Kris, Jyh-Hong Lai/15.05.1961

AN INTERACTIVE PROGRAM FOR IMPLEMENTING A PROJECT PLANNING AND CONTROL SYSTEM

J. H. Lai and R. Leonard

University of Manchester Institute of Science and Technology, UK

(IEE, UKACC International Conference on CONTROL '96 Conference Publication, Number 427, Volume I, pp96-101)

INTRODUCTION

PC-based project management software (PMS) packages exist for sophisticated applications, and have relatively low installation and running cost. This has made PMS more accessible for small and medium-sized contracting firms. Although the PMS packages' functionality have been enhanced in many areas, two problems have been encountered, namely that packages are difficult to learn and complex to use.

One solution is to develop stand-alone, knowledge-based systems for the construction industry. But, these are not available for small or medium-sized companies using PC-based PMS packages. This paper presents an effective approach to the implementation of PMS by means of an interactive program (IP) in a fire protection contracting company.

FACTORS HINDERING PMS APPLICATIONS

The factors resulting in failures to implement a PMS package have been widely discussed, and could be summarised into four major aspects (1, 2, 3, 4, 5):

The mismatch of PMS packages' specifications with the user's requirements. Most PMS packages are designed for different levels of sophistication. They lack flexibility to accommodate different users' specific needs and are either too simple or too complex for use in different PM environments.

The difficulty of learning and operating PMS packages. Most PMS packages emphasise their technical capabilities, rather than provide friendly operating interfaces. Subsequently, user confidence in the package is inhibited.

The inefficiency and ineffectiveness of the user's existing PM system. Many companies do not have any clear definition of their existing PM systems in terms of operating procedures, organisation structures, and information channels, etc. This results in ineffective interfaces between different operational functions if implementing the PMS application.

The lack of manpower training and senior management's support. Many companies lack a long-term information technology (IT) strategy, and senior management support for implementing the PMS application. Subsequently, little training is provided on the PM knowledge required for operating the PMS.

In addition, although PMS packages provide functions for automating PM techniques, they are not capable of automating tasks, such as constructing a work breakdown structure (WBS) and defining an organisation breakdown structure (OBS). Accordingly, the usability of PMS package is restricted.

RESEARCH OPPORTUNITIES

In the past, most PMS suppliers have tried to improve PMS packages' general operating functions, as well as providing advanced applications. However, these only give partial solutions to the problems encountered. In addition, many researchers have helped develop knowledge-based systems for automating the control of complex construction projects (6, 7, 8, 9). Yes, these knowledge-based systems are generally not available for small and medium-sized contracting companies using PC-based

PMS application. Therefore, opportunities exist in developing better methods for improving PC-based PMS applications in areas such as:

- Integrating the user's PM process into the operation of the PMS packages.
- Automating and standardising the construction of WBS and OBS.
- Simplifying data inputting and default setting.
- Enhancing user operating interfaces through the provision of useful procedural guidelines.
- Automating project analysis and problem diagnosis.
- Standardising project planning and control reports.
- Automating data exchanges between PMS and other systems.

INTERACTIVE PROGRAM

Definition

The term interactive program refers to an external application developed for interacting with a host program. In general, the host program provides a built-in Dynamic Link Library (DLL) allowing certain computer languages to communicate with it. Application programs with these specific languages are able to send date or commands to manipulate the operations of the host program. In addition, they might obtain data from both user inputs and the host program for further processing.

The use of IPs are generally found in many computer-aided design (CAD) systems. This involves creating user-input interfaces for obtaining design data, to generate drawings automatically. Among project management software packages, few low-end products have provided functionality for developing IPs. Those which do provide DLLs for writing IPs are generally compatible with Database or Basic languages. For example, both Microsoft Project 4 and CA-SuperProject 4 are able to communicate with Visual Basic.

Potential Applications and Benefits

The purposes of developing an IP for application with a PMS could be many, depending upon the extra operating capabilities that the user wants to add to the PMS applications. The IP could be used for automating a list of keystrokes as performed by macros. In contrast, it could perform complex tasks, such as retrieving data for the host PMS and reprocessing the data for analysis or diagnosis purposes. The following are some potential areas for developing the IP in the PM applications.

