.

Zahniser, Richard A., "Design by Walking Around." Communications of the ACM (October 1993): 115-122.

Simulation Literature

Jain, Raj. The Art of Computer Systems Performance Analysis. New York: John Wiley & Sons, Inc., 1991.

Miller, David P.; Firby, R. James, Fishwick, Paul A.; Franke, David W.; Rothenberg, Jeff, "AI: What Simulationists Need To Know." ACM Transactions on Modeling and Computer Simulation (October 1992): 269-284.

Appendix A
PS Admin Workstation Prototype

This appendix presents the user interface design and actual prototype screen captures for the *PS Admin Workstation* application. The screens were built using *Microsoft Visual Basic 3.0.*

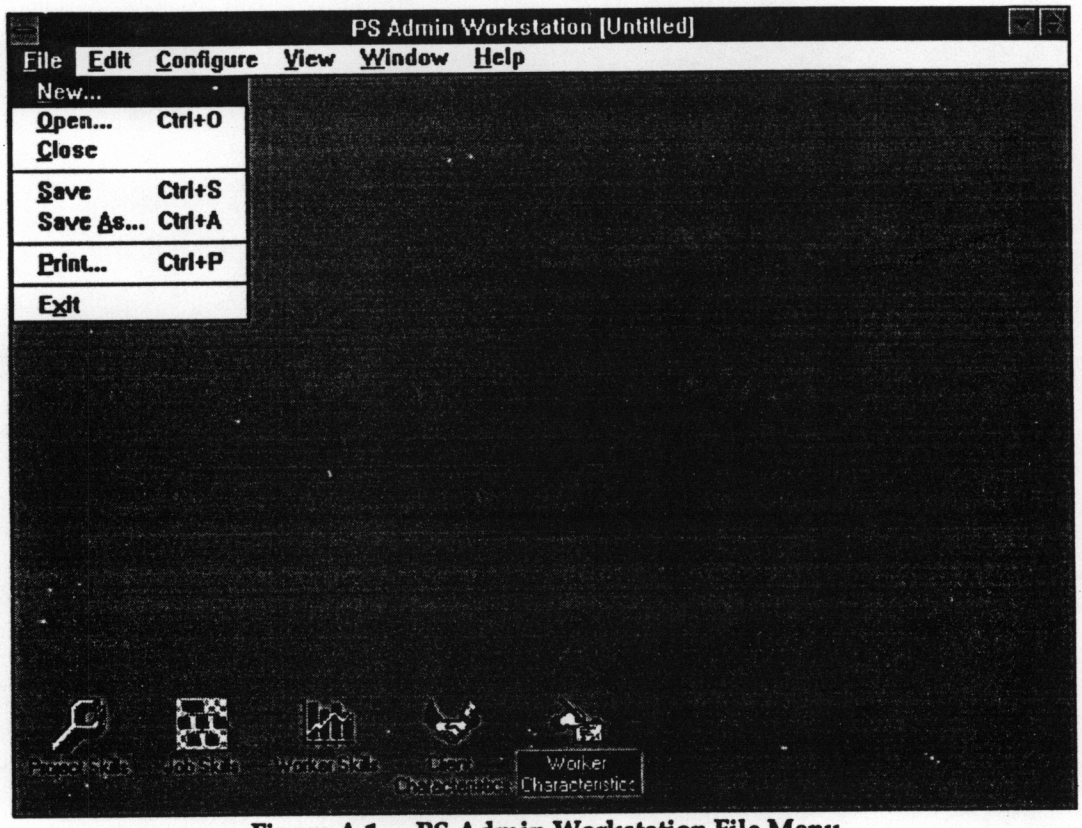


Figure A.1 - PS Admin Workstation File Menu

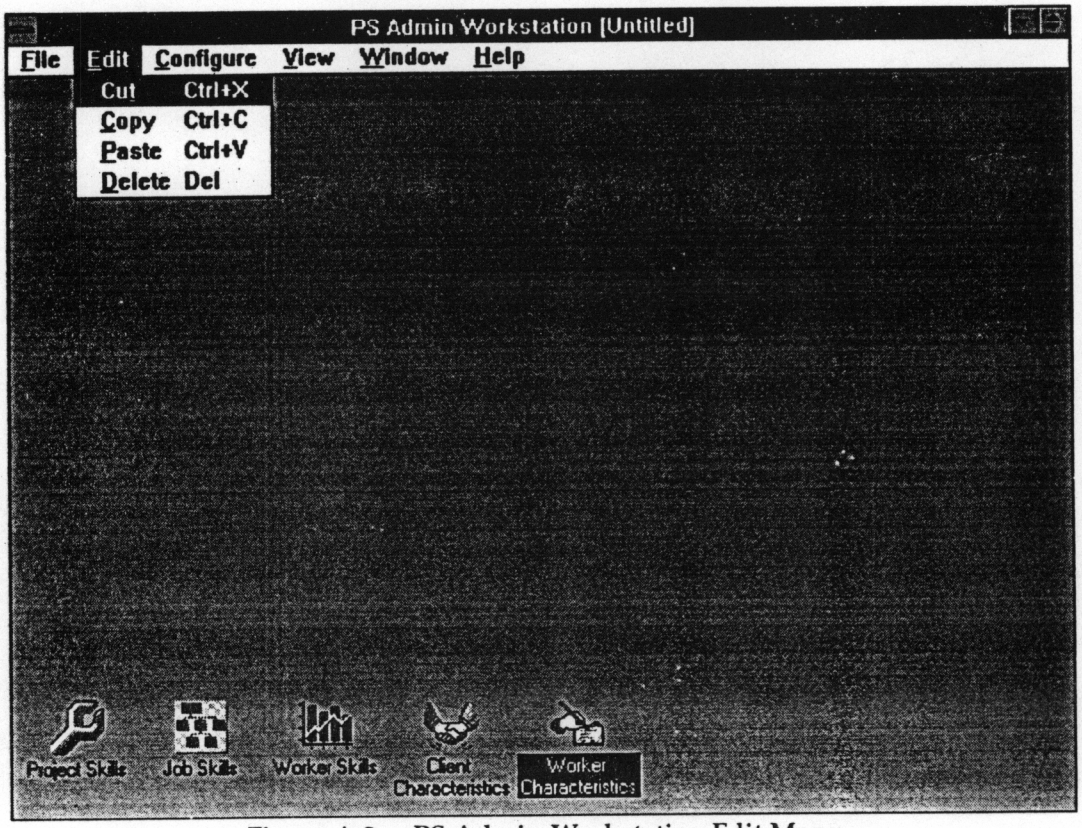


Figure A.2 - PS Admin Workstation Edit Menu

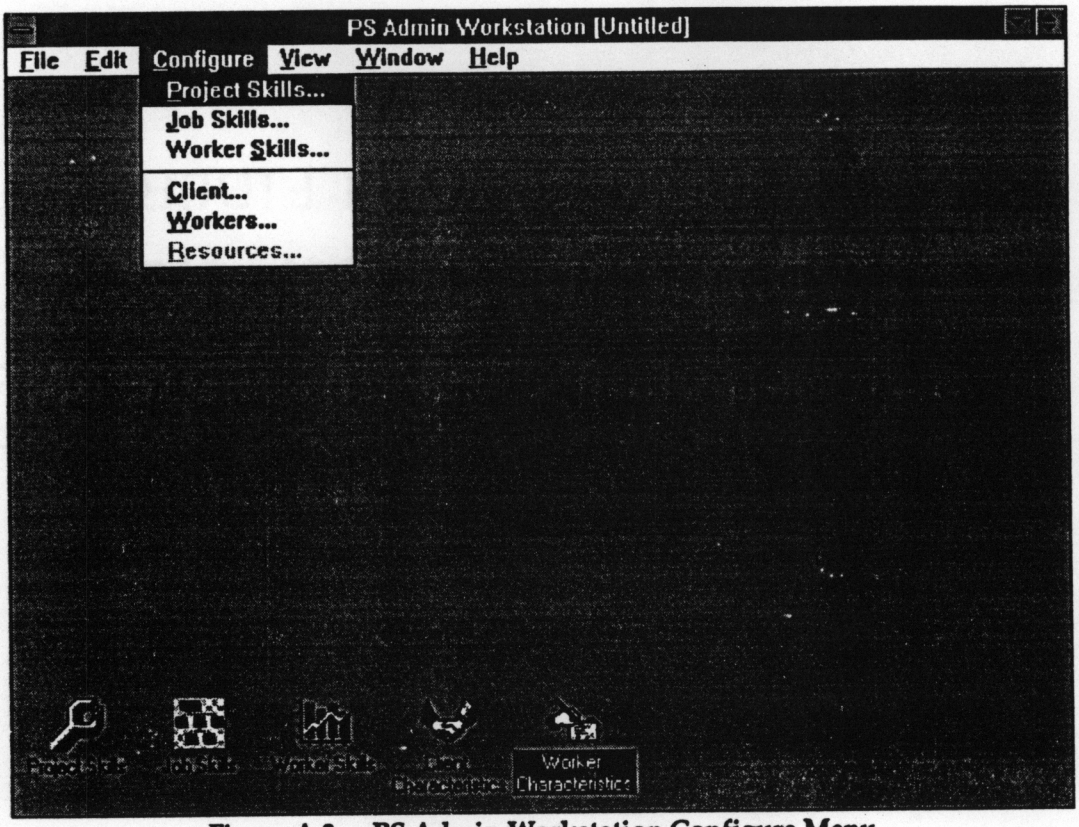


Figure A.3 - PS Admin Workstation Configure Menu

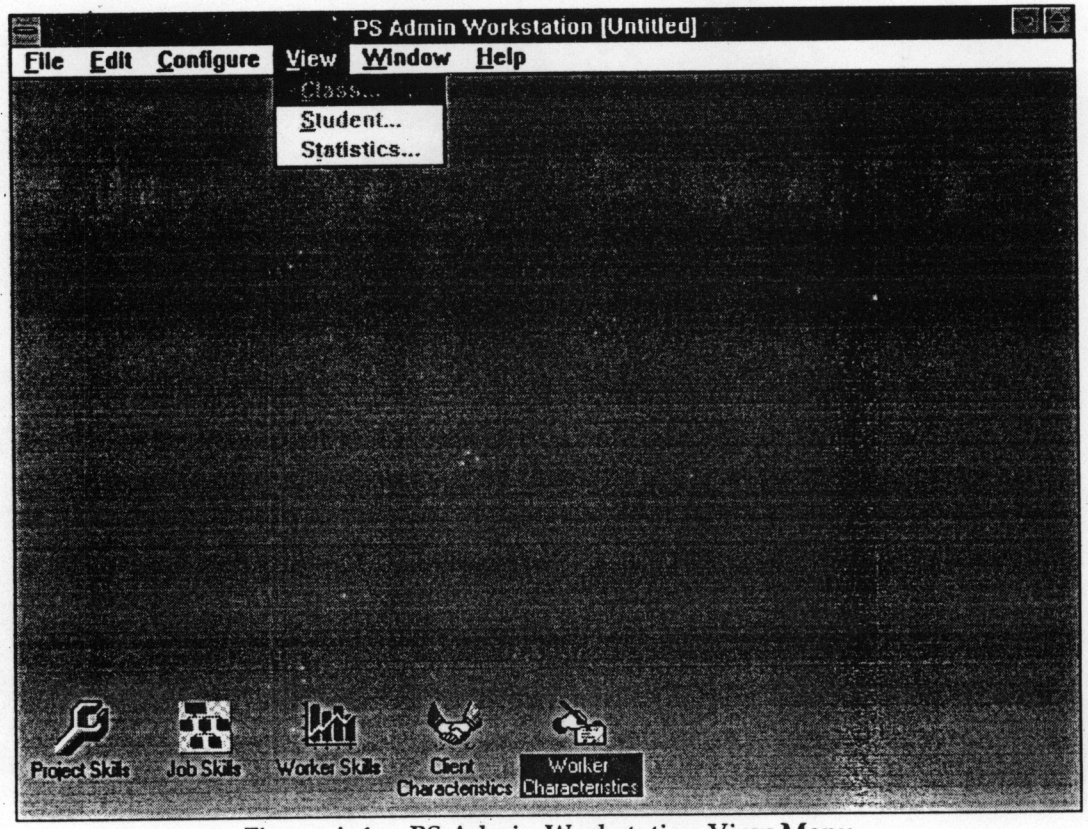


Figure A.4 - PS Admin Workstation View Menu

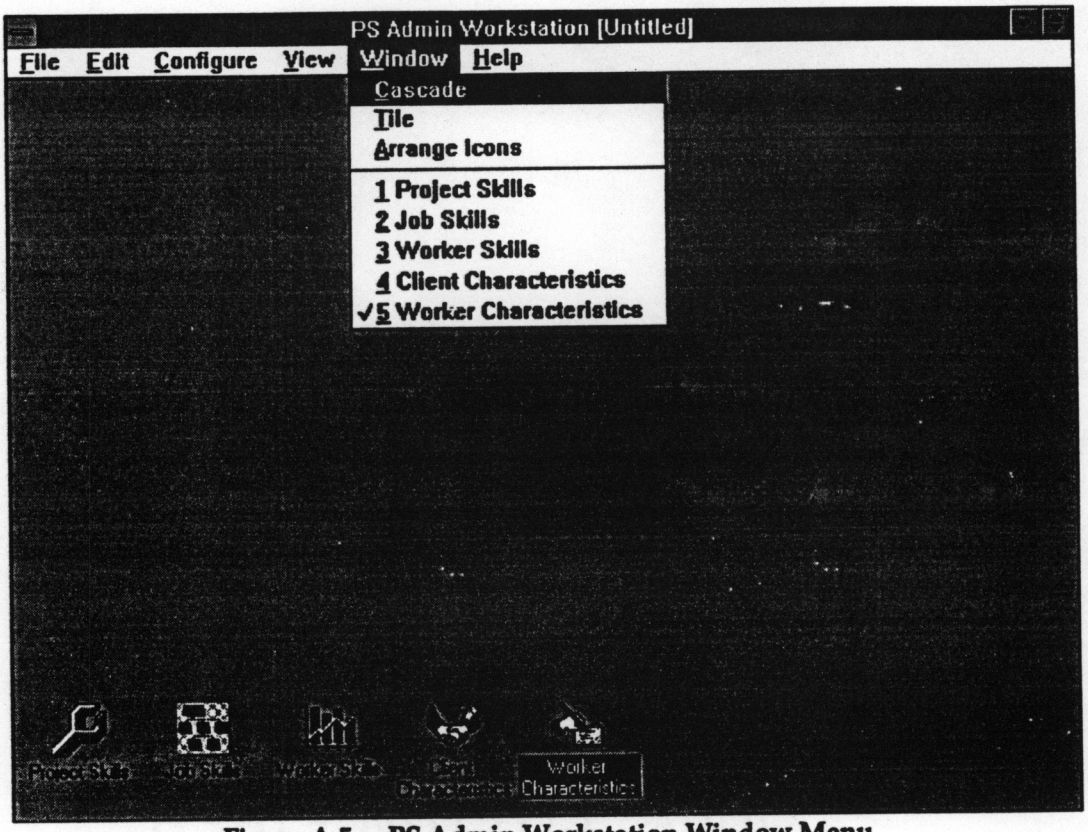


Figure A.5 - PS Admin Workstation Window Menu

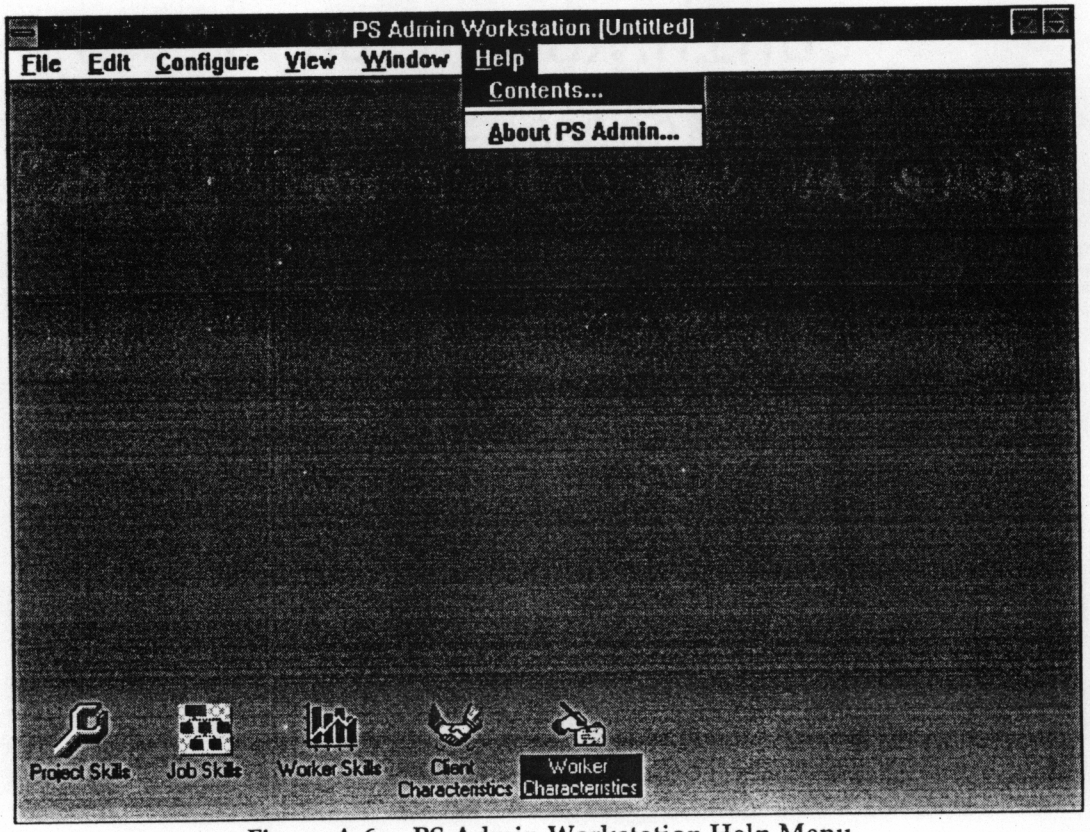


Figure A.6 - PS Admin Workstation Help Menu

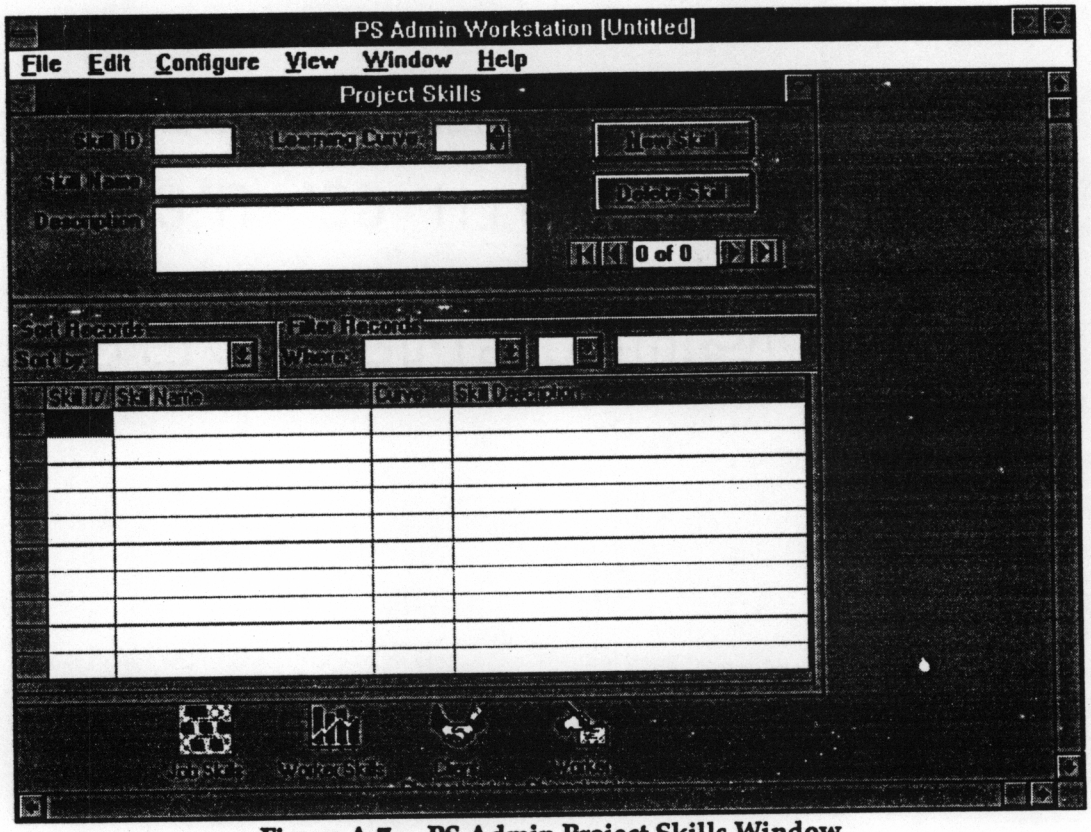


