PAMPA II ADVANCED CHARTING SYSTEM

A Thesis

by

PRABHU ANAND INBARAJAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2003

Major Subject: Computer Engineering

PAMPA II ADVANCED CHARTING SYSTEM

A Thesis

by

PRABHU ANAND INBARAJAN

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Dick B. Simmons (Co-Chair of Committee)	A. L. Narasimha Reddy (Co-Chair of Committee)
Pierce E. Cantrell (Member)	Chanan Singh (Head of Department)

December 2003

Major Subject: Computer Engineering

ABSTRACT

PAMPA II Advanced Charting System. (December 2003)

Prabhu Anand Inbarajan, B.E, Bharathiyar University

Co–Chairs of Advisory Committee: Dr. Dick B. Simmons
Dr. A. L. Narasimha Reddy

Project Management is the primary key to successful software development. In 1995 Caper Jones stated that the failure or cancellation rate of large software systems was over 20% in his article on patterns of large software systems. More than two thirds of the projects fail due to improper management of skills, activities, and personnel. One main reason is that software is not a tangible entity and is hard to visualize and hence to monitor. A manager has to be skilled in different CASE tools and technologies to track and manage a software development process successfully. The volume of results produced by these CASE tools is so huge that a high level manager cannot look into all the details. He has to get a high level picture of the project, to know where the project is heading, and if needed, then look into the finer level details by drilling down to locate and correct problems.

The objective of this thesis is to build an Advanced Charting System (ACS), which would act as a companion to PAMPA 2 (Project Attribute Monitoring and Prediction Associate) and help a manager visualize the state of a software project over a standard World Wide Web browser. The PAMPA 2 ACS will be responsible for visualizing and tracking of resources, tasks, schedules and milestones of a software project described in the plan. PAMPA 2 ACS will have the ability to depict the status of the project through a variety of graphs and charts. PAMPA 2 ACS implements a

novel charting technique called as DOT Chart to track the processes and activities of a software project.

PAMPA 2 ACS provides a multilevel view of the project status. PAMPA 2 ACS will be able to track any arbitrary plan starting from a collapsed / concise view of a whole project. This can be further drilled down to the lowest level of detail. The status can be viewed at the project version level, plan and work-breakdown levels, process, sub process, and activity level. Among all the process models, the DOT charts can be applied effectively to spiral process model where each spiral represents a project version.

To My Mom and Dad

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Dick B. Simmons, for his constant guidance, support and patience during this research work. My hours of interaction with him have helped me to understand many practical aspects of Software Process Improvement. I am deeply indebted to him for all the help that he has extended to me. I would like to thank Dr. Li Du for his extended support during my thesis. I would also, like to express my gratitude to my committee members for their cooperation and support.

I would like to thank all my colleagues in Software Process Improvement Laboratory who have helped me to enhance my knowledge through various discussions and interactions. Special mention go to Jinhwan Jung for his support on all the infrastructure related issues. I would also like to thank Mrs. Bertha Martinez for coordinating my thesis activities. Also, I would like to thank Yi Yang, who took a special interest in my research and provided valuable suggestions. Many thanks go to Prabha Acharya and Senthil Murugan for their support during the research process.

TABLE OF CONTENTS

Page

ABSTRACT iii

DEDICATION v

ACKNOWLEDGEMENTS vi

TABLE OF CONTENTS viii

LIST OF FIGURES ix

LIST OF TABLES xi

CHAPTER

I INTRODUCTION 1
- A. Software Measurement 1
- B. What are Metrics? 2
- C. Visualization 3
- D. Why Visualize Anything? 3
- E. What to Measure? 5
- F. Objectives 5
- G. Organization 6

II LITERATURE REVIEW 9
- A. Process Models 9
- B. Existing Systems 11
- C. Existing CASE Tools 12
- D. Visualization 13

III PAMPA 2 OBJECT MODEL 16
- A. PAMPA 2 Object Model 16
- B. Tracking the Plan 20

CHAPTER Page

IV STANDARD VISUALIZATION TECHNIQUES 23
A. Approach 23
B. Standard Techniques 25
1. Gantt Charts 26
2. PERT Charts 28
3. Work Breakdown Structure (WBS) Chart 29

V DOT CHARTS 33
A. Life Cycle Models 33
1. Waterfall Model 34
2. Spiral Lifecycle Model 34
3. Incremental Lifecycle Model 36
B. Dot Chart Visualization 38

VI ACS SYSTEM ARCHITECTURE AND DESIGN 43
A. Architectural Design 43
B. External Interfaces 46
C. Database Design 47

VII IMPLEMENTATION 51
A. Application Components 51
B. Visualization Engine 52
C. Data Gatherers 54
D. Data Broker 55

VIII EVALUATION, CONCLUSION AND SUMMARY 57
A. Limitations of Traditional CASE Tools 57
B. PAMPA 2 ACS Comparison 59
C. Case Study - Web Development Project 61
D. Problem Solving Procedure 64
E. Future Work 94
F. Conclusion 96

REFERENCES 98

VITA 105

LIST OF FIGURES

FIGURE		Page
1	PAMPA II Object Model	17
2	PAMPA Visualization Framework	24
3	PAMPA II Work Breakdown Structure Chart	31
4	Waterfall Model	35
5	Spiral Model	37
6	PAMPA 2 Dot Charts	40
7	Three Tier Architecture	44
8	PAMPA 2 ACS Architecture	45
9	PAMPA 2 Entity Relationship Diagram	49
10	Manager's Problem Solving Routine	63
11	Importing Microsoft Project Plan from Microsoft Project	66
12	Baseline Project	67
13	Viewing the Activities at Summary Level	69
14	Viewing the Activities at Detailed Level	70
15	Gantt Chart	72
16	Drilldown Gantt Chart	74
17	Cumulative Cost Chart	75
18	Summary Earned Value Chart	77
19	Detailed Earned Value Chart	78

FIGURE Page

20 PAMPA 2 Activity Network Chart 80

21 Reschedule Activity Screen 81

22 DOT Chart Displaying Version Status for a Project Lagging Behind Schedule 83

23 DOT Chart Displaying Version Status for a Healthy Project 84

24 DOT Chart Showing the Status of Different Teams in a Project Lagging Behind Schedule 86

25 DOT Chart Showing the Status of Different Teams in a Healthy Project 87

26 DOT Chart Displaying the Activities View for Project Following Spiral Model 89

27 DOT Chart Displaying the Activities View for Project Following Waterfall Model 90

28 Detailed Resource View Chart 92

29 Line Chart Parameter Configuration 93

30 Line chart Indicating Activity Progress 95

LIST OF TABLES

TABLE Page

I PAMPA 2 ACS DATABASE TABLES LIST 50

II TECHNICAL DETAILS 51

CHAPTER I

INTRODUCTION

Project Management is a key tool to successful software development. Weller [1] argues that many software projects fail because of lack of measures to track and control the processes involved in software development. In 1995, Caper Jones reported that the failure or cancellation rate of large software systems was over 20% [2]. More than two thirds of the projects fail due to improper management of skills, activities, and personnel. Of the 80% that are completed, approximately two thirds are late and experience cost overruns as much as 100%. Roughly two thirds have reliability and quality issues in the first year of development. Simmons states that projects failed either directly or indirectly from management problems [3]. The above projects failed because of breakdowns in their software development process and their inability to measure the process. The Management was not able to visualize the software systems that were being developed. In the case of failed projects, they did not know the status of the systems until it was too late to correct.

A. Software Measurement

Successful project management requires not only a good foresight, but also the ability to measure the software development process. A software development process can be measured and can continuously be improved. If only we can measure the key attributes of a software process, we will be able to have a more precise, predictable and repeatable control over the software development process. Software metrics is a part of software engineering that studies software artifacts in order to quantify

---

The journal model is *IEEE Transactions on Automatic Control.*

them by assigning meaningful values. It is the study of software related artifacts for the purpose of characterization, prediction, control, management or improvement through quantitative or qualitative analysis.

B. What are Metrics?

Fenton defined measurement [4] as *"the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules."* Software measurement is the process that measures the characteristics of a software product. There are five major objectives for software measurement

- To gather information about the current state of software product, project or process
- To provide a quantitative measure of the status of activities carried out in a project
- To assess the impact of process changes
- To track the progress of an organization towards its process improvement goals
- To allow managers to make timely, data driven decisions

Software Measurement not only enables us to estimate the cost and effort devoted to a project. It also helps us to plan, justify, track and control the cost and effort involved in the development process by giving quantitative measures of progress, thereby preventing overruns in cost and schedules. Estimating, measuring, and controlling are essential ingredients of a successful software project. Estimates determine feasible relationships among the work to be done, the schedule for doing it, and the resources

needed; measurement provides the data needed to determine whether a project is on schedule and whether the work products are of acceptable scope and quality; and control is exerted when actual values of project factors, product attributes, and risk indicators deviate from estimated or required values.

C. Visualization

Metrics so gathered in a project can either be presented directly or it can be transformed to an equivalent form that could be assimilated/understood more easily. Although large volumes of metrics are collected, much of it remains under utilized [5]. The volume of data makes it infeasible to read textually, and its lack of structure frustrates the analysis tools. New visual metaphors can simplify the process of extracting information and presenting it to users in a better form. This can be achieved by visualization, which is the graphical representation of metrics data. Visualization is as much as important as metrics. Einstein emphasized the importance of visualization to his own working methods when he said *"The words of language as they are spoken or written do not seem to play any role in the mechanism of thought. The psychical entities that serve as elements in thought are certain signs which can be voluntarily reproduced and combined."* It is empowering even for the most experienced project manager, to comprehend the complex project management process with a complete understanding of every piece, and an ability to visualize how each one fits into the overall project picture.

D. Why Visualize Anything?

Forsberg, Cotterman and Mooz have identified the essential characteristics of visualization [6] in their book on Visualizing Project Management.

- Visualization explains how things work. Abstractions can be personalized and explained by visualizations. In a graphic object is equivalent to many words of text.
- Visualization helps us assimilate a lot of information in a lesser time.
- Visualization broadens our perspective and provides a conceptual frame of reference just as the common vocabulary does for communications.
- Visualization can express rules and their implications in a more simple manner
- Visualizing a model clarifies relationships, helps to identify key element and consciously eliminate confusion factors.

A Software project is more difficult to visualize, monitor and control than other types of projects. This is mainly because a software project does not produce a tangible product that we can see or feel. Using regular tabular views for managing large software projects is not effective. For example the NASA Jet Propulsion Laboratory (JPL) [7] project had more than 10,000 activities, 38 deliverables and 10-phase life cycle model. Such huge projects are really difficult to track and control because of the huge amount of information involved. The underlying software project problems can get the attention of project managers if they can be visualized and depicted graphically. Even when graphic representations are used, conventional charting techniques like Gantt charts and activity networks might not be very effective because of their linear nature for a high-level project manager because he has to trace all the project activities to know the project status. An effective visual representation should roll up summarize, and present appropriate information. A survey of 600 firms has indicated that 35 percent of them had at least one runaway software project [8]. The frequency of these software project disasters is a serious concern. Most of these problems could

have been avoided or strongly reduced if there had been an explicit early concern for identifying and resolving their high-risk elements. This can be achieved with effective visual representation of project status.

E. What to Measure?

There might be many stages in a software development process based on the type of lifecycle model that is being followed. Each stage will produce one or more artifacts that can be measured. Among all these stages, gathering the requirements and planning the project activities are the most critical areas as these two stages lay the foundation for the rest of the project. Project planning is not restricted to one specific phase of the project. It spans through the whole software development lifecycle, therefore requiring constant attention of the project manager and needs to be modified and controlled by the manager carefully according to the current status of the project. A manager has to continuously monitor for such changes, track them and control them if necessary. For a complex project lifecycle with a lot of activities, it is an enormous task to gather the metrics, study them and take suitable measures. We believe that it would be very useful for a high level manager to have a tool which would help him visualize such a dynamic environment and assist him in the process of decision making. We will develop one such tool called PAMPA 2 Advanced charting system to help the manager track and control the project effectively.

F. Objectives

This thesis will have the following objectives

- To develop advanced charting mechanisms to visualize and track the planning cycle of software projects.

- To develop the capability to identify the plan based on its life cycle model (Waterfall, Spiral).
- To develop DOT charts that could be drilled down /collapsed at any level to provide a fish eye's view of the project at that level and track the plan using DOT charts.
- To develop conventional charts like Gantt charts, PERT/CPM charts as complementary views for DOT charts

The output of this thesis would be an Advanced Charting System, which will act as a companion to PAMPA 2 (Project Attribute Monitoring and Prediction Associate) and help a manager in visualizing the status of a software project over a standard worldwide web browser. The system implements a novel charting technique called as DOT Chart to track the processes and activities of a software project. Unlike a Gantt chart, which provides only two views (Summary and Interactive), Dot charts provide a multilevel view of the project and are more interactive. A manager can drill down at any level into the DOT Chart to view the status.

G. Organization

This thesis is organized in the following way.

Chapter 1 introduces the concepts, the main problem and the general solution proposed for the problem. We explain how management has difficulty in gathering metrics about process and visualizing the metrics so gathered during the software development process. Examples are presented where industry has learned how to improve the software development process by applying software improvement methodology.

Chapter 2 provides the background study for the thesis and the related work done in

this area so far. We classify the related work into four major sections. First section deals with prior work related to lifecycle models and planning. The second section deals with existing systems developed in this area. The third section deals with the CASE tools that will be used by the Advanced Charting System. The fourth section deals with related work on Information Visualization.

Chapter 3 provides an insight to the PAMPA 2 framework. We describe the related entities in the object model. We explain how we gather the metrics identified by different entities of the PAMPA 2 object model. We identify the CASE tools that we can use to gather the plan data. Then we explain how we can use these CASE tools to unobtrusively gather data from the software development process and populate the PAMPA 2 object model.

Chapter 4 explains the standard charting techniques used for visualization. We describe these charting techniques and explain how these techniques are incorporated as views into the PAMPA 2 ACS. Some of these views include Personal Gantt Charts, Organizational Gantt Charts, Summary and Detailed Gantt Charts, Activity Networks, Organizational and Personal Activity Networks and Work Breakdown Structure Charts.

Chapter 5 explains the idea behind DOT charts. We explain the history and usefulness of Dot Charts. We explain the Dot chart components and the navigation of Dot charts. We also explain how dot charts are related to other views such as Gantt Charts and Activity networks.

Chapter 6 explains the system design and architecture. We explain the extensible architecture of PAMPA 2 ACS. We also explain the techniques used in integrating the various CASE tools used with the ACS framework. We explain the data flow among ACS components.

Chapter 7 explains the implementation details behind ACS. We describe the technol-

ogy used for implementing ACS and its architecture.

Chapter 8 compares ACS with other case tools available in the market. We present a sample application of PAMPA 2 ACS for tracking a software project to support this argument. We explain how future changes and extensions could be easily accommodated into the PAMPA 2 framework. We present a set of possible future extensions that can be added to this framework. We then provide a summary and conclusion for the work.

CHAPTER II

LITERATURE REVIEW

This thesis proposes an extension to PAMPA 2 known as Advanced Charting System. With PAMPA 2 ACS a manager would be able to track a project over the Internet using a standard web browser. The following sections will discuss the motivation for PAMPA 2 ACS and related work. The first section discusses the related work in the area of software project planning that led to the development of PAMPA 2 system. The second section discusses the tools that were developed to support the PAMPA 2 framework. These tools will be used by the PAMPA 2 ACS to visualize and track a software project. The third section discusses the existing CASE tools available in the market with which PAMPA 2 ACS can integrate to gather project data unobtrusively. The last section explores the prior work in the area of information visualization.

A. Process Models

A process model defines the desired phases or activities in a project. It describes in detail about the phases to be followed in a project, inputs required for those phases and the desired output artifacts of every phase so that the processes could be measured. By following a pre-specified model, a project should increase the quality of its process such as repeatability, manageability and measurability. The interim work products are more clearly defined and justified. Activities are less likely to be forgotten. For measuring a project, process models are very important. We cannot measure a project that does not follow a specific process model because we do not know what to measure. Mills and Dyson point out that *"you can't control what you can't measure. That fundamental reality underlies the importance of software metrics"* [9]. Standardized guidelines can be used as a model for plan definitions and

formal document format. The IEEE Std 1058.1 [10] describes the format and content of software project management plans. The IEEE [11] has proposed a new version of the same standard, which is planned for release in 2002. IEEE also issued the "STD 12207.1-1997 Standard for software lifecycle processes" [12] for defining life cycle processes. Support tools are available for managers to make a software project plan. Barry Boehm proposed his famous spiral model [13] as a risk driven approach for guiding the software development process. Spiral model is based on the incremental evolutionary model, where each cycle is a mini waterfall and as a result each cycle ends with a usable system. It gives a better visibility of progress and early visibility of risks. The difference between spiral and incremental model is that in spiral model each cycle produces some artifact to be evaluated, but not necessarily a usable system. The management of risk is built into the model because each cycle ends only after passing through a quality gateway. Boehm also emphasized the need for formal risk management in software development in his paper on Software Risk Estimation: Practices and Principles [5, 14]. This paper also elaborates the steps involved in risk management, which involves risk identification, analysis, prioritization, planning, resolution and monitoring. It also defines the formal templates for the above-mentioned steps. The Rational Unified Process (RUP) [15] model incorporates formal methods to track the plan by assessing the risk associated with the Use Cases [16, 17]. These methods can be effectively applied to the spiral process model [13]. Simmons et al. have discussed extensively about knowledge-based approach for project planning and tracking in their paper on Software Project Planning and Associate [18]. These papers mainly discuss the PAMPA 2 framework, its support for tracking plan, and the attributes to be gathered using this framework, which would be helpful in tracking a plan. The prior work in these areas provides the motivation to develop a tool that can visualize the planning phase of a project.

B. Existing Systems

Researchers believe that modern technology can be applied to software projects to help managers continuously visualize the status of a project. Actions can be taken to early to help head off problems before they cause major modifications to plan. In 1992, Simmons emphasized a metric based management approach to help a manager visualize and understand the software development process [19]. He described a Computerized Life Cycle Advising, Monitoring and Predicting (CLAMP) [20] tool that was created by Richard Chen. In 1993 Simmons proposed a tool called Manager Associate [21] that can assist managers in planning, organizing, staffing, scheduling, measurement, visualization and control. The Manager Associate helps the managers anticipate problems that allow early corrective action. As a result, software projects can be described on time, within budget, and to customer satisfaction. A client server version of the PAMPA (Project Attribute Monitoring and Prediction Associate) framework was built in 1997 [3] to help gather project metrics from any software development environment, save it in an understandable object/attribute/relationship format, view it using an inexpensive workstation, and supply to expert system building tools used for creating intelligent agents. The PAMPA framework can gather project information from any software development project that can share directories over a network to a Microsoft Windows client workstation. Along with this, a client server version of the PAMPA visualization toolkit was created to help managers visualize software projects. This toolkit provided the basic visualization for the project metrics. This version of the PAMPA tools is available in the book on software measurement by Simmons, Ellis, Fujihira and Kuo [3]. Simmons et al. developed the Software Project Planning Associate (SPAA) for supporting knowledge based approach for dynamic software project planning and tracking [18]. SPPA was

developed according to a Knowledge Base Plan model to allow the manager to keep track of the software plan component. A prototype version of PAMPA 2 based on three-tier architecture was developed in 2001 to track the Software Engineering class project at the Department of Computer Science, Texas A&M University. The system proposed in this thesis is an extension of PAMPA 2 framework. It would provide an enhanced web based version where a manager would be able to track a project over the Internet using a standard web browser. It would also provide extensive charting capabilities that were not available in the client server version of PAMPA and thus assist the manager in visualizing the system better. PAMPA 2 ACS will identify the attributes essential for tracking a project from the earlier research work mentioned in this section.

C. Existing CASE Tools

CASE tools have played an important role in the area of project management. PAMPA 2 ACS will be using the useful features available in some of the CASE tools mentioned below and extend them to overcome the limitations posed by these CASE tools. Rational Corporation provides a suite of tools for requirements capturing, design, testing, configuration management and defect tracking. Rational provides a project management tool called as the Rational Project Console [22] which gathers metrics from the rational case tools and graphically depicts the collected metrics using different charting methods like dials, gauges, pie charts. Rational Corporation has also proposed a new Life Cycle model called as Rational Unified Process [15], which is a collection of industry standard best practices. These tools [23] act as valuable information sources from which facts about the project could be captured. Microsoft Corporation provides a project management tool called Microsoft Project, which is

used to create software project plans [24]. The Rational and Microsoft tools in their natural form are proprietary and do not allow external systems to share their knowledge. This is a big restriction when we have to extend or customize the functionality provided by these tools. The system proposed for this thesis provides a framework to integrate with all the above-mentioned case tools, gather their knowledge through data gatherers and use that knowledge to achieve the objectives of the PAMPA 2 system.

D. Visualization

Visualization provides an interface between two powerful information processing systems- the human mind and modern computer. Gershon [25] defines visualization as the process of transforming data, information and knowledge into visual form making use of humans' normal visual capabilities . Kiem [26] suggests that the basic idea of visualization is to present data in some visual form, allowing humans to get insight into the data and draw conclusions. With effective visual interfaces we can interact with large volumes of data rapidly and effectively discover hidden characteristics, patterns and trends. According to Grinstein and Ward [27], the purpose of visualization is not to replace solid quantitative analysis, but instead allow the quantitative analysis to be focused. Tegarden [28] defines the basic criteria for any visualization system as follows:

- Exploit human visual system to extract information from data
- Provide an overview of complex datasets
- Identify structures, patterns, trends, anomalies, and relationships in data
- Assist identifying the areas of "interest"

Though these CASE tools provide some visualization to the project data to a partial extent, they have some inherent disadvantages. They do not share the knowledge present in them with external tools and application. As a result, we need to have different user interfaces for different CASE tools. These CASE tools produce graphic charts that are not interactive. Studies in the area of Human Computer Interaction shows that interactive charts are more efficient than their static counter parts. Nakakoji, Takashima and Yamamoto [29] show that interactive visualization system has a better cognitive impact on the user and helps the user to understand the represented information better and faster. Interactive visualization systems, by definition, not only present information, but also allow the user to interact with the information so presented. Such a system can present the correct amount of information in every visual representation and expose further information on demand through navigational cues. These systems let the user see the information rolled up or drilled down to the last level of detail. The Gantt charts and activity networks produced by conventional CASE tools such as Microsoft Project [24] are static in nature. They do not let the user interact with the chart and get further information. Since the CASE tools themselves cannot be modified, an alternate solution is to generate these visualizations external to the tool. Also the conventional CASE tools enforce software requirements on the client side. As a result, these charts and visualizations can be viewed only from workstations where the tools are installed. The widespread popularity of web technology has created new information visualization technology model, in which browsers enable the widespread distribution of information using standard html techniques. Jern's [30] studies show that the explosive growth of web has changed user expectations concerning the delivery of information to the client. In the traditional information visualization approach any client to communicate as a peer to any available server. The data gets executed on the client to render the

visualization. Web introduces a new model in which the client GUI, based on html, is less functional and relies upon the data or application servers for visualization traditionally executed on the client. Keim's [31] study in data visualization shows that the visual data exploration follows a three-step process: Overview first, zoom and filter, and then details on demand (also known as information seeking). First, the user needs to get an overview of the data. In the overview, the user identifies interesting patterns and focuses on one or more of them. For analyzing the patterns, the user needs to drill down and access details of the data. Visualization technology can be used for all three steps of data exploration process. This study was conducted based on empirical data, user surveys, and user interaction pattern experiments. The results of this study could be used as an effective guide for designing our data visualization on project management data.

The prior work in these areas provides a strong motivation and guideline for development of PAMPA 2 ACS as an advanced visualization system for project management.