Standardisation. To use a PMS package efficiently, standardisation of PM methods and procedures are essential. For example, the standardisation of project task names, individual resource details, and assignment account codes will speed the construction of project plans, as well as facilitating the summary of project information for different managerial requirement. Similarly, the standardisation of scheduling methods, progress data inputting, and project reporting helps to maintain the consistency and integrity of project information. To a great extent, these requirements could be achieved by writing IPs with sub-routine procedures which provide the user with step-by-step guidance throughout the entire PM process.

Customisation. Much information and functionality in the PMS is not necessary for the user's specific PM requirements. By developing an IP, operating interfaces can be simplified. This involves the customisation of data input/output layouts, graphical views, data filtering/sorting criteria, and help prompting, etc. In addition, for training purposes, the operating procedures can be tailored for different levels of sophistication, thus enhancing the user's learning ability.

Automation. Operating the PMS is time-consuming, therefore, automating repetitive tasks into simple operations saves time, reduces errors, and increases users' satisfaction. Areas to be automated are setting project defaults, assigning resources and account codes to tasks, inputting progress data, performing what-if analysis, diagnosing project performance, and producing reports, etc. In general, these can be achieved through the development of IPs.

Integration. A difficult task in the PMS application is to integrate or exchange data with existing systems. Many PMS packages provide functions for data import or export, but the operating procedures are complicated. For instance, if the user wants to convert cost estimates from a spreadsheet into the PMS, for constructing a project WBS, at least three conditions are required: a) the methods for estimating project costs must be consistent with the methods for constructing the project WBS, b) the format used in the spreadsheet must be compatible with that used in the PMS; and c) there must be mechanisms in the PMS for data export or import. These conditions are not easy to meet because the diverse estimating methods are not suitable for data conversion (e.g. bill of quantity). One solution is to develop an IP for retrieving estimates from the spreadsheet, and converting them into the WBS in the PMS.

PROJECT BACKGROUND

Wormald Ansul UK is part of the world's largest fire protection company. It designs, manufactures and installs fire protection systems for domestic and overseas use in power generation, petrochemical, offshore and general industrial applications. The provision of timely information for effective project planning and control is vital for completing projects on time, within budget and to specification. In the past, these tasks used manual methods, spreadsheet databases and different drawing packages in separate areas of the company. This limited the effectiveness and flexibility of project control, as well as increasing the potential for contractual problems.

To resolve these problems, the company envisaged a PM system for integrating PM staff with project information. This involved collaboration With Total Technology at UMIST. The task needed an analysis of the company's requirements, followed by the selection and implementation of a PMS package. A PMS package, called CA-SuperProject 3.0 from Computer Associates was selected, and a pilot implementation constructed for integrating PM knowledge and techniques.

After the pilot study, PM staff became convinced of the potential benefits. The company thus decided to go ahead with full-scale implementation. However difficulties were envisaged in the actual application, listed as follows:

- Various PM procedures and control methods exist for different functional requirements, due to the diversity of business operations. This entails different focuses on the PMS application (i.e. time, cost, or resource).
- Many PM staff do not have adequate knowledge of using computers for project planning and control. This requires extensive training in PM techniques for the PMS application.
- There is a lack of manpower for the application. Most users are engineers who have limited time for learning the tool. This necessitates simplifying procedures for operating the PMS package.
- The proportion of project variations is high, which needs effective methods for analysis knock-on effects on all aspects of project execution.
- There is a need to standardise project planning and control reports for different managerial uses.
- Project information, generated between estimating, accounting, and site operations, for PM is manually connected. This requires integrating project information with the use of the PMS package.

Accordingly, an IP called 'Hands-On Wizard', was developed to resolve these potential difficulties. The results show that the effectiveness and efficiency of the PMS application has been greatly enhanced. The following sections detail the development process.

DEVELOPMENT OF 'HANDS-ON WIZARD'

Selection of Programming Language

A study was conducted on developing an IP to interact with CA-SuperProject. This showed that CA-SuperProject provided 180 built-function commands accessible by popular computer languages, including CA-Realizer (a programming language provided by Computer Associates). This allowed the author to develop an IP to link with CA-SuperProject, for performing simple macros and sophisticated applications.