Figure A.7 - PS Admin Project Skills Window

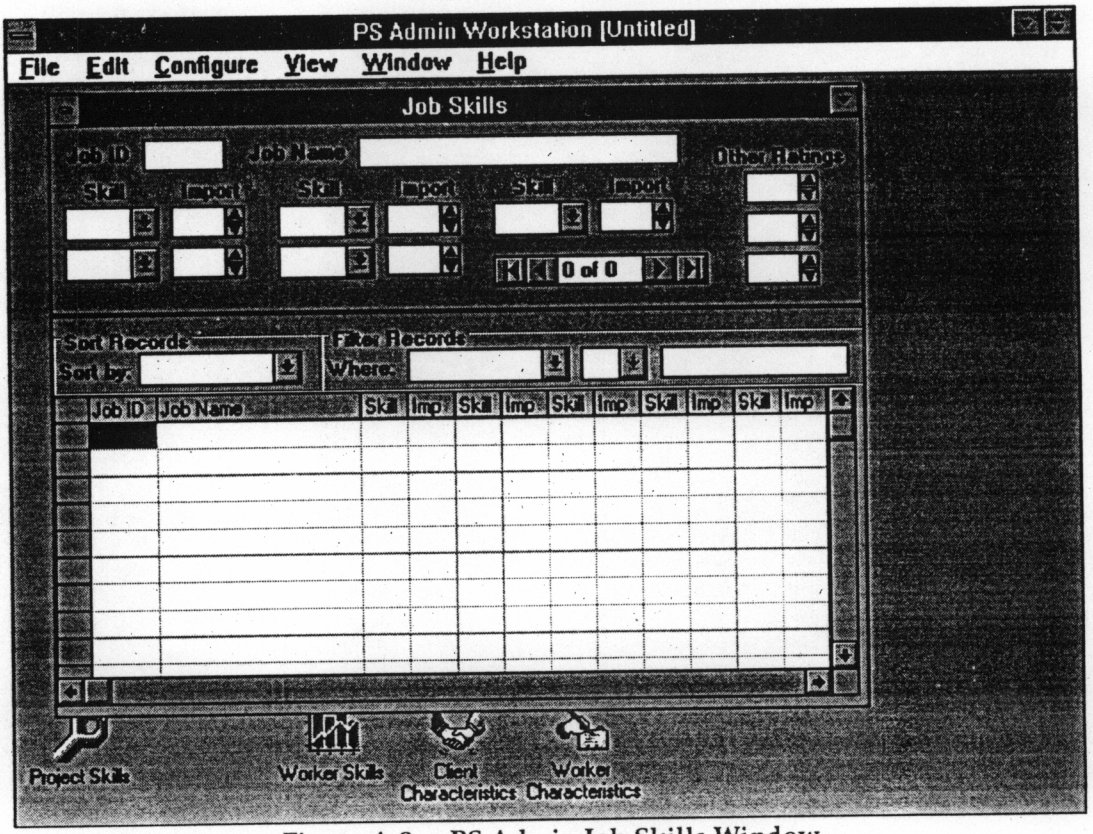


Figure A.8 - PS Admin Job Skills Window

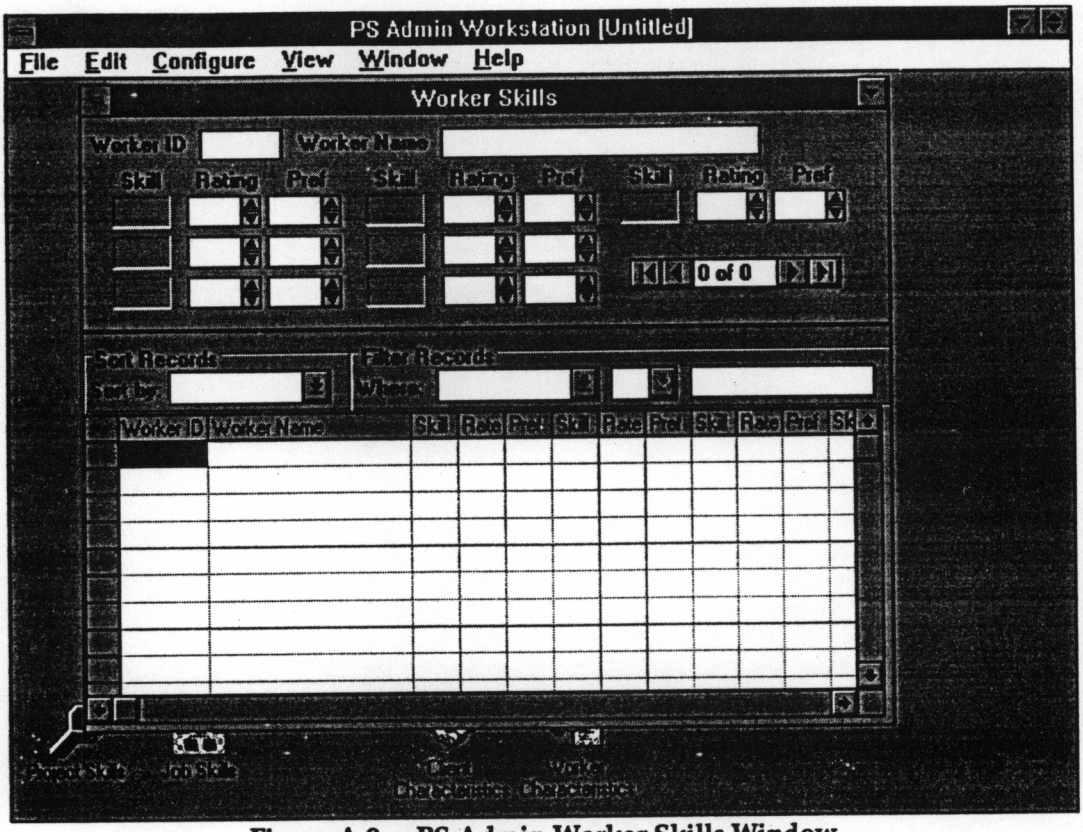


Figure A.9 - PS Admin Worker Skills Window

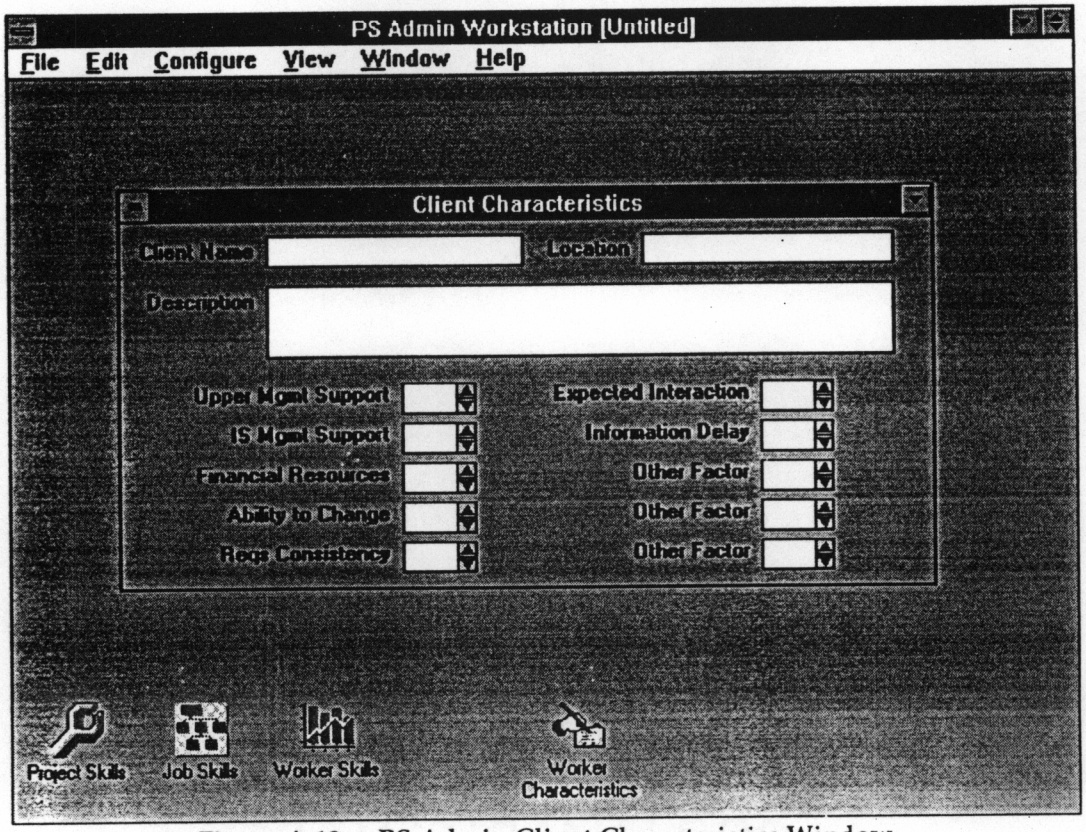


Figure A.10 - PS Admin Client Characteristics Window

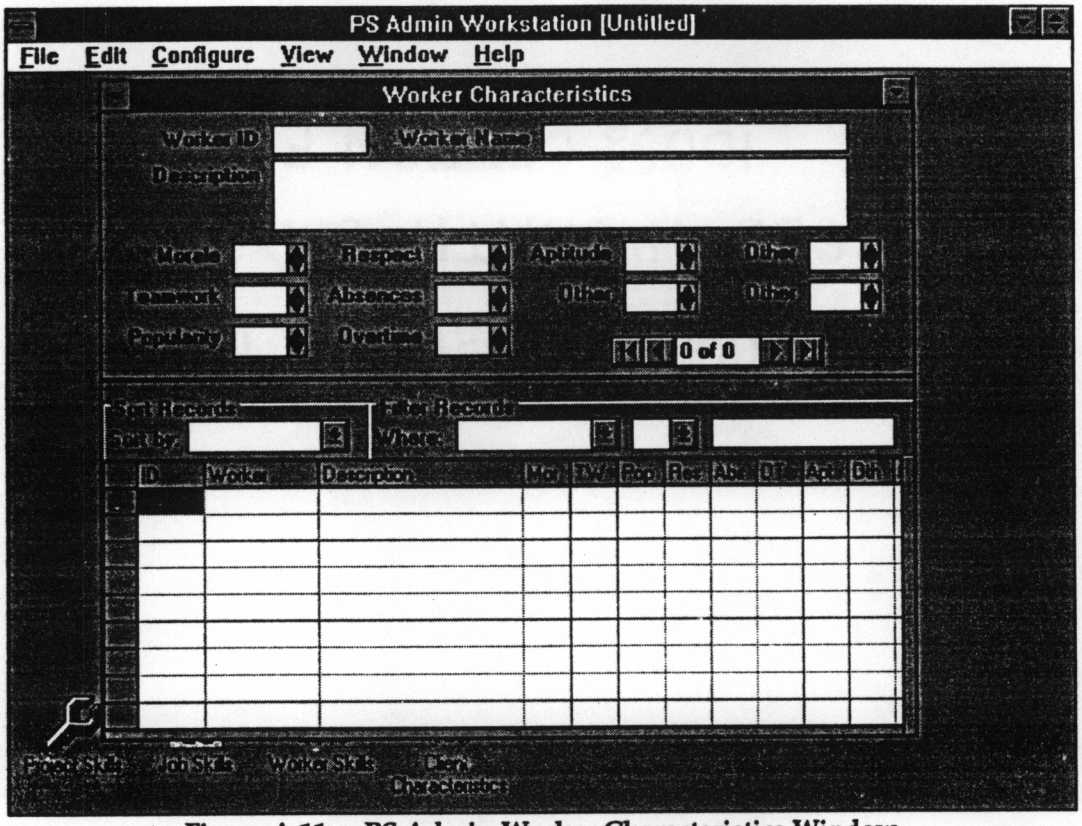


Figure A.11 - PS Admin Worker Characteristics Window

Appendix B
Project Simulator Prototype

This appendix presents the user interface design and actual prototype screen captures for the *Project Simulator* application. The screens were built using *Microsoft Visual Basic 3.0.*