CHAPTER III

PAMPA 2 OBJECT MODEL

In 1997 the PAMPA tool was developed to help managers gather project information from any software development environment. The data so gathered would be stored in the PAMPA knowledgebase and can then be viewed using custom viewers from inexpensive workstation. The data can also be supplied to expert system building tools to create intelligent agents [18]. Later, new features were added to PAMPA to create version PAMPA 2. Some of the significant features include the ability to run from Internet browsers and adapting a three-tier architecture. PAMPA 2 framework monitors, controls and tracks a software project by gathering data from the project, calculates the attributes and presents the status of the project to the manager. Objects along with their attributes, relationships and properties are saved in the PAMPA 2 knowledgebase. The advanced charting system proposed for this thesis extends the PAMPA 2 framework and provides visualization for the plan tracking project attributes of PAMPA 2 knowledgebase. Using ACS management reports and charts can be generated from the knowledgebase.

A. PAMPA 2 Object Model

The PAMPA 2 object model extends the PAMPA project knowledge base to include project version, plan and milestone object classes [32]. Fig 1 shows the PAMPA 2 object model that includes the expanded object classes. The shaded portions in the figure indicate the entities we will use to track an arbitrary plan. PAMPA 2 knowledgebase reflects all attributes and relationships of a software *Project*. A *ProjectList* is made of multiple *Projects*. Each Project contains multiple *ProjectVersion*s.

Each of these *ProjectVersion*s is mapped to a *SLCModel* (and vice versa) and is

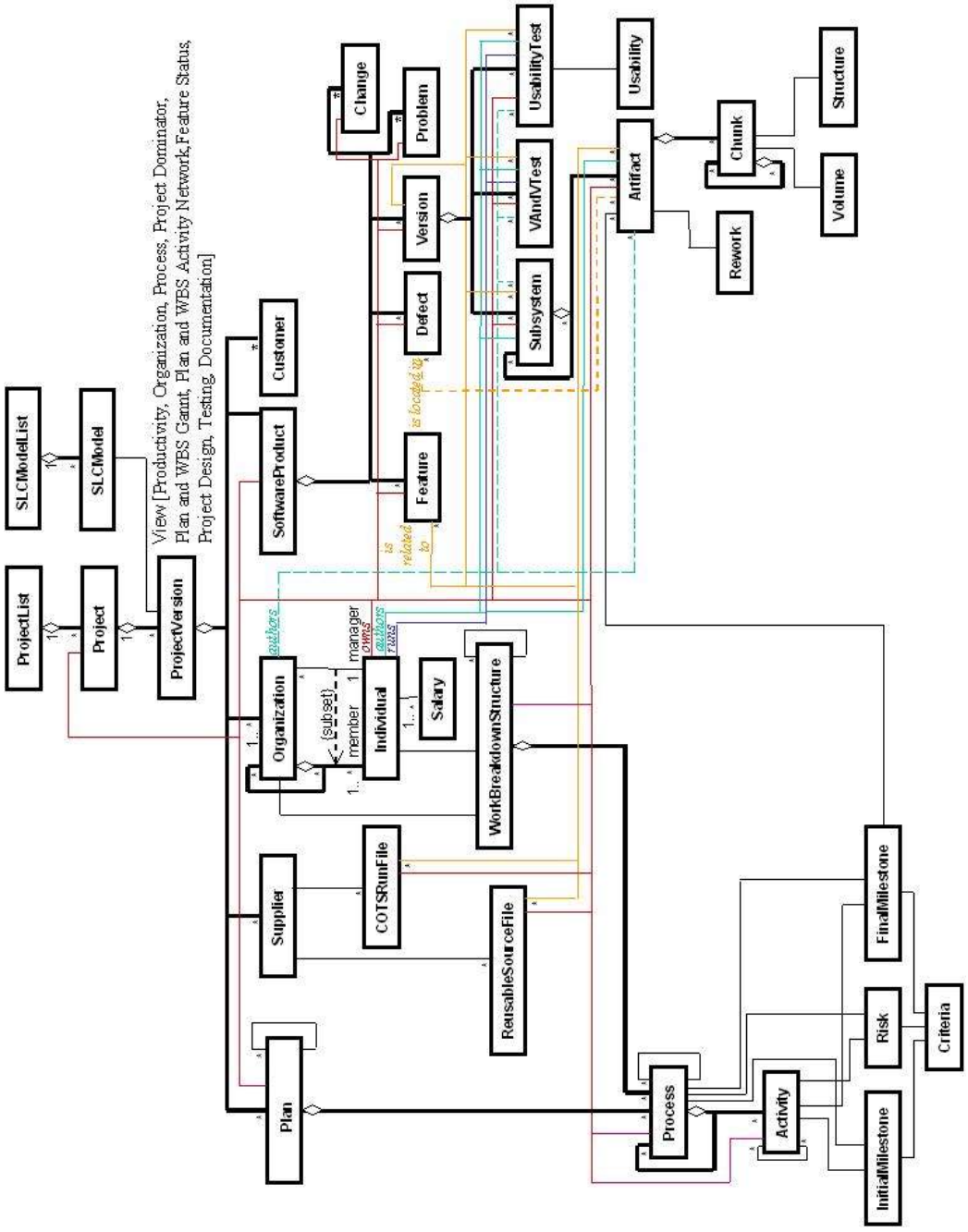


Fig. 1. PAMPA II Object Model

also mapped to zero or more *Plan*s. *SLCModel* class lets us track an arbitrary plan by identifying the lifecycle model from a given plan. Currently we support waterfall, incremental and spiral lifecycle models. But this can be extended to provide support for new lifecycle models such as the eXtreme Programming model [33] or the Rational Unified Process model [15]. Support for new lifecycle models can be provided by defining the attributes of the new lifecycle model in the knowledgebase. The *ProjectVersion* class allows the *Plan* to be modified after a project has started. Each modified project *Plan* results in a new *ProjectVersion*. A *ProjectVersion* is linked to the *WorkBreakDownStructure* through the Individuals and the Organization. The *ProjectVersion* class allows tracking of incremental and spiral life cycle models.

A *Plan* contains the attributes of a planned software project. This class can be used to predict *Volume, Reliability and Usability* attributes of the software product. These planned attributes can then later be used in deciding the *Criteria* for completion of *Activities*. The *Plan* is linked to the *Process*, which in turn is made up of *Activities* and *Milestones*. When planning for a software product, we have to estimate cost and schedule. The cost involves paying for human resources and other hardware and software assets required for developing the product. The *Salary* class calculates those attributes from the *Organization* and *Individual*s. The average salary of *Individual*s that make up the project Organization is estimated. Once the effort (the time required for developing the software product), is estimated, we can compute the schedule and resource assignments.

A *Process* describes how a project intends to create the *SoftwareProduct*. Each *Activity* is tied to a *Milestone* and *Criteria*. An *Organization* is composed of other *Organization*s or *Individual*s and has a *WorkBreakdownStructure* assigned to it. These details can be captured from a standard project-planning tool like Microsoft project [24]. The data so captured will be stored in a relational database. This data can be

used for later analysis and tracking. An *Activity* may be composed of other Activities or it can be the lowest level of work item. Each *Activity* will have an *InitialMilestone* and a *FinalMileStone*. The *InitialMilestone* and *FinalMilestone*'s are linked to the *Criteria*, which determines whether the milestones have been achieved. The Activity is also tied to a *Risk*, which can be determined quantitatively. By setting the optimal levels for the *Risk* we will be able to control the risk associated by taking alternate measures when the current assessment of the risk parameter exceeds the optimal limits.

The above framework consists of three subsystems.

- Planning
- Group / Personnel
- Software Product

Planning subsystem maintains the plan and the entities related to the plan such as processes, sub-processes, activities, milestones, risk and criteria. Personnel subsystem maintains personnel information related to the project for grouping and staffing (skill, salary, experience, task assignments etc). Software Product subsystem contains data about the project/product developed and the related artifacts that are produced in the process. The knowledgebase also analyzes the artifacts and populates the metadata about the software product. This metadata can also be called as metrics about the software product. These metrics include volume, structure, complexity, usability and reliability.

PAMPA 2 framework compares these metrics against their planned values to determine the status of a software project. The planned values for these metrics can be obtained from the planning subsystem. This comparison serves as a criteria

for completion of milestones. The facts *Salary* and average number of *Individual*s are used to compute actual *Cost*, which will be compared against the planned cost. PAMPA 2 framework determines when an *InitialMilestone* initiates an *Activity* or *Process* and when a *FinalMileStone* terminates one. During all phases of a software development *Project*, snapshots are taken and saved as *ProjectVersion*s. At anytime during a project or after a *Project* is complete, progress of the *Process*es and *Activities* can be reconstructed and analyzed using archived *ProjectVersion*s.

B. Tracking the Plan

To deliver a project successfully, a manager must plan and control activities according to the plan. PAMPA 2 object model can be used to track and control a project plan. The planning phase involves four major activities

- Creation
- Scheduling
- Tracking
- Controlling

The objective of the creation activity is to construct a plan containing activities, milestones and artifacts. Scheduling is the process of including a timeline to the activities. A schedule shows the start and finish times and the relationship between activities. The first two phases involves defining of goals and activities for the project, then estimating durations, risks and assigning suitable resources - people, equipment, facilities and materials to each activity. When the project is underway, there is a need to monitor and control the progress of every activity.

To effectively track and control the project, relevant charts and graphs should be generated from the actual start and finish times and cost. CASE tools like Microsoft Project [24] can be used to make project plans. The plans made using such CASE tools can be gathered into the PAMPA 2 knowledge base. With this input, the PAMPA 2 framework will be able to track projects. The major advantage of this approach is that we are abstracting the CASE tools from the knowledgebase and hence we will be able to use this approach with any standard project management tool in any platform. This platform and CASE tool independence provides an additional level of flexibility. There are some drawbacks/limitations in using conventional CASE tools to track plans. They are as follows:

- Cost of the CASE tools are affiliated to one type of process model, For example, Rational suite of products support Rational Unified Process model very well but provides little support for a waterfall based model. Microsoft Project supports waterfall model extensively but provides little support for other life cycle models. As a result we loose the ability to track arbitrary plans using a single tool. There is a need for a generic tool that can trace arbitrary plan.

- Microsoft Project lacks the capability to address multiple project versions and life cycle model association. It does not distinguish between processes and activities.

- Most of the CASE tools are standalone and platform dependent. We do not have a browser-based version of CASE tools. We might get into the issues of Software installation, licensing and mobility of client/server, middleware and servers if we are to use a standalone CASE tool. PAMPA 2 is a platform independent and browser-based plan tracking system. Using this system, a manager will be

able to track the project status from any location, with an Internet connection and a standard Internet browser.

- Projects happen in dynamic environments and if a project falls behind schedule, it is important that decisions are made quickly to identify alternative course of action to modify the plan. The standard project management tools available in the market lack the capability to dynamically track changes in plans. As a result these tools cannot dynamically assess the risk created because of changes in plans.

PAMPA 2 ACS addresses all these issues by isolating the presentation aspect from the CASE tools. PAMPA 2 gathers data from the CASE tool unobtrusively and generates graphs and charts external to these tools, thereby separates the presentation tier from the data tier and stores the data in a knowledgebase, which can then be used for analysis. It can be presented over an Internet browser, thereby making them wide area compatible. Thus the planners get to use the planning tools of their choice but still overcome the limitation posed by traditional planning tools. PAMPA 2 ACS provides a platform independent framework for project tracking. PAMPA 2 framework's extended object model tracks an arbitrary lifecycle and also provides the manager ways to define new lifecycle models without making changes to the framework. The agent based approach followed by the PAMPA 2 framework allows to monitor and track the projects continuously for changes.

Since PAMPA 2 ACS gathers data automatically from the CASE tools without external inputs, it avoids delays due to human factors. Since there are no human inputs, the analysis results in objective determination of metrics. This leads to more accurate prediction. It also improves the productivity as it saves a lot of time, which would have otherwise been spent on gathering and analyzing these metrics.

CHAPTER IV

STANDARD VISUALIZATION TECHNIQUES

PAMPA 2 ACS extends then PAMPA 2 framework to provide visualization of project data. Some of the standard visualization techniques to track plans include Gantt Charts, Tracking Gantt Charts, Personal Gantt Charts and PERT/CPM charts. ACS also provides custom visualization techniques such as DOT charts. DOT charts are handled in detail in the next chapter. This chapter describes the standard visualization techniques provided by the PAMPA 2 framework.

A. Approach

Fig 2 describes the highlevel architecture of the PAMPA 2 ACS framework. The project manager creates a project plan using CASE tools of his choice (1). PAMPA 2 data gatherers are used to gather the plan details from the CASE tools (2). The PAMPA 2 knowledgebase analyzes the data (3), inputs the result to the chart engine (4) and generates interactive Gantt charts and Activity Networks. These charts are published to a standard web browser (5). This approach has several benefits.

- This provides the project managers the ability to monitor the project status using any web browser, with minimum client side requirements.
- The schedule so captured can be analyzed and process effectiveness predictions can be made based on the compliance of the actual schedule to the planned schedule.
- The interactive charts provides the right amount of information and lets the manager interact with the charts to get more detail by either zooming out

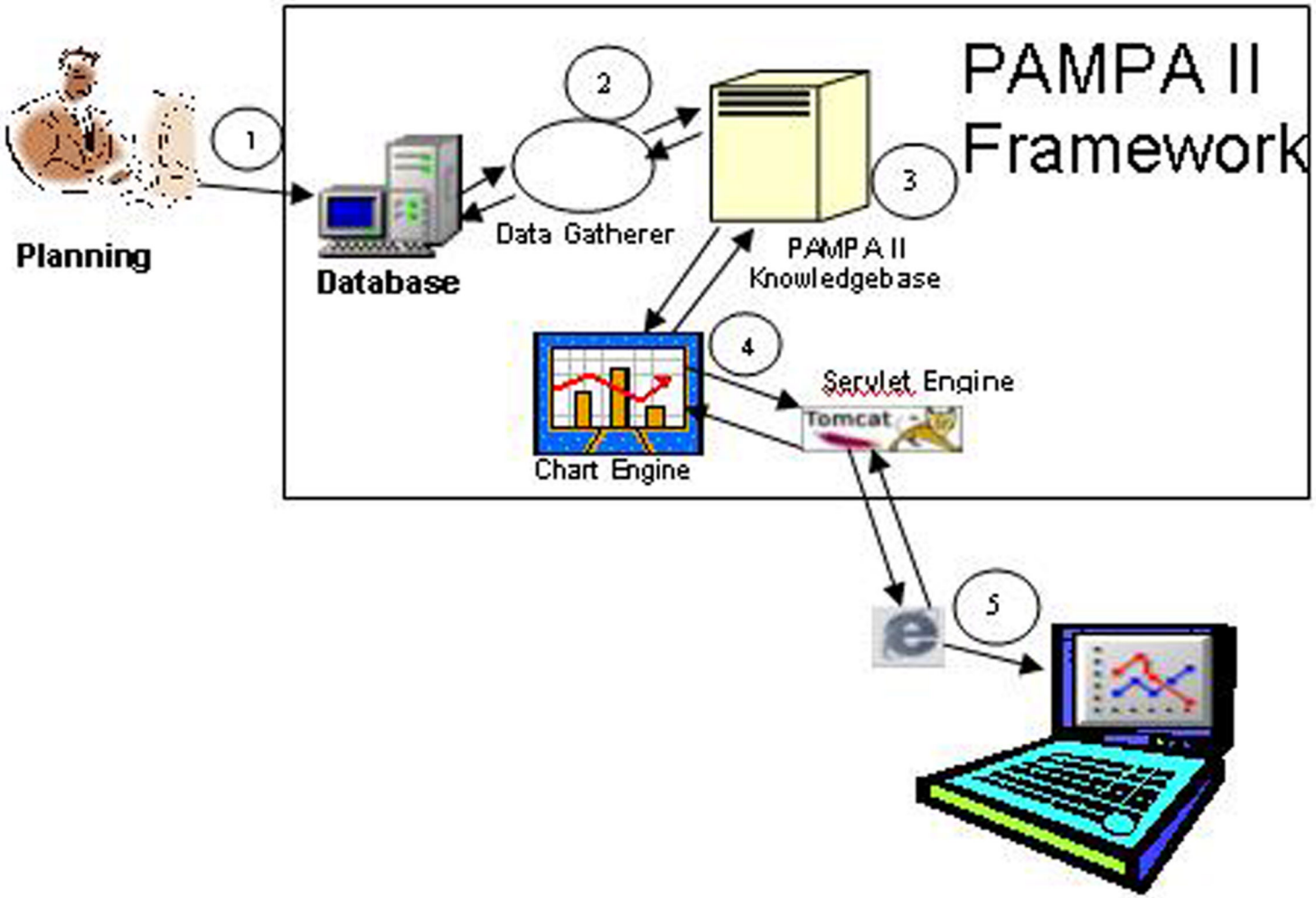


Fig. 2. PAMPA Visualization Framework

or drilling down further. Thereby provides much easier assimilation of the information presented.

B. Standard Techniques

We would be incorporating the following views in the PAMPA 2 framework. Some of the charts mentioned below are conventional project management charts, which are used to track plans. We will include them as a part of the system to be developed for making PAMPA 2 as a comprehensive project management tool. The charts generated by PAMPA 2 ACS are listed below.

- WBS Chart
- Gantt Charts
- Tracking Gantt Chart
- Personal Gantt Chart
- Cumulative Cost Chart
- Earned Value Chart
- PERT/CPM Charts
- DOT Charts

WBS gives a graphical view of the plan. It decomposes the project into hierarchically structured well-defined manageable task or activities.

The Gantt chart (named after its developer Henry L. Gantt) is the standard format for displaying a schedule graphically. It consists of a horizontal bar chart with time as the horizontal axis and either resources, activities, or orders as the vertical

axis. Individual operations are displayed as horizontal bars in the chart, indicating the time at which the activity begins and ends. Many variations on the Gantt chart exist to display additional information. Gantt charts are described in detail later in this section.

The Special Projects Office of the US Navy developed the Project Evaluation and Review Technique (PERT) during the Polaris missile program in 1958 and 1959 [34, 35]. PERT charts present a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project. These nodes are linked by labeled vectors (directional lines) representing tasks in the project.

In PAMPA 2 system, the manager would be able to navigate between these views through embedded links within the charts. By clicking an activity in a Gantt chart, the manager would be able to get the corresponding CPM chart or the activity information in tabular format and vice versa.

1. Gantt Charts

A Gantt chart maps project tasks against time [34, 35]. McLeod and Smith [34] explain that Gantt charts were developed and first used by Henri Gantt for military purposes in 1903. Gantt charts were the first formal, deterministic system of scheduling. Anton [36] explains that a Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a vertical axis representing the activities that make up the project. Project activities, depicted as bars, are mapped against their scheduled start and finish times on the chart. The timing, sequences and the time span for each task is represented in the horizontal axis as bars of varying length, which may overlap. Tracking Gantt Charts display the planned vs. actual activity schedule.

Color-coding is used to indicate completed activities. Comparison between estimated start and end and actual start and end indicates the project status on a task-by-task basis. By looking at the color coded task bars in the Gantt chart, we will be able to identify which activities are running late and which activities are progressing as per the plan. With this chart, a manager will be able to take corrective action against activities that are running late.

Variation of this chart called as Personal Gantt chart, shows personnel allocation on a person-by-person basis. Personal Gantt chart gives a Gantt chart view of activities for each individual or organization. With this chart, a manager will be able to track the performance of each individual / organization within a project team. This chart shows the slack time for the project personnel, that is, times when they are not actually working. It also shows the progress of individuals or organization on a specific set of tasks. This can be used to estimating programmer productivity.

Gantt Charts give a clear illustration of the project status. But they do not indicate task dependencies. Numerous authors have exposed the limitations of Gantt charts [34, 35, 36, 37, 38]. Gantt charts do not show dependencies easily and therefore cannot represent sequences of activities. Using Gantt charts we cannot visualize how one task falling behind schedule affects other tasks. Also Gantt charts provided by the standard CASE tools are not interactive. Hence they are not suitable for projects having large number of activities because the activities cannot be viewed in a single screen.

The Advanced Charting System overcomes these limitations by presenting complementary views to Gantt Charts. PAMPA 2 ACS generates interactive Gantt charts that will let the manager view the Gantt chart either at a summary level or at a detailed level or at any of the intermediate levels. A manager will be able to drill down the Gantt chart and view the required level of information to assess the project

status. PAMPA 2 ACS embeds cross reference links in the Gantt chart which will help the manager to navigate to the corresponding activity networks view to get the task dependency information. PAMPA 2 ACS also provides zoom features to let the manager enlarge a specific region of the plan in the Gantt chart to get a better view of the project.

2. PERT Charts

Program Evaluation and Review Technique (PERT) chart depicts task, duration, and dependency information. A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labeled vectors (directional lines) representing tasks in the project. According to McLeod and Smith [34] the technique is intended to provide better information on the status of projects. PERT involves breaking a large project into component tasks, determining task dependencies, estimating the time required to perform each task and summarizing the information into a chart form [35, 36, 37]. Each chart starts with an initiation node from which first task, or tasks originate. If multiple tasks begin at the same time, they are all started from the node or branch, or fork out from the starting point. Each task is represented in a box, which contains information about the task such as name, duration, resources, and start and end dates. The other end of the task line is terminated by another node, which identifies the start of another task, or the beginning of any slack time (waiting time between tasks). Each task is connected to its successor task in this manner forming a network of nodes and connecting lines.

An advantage of activity networks is that they can be used to identify and track critical path of a project. The critical path is the set of activities along the path that takes the longest time to complete. In essence, the critical path determines when the

project will be completed. According to McLeod and Smith [34], a project manager can avoid the delay of entire project by monitoring the activities on the critical path. Activities that are off the critical path have some freedom of activity. These activities can move slightly out of time without affecting the overall schedule of the project - this time is known as slack time. When slack time exists between the end of one task and the start of another, it is represented as a broken or dotted line between the end of the first task and the start of the next dependent task. The tasks on the critical path require special management attention because any schedule slippage in these tasks leads to a corresponding slippage in the project completion. A project can have multiple critical paths and that variations in the duration of tasks can shift a critical path from one set of activities to another. The critical path is used to identify various schedule risks.

The PAMPA 2 ACS generates interactive PERT charts, which lets the manager to cross-reference the Gantt charts by clicking on specific tasks in the activity network. Using PAMPA 2, ACS can continuously monitor the project and shift the critical path as the activities are completed. Also ACS provides special visual cues on the critical path along the activity network in order to get the attention of the project manager on the activities that lie along the critical path. Conventional PERT charts are very difficult to interpret especially if the projects are big and complex. ACS generates interactive chart, which will let the user roll up the activities to form a summary activity or drill down, to get a better view of the project.

3. Work Breakdown Structure (WBS) Chart

Work Breakdown Structure is the basis for all planning activities. The WBS was initially developed by the U.S. defense establishment, and it is described in Military Standard (MIL-HDBK-881) [39] as follows: *"A work breakdown structure is a*

product-oriented family tree composed of hardware, software, services, data and facilities. It displays and defines the product(s) to be developed and/or produced and relates the elements of work to be accomplished to each other and to the end product(s)."

According to the Center for Project Management, Dekom [36], Heizer and Render [37], Kerzner [35] and McLeod and Smith [34], the work breakdown structure [WBS] decomposes a project into its composite parts. Kerzner [35] continues by explaining that a WBS provides a common framework that can be utilized for planning, for establishing costs and budgets and for monitoring the progress of the project. Both plan and work breakdown structure deals with activities and processes. The difference is that the activities and processes detailed in a plan have one to one mapping to the activities in the work breakdown structure. Whereas the activities and processes in work breakdown structure have a many to one mapping to planned activities and processes. In other words work breakdown structure is a detailed representation of the activities and processes in a plan. Each high level process / activity is broken down to a detailed level in a work breakdown structure.PAMPA 2 generates a work breakdown structure chart similar to Fig 3.

A WBS can be decomposed into several levels. The number of levels of detail depends mostly on the project size and personal preference of the project manager. A manager should be able to view the work breakdown structure at different levels The system should display a high level or a summary view of the project, i.e, display all levels of detail or roll-up the detail to display a high level (summary) plan thus giving a macroscopic/microscopic view of the project. It is important that all the work items necessary to complete the project should be included in the WBS and assigned to an individual or a specific organization in an unambiguous manner.