Meanwhile, a programming language called Visual Basic, from Microsoft, was chosen for developing the Hands-On Wizard system. The reasons for Visual Basic were that: a) Visual Basic provides design tools for developing graphical applications, which were easy to operate within the Windows environment, b) Visual Basic is a simplified, high level language (similar to all Basic languages), by which applications were easy to be developed and maintained, and c) Visual Basic has excellent functionality for designing event-driven programs, suitable for designing training applications with heuristic scenarios.

Development of Program

The development of the Hands-On Wizard system are depicted as follows:

Standardisation of WBS, OBS, and CBS. The first step was to examine the detailed operations in Wormald's current PM system, in terms of: a) general contracting and tendering methods, b) cost estimating methods, c) project planning methods, d) resource utilisation, e) cost control methods, f) performance measurements, g) project reporting systems, and h) existing computer systems and manpower. This resulted in:

- All common project task names were identified and standardised. They were also categorised into different headings, similar to a project WBS. In addition, several template WBSs were constructed for different operation functions.
- All common in-house manpower resources were identified and compiled into a standard OBS. At the same time, individual job descriptions, cost types, hour rates, work hours, calendars, etc. were also collected and combined into the standard OBS.
- All common cost codes were standardised and compiled into a standard cost breakdown structure (CBS). Meanwhile, detailed rules were laid down on how to correctly allocate a cost code to an assignment.

Many benefits came from these standardisations. First, project information was more consistent and accurate, and the collection and aggregation of useful project information was easier. Second, it helped other supporting departments to do their tasks more effectively through the receipt of consistent and accurate information (e.g. procurement, accounting, production). Third, it facilitated PM staff's understanding of any assumptions defined in a project programme through the use of a common PM language. And lastly, it made the automation of WBS, OBS, and CBS more possible.

All standardised data was stored separately in several template projects inside the computer. One program module was designed to guide the user on the use of these pre-defined templates properly. Consequently, time and cost was saved when constructing the initial project programmes, as well as the quality of the constructed project programmes being enhanced.

Automation of project analysis and problem resolving. The second step was to identify all the criteria by which a project programme was to be analysed throughout every stage of project control. Criteria were grouped under two main headings, namely PLANNING and TRACKING (see ***Table 1*** for the details of their subheadings and criteria).

All operations within these measuring criteria have been automated into a program module (see ***Figure 1***). This allows the user to analyse their project programmes more systematically and accurately, thus making better decisions for resolving problems.

Advanced tools for what-if analysis, performance summary, and project diagnosis. Three advanced tools were designed to assist the user in conducting what-if analysis, summarising project performance, and diagnosing potential problems. Their functions and advantages are described as follows:

- What-if analysis module (WIM) (see ***Figure 2***) provides five sets of project scenarios for the user to extract important scheduling data from the PMS. This allows the user to compare relevant project data while selecting a desirable project schedule (e.g. least cost, shortest duration, least resource conflict). It also provides functions for saving and retrieving data for later uses. This tool adds great power to the user for performing project scheduling, in areas such as compiling a new project schedule, and assessing project variations.
- Project performance summary module (PPSM) (see ***Figure 3***) presents all the important data required for reviewing a project's performance. These are baseline, scheduled, actual, remaining, and variance data, in relation to time, resources, cost, performance, and forecast. This tool compiles data into a summary format to give an overview of project developments. There are also two sets of histograms visually showing data comparisons. Thy add simplicity to the interpretation of the messages conveyed, thus enabling the user to spot deviations.
- Project diagnosis module (PDM) (see ***Figure 4***) uses the data generated in PPSM to create a set of narrative descriptions for indicating general project symptoms (shown in ***Table 2***). The data includes Cost Performance Index (CPI), Schedule Performance (SPI), Variance at Completion (VAC), Schedule Variance (SV), Cost Variance (CV), and resource conflict hours. These are calculated and compared by simple algorithms within the PDM. The PDM also provides two functions for storing cause and action criteria, relating to the symptoms diagnosed. These criteria are identified in connection with the user's key project operations, according to his/her past experiences. The tool provides a quick way of diagnosing project symptoms, identifying causes, and proposing actions, thereby adding great efficiency into the PMS applications.