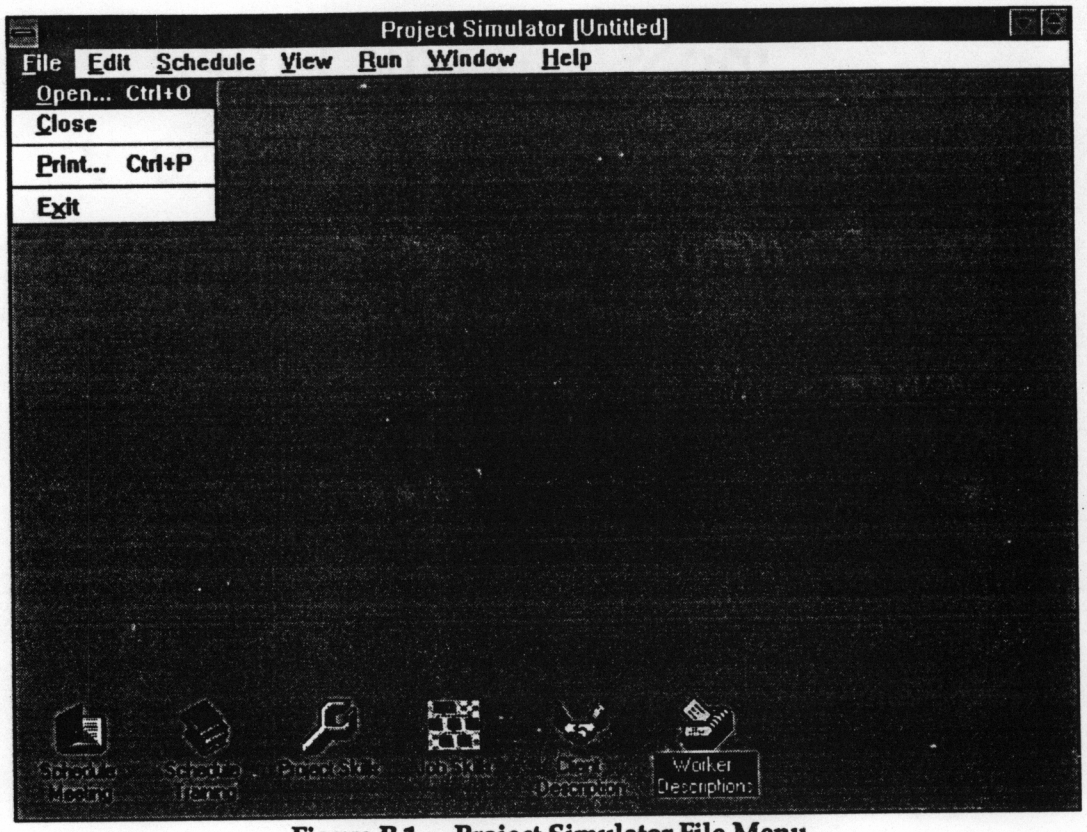


Figure B.1 - Project Simulator File Menu

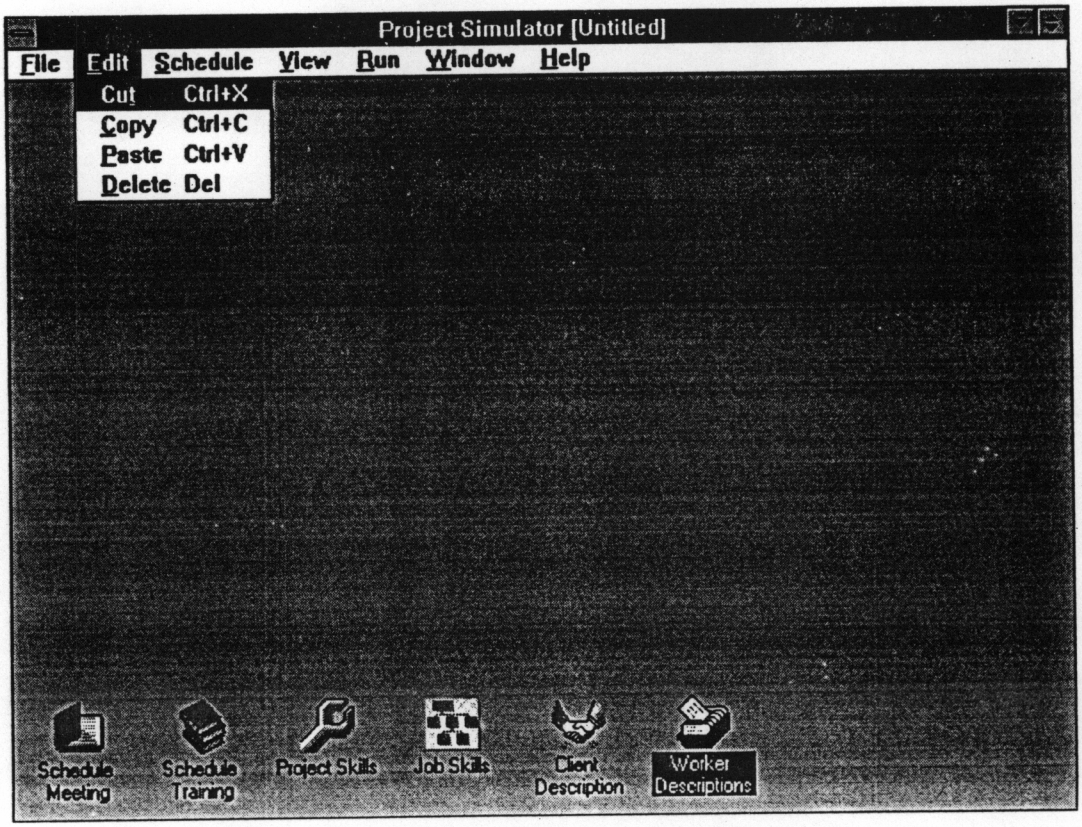


Figure B.2 - Project Simulator Edit Menu

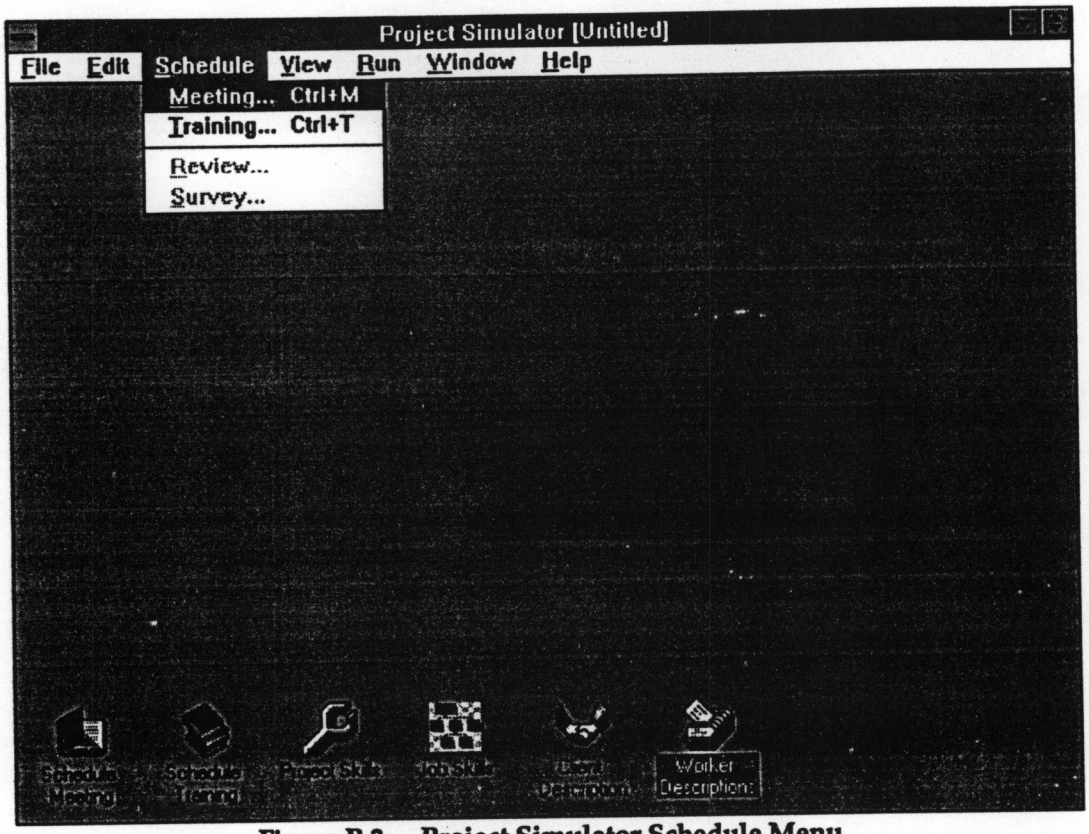


Figure B.3 - Project Simulator Schedule Menu

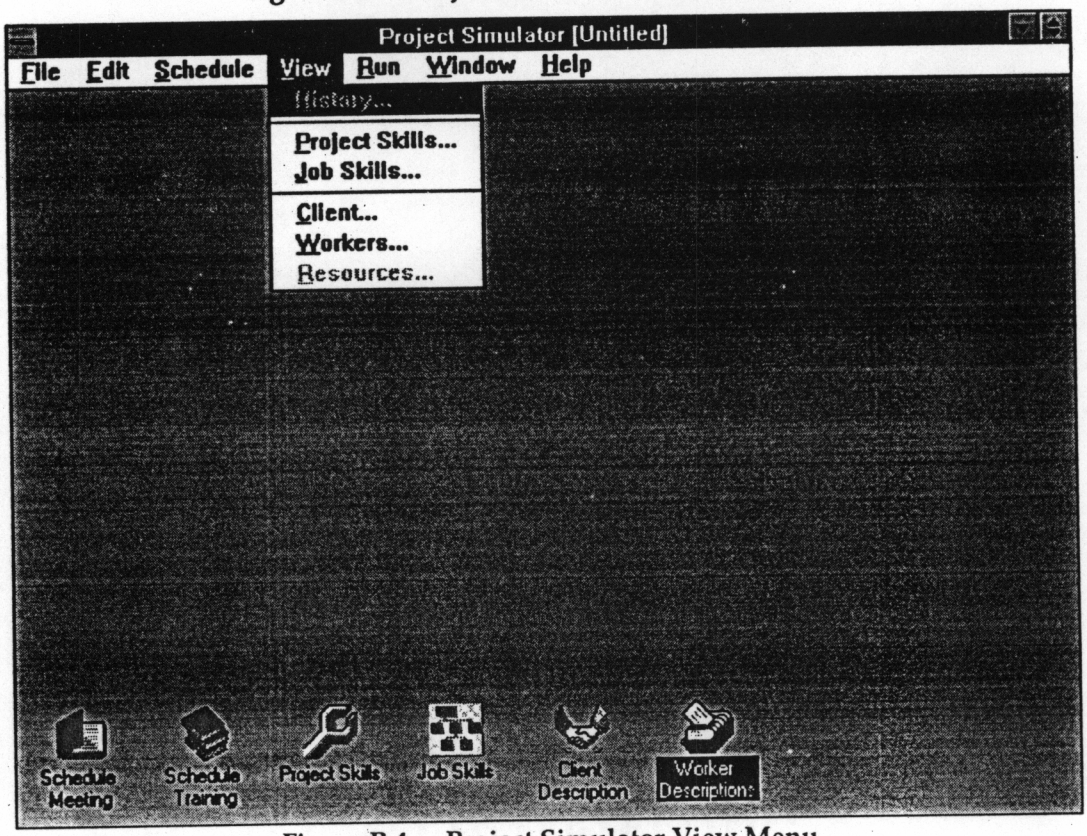


Figure B.4 - Project Simulator View Menu

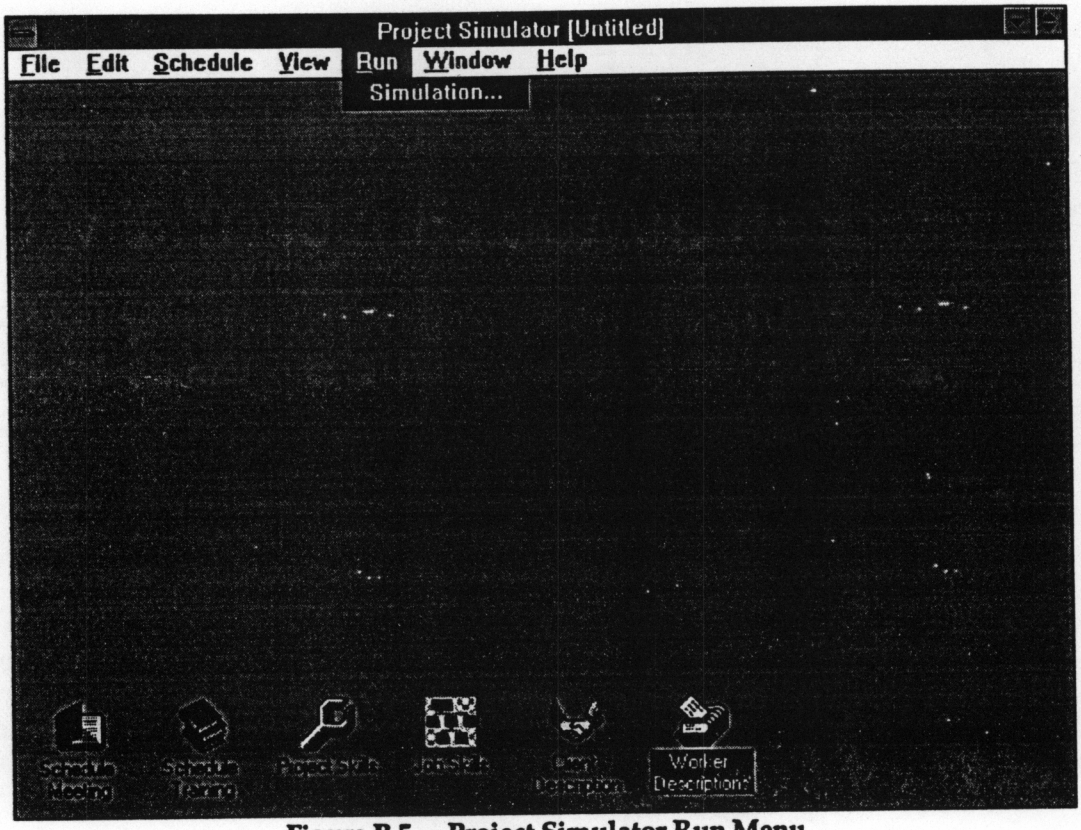