Also a manager should be able to zoom into specific sections of the plan. This feature allows us to isolate a section of the chart for viewing and/or printing sections

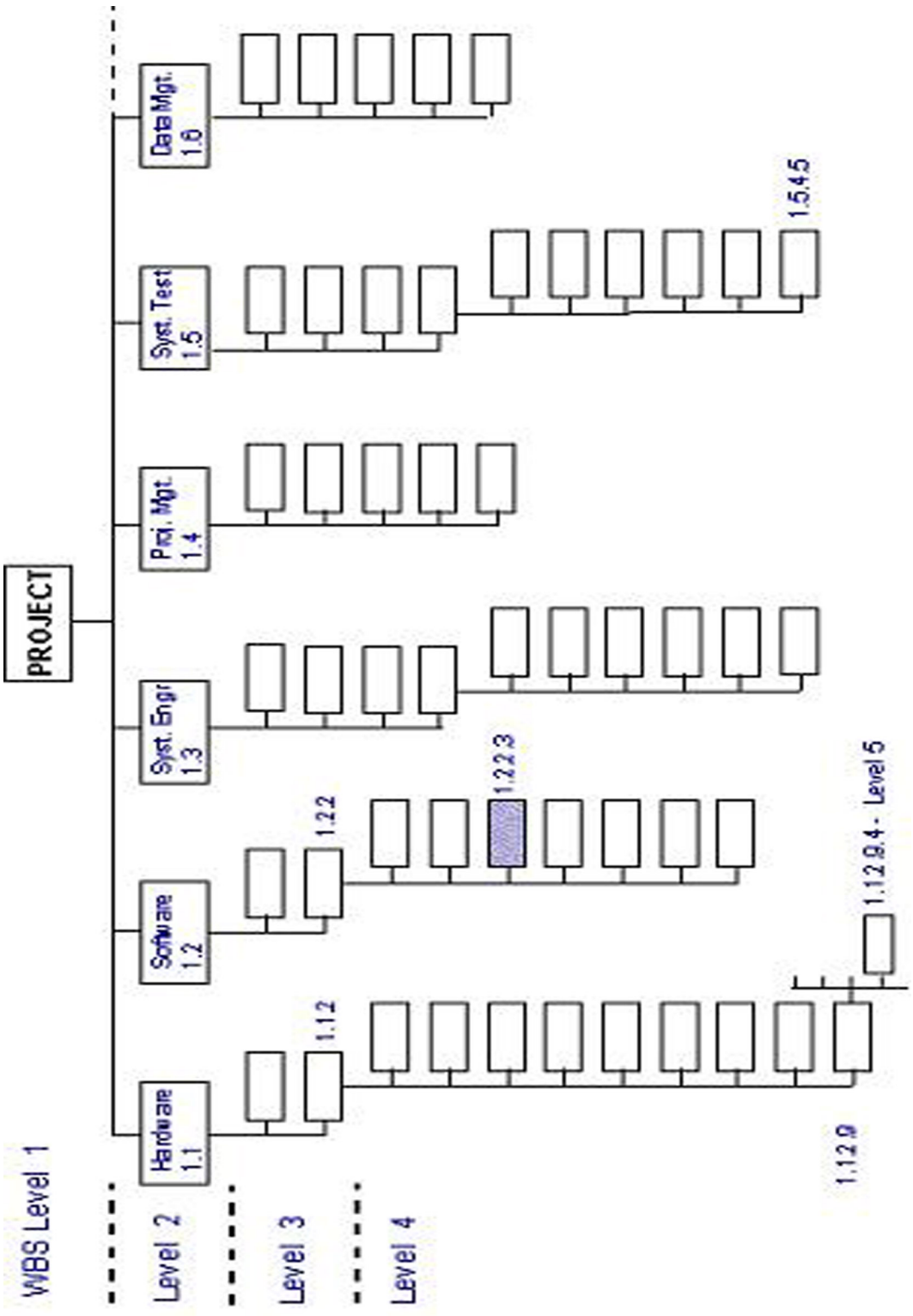


Fig. 3. PAMPA II Work Breakdown Structure Chart

of the plan that needs immediate attention. This is ideal when working with groups of people for displaying only the sub-tree that pertains to that group

This chapter discussed the standard project management visualization techniques supported by ACS. The next section will discuss DOT chart, which is a custom visualization technique and flagship feature of ACS.

CHAPTER V

DOT CHARTS

The DOT Chart is a novel charting idea, and a unique feature of PAMPA 2. D.B. Simmons developed the DOT chart concept in support of an R&D project for Hewlett Packard [40]. DOT charts combines the benefits of Gantt Charts and Activity networks. DOT chart represents the timeline of activities and the task dependencies as concentric dots which can be drilled down progressively . In addition to this DOT chart provides a natural way of visualizing the project activities based on the type of lifecycle model being followed. For example, if a project follows a spiral model, the activities are visualized as spirals having varying number of phases. In a plan following waterfall model, the activities are represented in a linear fashion. Though this chart allows us to track an arbitrary plan, it is particularly useful in tracking projects that follow the spiral process model. These aspects of the lifecycle model are represented graphically using DOT charts.

A. Life Cycle Models

A basic understanding of these lifecycle models is essential for visualizing them. We will be able to understand the processes better if we can visualize them in their natural form. A series of "what if" conditions can be simulated and their impact can be realized visually if we can get a graphical model of the task dependencies. The following sections give us a brief overview of the popular lifecycle models to show us how they can be visualized. The evolution of a software product between the beginning and ending of a process is called the software life cycle (SLC). For a specific software project, the SLC is a sequenced map of activities (or tasks).

1. Waterfall Model

The Waterfall Model abstracts the essential activities of software development and lists them in their most primitive sequence of dependency. According to Royce and Winston [41], Waterfall Model describes a development method that is linear and sequential. Fig 4 describes the waterfall model.

Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back (theoretically) or it is very expensive in terms of cost and time to revisit the earlier phases.

2. Spiral Lifecycle Model

The original spiral model [13] uses a cyclic approach to develop increasingly detailed elaborations of a software system's definition. The spiral lifecycle combines elements of the waterfall lifecycle model, along with an emphasis on the use of risk management techniques. Its chief distinguishing feature is that it is primarily risk-avoidance driven rather than document-driven. Fig 5 illustrates the spiral model graphically. Each cycle of the spiral uses the same high-level processes: determine objectives, alternatives and constraints; evaluate alternatives, identify and resolve risks; develop, verify next-level product; and plan next phase. Within this framework, the tasks may resemble those of the waterfall lifecycle model with the additional of specific risk management techniques. A review occurs as a project management checkpoint after each cycle is completed. The primary advantage of the spiral model is that its range

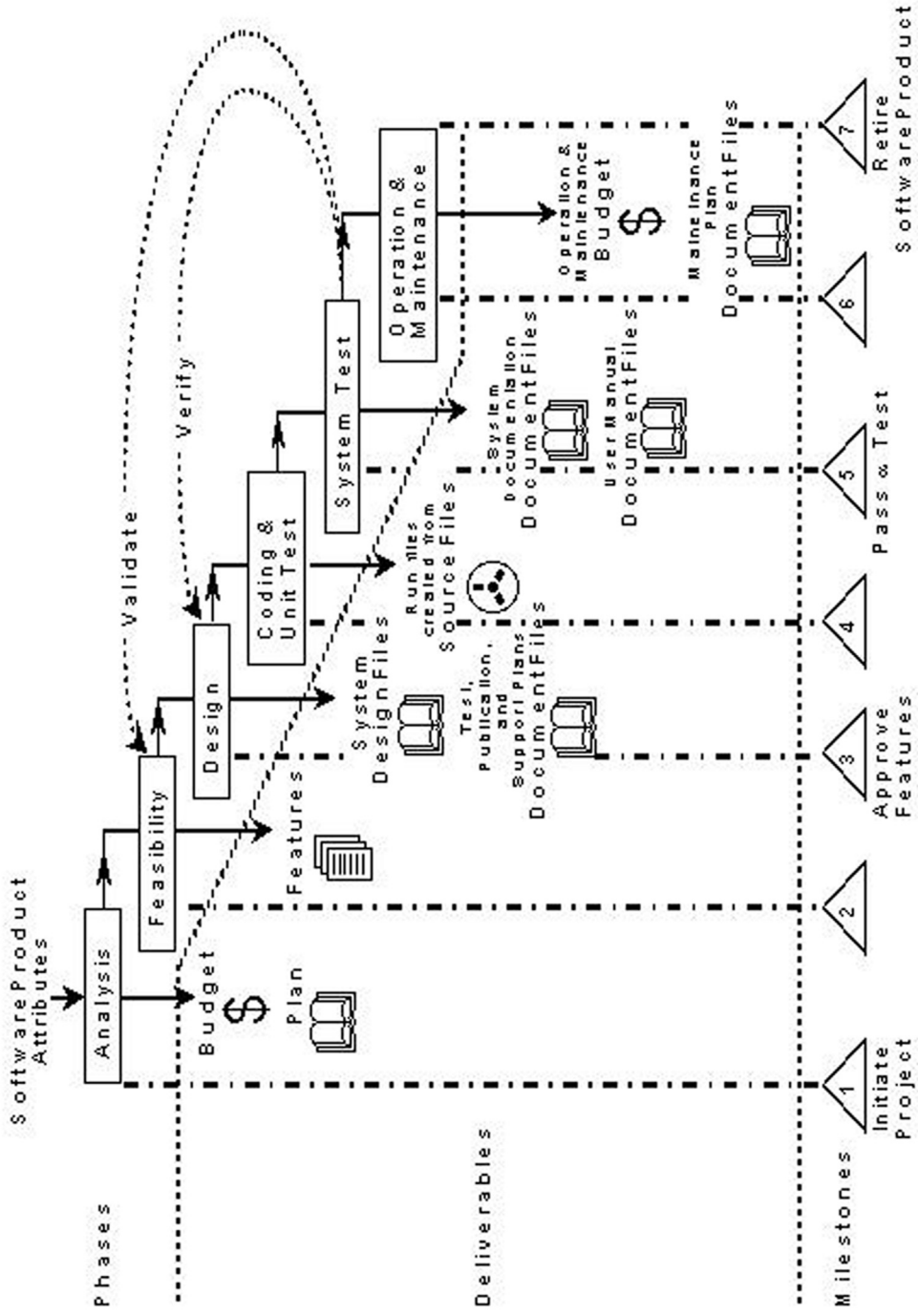


Fig. 4. Waterfall Model

of options allows it to accommodate the best features of existing software process models, while its risk-driven approach helps it to avoid most of their difficulties. In appropriate situations, the spiral model becomes equivalent to one of the existing process models. In other situations, it provides guidance on the best mix of existing approaches to be applied to a given project.

The spiral model has been extensively elaborated [42], and successfully applied in numerous projects [43, 44]. However, some common difficulties have led to some further extensions to the model.

The spiral model is a risk driven process model. It has three main distinguishing features.

- Cyclic approach for incrementally growing a system's degree of definition and implementation.
- Set of anchor point milestones for ensuring stakeholder commitment and mutually satisfactory system solutions.
- The level of effort on each activity within each cycle is driven by risk considerations.

3. Incremental Lifecycle Model

An incremental model has a subset of properties possessed by a spiral lifecycle model. In incremental development, each phase consists of a series of relatively small steps. Fowler [45] suggests that every step is a mini-project consisting of the following activities: analyzing the work to be done, designing the incremental solution, coding, testing, and documenting. In an incremental model the detailed increments and releases are planned. In incremental model, each increment has its own complete lifecycle (usually shown as a waterfall lifecycle in examples). The increments may

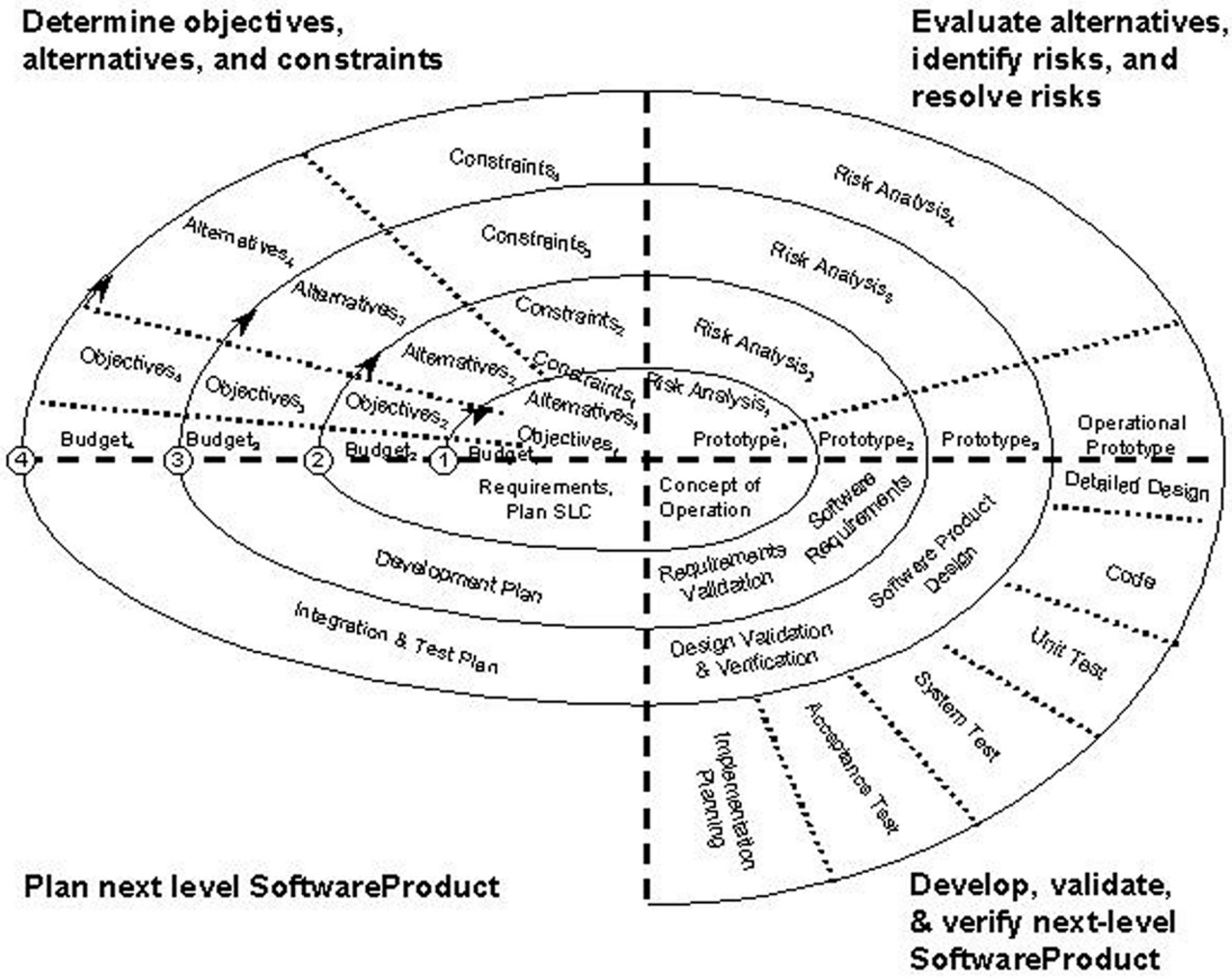


Fig. 5. Spiral Model

be built serially or in parallel depending on the nature of the dependencies among releases and the availability of resources. Each increment adds additional or improved functionality to the system. Every iteration results in a working product with reduced functionality.

B. Dot Chart Visualization

The PAMPA 2 ACS can detect the lifecycle model of an arbitrary plan using PAMPA 2. PAMPA 2 uses the basic characteristics of the lifecycle model to detect an arbitrary plan and uses the recognized lifecycle model as a template for visualizing the plan. Visualizing a plan conforming to waterfall model is easy. All the activities will be represented in a linear fashion. Visualizing a spiral model is more complex. In a project plan conforming to spiral model, a set of recurring activities can be grouped into a spiral and the spirals themselves can be grouped to form concentric spirals, if the member spirals have the same number of activities. Based on the task dependencies, each spiral will be preceded and/or succeeded by one or more spiral groups. Each of these spiral groups may vary in the number of activities contained in its spirals. These spirals are ordered on a time line and also based on Organization (refer the object model). In order to track the progress of activities contained in the spiral, the spirals are color-coded. A red indicator in a specific activity in a spiral would mean that the activity is lagging behind schedule; a blue indicator would mean that the activity has been completed on schedule and a grey indicator would mean that the activity is planned but not actually started. This chart gives a fish eye view of who is working on the project, what activities are lagging behind and, what other activities are dependent on completion of this activity and hence combines the information presented in Gantt Charts and PERT charts in addition to visualizing the whole

project life cycle. Fig 6 depicts a multilevel view of DOT chart. This particular plan was made for the HP R&D development project where an incremental development was done using the spiral process model and features were gradually built into the system. Step 1 of Fig 6 shows a single collapsed dot that represents the whole project as a PI chart. This dot represents the whole project and the color of the dot indicates the status of the project. The portion of the PI chart that is in blue color represents the percentage of activities that are completed. The portion of the chart in red indicates the percentage of incomplete activities. The remaining portion in gray indicates the planned activities that haven't started. Step 2 provides a zoomed in (enlarged) version of step 1. The PI chart in step 3 is a drill down view on the project. This displays the status of individual project versions. Each of these versions can be individually drilled down further to get detailed information about their activities. Step 6 displays the status of activities according to their organization in the project plan. Step 7 provides a detailed view of the project activities by generating the spirals and activities as DOTs. Each dot in figure 6 is a collection of spirals, where each spiral is a collection of recurring activities that logically completes an iteration in the spiral process model. These spirals are grouped to form DOTs and these DOTs are linked to each other as per the task dependencies of their member activities.

The DOT charts provide a multi level interactive chart that can be drilled down to their lowest level until the DOTs are collapsed into activities. The lowest level of this chart is represented by stage 7 where the activities are grouped by organization, time lines and spirals. Each of these spirals represents a set of activities that occur together. In this example, the 4-faced spirals represent the design, development, alpha test and beta test of a specific feature. Since the whole system is built incrementally by adding features, the plan contains a set of recurring activities of design, development, alpha and beta tests that are grouped in to concentric spirals having four phases in

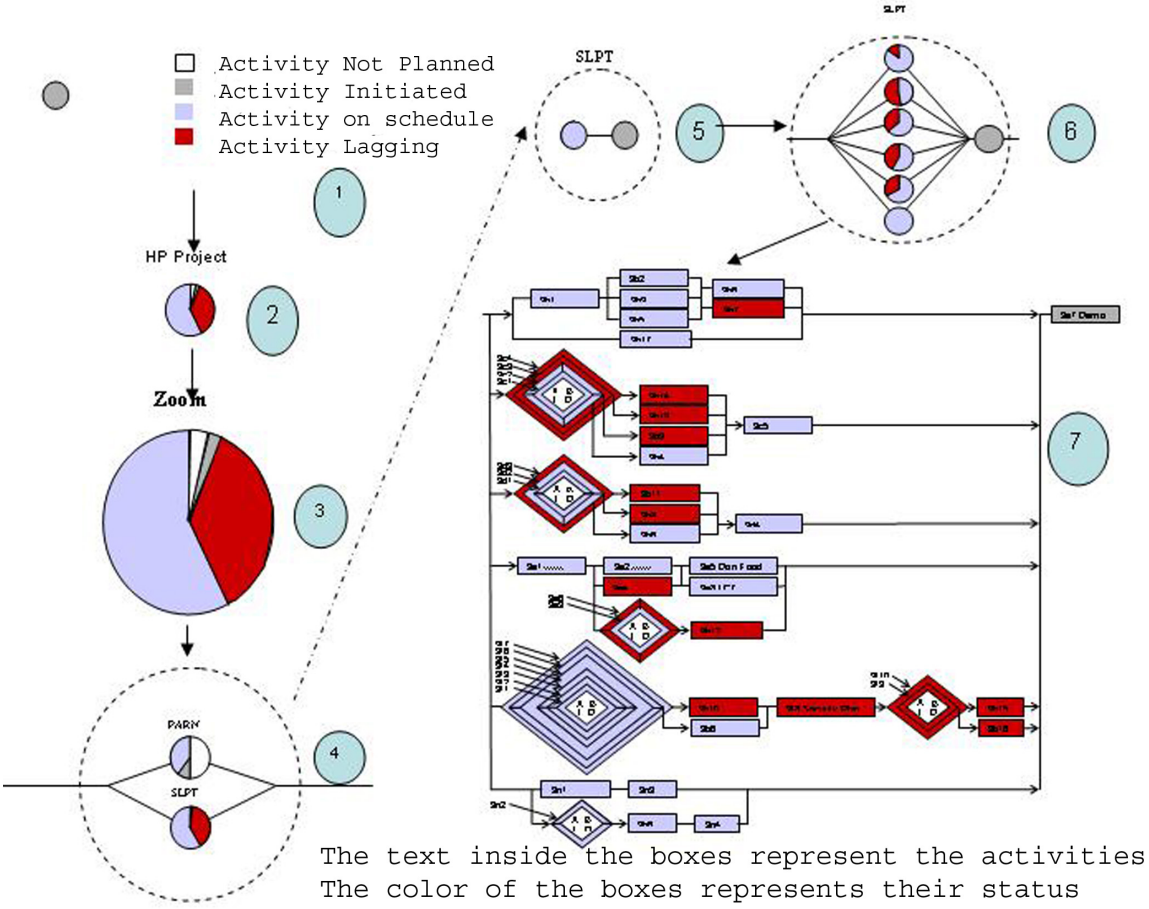


Fig. 6. PAMPA 2 Dot Charts

each spiral. The arrows out of the spirals represent the next set of activities following this feature. In this example, it might be an external testing activity happening in parallel to the internal test or it can be the next feature having a dependency on this one.

DOT charts allow drill down on lagging activities to view the project, processes, activities, schedule, PERT/CPM charts and Gantt Charts. We can also zoom up in to a higher-level view, for example from activity level to a pie chart containing various sub-processes and their status (Note that we will also be able to drill down from this view to the spiral view). This can be further zoomed out to get a bigger picture of the whole project in terms of it subprojects. When a plan following a waterfall model is tracked using DOT charts, it will not have spirals because of the sequential nature of activities and hence it will be identical to the Gantt chart. DOT charts are ideal to track projects following spiral process model [13] where the development happens incrementally. The activities shown in red are the delayed activities and the ones in gray are on schedule. If there are a lot of red activities, then the project has a problem and appropriate control measures needs to be applied in order to deliver the project on time. From this DOT Chart view, a manager will also be able navigate to a Gantt chart or WBS chart view to get an overall picture of the project from multiple dimensions.

The following are the characteristics of DOT charts

- The DOT charts can be used in visualizing a spiral process model in its natural form and hence promote a better understanding of project activities.

- DOT charts provide a multilevel view of the project initially starting with the project level. The interactive behavior of DOT chart lets the user drill down the project view to project version view, process view, sub process and activity

level views hence providing a multi level view of the project.

- DOT charts have zooming feature to enlarge and view specific portions of plan
- DOT charts provide tracking features by providing color-coding of activity status. Using this, a manager will be able to track the progress of activities.
- Dot charts also indicate the critical path, which comprises the major risk in the project and also continuously monitors a project for change in the critical path.
- Dot charts present a multi chart representation unlike Gantt charts and PERT charts, which have a unique form of representation. DOT charts provide a mixed representation comprising of pie charts, bar charts and other graphical forms to represent a project in its natural form.
- Dot charts help the manager in identifying sequential activities that can otherwise be performed in parallel. According to Simmons, this improves group efficiency [19]. If the ratio of time spent on sequential tasks to the total time spent on all the tasks is fs, then the highest efficiency occurs when fs is equal to zero. If we eliminate more number of sequential activities and replace them with parallel activities, we will be able to reduce the schedule time and hence indirectly the critical path. By looking at the DOT charts a manager will be easily able to identify which sequential activities can be reorganized to concurrently occurring activities in the plan.

Thus DOT charts provide a useful visualization of project activities and hence can help a manager in understanding project activities better and thereby assist a manager in process improvement, tracking and controlling a project.