Standardisation of PM reports. In the past, diverse project report formats were employed by different engineers. This caused difficulties in compiling documents for aspects of project control. A standard report module (SRM) (see ***Figure 5***) was developed, containing 21 standard reports for formal use. These were categorised into 6 groups, i.e. Schedules, Assignments, Resources, Costs, Performance, and Forecast. The SRM module uses a graphical symbol to show each report format and allows the user to place the mouse cursor on the symbol for previewing or printing the report. The user can define his/her own report formats and save them for specific purposes. The SRM helps unify project reports and simplify the output process, hence increasing the planner's productivity in producing and circulating project information.

Standardisation of PM procedures. The Hands-On Wizard system integrated the user's PM procedures into a flow chart diagram (shown in ***Figure 6***). This diagram contains all the program modules required for undertaking project scheduling and control. Each program module is represented by a graphical tool bar, which enable users to access the program modules easily. For example, if the user clicks the mouse cursor on the "Edit/Revise Project" tool bar, the module shown on ***Figure** 7* will appear. This allows the user to carry out any sequential tasks for editing and revising the project programme. In addition, necessary customer dialogs and input boxes have been designed to guide the user to perform his/her work properly, thus increasing user confidence in the PMS tool.

The approach has greatly simplified the procedures required for the PMS application. It has also created an operating environment in which the user operates tasks by using his/her own PM philosophy and methods. As a result, this gives the user more confidence in using the PMS package, and hence has a better chance of implementing the PMS application successfully.

Testing and Implementation

The Hands-On Wizard system has been tested by using live projects in different operational functions. The results show that considerable time and cost will be saved. The following section gives a comparison of the time required for creating the initial schedule for a simple fire protection project (63 tasks) in CA-SuperProject by using manual-driven methods and by the Hands-On Wizard procedural guidance. The test was conducted by an experienced planner. The result is shown in ***Table 3***.

The time needed by manual-driven methods is almost three times of that of the Hands-On Wizard system. In addition, this takes no account of the quality of the constructed plan, as well as the time required for producing reports, tracking progress, analysing problems, etc.

The Hands-On Wizard system has been installed in different functional areas for further testing. This will determine its performance and suitability in different operating environments, and might lead to modifications for different requirements. Nevertheless, the benefits arising from this system are manifest in many aspects of the PMS applications.

Conclusions and Acknowledgement

The benefits of developing the Hands-On Wizard system have been widely recognised by the PM staff. Meanwhile, a further development is proposed regarding the integration of project scheduling data with the cost estimates and site progress reports. This gives the authors the opportunity to further improve the capability of the Hands-On Wizard system. The authors would like to express their gratitude for the company's support, without which this work could not have progressed so successfully.

References

1. Kidd, J., 1990, "Project management software – are we being over-persuaded?", Int. J. PM, 8, 109-115.
2. Pyane, J. H., 1993, "Introducing formal project management into a traditional functionally structured organisation", Int. J. PM, 11, 239-243.
3. Ware R., 1991, "Project management software: project panacea?", J. Info. Sys. Man., 8, 79-83.
4. Yeo, K. T., 1991, "Implementing a successful IT strategy for contracting firms", Int. J. PM, 9, 34-38.
5. Skibniewski, M. J., 1990, "On the use of microcomputers by small contractors", PMJ, 21, 25-31.
6. Benjamin, C. O., Yunus, N. B., and Babcock, D. L., 1989, "A knowledge-based pedagogical system for project management", Int. J. Appl. Engng Ed., 5, 507-513.
7. Brackett, S. W. and Isbell, A. M., 1989, "PMIS: an integrated approach for the management and distribution of project information", PMJ, 20, 5-10.
8. Gilmore, M. L., 1989, "Knowledge-based systems in construction and civil engineering", Int. J. PM, 7, 147-153.
9. Yates, J. K., 1992, "A computerized inquiry feedback knowledge engineering system", AACE Trans., R.1.1-R1.7.

TABLE 1 – Measuring criteria for project analysis

PLANNING

Schedule:

Critical Tasks, Must Start, Must Finish, Milestones, Negative Floats, High Priority Tasks, Conflict Hours, Total Hours, Summary (Gantt/Table), Schedule Histogram.