Figure B.5 - Project Simulator Run Menu

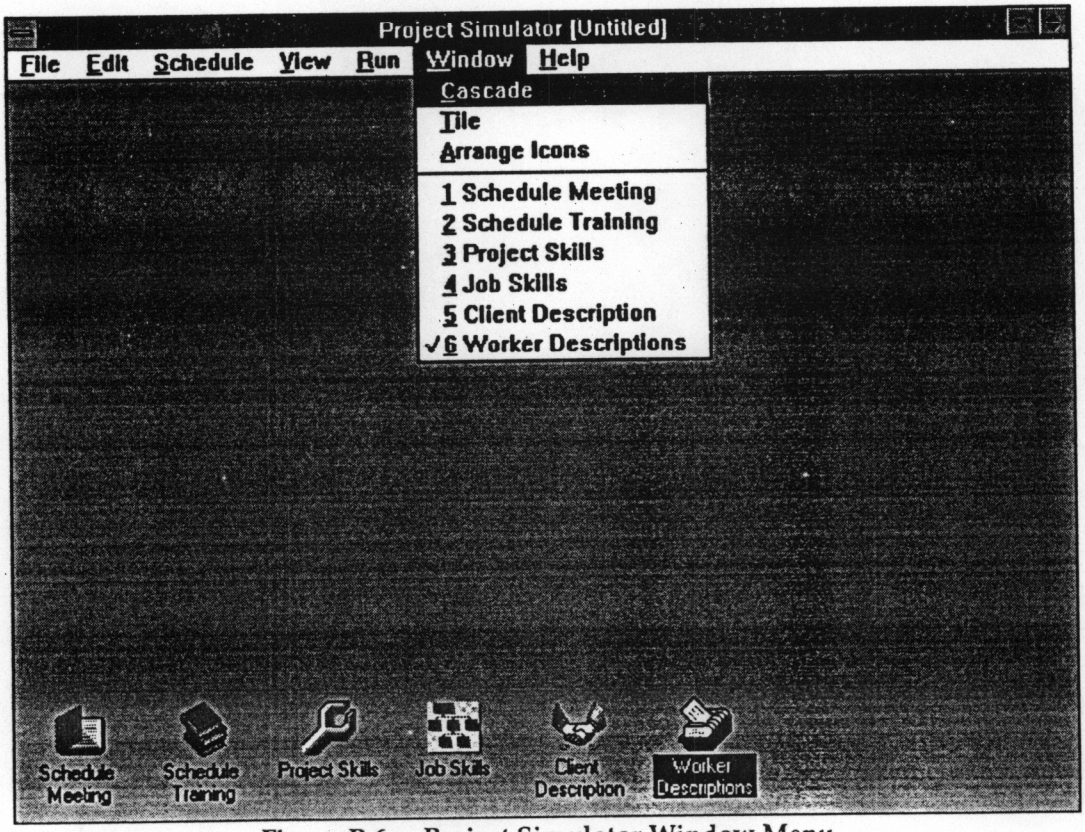


Figure B.6 - Project Simulator Window Menu

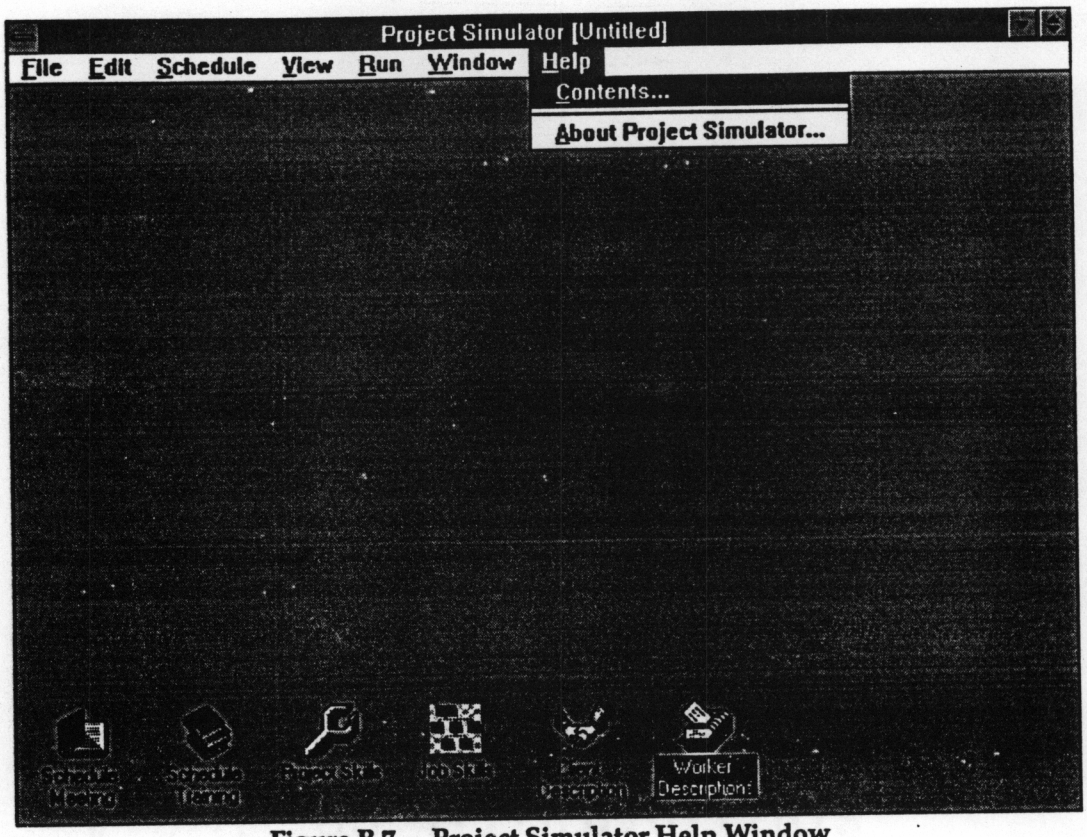


Figure B.7 - Project Simulator Help Window

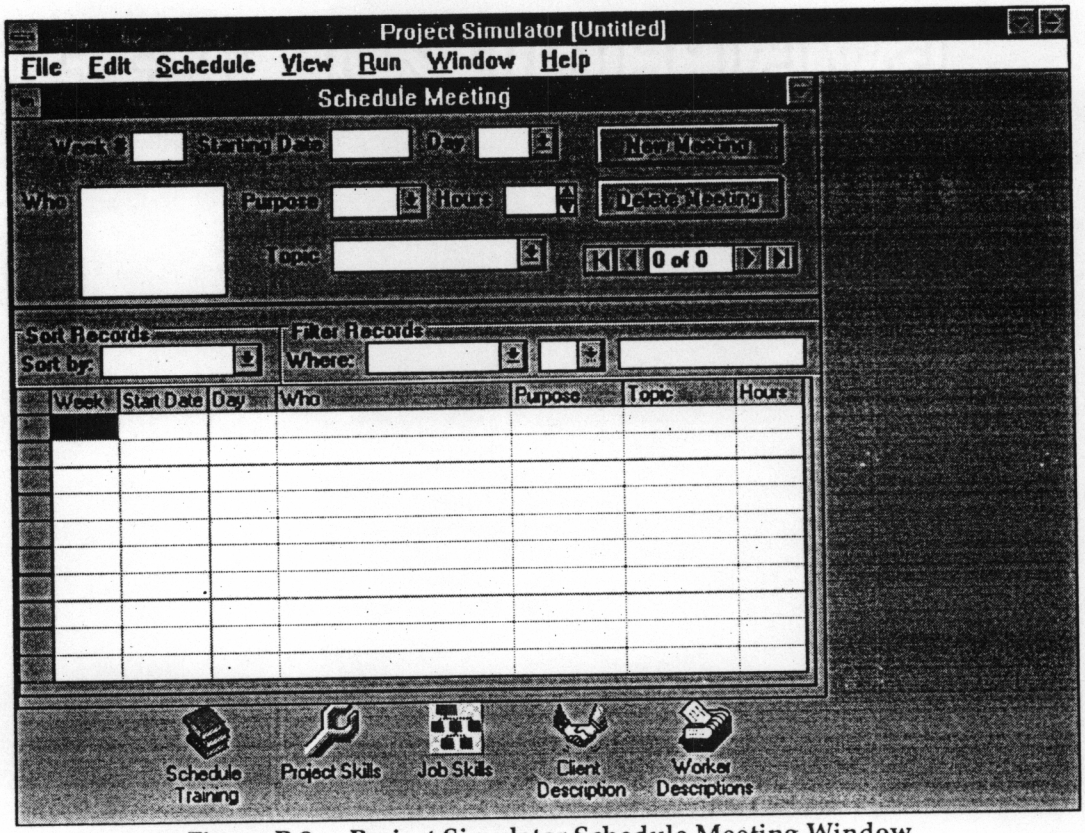


Figure B.8 - Project Simulator Schedule Meeting Window

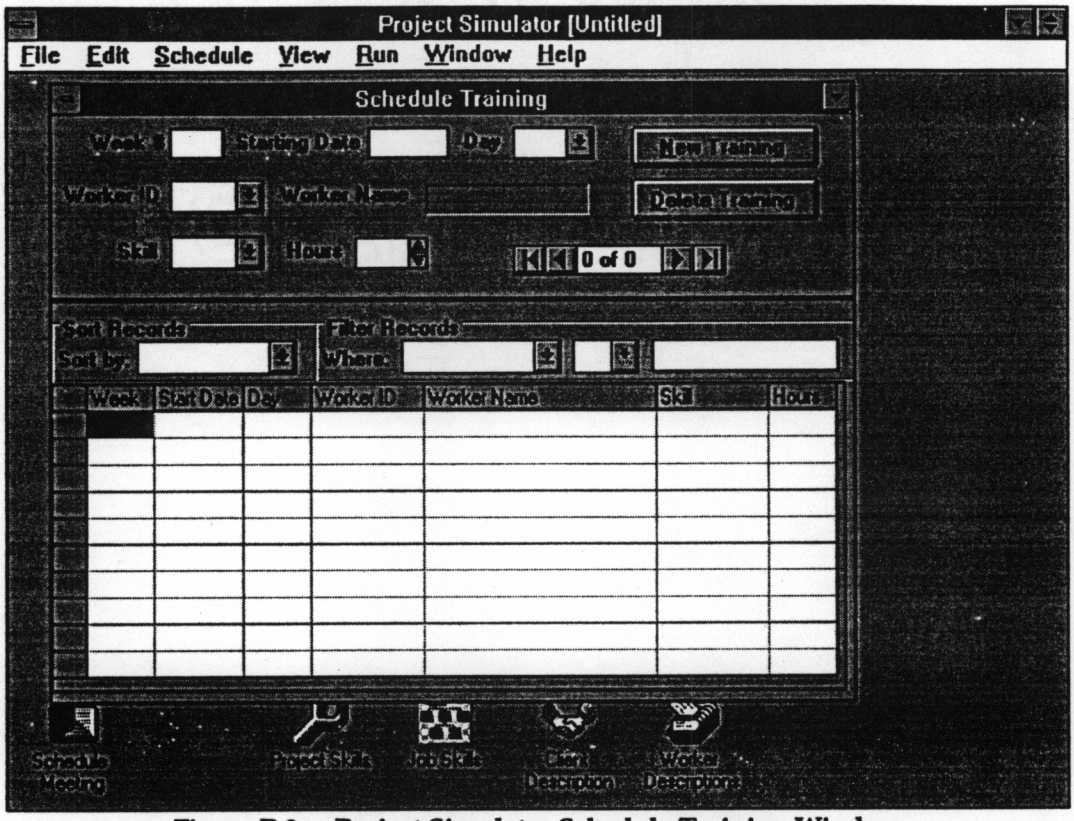


Figure B.9 - Project Simulator Schedule Training Window