CHAPTER VI

ACS SYSTEM ARCHITECTURE AND DESIGN

Software architecture forms the backbone for building successful software-intensive systems. Architecture is the framework of the system, detailing interconnections, expected behaviors, and overall control mechanisms. If done right, it lets the developers concentrate on specific module implementations by freeing them of the need to design and implement these interconnections, data flow routines, access synchronization mechanisms, and other system functions. Architecture represents a common vehicle for communication among a system's stakeholders, and is the arena in which conflicting goals and requirements are mediated.

Design involves the use of scientific principles, technical information, heuristics, and Imagination in the definition of a software system to perform pre-specified functions with maximum economy and efficiency. Design defines a software system in detail and embeds the essential qualities of software such as reliability, usability, performance, maintenance etc. Design is as important as architecture. Design and Architecture on top of a clear requirement specification forms the foundation for software of superior quality. The following sections define how we use these principles for developing the PAMPA 2 system.

A. Architectural Design

Figure 8 shows the simplified architecture of Three-Tier architecture. ACS follows a three-tier architecture. A three-tier architecture [46, 47] is a flexible way of organizing distributed client-server systems. In a client-server system, every client is connected to a server and the business logic gets executed on the client. In a three-tier architecture the middle tier server is between the user interface (client) and the data management

(server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users by providing functionality such as queuing, application execution, and database staging. The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the end user.

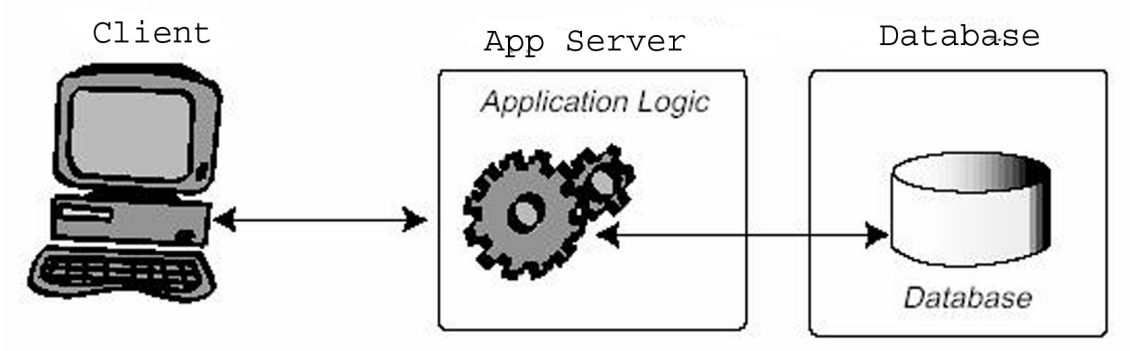


Fig. 7. Three Tier Architecture

PAMPA 2 ACS currently has three tiers. Fig 7 shows the Three-Tier architecture as implemented in ACS. The presentation tier has the web browsers, which are thin clients, to access the PAMPA 2 framework on the Internet. The requests from these browsers are received by the web server, which acts as a gateway to PAMPA 2 application components in the middle tier. The three-tier web based architecture makes PAMPA 2 platform independent. The middle tier comprises of PAMPA 2 application components and the visualization engine. The requests from the web server are forwarded to the application components. These application components dispatch the data gatherers, which will analyze the project data in the CASE tools and store them in the PAMPA 2 knowledgebase. On request from the clients, PAMPA 2 application

components accesses the knowledgebase and process the data and sends the processed data to the visualization engine. The visualization engine is the core component of the PAMPA 2 ACS that generates charts and graphs as per instructions from the application components. These visualizations are then sent to the web server, which composes the response and sends it back to the client.

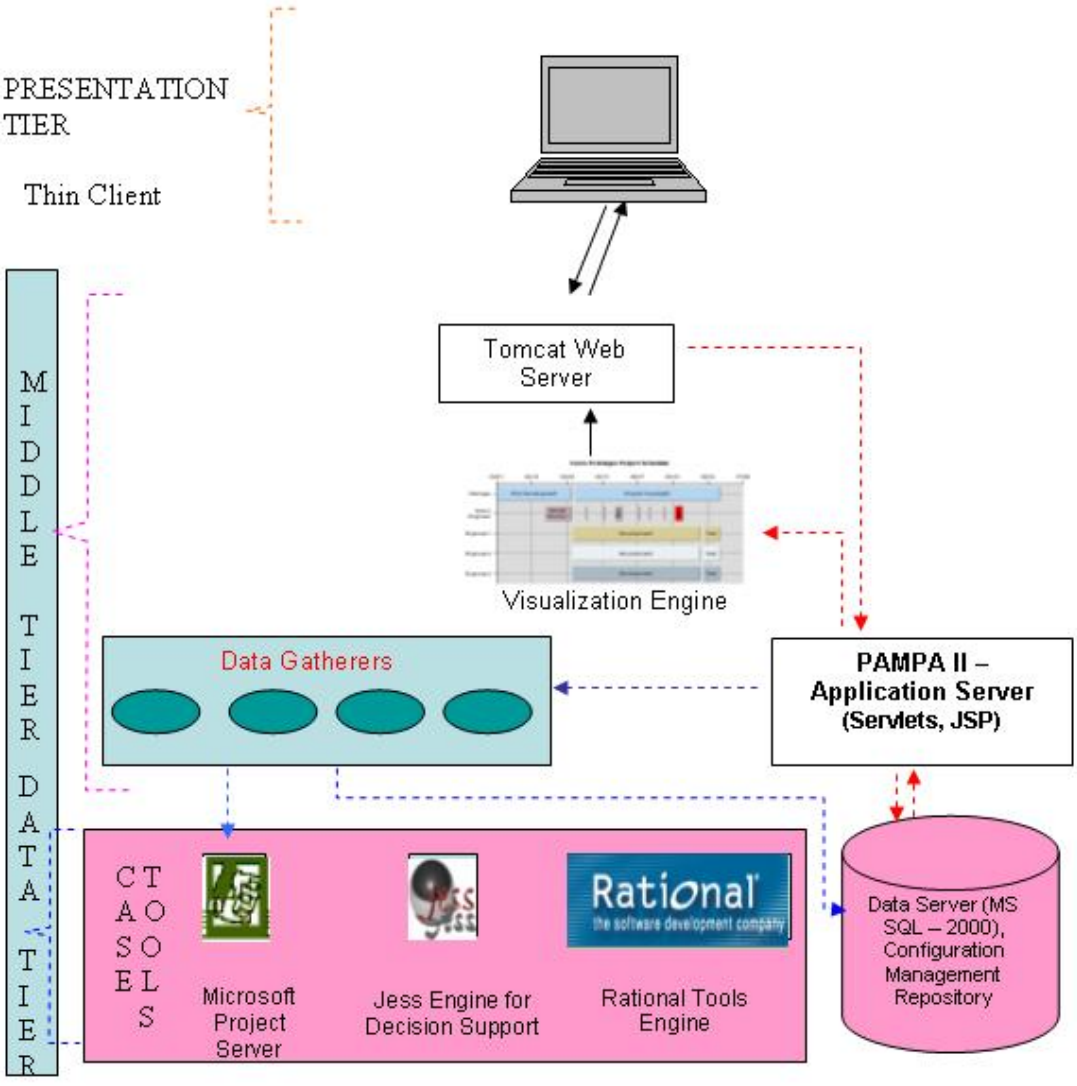


Fig. 8. PAMPA 2 ACS Architecture

Fig 8 shows the PAMPA application components interacting with the visual-

ization engine to produce the charts and graphs. The visualization engine composes html compatible images so that they can be viewed in all browsers independent of the platform. For interactive features such as zooming and drilldown the client browser communicates with ACS and supplies the appropriate parameters for zooming in or drilldown. These requests are processed at the ACS and the new images are generated and pushed out to the client browser.

Data Gatherers (DG) are agent programs that can understand the documents produced by the CASE tools. These programs are invoked by the PAMPA 2 framework components periodically or based on request from the client browser. The DG's locate the project data from the CASE tools, analyzes it, extracts the project metrics and stores it into the PAMPA 2 knowledgebase. PAMPA 2 integrates with many CASE tools in order to gather the project metrics. Some of the CASE tools include Microsoft Project (Planning), Rational Requisite Pro (Requirements), Rational Clearcase (Project Artifacts), Rational Clearquest (Defect Tracking) and Rational Rose (Use Cases and Design Documents). Some of these tools like Requisite Pro directly integrate with the database and some others like Clearcase and Rational Rose needs DG's to analyze its log files or artifacts to gather metrics. In order to accommodate disparate information sources, PAMPA 2 knowledgebase acts as a data repository and gathers facts from all the CASE tools and stores in its custom database. This data is processed by PAMPA 2 application components to generate charts.

B. External Interfaces

PAMPA 2 ACS interfaces with external systems in order to provide visualization for project data. The external systems include Microsoft Project [24] (Stores Plan data)

and Statistical Analysis Software a.k.a SAS [48] (visualization for Gantt charts, Activity Networks and WBS charts). PAMPA 2 uses Data Gatherers as external interfaces to gather data from Microsoft Project, Rational, and SAS tools. The DG's communicate with the CASE tools through three mechanisms

- ODBC (Open Database Connectivity) - Microsoft Project
- Native Database Drivers - Rational Requisite Pro
- Custom parsers - Rational rose and Clearcase

SAS contains a suite of project management tools, which can be used to visualize the project metrics. SAS contains several inbuilt procedures to compute critical paths (proc cpm), Gantt chart (proc gantt), and WBS (proc PM). These procedures can be externally accessed using Application Programming Interface (API) calls. The visualization engine composes the SAS queries to generate the appropriate type of chart and issues an API call to SAS. SAS generates these charts and returns the charts to Visualization engine. These charts are then converted into a suitable web compatible format by the visualization engine and are returned to the client browser via the web server. SAS communicates to the visualization engine through an interface mechanism called Output Delivery System (ODS) [49]. This ODS protocol will write the SAS results to the standard output device in a pre-specified format. The visualization engine composes SAS data sets and sends it to SAS to generate the output charts.

C. Database Design

PAMPA 2 ACS extends the PAMPA 2 knowledgebase and has added more entities in order to accommodate the objectives of the tool. The new schema contains tables

to hold requirements data, plan data, configuration management and defect tracking data. Since PAMPA 2 ACS currently tracks only the plan data, we will concentrate only on that portion of the schema concerning the project plan. Detailed information about the Microsoft Project plan tracking tables can be obtained from Project MVP website [50]. Fig 9 shows the entity relationship diagram for the Microsoft project plan data integrated with PAMPA 2 knowledgebase entities.

Plan tracking is done using 13 main tables. Native tables of PAMPA 2 are prefixed by PAMPA. Microsoft Project have an MSP prefix. Table I provides a description of the tables.

The Data Gatherers extract information from Microsoft Project data files and populate them into the PAMPA 2 knowledgebase tables. These tables contain essential information about the plan. Using these tables the visualization engine will be able to generate charts for the project schedule and its activities. Essentially PAMPA 2 ACS externalizes the knowledge present in the CASE tools and shares it with other applications.

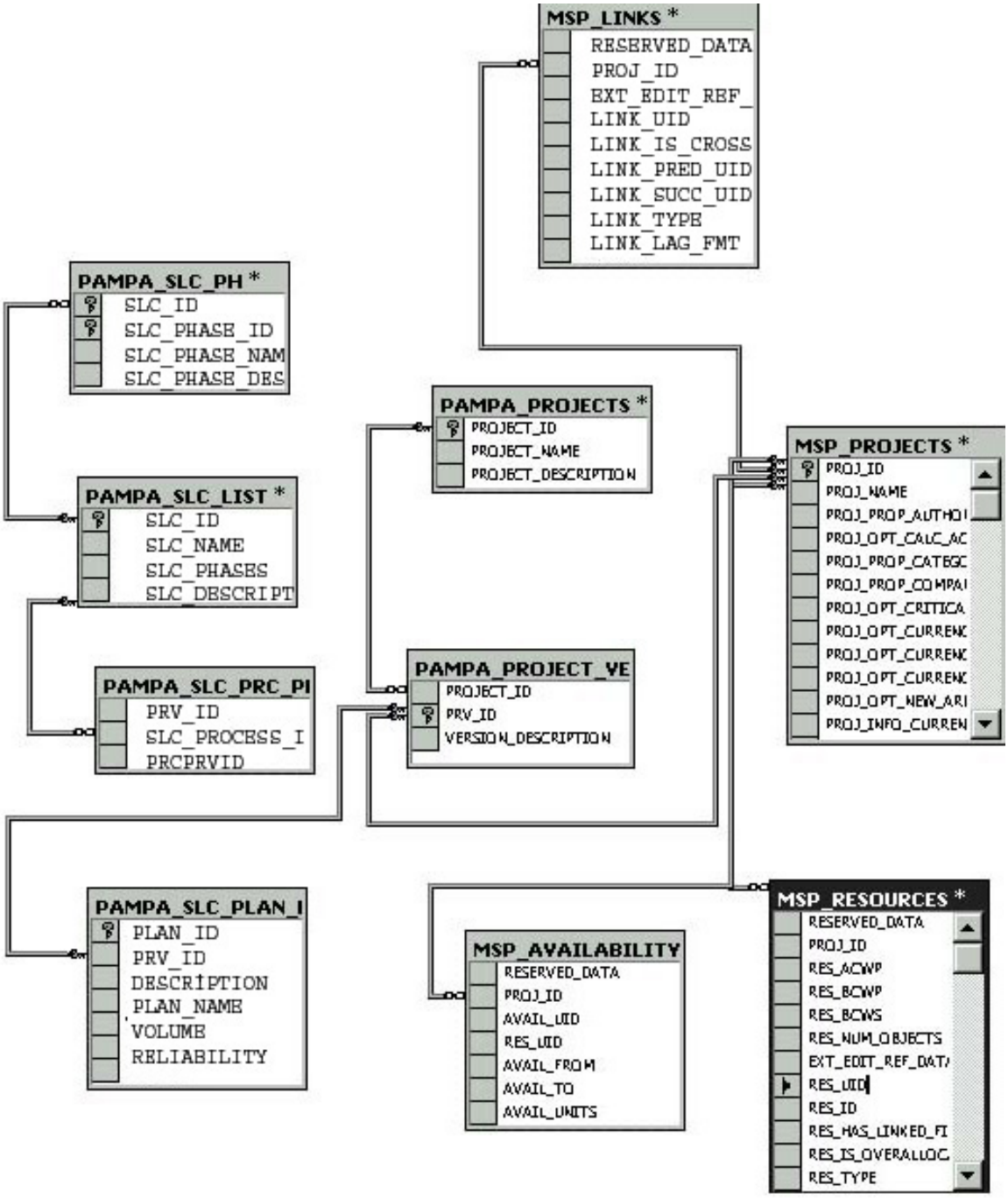


Fig. 9. PAMPA 2 Entity Relationship Diagram

Table I. PAMPA 2 ACS DATABASE TABLES LIST

TABLE NAME	**PURPOSE**
PAMPA_SLC_LIST	Contains the list of supported software lifecycle models, its description
PAMPA_SLC_PH	Contains the lifecycle model definitions and phases
PAMPA_SLC_PLAN_PRV	Contains the expected planned values of project attributes such as reliability, usability, volume etc.
PAMPA_SLC_PRC_PRV	Associates a project version with a SLC model
PAMPA_PROJECTS	Contains master projects detail
PAMPA_PROJECT_VERSIONS	Tracks the various versions of a project by linking them to a master project
MSP_PROJECTS	Contains project version details, extracted from MS Project
MSP_RESOURCES	Contains the resource details for a specific project
MSP_TASKS	Contains the activity details for a specific project
MSP_ASSIGNMENTS	Contains resource assignments to tasks
MSP_RESOURCE_RATES	Contains the salary details for resources
MSP_LINKS	Contains the activity dependency information

CHAPTER VII

IMPLEMENTATION

This chapter discusses the implementation and technology details of PAMPA 2 ACS.

A. Application Components

PAMPA 2 Advanced charting system has been implemented with Java Servlets and JSP. Table II describes the technology, tools and libraries used for implementing PAMPA 2 ACS.

Table II. TECHNICAL DETAILS

Category	**Specification**
Implementation Language	Java, C++
Technologies	JNI, Servlets, JSP, SAS IntrNet, Macromedia Flash
Platforms	Windows 2000, Solaris 2.8
Database	SQL Server 2000
Libraries	Ming Library for flash extensions, JDBC Type III drivers for data access, JPEG, PNG and GIF image encoding libraries, Xerces XML parser
Web Server	Apache Tomcat 4.1.1.0
Development tools	Textpad, VisualCafe 4.0
Build Tools	Jakarta Ant

As discussed in the previous chapters, ACS architecture contains three critical components.

- Visualization engine
- Data Gatherer
- Data Broker

Visualization engine is responsible for generating interactive charts from the data received through PAMPA 2. The Data Gatherers are responsible for gathering data from CASE tools and storing it in PAMPA 2 knowledgebase. Data Broker module is responsible for all database interactions. PAMPA 2 application framework would be accessing the data repository through calls to Data Broker. This abstracts the PAMPA 2 framework and application components from the database and hence provides a flexible framework that can work with any database. The following sections discuss the implementation details of each of these modules.

B. Visualization Engine

The visualization engine is responsible for generating the charts from the data received through PAMPA 2 and SAS. The visualization engine uses SAS to compute Gantt charts, PERT/CPM charts and WBS charts. SAS (Statistical Analysis Software) Integration technologies delivers a base service component hierarchy with an integrated object model (IOM) [51] that provides access to the SAS procedural scripting language, data, file system, results content, and formatting services through Java API calls. Java applications can connect to SAS system through SAS/Connect, which is a SAS service that allows programmatic connection to SAS system. SAS IOM provides API's for connecting to a SAS system, running a SAS application remotely, receiving

the output of a SAS program and to access the log files for getting error/status messages. Using this, the PAMPA 2 ACS visualization engine running in a remote java environment will be able to connect to the SAS system, convert the data received from PAMPA 2 into SAS datasets, and pass the SAS datasets to SAS system to generate the charts. Visualization engine can access the graphs generated by SAS using SAS ODS [49].

The visualization engine converts the graphs generated by SAS into a HTML compliant image format using image-encoding libraries such as JPEG/PNG/GIF encoders. The type of image format can be pre-configured in the visualization engine or can be changed at runtime by user configuration. In case of DOT charts, the chart generation is not delegated to other systems. The Visualization Engine recognizes the lifecycle model of the plan and renders the charts as per the model. For example, while tracking a plan following a spiral model, the DOT charts renders the spirals with varying phases on a graphic canvas. After rendering all the spirals, the visualization engine encodes this canvas using image-encoding libraries.

Irrespective of whether the graphs come from SAS or from visualization engine, they are treated in the same way by the encoding libraries. The visualization engine also provides support for Flash [52] images. The advantage of flash images is that it supports more interactive features than regular JPEG images. Zoom in and Zoom out can be performed locally on the client without having to make a request to the server again. The image gets rendered locally on the client while using flash because the flash viewer is present on the client side. Flash also supports animation and hence we can provide more interactive charts. Flash also supports image maps similar to the regular html images. Hence it can produce drill down charts. PAMPA 2 ACS has a flash image generation module inside the visualization engine. The flash module accesses native flash libraries written in C++ through a JNI interface to produce

flash images. Java Native Interface (JNI) [53] is a protocol to access native libraries written in languages other than Java from JAVA. If a user chooses to receive the visualizations through flash, the images received from SAS/ or Java are encoded into flash images through these extensions and pushed out to the client browser through the web server.

C. Data Gatherers

Data Gatherers are used to access data from CASE tools. ACS deals mainly with tracking plans. we will mainly discuss the Data Gatherer interface for Microsoft Project, which is a CASE tool widely used by many managers to make project plan. There are two ways in which Microsoft Project Plan data can be captured into the PAMPA 2 knowledgebase.

- Directly store the data from Microsoft Project into a ODBC data source. If the data source points to the PAMPA 2 knowledgebase, we can directly receive the plan data

- Microsoft Project provides COM API's [54] to programmatically access plan data. Using these API's the DG will be able to programmatically save the data in a project file into the data source pointing to PAMPA 2 knowledgebase. The only challenge here is that Java has a poor support for COM. PAMPA 2 ACS solves this problem by building a Java-COM bridge (JACOB) [55] that will help Java application components to interact with COM components. Using this bridge, the Java components in the Data Gatherers will be able to talk to COM extensions of Microsoft Project and retrieve the plan data. Then the data gatherers will store the captured project plan into the PAMPA 2 knowledgebase.

D. Data Broker

In both the approaches mentioned above, storing the data into the PAMPA 2 knowledgebase involves database calls. Instead of letting each component in PAMPA 2 framework perform its own database operations, PAMPA 2 provides a module called data broker that acts as the data access layer. All application components access the database through calls to the Data Broker. Data Broker abstracts the data access logic and provides data services to all the application components. This has two benefits:

- Abstraction of data access logic from the application components improves portability. The application can use any database as its data store by switching the data broker component that is compatible to the database.
- Data Broker also provides data transformation service. The result sets from the database are converted into other formats at the Data Broker depending on which module needs the data. If the data needs to be passed on to SAS, Data Broker generates SAS datasets instead of regular SQL result sets. This provides an additional flexibility.

Data Broker accesses the database using Java Database Connectivity (JDBC) [56, 57]. JDBC is a protocol for accessing databases programmatically from Java. The Data Broker module in PAMPA 2 ACS can be configured to access the database either through Type I or Type III JDBC.

- Type I JDBC: Data Broker uses ODBC [58] to access the database. The ODBC data source must point to PAMPA 2 knowledgebase.
- Type III JDBC: Data Broker uses native drivers provided by the database

vendor to access the data from the database. This is a better solution in terms of performance.

PAMPA 2 ACS can be configured to use Type I or Type III JDBC. ACS also exposes certain parameters to configure the connection information. The configuration parameters are available in XML format and can be manipulated by a Java application during runtime using Xerces XML parser.

CHAPTER VIII

EVALUATION, CONCLUSION AND SUMMARY

The importance of information visualization has increased in the recent years, as people have access to larger and more diverse information systems and due to the nature of increasingly complicated interactions between these information systems. Human brain cannot perceive and interpret such high volume information in a textual form. In the previous chapters, we presented PAMPA 2 ACS as an advanced visualization mechanism which will allow the managers to graphically visualize the project metrics using improved visualization techniques.

In this chapter, we will compare PAMPA 2 ACS with other project management CASE tools. We will demonstrate how PAMPA 2 ACS overcomes the limitations posed by traditional project management tools. We will also show how PAMPA 2 ACS is effective in tracking projects using a sample application. Finally, we will suggest possible extensions to the PAMPA 2 framework and summarize the results

A. Limitations of Traditional CASE Tools

- Most of the CASE tools are affiliated with one type of process model. For example, Rational CASE tools support the Rational Unified Process model very well but provide little support for a waterfall based model. Microsoft Project supports waterfall model extensively but provides little support for other life cycle models. As a result, we loose the ability to track arbitrary plans using a single tool. There is a need for a generic tool that can trace arbitrary plans.

- Microsoft Project lacks the capability to address multiple project versions and life cycle model association. It does not distinguish between processes and

activities.

- Most of the CASE tools are standalone and platform dependent. We do not have a browser-based version of CASE tools. We might get into the issues of Software installation, licensing and mobility of client/server, middleware and servers if we are to use a standalone CASE tool.

- If a project falls behind schedule, it is important that decisions are made quickly to identify alternative course of action to modify the plan. The standard project management tools available in the market lack the capability to dynamically track changes in plans. As a result these tools cannot dynamically assess the risk created because of changes in plans.

- Charts generated by these CASE tools are not interactive. The static nature of the graphs makes it difficult for the viewer to assimilate the information presented, especially if the volume of the information is high.

- Charts produced by these tools are fixed in nature. New chart types cannot be plugged into the tools easily. Complex lifecycle models often require new visualization methods for effective representation.

- CASE tools do not integrate with expert systems like JESS. In other words, the CASE tools do not share their knowledge with other systems. This limits the ability to make process improvement predictions through the expert systems.

- CASE tools like Microsoft Project support a limited number of data source types. For example, MS Project supports Microsoft Project, Excel and ODBC type data sources. This is a limitation when we are porting the plan details from another CASE tool running on a different platform.