Resource:

Resource Conflict, Resource Overtime, Resource Overscheduled, Overtime Hours, Underscheduled Hours, Total Hours, Resource Histogram.

Cost:

Overtime Cost, Total Cost, Task Cash Flow, Resource Cash Flow, Cost Histogram

TRACKING

Progress:

Scheduled, Should Have Started, Starting Within 10/20 Workdays, In-progress, Started Earlier In-progress, Started Late In-progress, Ahead-of-Schedule, Behind-Schedule, Behind Schedule Critical, Completed, Completed Late, Completed Earlier.

Cost:

Overbudgeted Tasks, Overbudgeted Assignments, Assignment Cash Flow, Progress Cost Summary (Gantt/Table/S-curve).

Performance:

Earned Value Analysis

Forecast:

Starting Earlier Scheduled Tasks, Starting Late Scheduled Tasks, Behind Schedule at Completion, Overbudgeted at Completion, Behind Schedule/Overbudgeted at Completion, Estimated Cost at Completion.

FIGURE 1 – Project analysis module

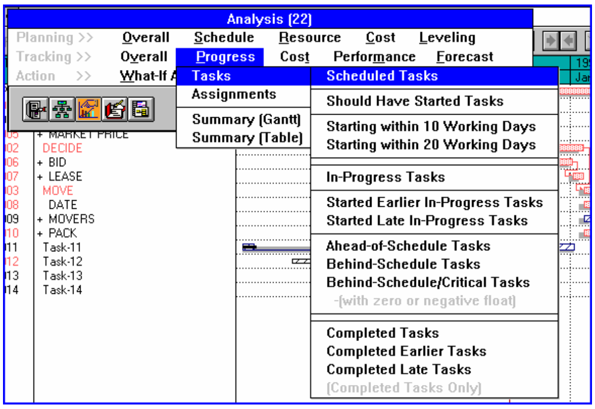
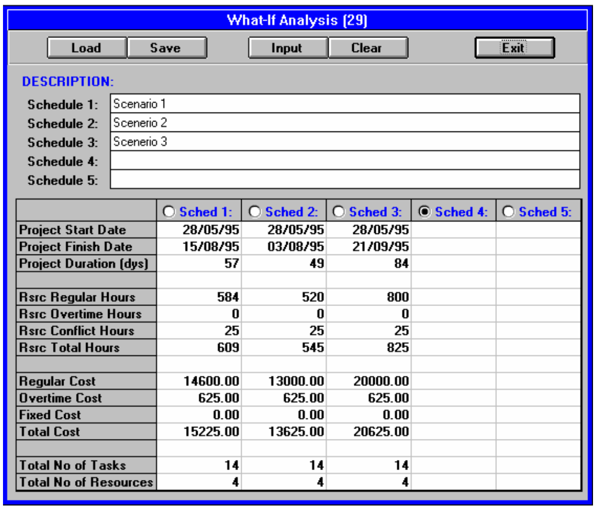


FIGURE 2 – What-if analysis model

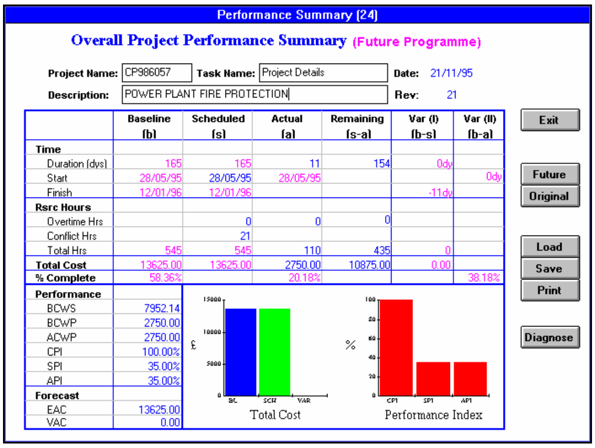


DESCRIPTION:

Schedule 1: Scenario 1
Schedule 2: Scenerio 2
Schedule 3: Scenerio 3
Schedule 4:
Schedule 5:

	○ Sched 1:	○ Sched 2:	○ Sched 3:	● Sched 4:	○ Sched 5:
Project Start Date	28/05/95	28/05/95	28/05/95		
Project Finish Date	15/08/95	03/08/95	21/09/95		
Project Duration (dys)	57	49	84		
Rsrc Regular Hours	584	520	800		
Rsrc Overtime Hours	0	0	0		
Rsrc Conflict Hours	25	25	25		
Rsrc Total Hours	609	545	825		
Regular Cost	14600.00	13000.00	20000.00		
Overtime Cost	625.00	625.00	625.00		
Fixed Cost	0.00	0.00	0.00		
Total Cost	15225.00	13625.00	20625.00		
Total No of Tasks	14	14	14		
Total No of Resources	4	4	4		

FIGURE 3 – Performance summary module



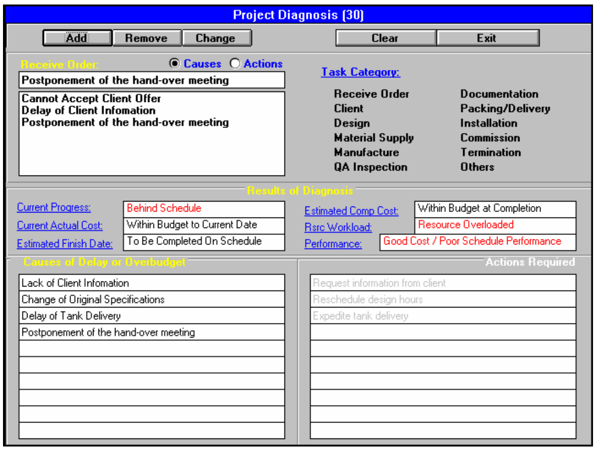
Overall Project Performance Summary (Future Programme)

Project Name: CP986057 Task Name: Project Details Date: 21/11/95
Description: POWER PLANT FIRE PROTECTION Rev: 21

	Baseline (b)	Scheduled (s)	Actual (a)	Remaining (s-a)	Var (I) (b-s)	Var (II) (b-a)
Time						
Duration (dys)	165	165	11	154	0dy	
Start	28/05/95	28/05/95	28/05/95			0dy
Finish	12/01/96	12/01/96			-11dy	
Rsrc Hours						
Overtime Hrs		0	0	0		
Conflict Hrs		21				
Total Hrs	545	545	110	435	0	
Total Cost	13625.00	13625.00	2750.00	10875.00	0.00	
% Complete	58.36%		20.18%			38.18%

Performance	
BCWS	7952.14
BCWP	2750.00
ACWP	2750.00
CPI	100.00%
SPI	35.00%
API	35.00%
Forecast	
EAC	13625.00
VAC	0.00

FIGURE 4 - Project diagnosis module



Receive Order: ● Causes ○ Actions

Postponement of the hand-over meeting

Cannot Accept Client Offer
Delay of Client Infomation
Postponement of the hand-over meeting

Task Category:

Receive Order	Documentation
Client	Packing/Delivery
Design	Installation
Material Supply	Commission
Manufacture	Termination
QA Inspection	Others

Results of Diagnosis

Current Progress:	Behind Schedule	Estimated Comp Cost:	Within Budget at Completion
Current Actual Cost:	Within Budget to Current Date	Rsrc Workload:	Resource Overloaded
Estimated Finish Date:	To Be Completed On Schedule	Performance:	Good Cost / Poor Schedule Performance

Causes of Delay or Overbudget	Actions Required
Lack of Client Infomation	Request information from client
Change of Original Specifications	Reschedule design hours
Delay of Tank Delivery	Expedite tank delivery
Postponement of the hand-over meeting	

TABLE 2 – Descriptions of project symptoms

Current Progress:
Ahead of Schedule, On Schedule, Behind Schedule.

Current Actual Cost:
Within Budget to Current Date, Equal to Budget to Current Date, Over Budget to Current Date.