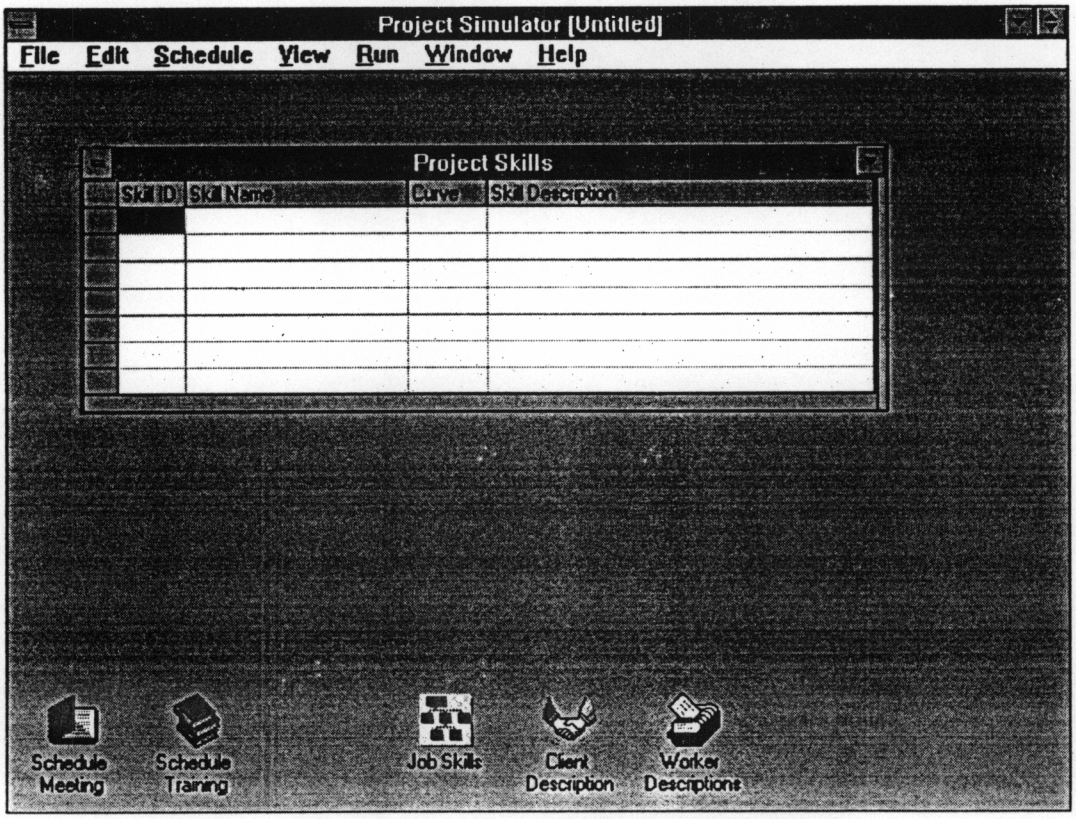


Figure B.10 - Project Simulator Project Skills Window

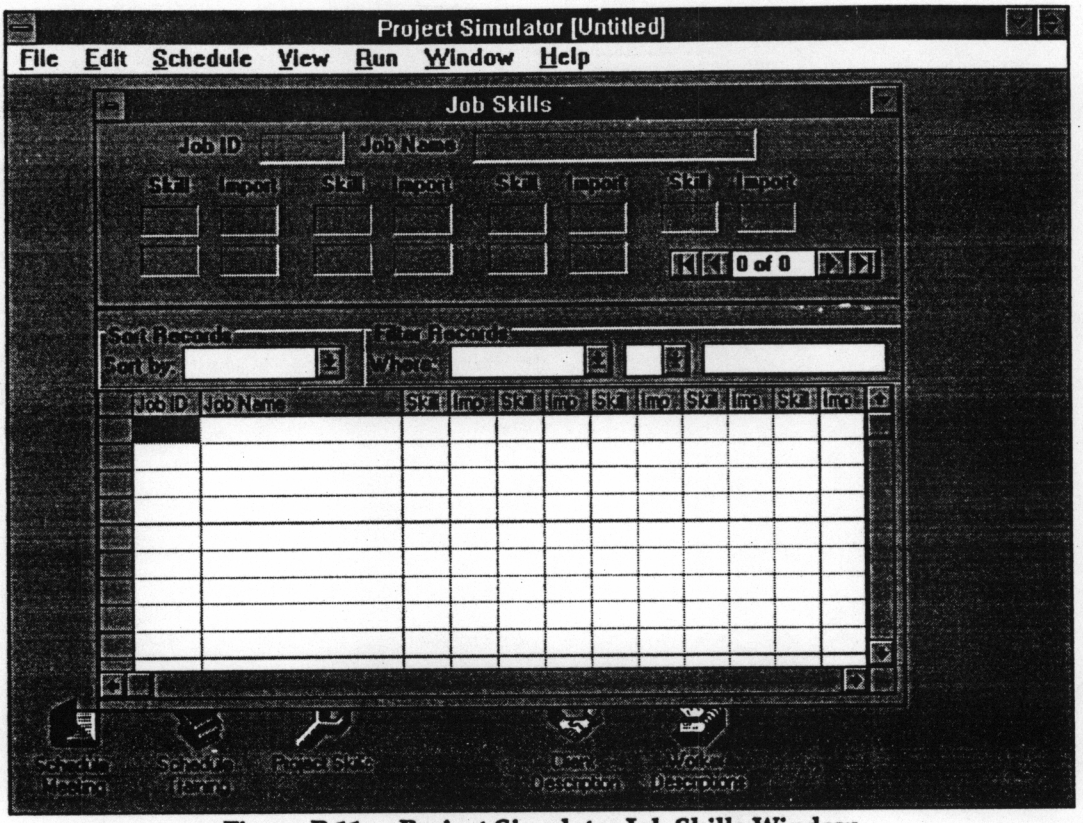


Figure B.11 - Project Simulator Job Skills Window

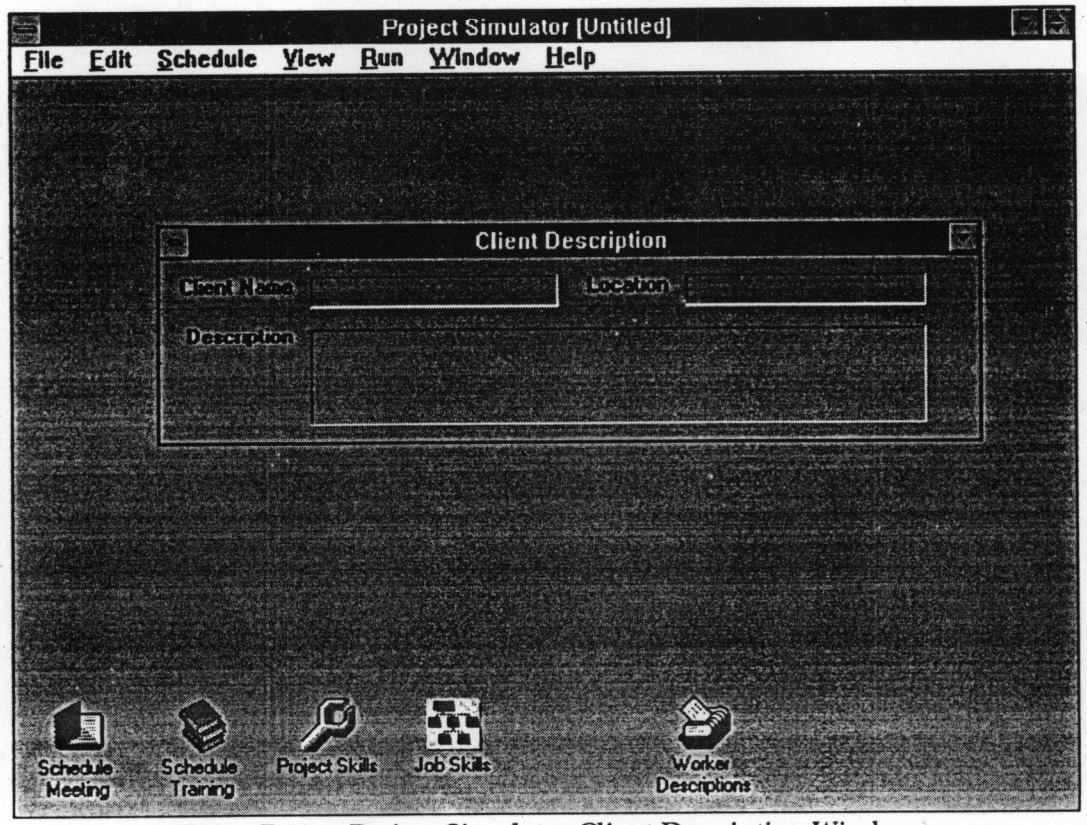


Figure B.12 - Project Simulator Client Description Window

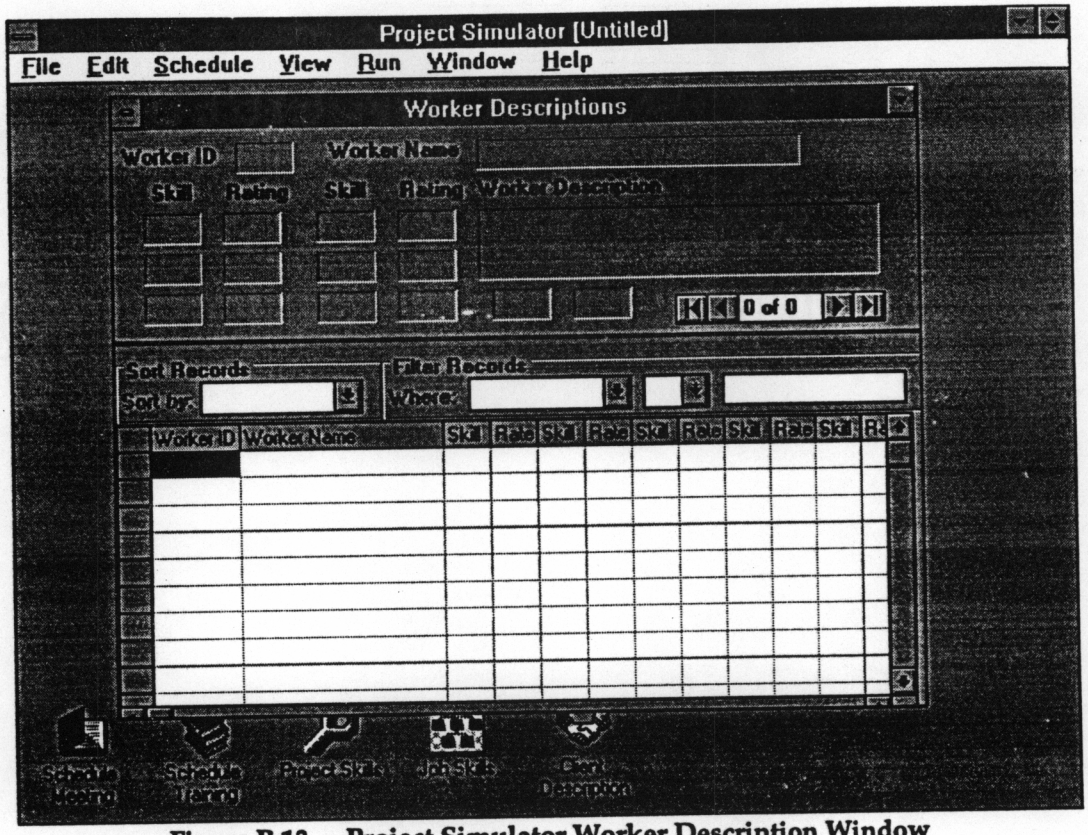


Figure B.13 - Project Simulator Worker Description Window