B. PAMPA 2 ACS Comparison

Given these limitations of conventional CASE tools, we will see how these limitations are resolved in PAMPA 2 ACS in the following section.

- PAMPA 2 ACS provides support for adding new life cycle models to the system. The object oriented and modular design of PAMPA 2 ACS allows a manager to plug-in /configure new life cycle definitions. The open system's approach in PAMPA 2 ACS design allows any interested developer to follow the documented open interface standards of ACS and write additional plug-ins. The system is open and extensible by its design.

- PAMPA 2 ACS provides support for tracking multiple versions of the same project. In addition to Microsoft Project tables, PAMPA 2 ACS provides additional tables (PAMPA_PROJECTS, PAMPA_PROJECT_VERSIONS) that are used to store the version details of the same master project. Additional tables such as PAMPA_SLC_LIST, PAMPA_SLC_PRC_PRV maintain the lifecycle information associated with a specific project. Detailed information about the Microsoft Project plan tracking tables can be obtained from the Project MVP website [50].

- PAMPA 2 ACS has been implemented using Java Server Pages and Java Servlets. Hence it leverages the main benefits of the Java programming language such as platform independence, portability and scalability [59]. The web based, three-tier architecture allows PAMPA 2 ACS to be accessed from any World Wide Web browser that supports HTML 4.0 standards [60].

- PAMPA 2 ACS has been implemented using an agent-based approach. Agent programs continuously monitors the projects for changes and dynamically assess

the risk associated with projects. If the estimated value of risk exceeds the pre-specified limits, PAMPA 2 ACS displays the areas that require attention using special colors. For example, a manager can set the maximum permissible delay in schedule along the critical path. If the actual delay exceeds the preset limits, then the critical path is indicated with special colors so as to capture the attention of the viewer. Also additional information about the delay is displayed as tool tips.

- PAMPA 2 ACS generates interactive charts that support drilldown, tool tips, zoom in and zoom outs etc. These interactive features allow viewing of information at a detailed level (activity) or a summary level (project) or any one of the intermediate levels (version, process, sub process etc.). According to Kiem [31], the user needs to get an overview of the data. In the overview, the user identifies interesting patterns and focuses on one or more of them. For analyzing the patterns, the user needs to drill down and access details of the data. PAMPA 2 ACS supports Kiem's analysis on Visual Data Exploration. The interactive features in PAMPA 2 ACS allow the presentation of information on demand, presenting the right amount of information in every screen. Thus the viewer will easily absorb the information presented.

- The extensible architecture of PAMPA 2 ACS allows easy plug-in of additional chart types. The JSP based approach [61] facilitates zero code change for the existing modules while allowing additional modules/ pages to be plugged in to the ACS.

- PAMPA 2 ACS essentially externalizes the knowledge that is currently proprietary to the CASE tools. The knowledge so extracted can easily be shared with other systems. For example, the knowledge extracted from Microsoft Project

Plan is formatted into SAS compliant data sets and is passed to SAS for chart generation. Though PAMPA 2 ACS does not currently provide an expert system support, it has a design level support that allows expert systems such as JESS to be integrated with it.

- PAMPA 2 supports Microsoft Project Plan (.mpp) format, ODBC, Comma Separated Values (.csv) format, SAS data sets, and Excel type data sources. More importantly the open source architecture of PAMPA 2 ACS allows new data source formats to be plugged in very easily. The knowledge extracted from these data sources is stored in a common format in SQL server. This data can be can be interchanged with external systems. For example PAMPA 2 ACS generates SAS compliant data sets from Microsoft Project Plan to provide to SAS runtime for generating PERT/CPM charts. This approach facilitates knowledge sharing between different CASE tools.

C. Case Study - Web Development Project

This section demonstrates how projects can be visually managed using PAMPA II ACS. This section demonstrates the following aspects of PAMPA II ACS.

- Visualize the planning and tracking phases of a project.
- Visualize project data using conventional charting methods such as Gantt Charts and Activity Networks.
- Modify conventional charts into an interactive form where the users can drill down and get information on demand.
- Mark critical areas of projects visually with special colors so that it is distinctly visible.

- Allow new chart types such as DOT charts to be plugged in so that the project management data can be effectively visualized.
- Allow cross referencing among multiple charts so that the related information can be viewed quickly.
- Integrate various CASE tools and extract data automatically, so that the users can still use their familiar CASE tools without having to reenter the data into the PAMPA 2 system.
- Allow the users to set alert levels, schedule parameters etc. and study its effect by simulating the effects of these parameters on the projects visually.

Figure Fig 10 shows the classic approach to tracking projects. Usually, the project time sheets is updated in a spread sheet against the activities in the plan. The Gantt chart is used to track the schedule and is used to check if any activity is lagging behind schedule by comparing the current status with the baseline schedule.

Different visual mechanisms such as cost control charts, activity networks etc. are used to track different aspects of a project. The cumulative cost charts are used to see if the activities lagging behind schedule have any cost impact on the project. Cumulative cost chart also plots the current cost vs. scheduled budget vs. forecasted cost of the project. The total revenue earned by the project is displayed by the earned value charts. The dependency between various project activities is and the critical path are depicted by the activity network charts.

A single project management tool does not provide a comprehensive coverage of all aspects of project management. Each tool specializes in one aspect of management. For example, Microsoft Project is used for managing the schedule. Cost eXpert is used to manage the costs associated with a project. As a result, a man-

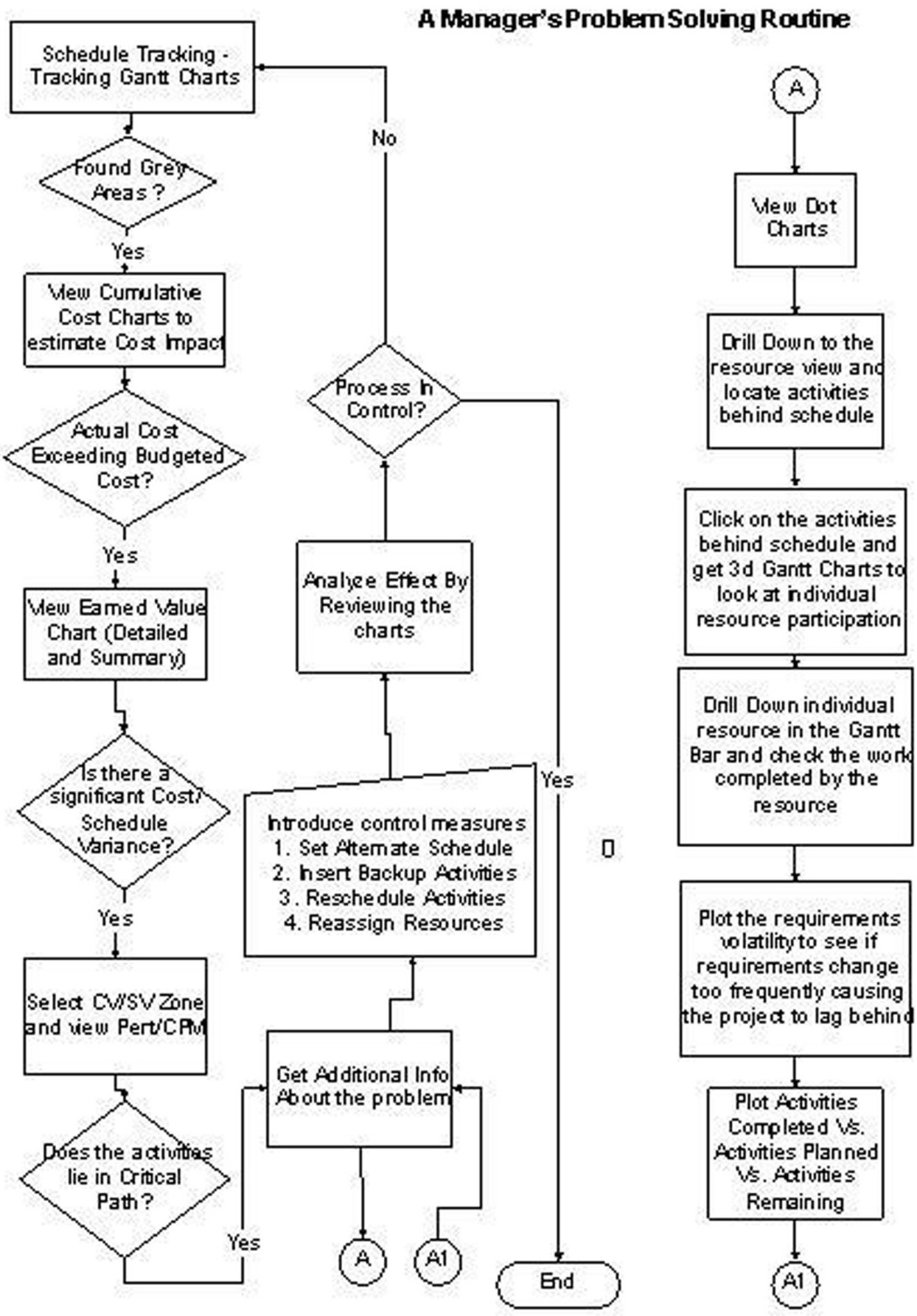


Fig. 10. Manager's Problem Solving Routine

ager has to use different tools to track different aspects of a project. PAMPA 2 ACS addresses these issues by integrating these tools to provide a comprehensive visual project management solution. Currently, PAMPA 2 ACS integrates five CASE tools including Microsoft Project, SAS, Milestone Professional, Cost Expert and Project Commander. Microsoft Project is used for managing a project's schedule. The Project Management module in Statistical Analysis Software is used to create activity networks. Milestone Professional is used to manage resource allocation and milestone information from the schedule. Cost Expert is used to manage the budget information related to a project. These are the popular CASE tools that are used by project managers in day to day project management activities. PAMPA 2 ACS not only gathers data unobtrusively from these CASE tools, but also uses their visual chart generation mechanisms to generate graphical charts.

D. Problem Solving Procedure

In a Gantt chart, each activity is tracked against its scheduled/ baseline start and end dates. In Fig 10 , When the project manager reviews a Gantt chart, he can find whether a project is lagging behind schedule. If an activity starts late or finishes late, PAMPA 2 ACS marks that activity using special colors in order to alert the viewer. task bar for that activity using special colors. The schedule information alone is not sufficient to apply control measures to lagging activities. A manager needs to compare the planned cost of the project to the actual cost to see if the project is not exceeding the budgeted costs. He also needs to forecast what will be the total cost incurred at the end of the project. This information helps in the early identification of cost overruns.This information is provided by the cumulative cost chart which plots the planned budget vs. current budget vs. forecasted budget. But cumulative cost charts

cannot reveal information like - which particular activity has the highest cost impact or what is the total earned value of the project on a detailed basis. PAMPA 2 ACS provides a new chart mechanism called as earned value chart which provides detailed activity information against its earned value. The activities in critical path present an impact in schedule and hence should be treated with the highest priority.

Additional information about the cause of the delay can be obtained by studying the resource progress charts, DOT charts, project volatility charts and other adhoc (user customizable) charts that can be generated by PAMPA 2 ACS. Thus by using PAMPA 2 ACS, a manager will be able to completely visually manage a project using its different graphical charts. The following sections we will demonstrate how to use the various PAMPA 2 ACS visual charts to track and control a project.

The project to be tracked should be imported into PAMPA 2 ACS. The process of importing extracts the details automatically from an existing Microsoft Project Plan. Currently, PAMPA 2 ACS allows only Microsoft Project plans to be imported. But the architecture of PAMPA 2 system allows new import mechanisms to be plugged in transparently. This lets the managers to plan using their favorite project management tools and still use PAMPA 2 ACS to track the projects without having to input the details of the plan again. Fig 11 shows how to import a Microsoft project plan using PAMPA 2 ACS. Already existing projects can be imported to get snapshots of the project as it is progressing. Thus we can observe the effects of our control policies on the project, by reviewing the different snapshots of the project.

The project plan must be baselined before it can be tracked. By baselining a project, we store the scheduled start and end dates of every activity in the plan and when the actual values are put in, we will be able to track them against the baselines. Fig 12 shows how the Web Development project is baselined using PAMPA 2 ACS.

We will be able to view the activities in the Project Plan through PAMPA 2

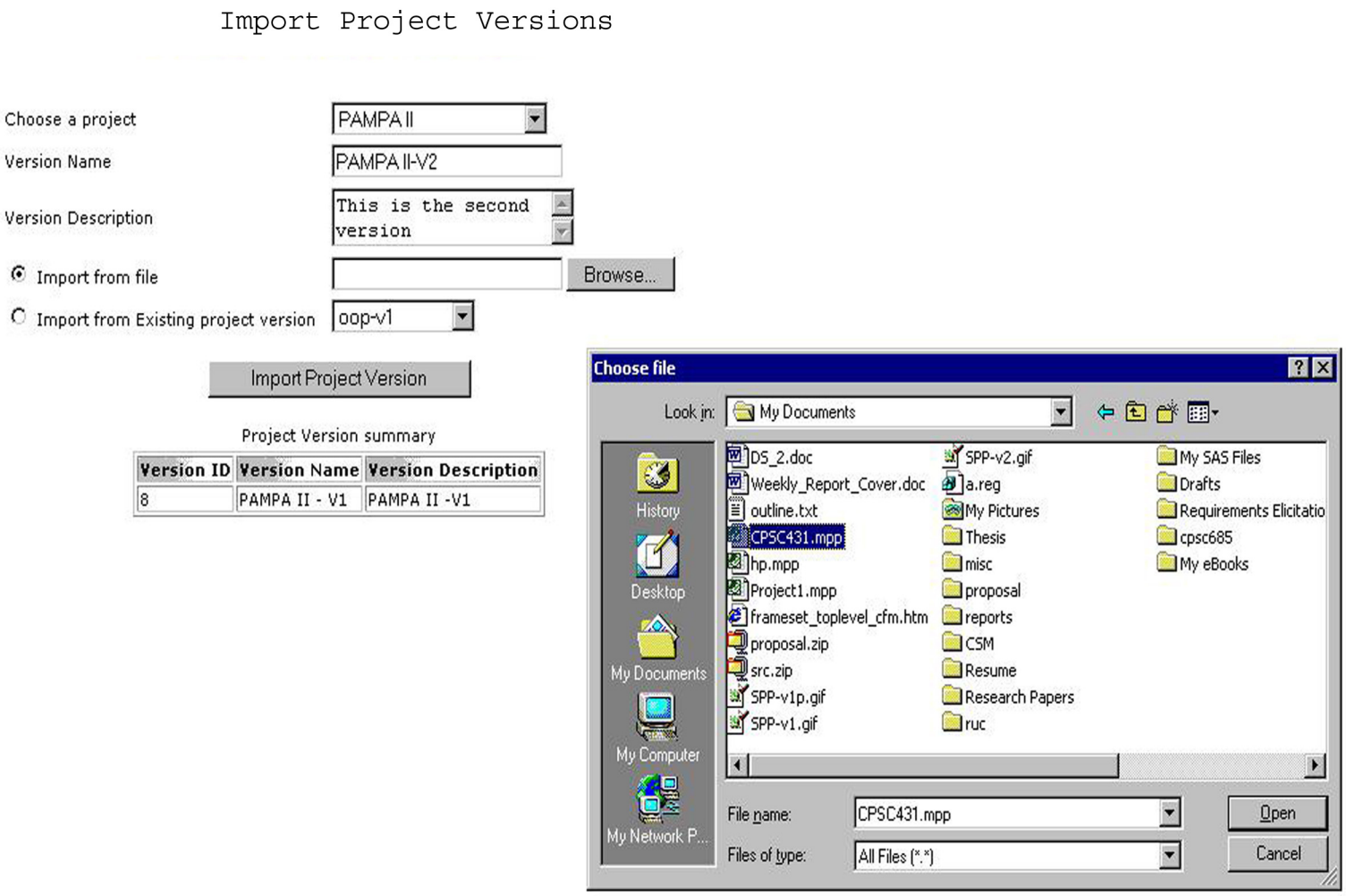


Fig. 11. Importing Microsoft Project Plan from Microsoft Project

Fig. 12. Baseline Project

ACS and modify the actual start and end dates of specific activities. Fig 13 and Fig 14 shows the activity details screen. The activities can be viewed at a detailed level or at a summary level. While the summary view shows only the planned details of an activity, detailed view allows the user to view additional information about the activity such as the actual start and finish dates, variances in duration, start and end dates. Detailed view also allows the user to edit actual values and to create new activities. This feature allows the plan to be rescheduled in case of contingencies . The Automatic rollback option allows the modified plan to be rolled back to its check point, thus allowing a project manager to experiment with alternate schedules, to perform simulations and study the effect without causing a permanent change in schedule. Fig 14 shows the activities at their detailed level, which displays information such as actual start date, actual finish date, start variance, finish variance etc. Also Activities can be listed in drilldown mode. When a manager chooses the drilldown option from the select box, he can view the summary level activity status and can progressively drill down the summary level activities to reveal information about its sub activities without overwhelming the user with the complete plan.

List Activities

Version Name : Web Development Version 2

Display Mode List All Detailed

Task ID	Task Outline Number	Task Name	Scheduled Start	Scheduled Finish	Duration day (s)
0	**0**		**11/20/2001**	**04/26/2002**	**114.0**
1	**1**	**1. Initiation (Inception) Phase**	**11/20/2001**	**12/18/2001**	**21.0**
2	**1.1**	**Concept Development**	**11/20/2001**	**12/10/2001**	**14.0**
3	1.1.1	Define Project Strategy	11/20/2001	11/21/2001	2.0
4	1.1.2	Review	11/22/2001	11/22/2001	1.0
5	1.1.3	Storyboard use case scenario design	11/29/2001	12/07/2001	6.0
6	1.1.4	Concept approved	12/10/2001	12/10/2001	0.0
7	1.2	Risk Evaluation	12/11/2001	12/11/2001	0.0
8	1.3	Client concept review	12/11/2001	12/11/2001	1.0
9	1.4	Finalize contract and project plan	12/12/2001	12/17/2001	4.0
10	1.5	Client acceptance	12/18/2001	12/18/2001	1.0
11	**2**	**Planning / Design (Elaboration) Phase**	**12/19/2001**	**02/15/2002**	**43.0**
12	**2.1**	**Preproduction**	**12/19/2001**	**01/04/2002**	**13.0**
13	**2.1.1**	**Receive package**	**12/19/2001**	**12/24/2001**	**4.0**
14	2.1.1.1	Feasibility	12/19/2001	12/19/2001	1.0
15	2.1.1.2	Implementation	12/20/2001	12/24/2001	3.0
16	**2.1.2**	**Create staging Environment**	**12/25/2001**	**12/27/2001**	**3.0**
17	2.1.2.1	Implementation	12/25/2001	12/26/2001	2.0
18	2.1.2.2	testing	12/27/2001	12/27/2001	1.0
19	**2.1.3**	**Begin Production Guide**	**12/28/2001**	**01/01/2002**	**3.0**

Fig. 13. Viewing the Activities at Summary Level

List Activities

Version Name : Web Development Version 2

Display Mode List All ☑ Detailed

Baseline Finish	Scheduled Start	Scheduled Finish	Actual Start	Actual Finish	Duration day (s)	Start Variance day(s)	Finish Variance day(s)	Edit Activity
02/06/2002	11/20/2001	04/26/2002	11/20/2001	N/A	114.0	0.0	57.0	
12/06/2001	11/20/2001	12/18/2001	11/20/2001	12/18/2001	21.0	0.0	8.0	
11/29/2001	11/20/2001	12/10/2001	11/20/2001	12/10/2001	14.0	0.0	6.0	
11/21/2001	11/20/2001	11/21/2001	11/20/2001	11/21/2001	2.0	0.0	0.0	Edit Activity
11/22/2001	11/22/2001	11/22/2001	11/22/2001	11/22/2001	1.0	0.0	0.0	Edit Activity
11/29/2001	11/29/2001	12/07/2001	11/29/2001	12/07/2001	6.0	5.0	6.0	Edit Activity
11/29/2001	12/10/2001	12/10/2001	12/10/2001	12/10/2001	0.0	6.0	6.0	Edit Activity
12/03/2001	12/11/2001	12/11/2001	12/11/2001	12/11/2001	0.0	7.0	5.0	Edit Activity
12/04/2001	12/11/2001	12/11/2001	12/11/2001	12/11/2001	1.0	5.0	5.0	Edit Activity
12/06/2001	12/12/2001	12/17/2001	12/12/2001	12/17/2001	4.0	5.0	7.0	Edit Activity

Fig. 14. Viewing the Activities at Detailed Level

Fig 15 shows the Gantt chart for the web development project. This feature can be accessed by clicking the Gantt chart link in the menu. This corresponds to the first step in the problem solving flow chart. The tracking Gantt chart displays the activity, the planned start date, planned finish date and duration. On the right hand side it displays a grey graphical bar of length proportional to its duration against its activity. It also displays the actual values as a textured blue color bar. The difference in start or end position of the planned and actual bars indicate the lag in starting or finishing the activity. The percentage of work complete is indicated by a solid blue bar which overlaps the actual value bar. Thus by looking at the Gantt chart, a manager can determine whether a project is lagging behind schedule and the activities that are causing the delay. PAMPA 2 ACS provides the following special contributions to the conventional Gantt charts.

- Tracking mode - Conventional Gantt charts does not allow tracking projects. It also does not summarize at various levels, the percentage of work that is completed. PAMPA 2 ACS adds the percentage work completed data at the mile stone level, process level and activity level. In the Gantt chart, milestone and process indicators are displayed as black bars with the percentage or work complete next to it. This provides an overview of what percentage of a milestone is complete, how much work in a process has been completed etc.

- Gantt charts do not provide the resource information, cost impact of an activity and the dependencies for a specific activity (what task should be performed before or after this activity). PAMPA 2 ACS fills in these gaps by embedding cross references for the activities in the Gantt charts. When these Gantt charts are generated by PAMPA 2 ACS, hyperlinks are embedded to other chart types. A manager can right click on any of these activities and navigate to other charts.