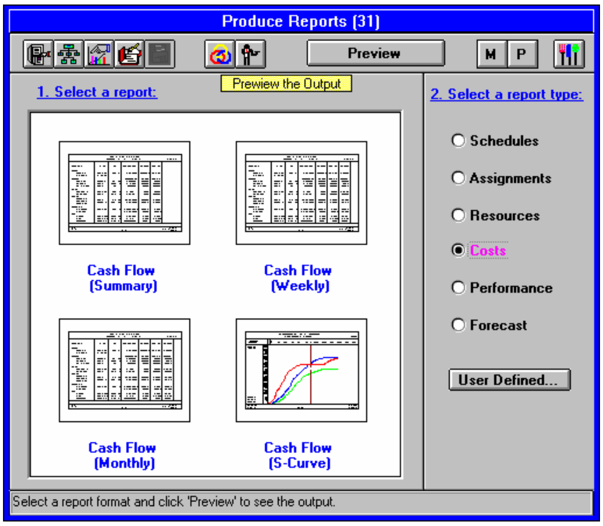
Estimated Finish Date:
To Be Completed Late, To Be Completed on Schedule, To Be Completed Earlier.

Estimated Completion Cost:
Within Budget at Completion, Equal to Budget at Completion, Over Budget at Completion.

Resource Workload:
No Overloaded Resource Hours, Resource Overloaded.

Performance:
Good Cost and Schedule Performance, Good Cost but Poor Schedule Performance, Poor Cost but Good Schedule Performance, Poor Cost and Schedule Performance.

FIGURE 5 – Standard reports module



Select a report format and click 'Preview' to see the output.

FIGURE 6 – Standard procedure flow chart

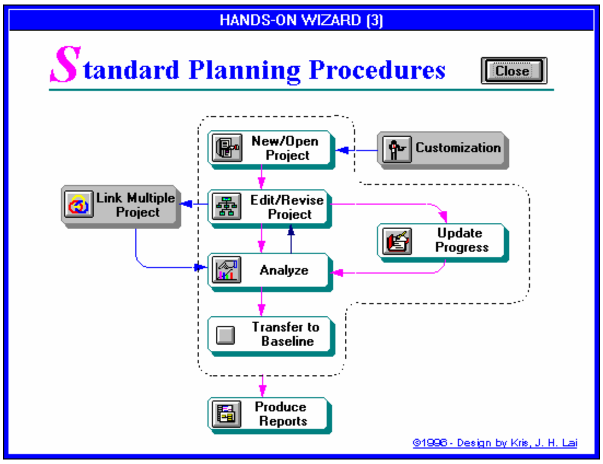
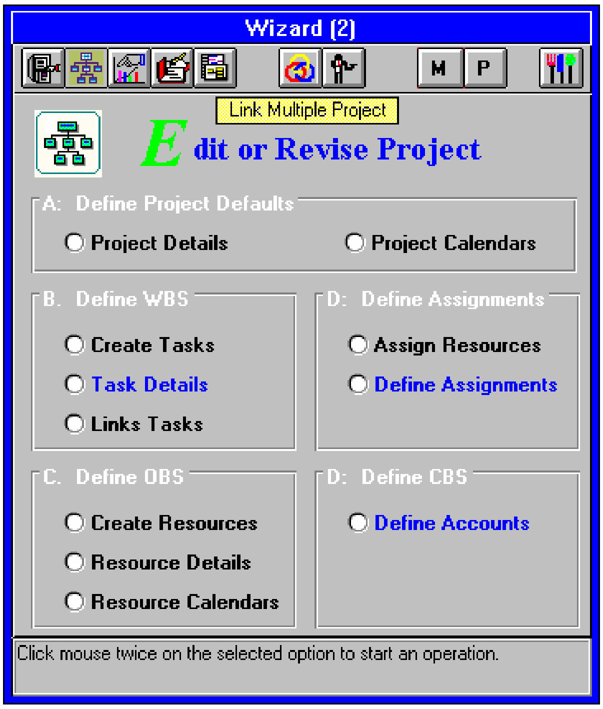


FIGURE 7 – Project editing/revising module



Click mouse twice on the selected option to start an operation.

TABLE 3 – Comparison of scheduling time by using manual-driven and procedural-driven project methods

Tasks/Methods	Manual-driven	Hands-On Wizard
Create Tasks	45-60	15-20
Define Task Details	20-30	5-10
Link Tasks	20	20
Create Resources	10	5
Define Resources	30	5
Define Calendars	20	5
Assign Resources	20	15
Define Assignments	30-40	15
Define Accounts	30	5
Total	**225-260**	**90-100**
(unit: min)		