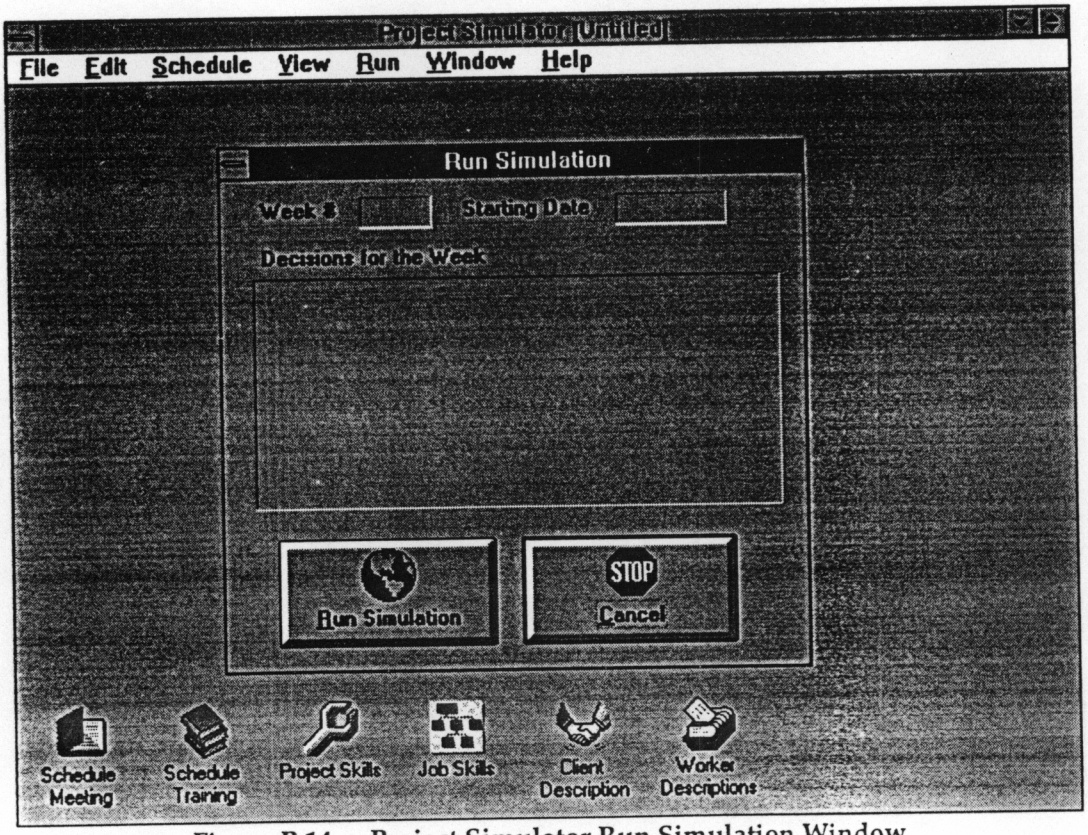


Figure B.14 - Project Simulator Run Simulation Window

Appendix C
Project Scheduler 6 Interface

This appendix presents screen captures from the *Project Scheduler 6 for Windows* beta release from Scitor Corporation. This project tracking software application will be integrated with the *Project Simulator* applications.