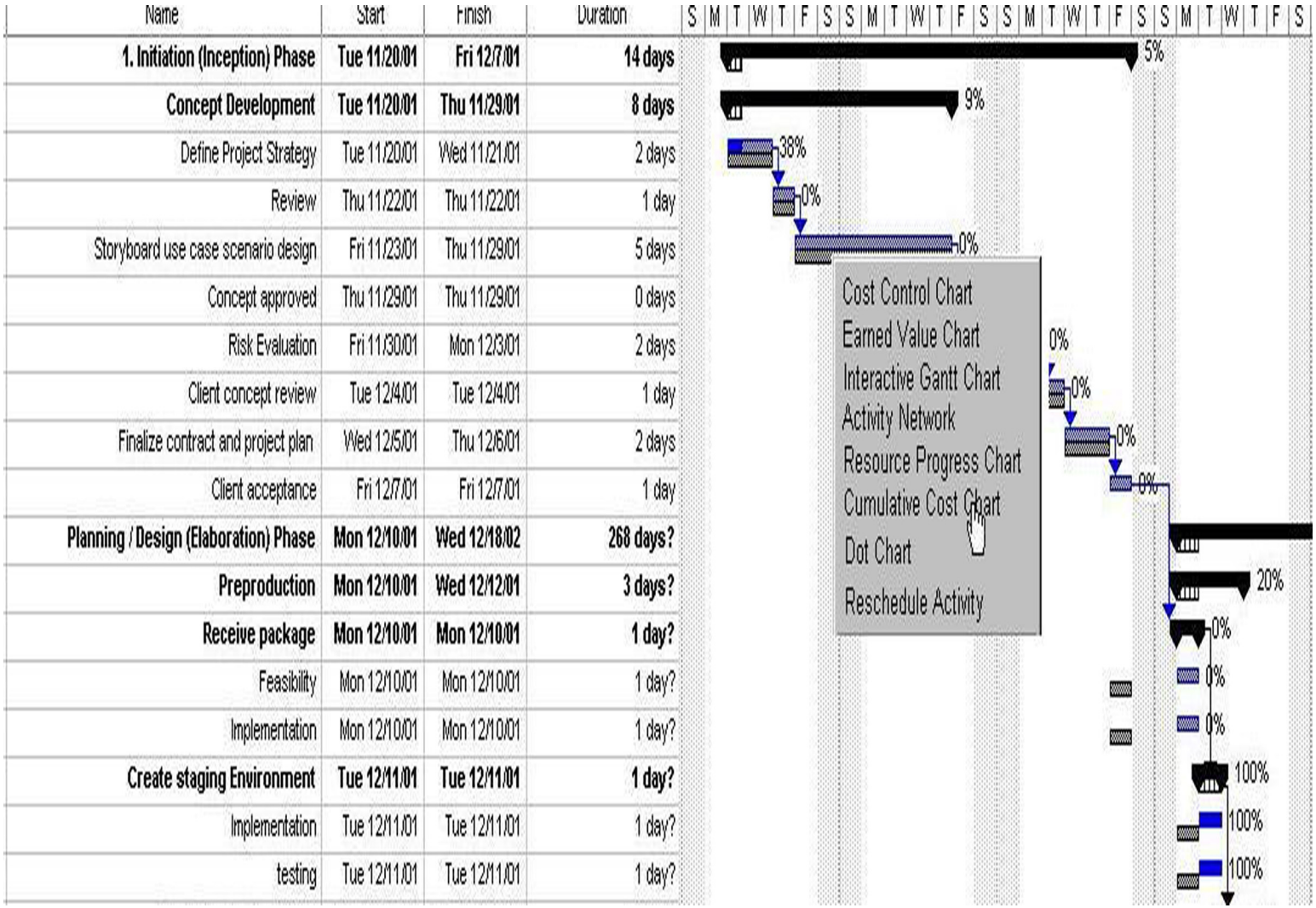


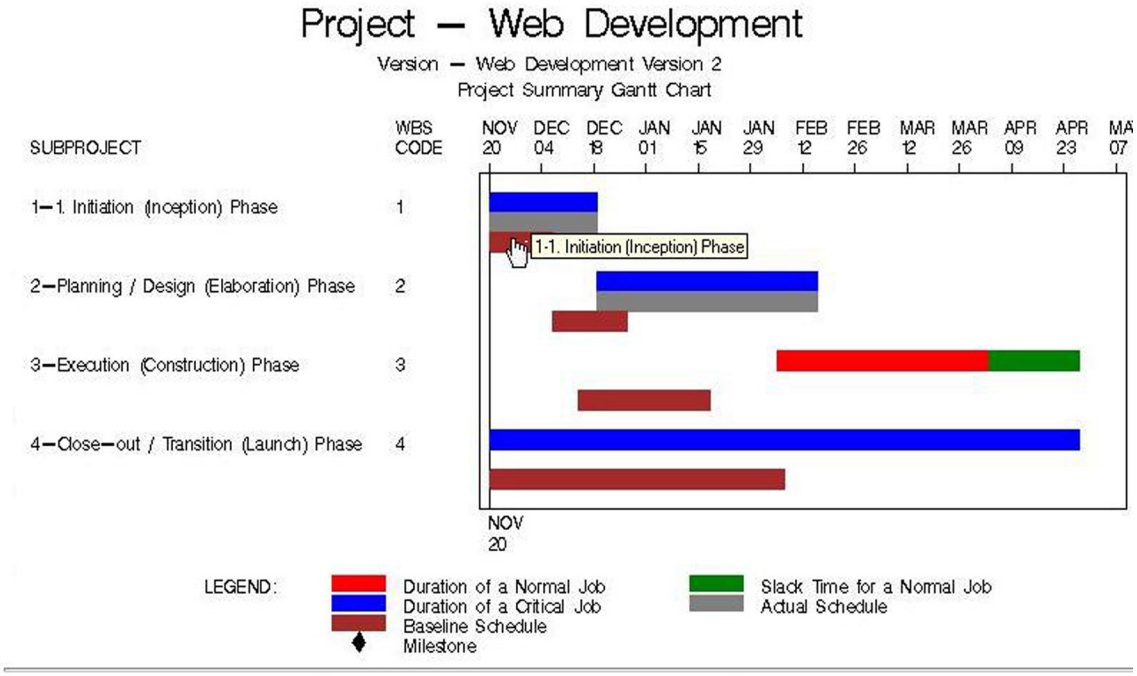
Fig. 15. Gantt Chart

Thus a manager will be able to navigate to various other charts without losing the context and thus obtain different perspectives of the problem to determine the cause of the problem. The order in which these charts are to be visited totally depends on the personal preferences and field experience of the manager. Even though the flow chart in Fig 10 specifies an order, it need not be followed verbatim.

- In cases where where there are too many activities, viewing the Gantt chart as a whole might not be effective. The end user needs to analyze and interpret data by 'Drilldown' navigation, in a meaningful way. PAMPA 2 ACS provides an interactive summary gantt chart that can be drilled down to individual/workbreakdown structure level (Fig 16.)

In order to view if the delayed activities indicated by the grey color in the Gantt chart have any schedule impact, the manager can view the cumulative cost chart in the context menu. Cumulative cost chart forecasts the cost of the project based on the current information (scheduled cost, actual cost, expected delay, cost of delay etc). Fig 17 displays a cumulative cost control chart for the web development project. We can see that the actual cost of the project does not follow the budgeted cost. There is a deviation starting from the time period where the Gantt chart indicated a delay in schedule. PAMPA 2 ACS also forecasts the spending that the manager might have to do if the project is let to continue like this. When there is a significant deviation from the planned cost and there is a sharp rise in the forecasted spending for the project, a manager should take suitable control measures.

From a cumulative cost chart we cannot learn the status of each activity. Earned Value Charts play an important part in determining the cost significance of individual activities in a project. By looking at the cost of individual activities and at the



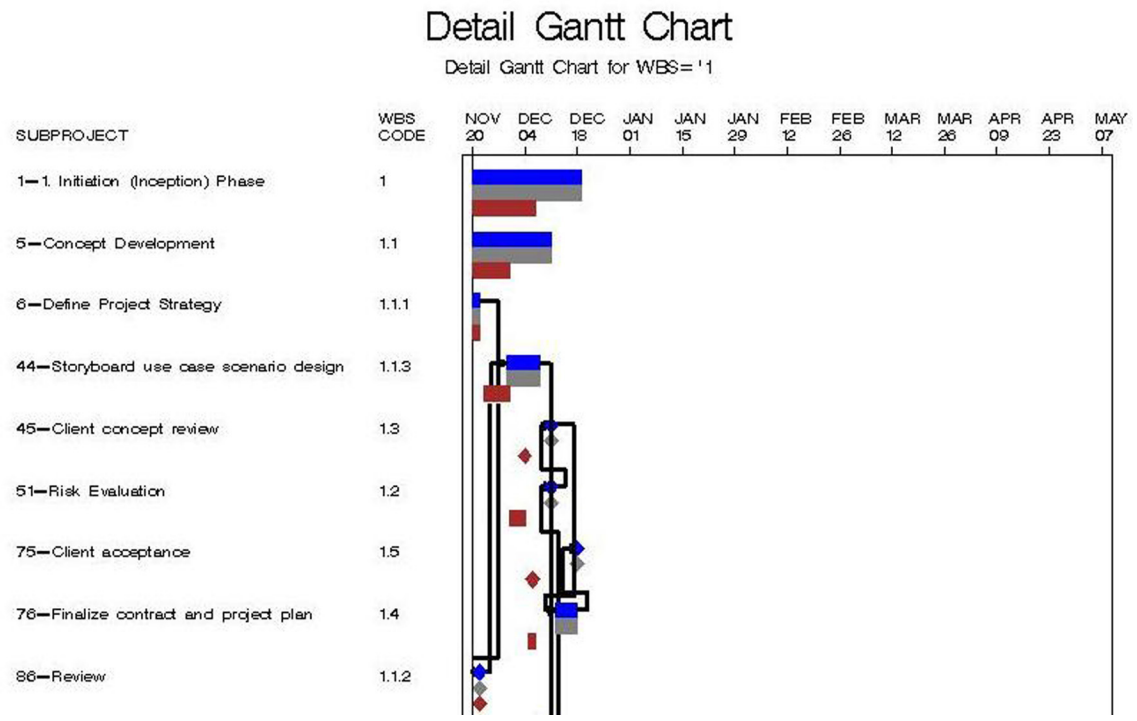


Fig. 16. Drilldown Gantt Chart

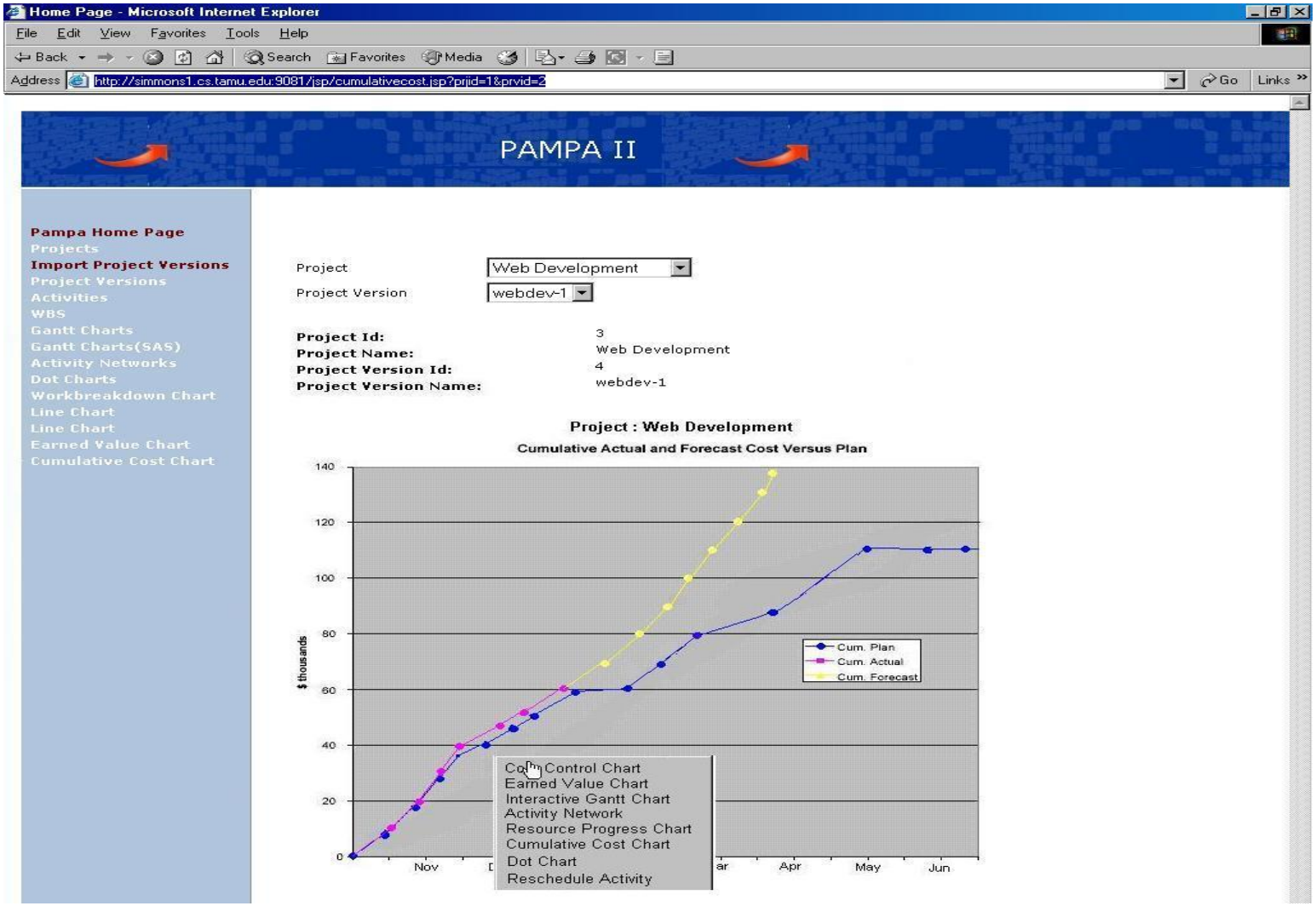


Fig. 17. Cumulative Cost Chart

summary level (milestones, processes etc), we can decide which areas need immediate attention. In order to view earned value of individual activities, PAMPA 2 ACS provides both summary and detailed earned value charts (Fig 18 and Fig 19). PAMPA 2 ACS uses the information extracted from Cost eXpert and Microsoft Project tools to draw the earned value chart. The contributions made by PAMPA 2 ACS in earned value charts are as follows:

- Earned value charts are directly superimposed against the activity Gantt charts. Hence it is easy to identify activities that are causing a decrease in earned value (an increase in spending of the project).

- Provides a tabular view of metrics such as Baseline Cost of Work Performed (BCWP), Baseline Cost of Work Scheduled (BCWS), Actual Cost of Work Scheduled (ACWS) and Percentage of Work Complete (PWC) against the earned value curves in order to easily get the numerical cost values.

- The percentage work completed (Fig 19) is displayed graphically to make it easier for the manager to locate activities having significant cost impact (that are not complete yet).

- The summary earned value charts also plots the cost and the schedule performance index (which can be calculated from the earned value) of the project. Fig 19 shows that the Cost Performance Index of the project has decreased sharply starting December - January, which is exactly during the time period when the planning phase started lagging behind the baseline dates significantly.

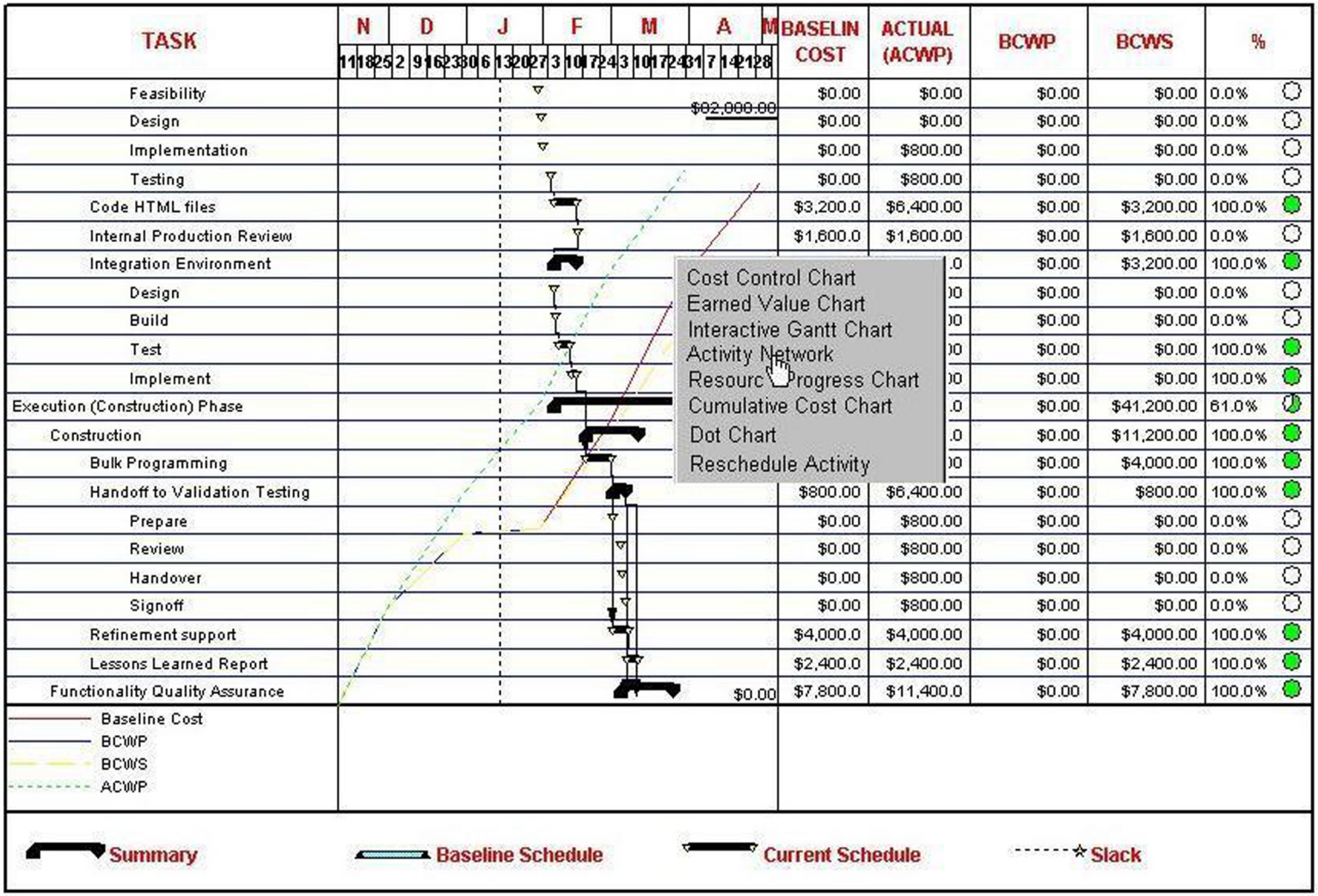


Fig. 18. Summary Earned Value Chart

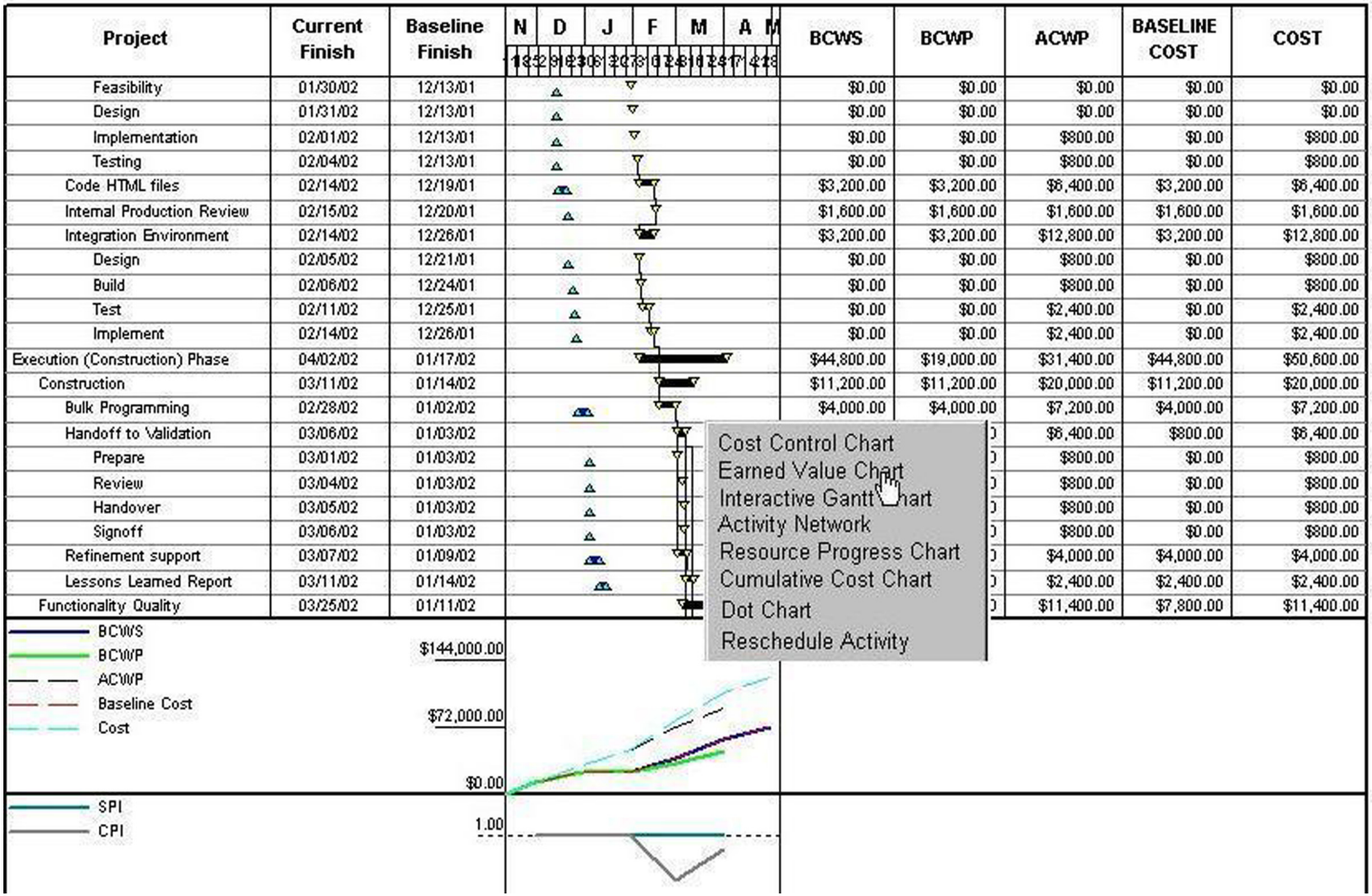


Fig. 19. Detailed Earned Value Chart

The earned value charts provides information only about the cost aspect of a project. In order to study the impact of a lagging activity on the overall plan, we need a bigger picture of the plan. PAMPA 2 ACS provides activity networks in order to provide a network view of the project. Fig 20 shows the activity network chart for the web development project. In an activity network, each node represent an activity and directional lines leading in and out of every node denote the successors and predecessors of that activity. It also displays the critical path of a project, which contains the chain of activities that take the longest time to execute. Activities that are not in the critical path can be rescheduled within the slack period without impacting the overall schedule of the project. Depending on whether an activity is in the critical path or not, the reschedule plan would change. PAMPA 2 ACS displays activities along the critical path with special colors so that it can be easily identified. Since the changes made can be easily rolled back, we can input various values for the control parameters through the web interface and simulate the effects. Using PAMPA 2 ACS the activities can be rescheduled. Fig 21 shows the reschedule activity screen. Using PAMPA 2 web interface, various control parameters such as the actual start date,actual end date, early start date, early finish date, latest start date and latest finish date can be varied for non critical activities in order to provide buffer for the critical activities and if required they can be rolled back to their original values.