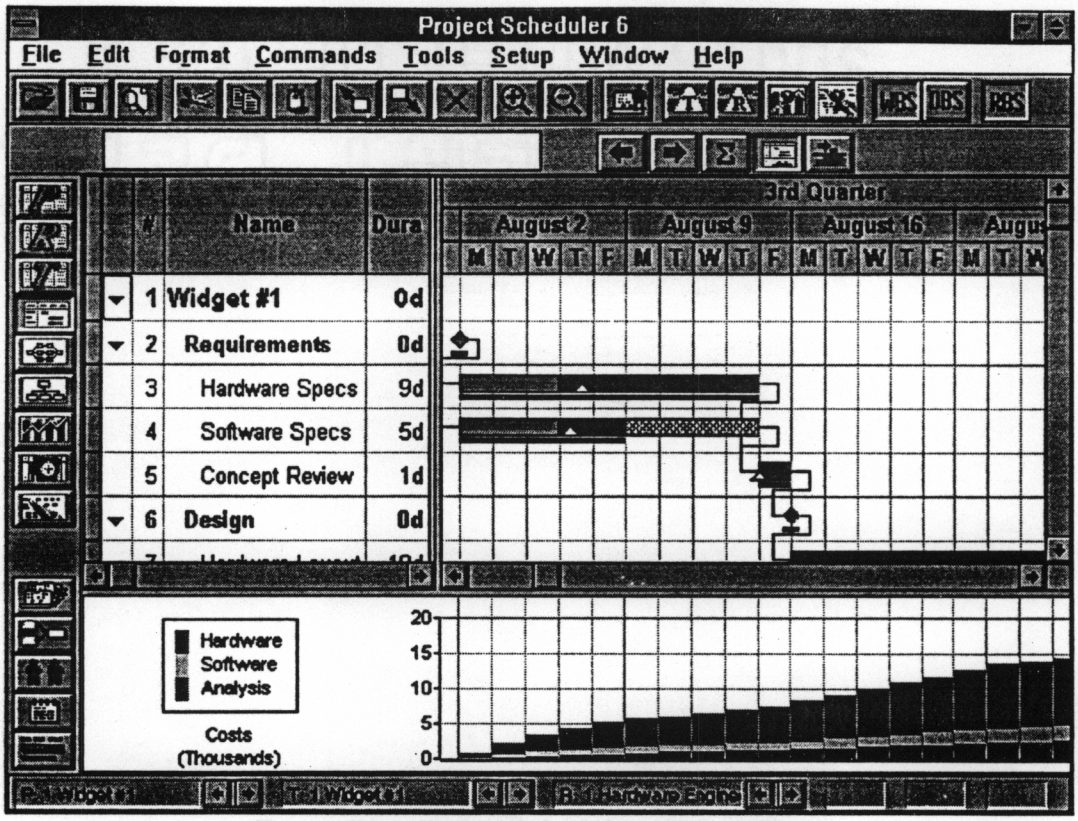


Figure C.1 - Project Scheduler 6 Gannt Chart

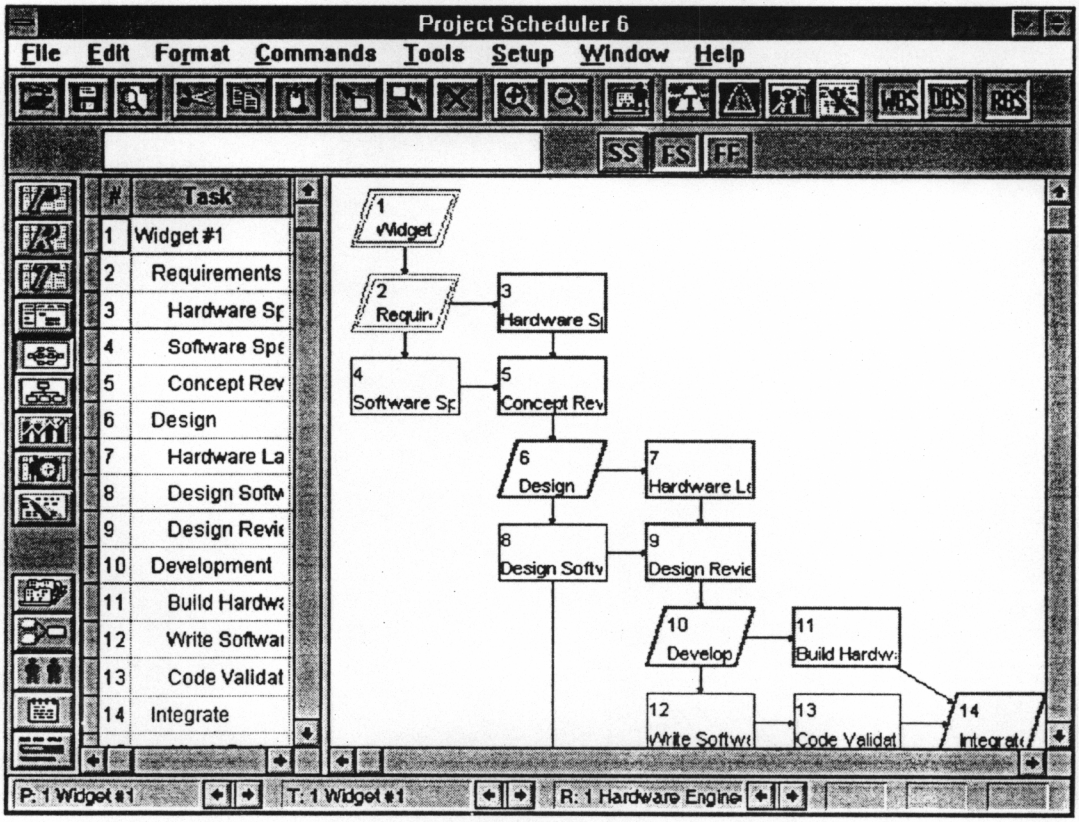


Figure C.2 - Project Scheduler 6 Network Chart

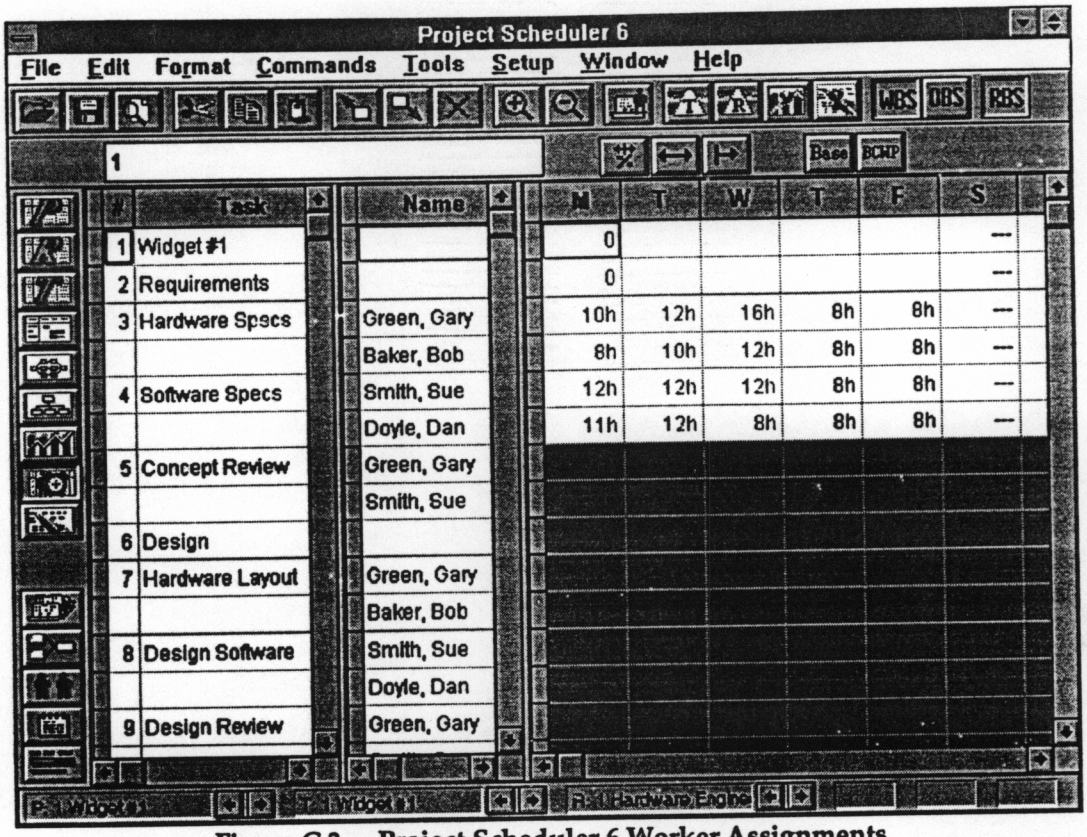


Figure C.3 - Project Scheduler 6 Worker Assignments

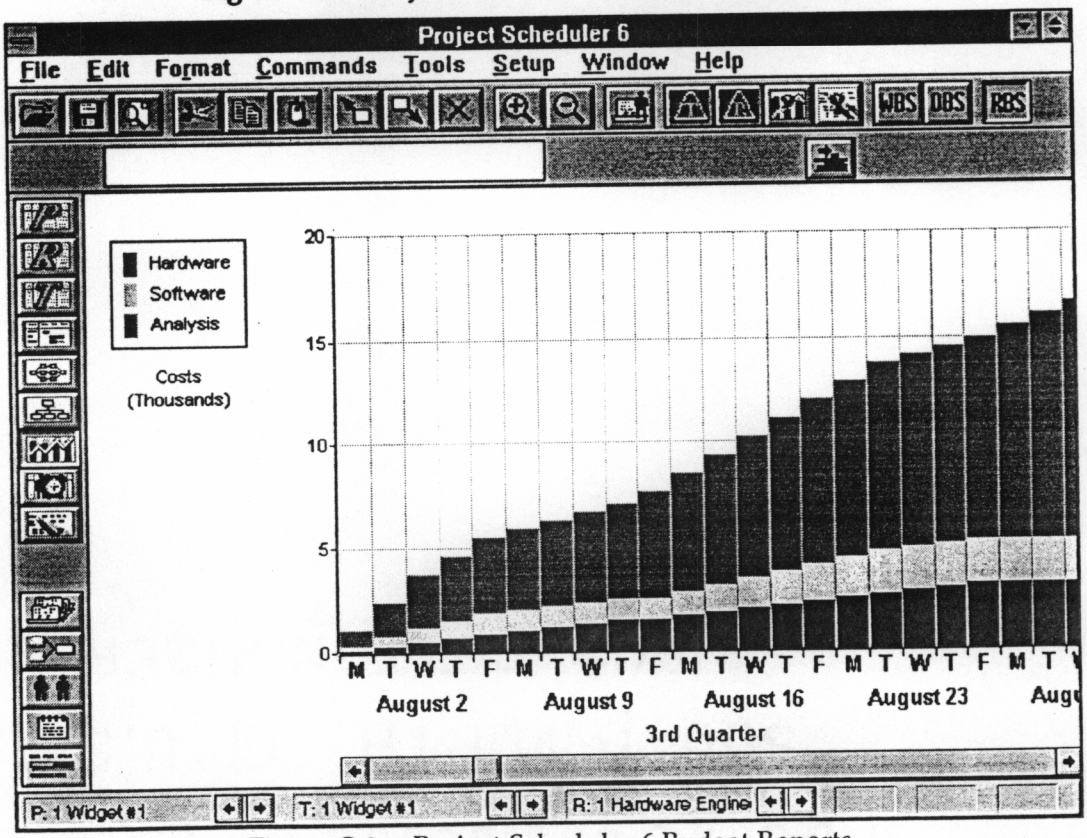


Figure C.4 - Project Scheduler 6 Budget Reports