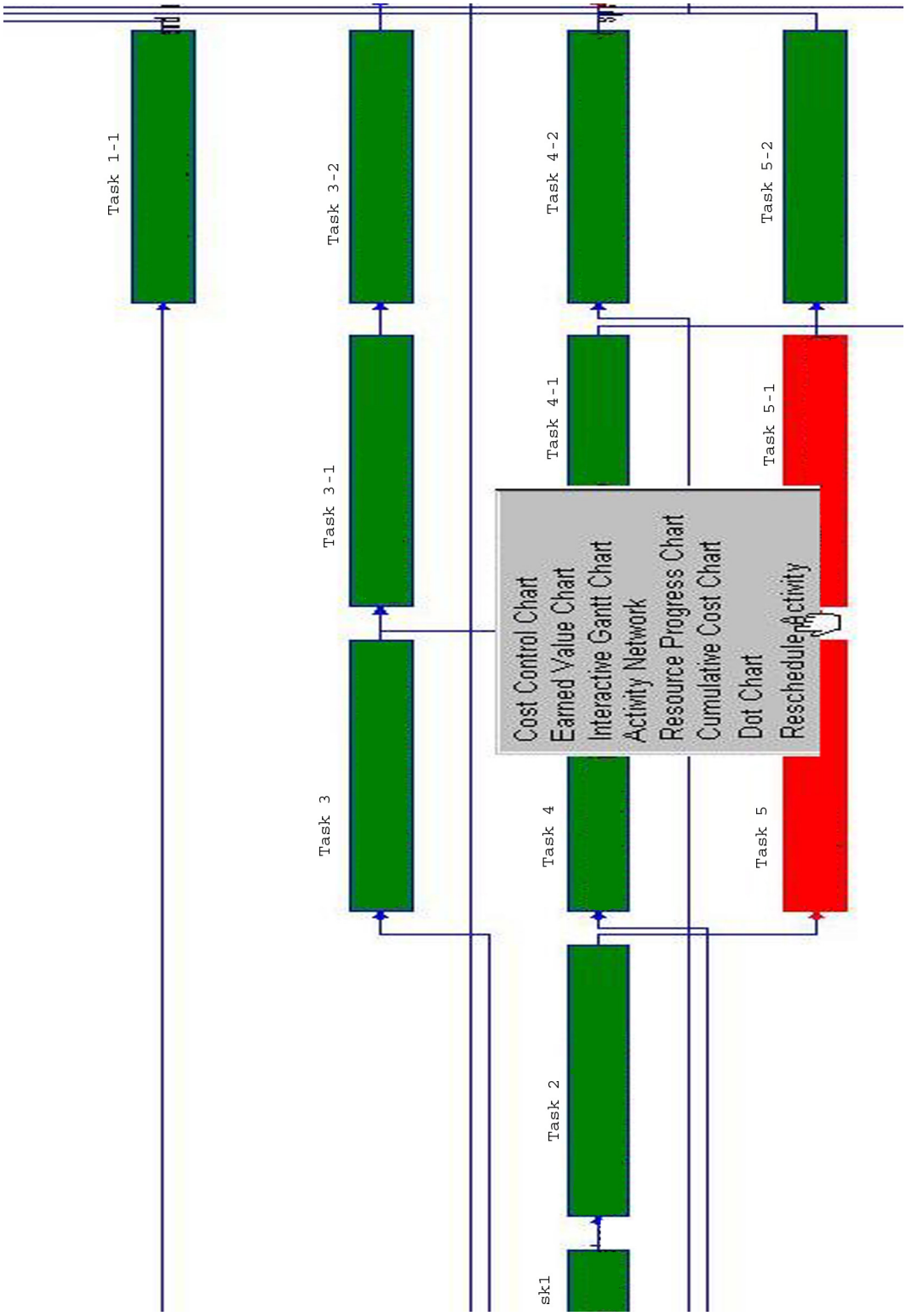


Fig. 20. PAMPA 2 Activity Network Chart

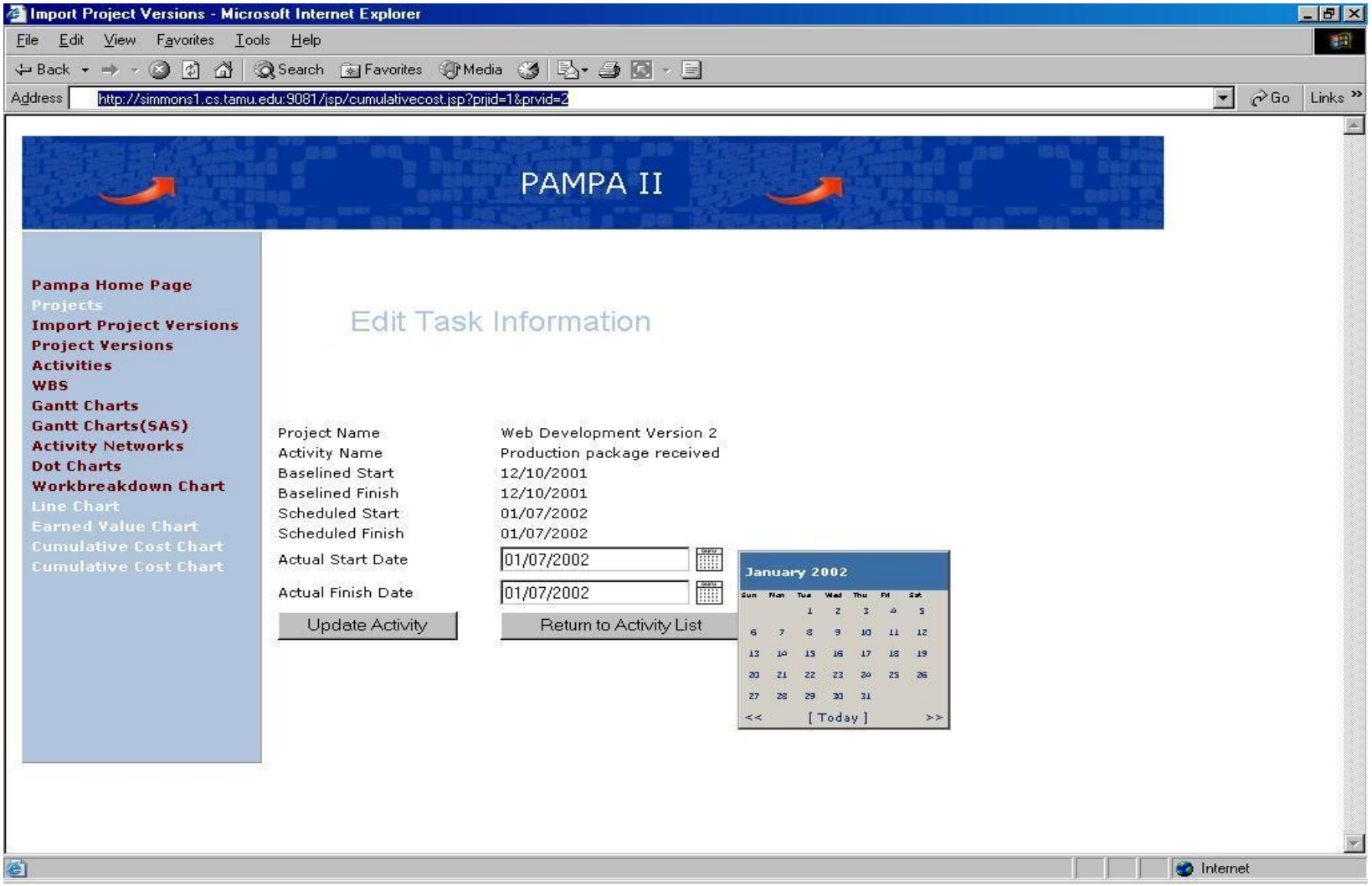


Fig. 21. Reschedule Activity Screen

Apart from conventional project management techniques, new chart types can also be plugged into PAMPA 2 ACS. We demonstrate this feature by plugging in DOT charts into the PAMPA 2 ACS framework. Dot charts let the manager view the project status at different levels. At the summary level, the project version view provides the status of different versions of a project in a single view using pie chart. For a given project or a product, there can be several versions which might simulataneously be under development. Each project version is represented as a pie. Each sector in the pie denotes the status of the processes in that project version. Fig 22 and Fig 23 displays the project version view of the same project at two different times. In Fig 22, we see two dots, one for each version. The red area indicates the percentage of activities in that version that are running behind schedule. The grey area indicates the percentage of activities that are planned but not executed. Blue area indicates the percentage of activities that are complete. Hovering the mouse over the dot reveals a tool tip, which gives detailed information about the version corresponding to that dot. The first dot in Fig 22 looks red and grey. None of the activities in this dot have completed yet. This indicates that the project is not doing well and hence needs to be controlled . The second dot in Fig 23 represents the same project after the control measures were applied. Hence, we can see that the second version of the project is in a better shape with a major portion of the dot in blue (the activities represented in blue are the completed activities).

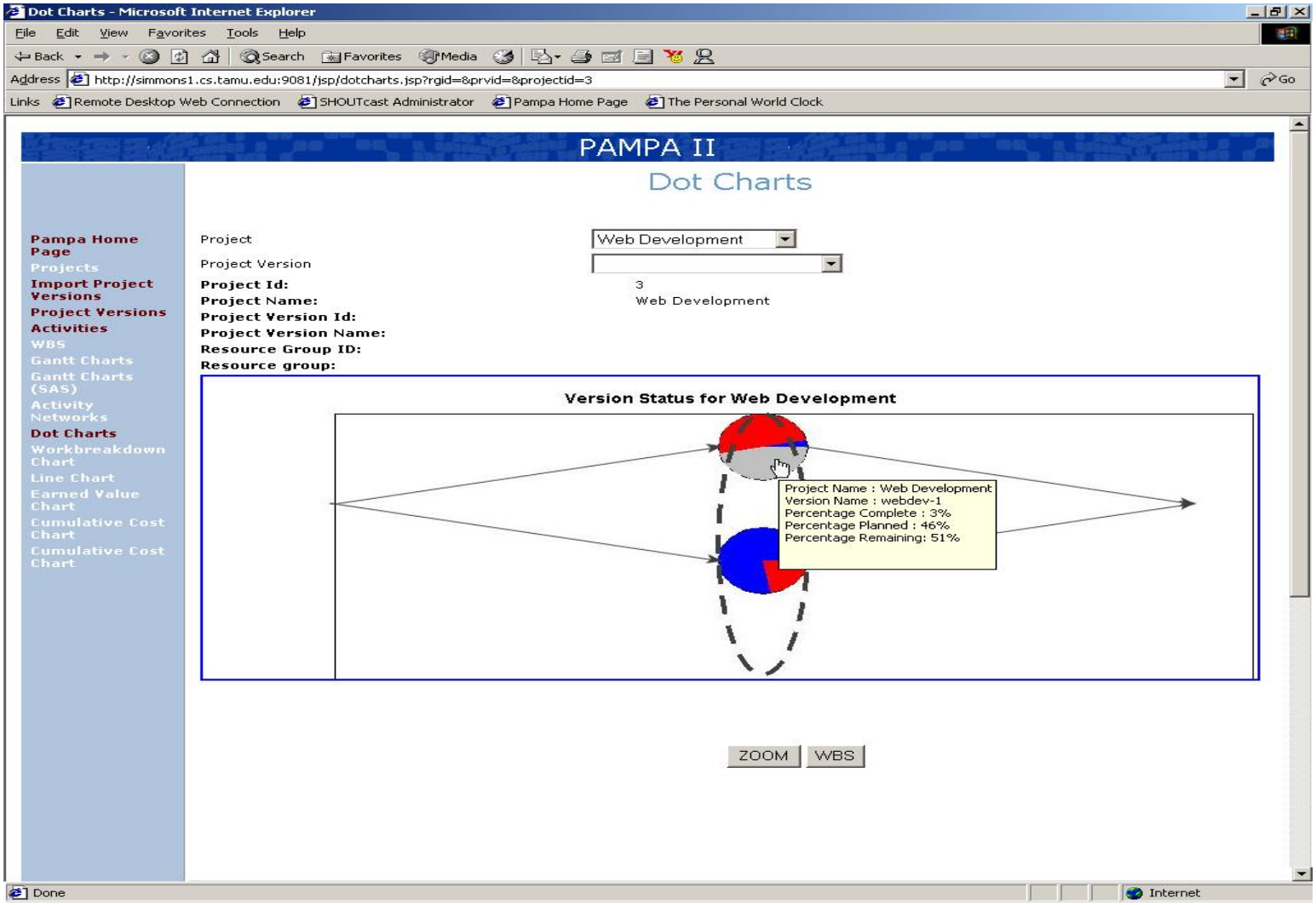


Fig. 22. DOT Chart Displaying Version Status for a Project Lagging Behind Schedule

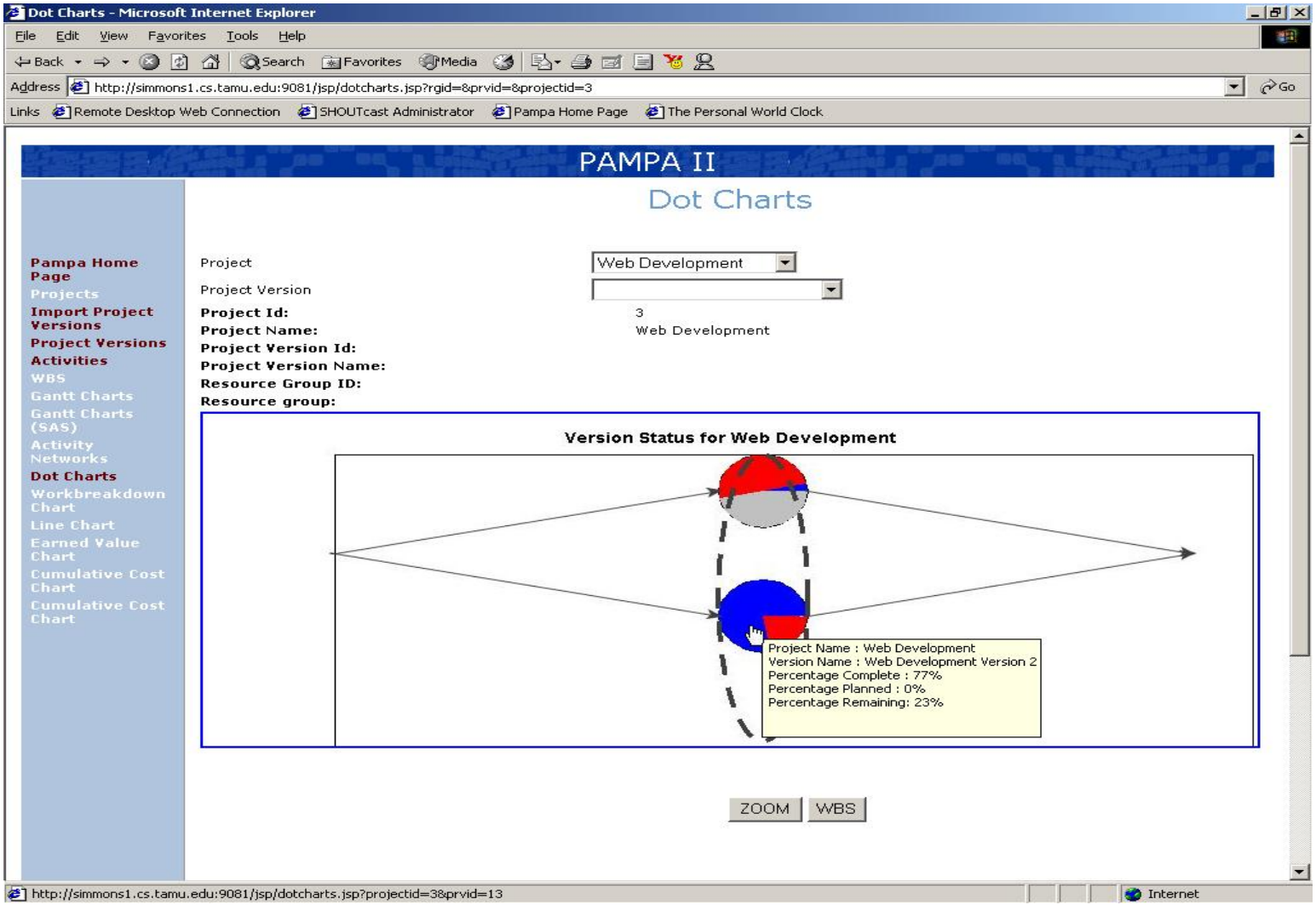


Fig. 23. DOT Chart Displaying Version Status for a Healthy Project

By clicking on the corresponding project version dot, we can see the status of different teams working in the project. This view is called as the organization view of the project version. Fig 24 and Fig 25 shows the status of various teams working in the PAMPA 2 project before and after the control measures were applied on the project.

This chart denotes the status of every team working in the project as a dot and provides a unified view of the status of individual teams. By viewing this screen, a manager would know how the various teams in his project are performing. When mouse is brought over a dot, it provides a tool tip, which reveals more information about the team and its activities. By clicking on a dot corresponding to a particular team, we will be able to get information such as activity status, the resources and other teams on which, this team is dependent to perform the activities for that team. This view is very useful in determining the productivity of various teams working in a project. In process models such as extreme programming, we will also be able to study the effect of team sizes on their productivity factors and hence select an optimal team size. we can also observe the fact that the individual team productiveness has a profound effect on the overall project itself. Thus the organization view provides a detailed breakdown of the project version status, and helps us to narrow down the problems to specific team(s).

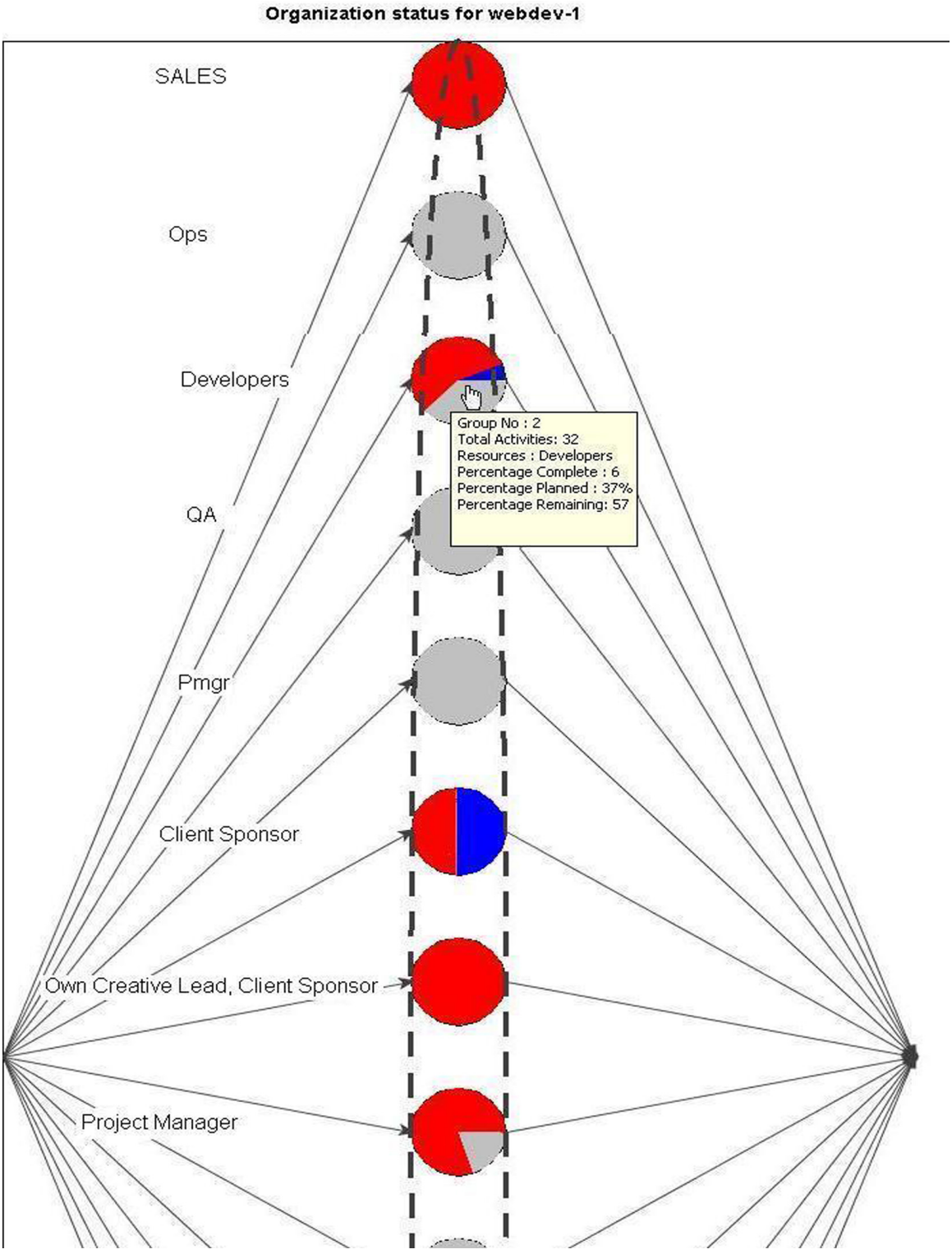


Fig. 24. DOT Chart Showing the Status of Different Teams in a Project Lagging Behind Schedule

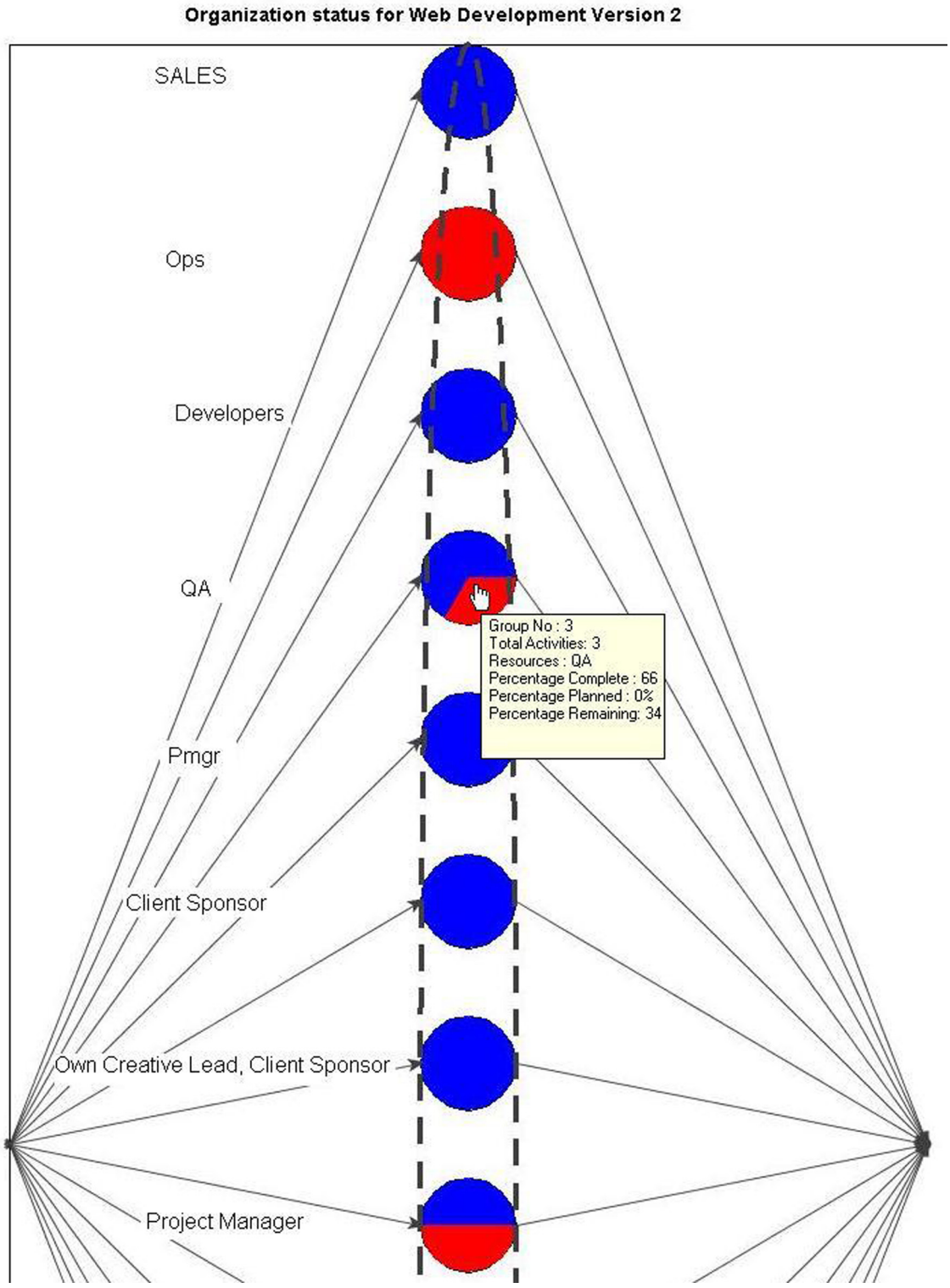


Fig. 25. DOT Chart Showing the Status of Different Teams in a Healthy Project

In the activity level view, the various activities carried out by that particular team are visualized in detail. This view also displays the dependency relationship between the activities. Bringing the mouse over the individual activities reveals further information about the activity. Through the context menu, the user will be able to navigate to the activity network or Gantt charts view for that particular activity. Fig 26 shows the activities view of dot charts for the "developers" group. From Fig 26, we can see that the developers have completed activities 2.1.2.1 and 2.1.2.2 on schedule, but are running late on all the other activities. Also we can see that the other activities are dependent on activities 2.1.2.1 and 2.1.2.2. In dot charts, each polygonal shaped dot represents a recurring pattern of activities. For example, the diamond shaped spiral has four sides which represents activities corresponding to design, implementation, alpha and beta tests, where as the sides of the triangle shaped spiral represent activities corresponding to design, implementation and testing. In a project following incremental development model, we will be able to see concentric polygonal shapes such as concentric squares, where each square represents a specific iteration. Fig 26 displays the project which follows an incremental development process and as a result of this process models, we see interesting geometric pattern. Fig 27 displays the activity view of a project following a waterfall lifecycle model. From the above examples, we can see that the dot charts allows us to breakdown the project management information into manageable chunks and displays it visually, thereby letting the user to progressively explore the information. Dot charts also visualize the life cycle of the project.

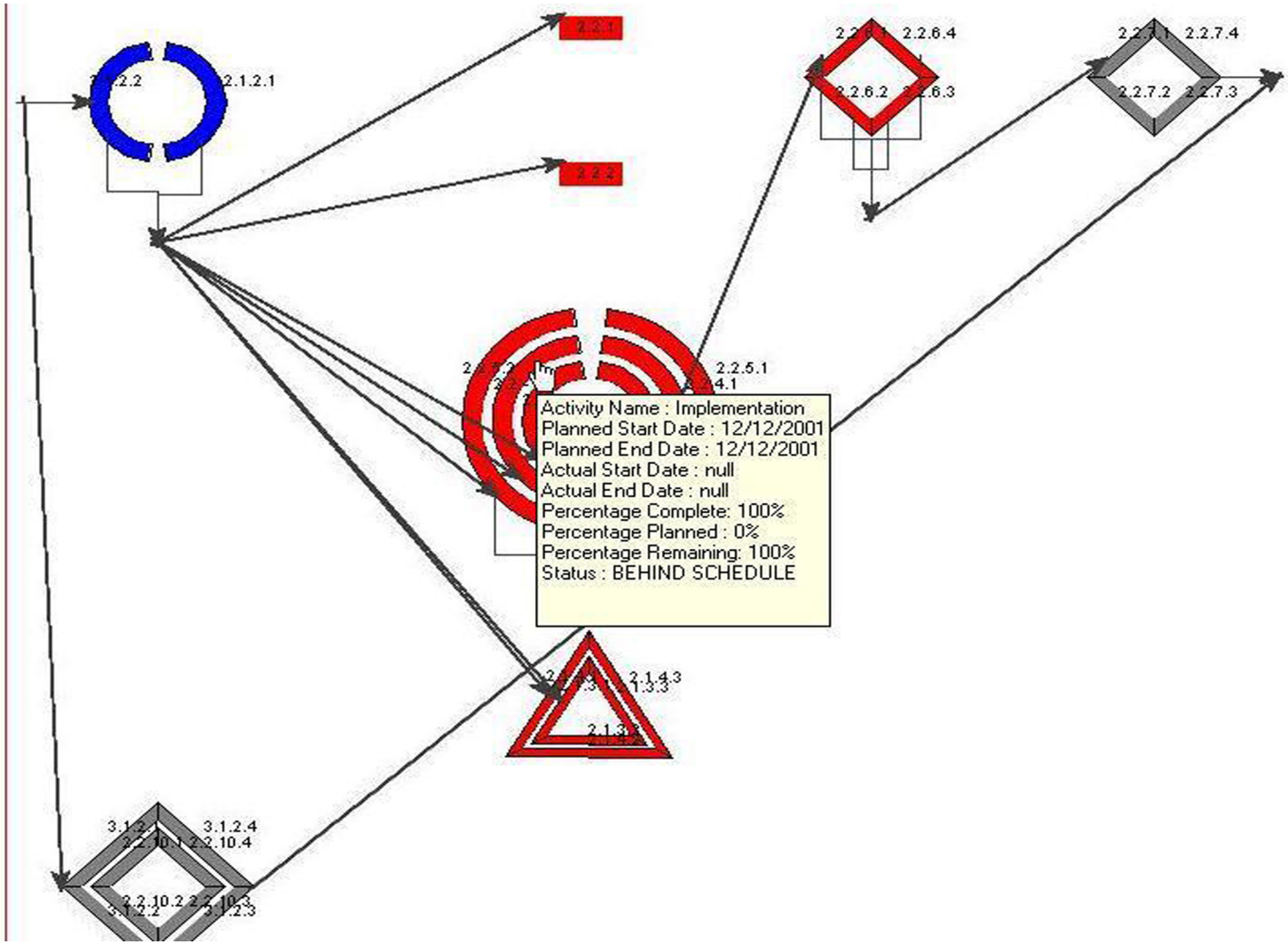


Fig. 26. DOT Chart Displaying the Activities View for Project Following Spiral Model

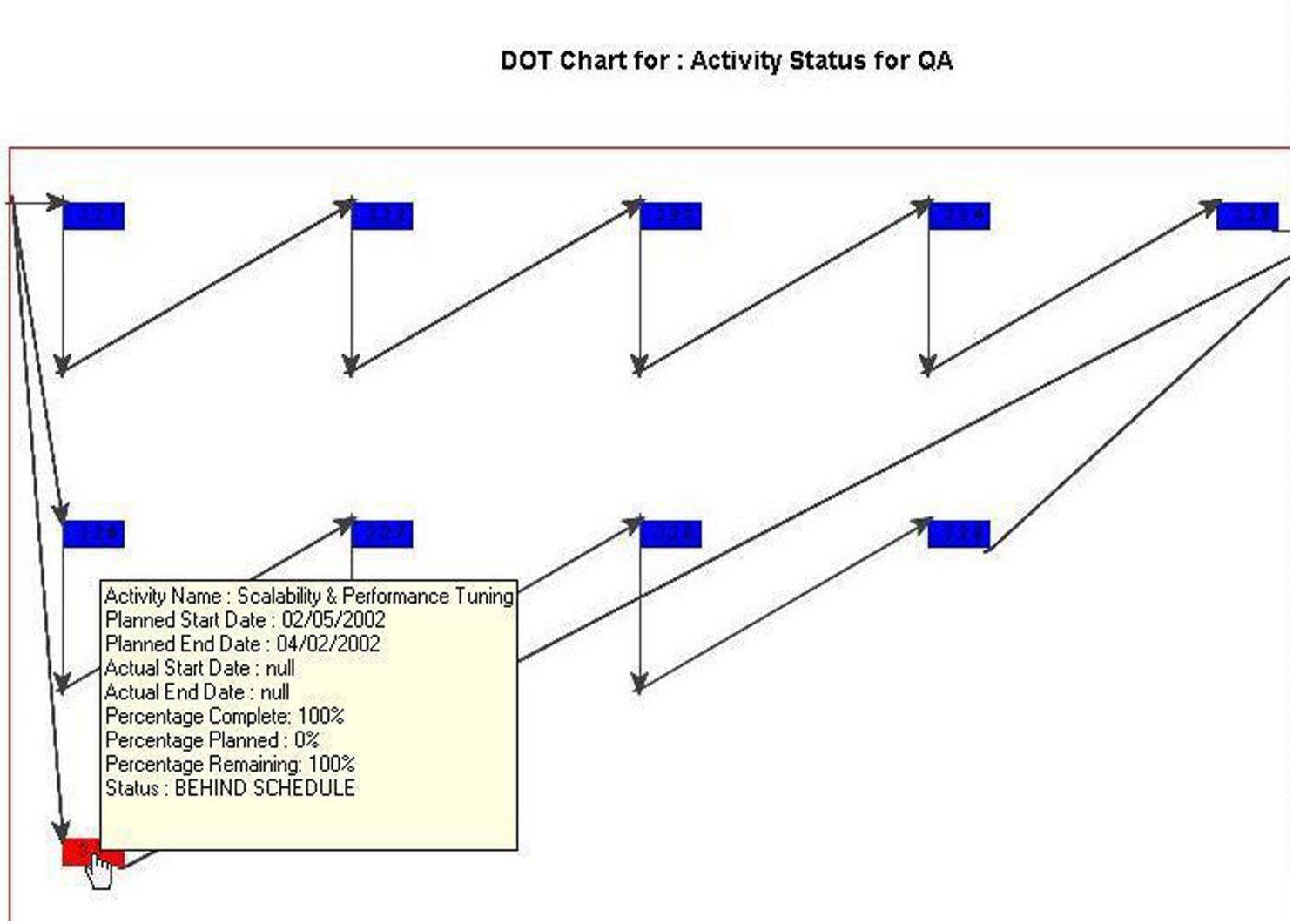


Fig. 27. DOT Chart Displaying the Activities View for Project Following Waterfall Model

PAMPA 2 ACS provides a detailed resource level view, which lists the status of various activities against the resources responsible for performing the activities . For each resource, this chart represents graphically the percentage of work completed, project status etc. Using this chart, a manager will be able to track the resources working on a specific activity, their progress and the percentage of work completed by a specific resource. Fig 28 shows a detailed resource view chart.

In addition to this, ad hoc charts can be generated by PAMPA 2 ACS. The user can plot the project parameters against project dates in the form of line graphs. For example, we can generate an activity progress chart which compares the number of activities that were planned ,number of activities that have actually completed ,and number of activities that are remaining. The plots will be against the project dates. This is a trend graph which displays the status of the project in terms of the number of activities to be completed or that have already completed. These trends can be projected to visualize the future state of the project. If the activities planned and activities scheduled curve do not have a large variance, then we can safely assume that a project is on track. Fig 29 displays how the parameters are set for plotting the activity progress graph.Fig 30 displays a line chart indicating activity progress.

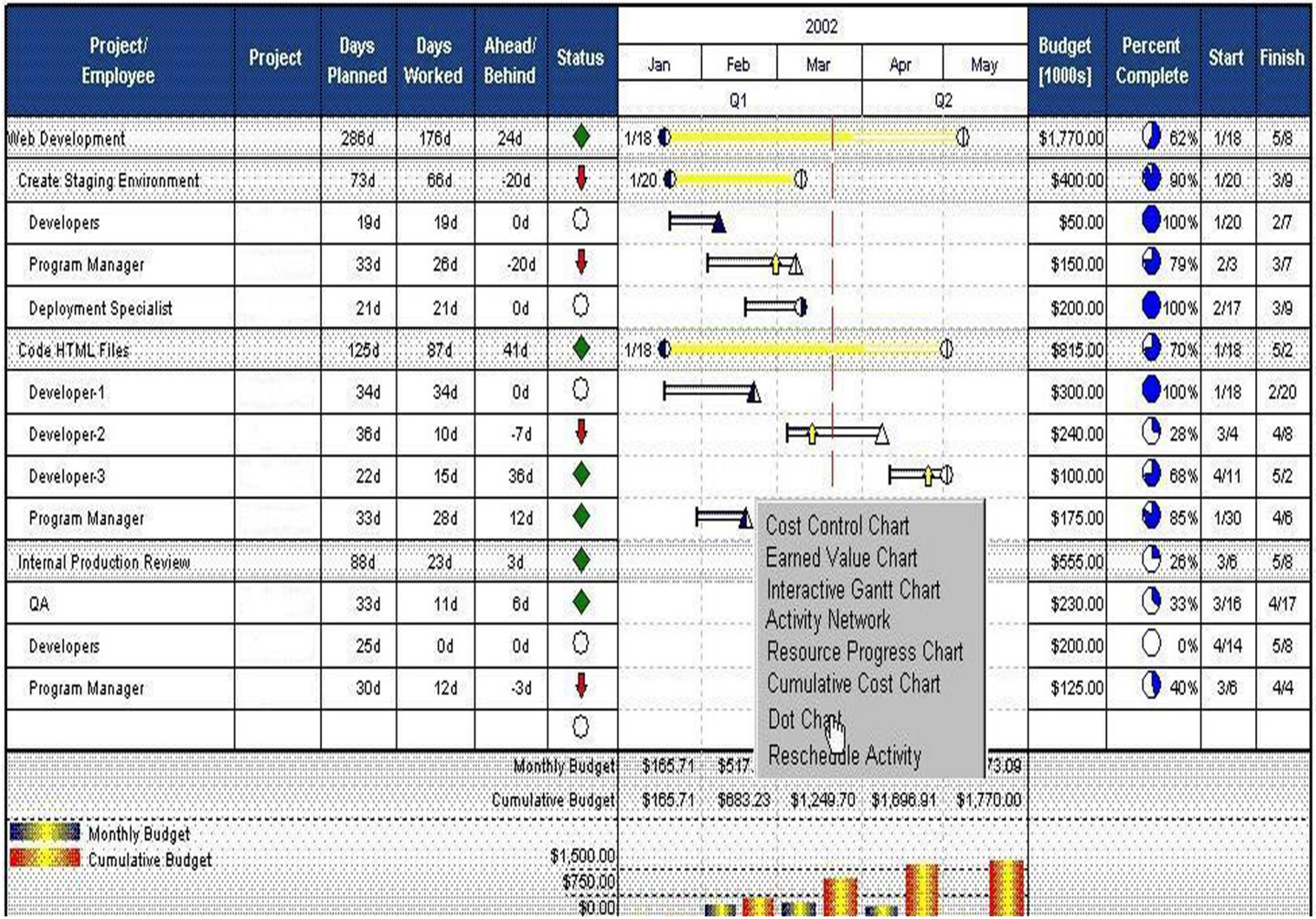


Fig. 28. Detailed Resource View Chart

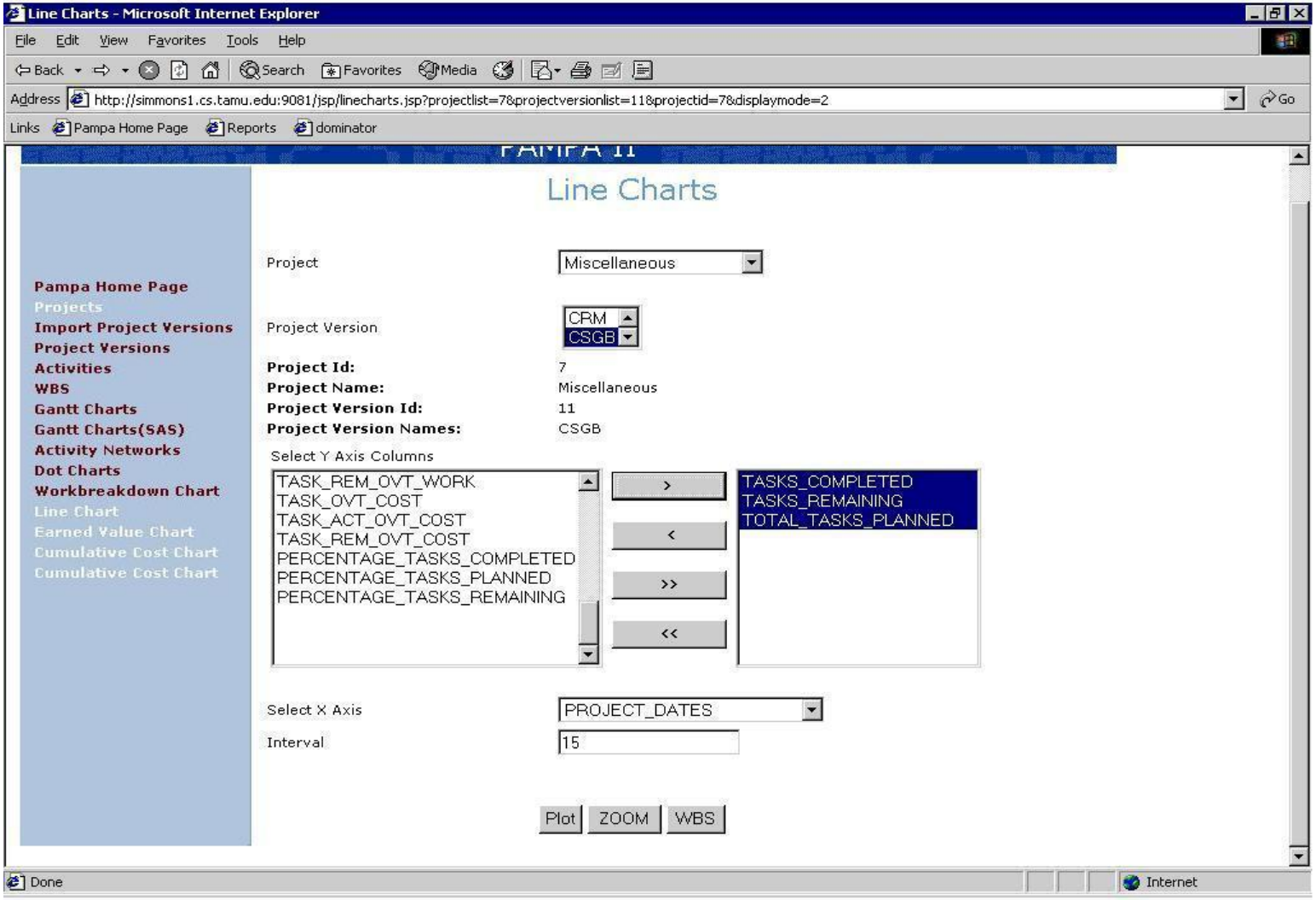


Fig. 29. Line Chart Parameter Configuration

These charts help a manager visualize the status of a project better than standard CASE tools, which provide only Gantt chart, and activity network view of the project. New chart types such as dot charts combine the benefits of both Gantt Charts and activity networks and also provides interactive features such as drill down and multi level view of the project. Hence it can be a useful tool for a manager tracking projects involving complex life cycles.

E. Future Work

PAMPA 2 ACS currently provides support for only a limited number of life cycle models. PAMPA 2 is extensible by design and can support plug and play of new life cycle definitions and their corresponding visualizations. PAMPA 2 ACS provides a framework that can support evolving SLC's such as eXtreme Programming [33]. PAMPA 2 ACS also provides design level support for expert system integration. Java based expert systems such as JESS can be easily integrated with PAMPA 2 ACS. This could be used in making process improvement predictions and hence help a manager in making decisions. Thus PAMPA 2 ACS can evolve as a decision support system than just being a visualization tool. PAMPA 2 ACS can also be extended to provide support for additional CASE tools such as Primavera [62], Metrics Center [63], Cost expert [64], SLIM etc. PAMPA 2 ACS's visualization capabilities can be used in areas other than life cycle management such as estimation, risk analysis. Though PAMPA 2 ACS unobtrusively gathers project data, human inputs are still required to update the project plan. One potential area of improvement could be analysis of artifacts to update the project plan. For example, PAMPA 2 ACS can analyze requirements, design and source code documents from predefined locations to estimate how much amount of work has been done. Once this has been estimated, PAMPA 2 ACS can

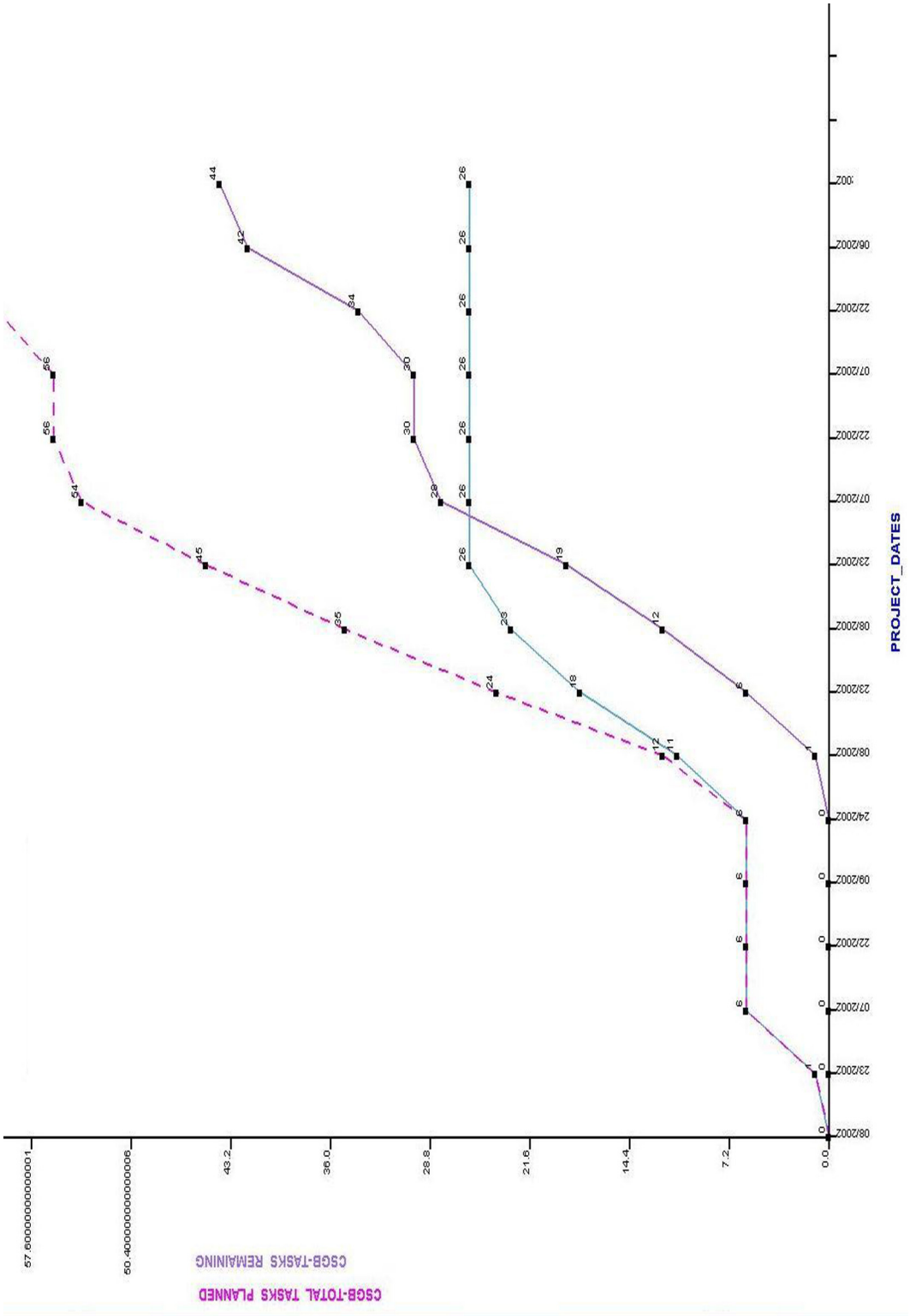


Fig. 30. Line chart Indicating Activity Progress

automatically update the project plan. Hence it can track the actuals against the planned/baselined values totally eliminating external inputs.

F. Conclusion

In this thesis, we established the need for a visualization tool that helps in depicting the project life cycle, controlling and tracking a project. We showed the limitations of conventional project management tools in the area of project visualization. We studied the relevant work in the field of life cycle models, existing systems, CASE tools and project visualization. Based on the previous work, we used the PAMPA 2 Object model to develop the PAMPA 2 Advanced Charting System. We explained the basic and extended PAMPA 2 object model to facilitate the understanding of PAMPA framework. We explained the basic techniques in Project Visualization. We described how PAMPA 2 ACS supports basic charting types such as Gantt Charts, Activity Networks, and Work Breakdown Structure charts. We also explained their variants and how they are useful in tracking a project. We discussed the limitations of these basic charting types. We also showed how dot charts complement the basic charting types in visualizing project life cycles and thereby overcome the limitation of conventional charts. We explained the design, architecture and implementation of PAMPA 2 ACS. Finally, we made a comparison between conventional project management tools and PAMPA 2 ACS and we concluded with a sample application demonstrating how projects can be tracked visually using PAMPA 2 ACS. We also suggested future improvements to PAMPA 2 ACS.

PAMPA 2 ACS gathers data unobtrusively from CASE tools such as Microsoft Project and generates special charts such as DOT charts and hence promotes the understanding of complex life cycles. DOT charts overcomes the limitation of con-

ventional project management charts and provides additional interactive features such as drilldown, zoom in and out, and tool tips. This helps in the better understanding of the project dependencies and easy assimilation of project management data.

REFERENCES

[1] E.F.Weller, "Using metrics to manage software projects," *IEEE Computer*, vol. 27, no. 9, pp. 27–33, 1994.

[2] C.Jones, "Patterns of large software systems: failure and success," *IEEE Computer*, vol. 28, no. 3, pp. 86–87, 1995.

[3] Hiroko Fujihara D.B. Simmons, Newton C. Ellis and Way Kuo, *Software Measurement: A Visualization Toolkit for Project Control & Process Improvement*, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[4] N. E. Fenton, "Software metrics; theory, tools and validation," *Software Engineering Journal*, vol. 5, no. 1, pp. 65 – 78, 1990.

[5] B.W.Boehm, "Software risk management: Principles and practices," *IEEE Software*, vol. 8, no. 1, pp. 32 – 41, 1991.

[6] H.Mooz K.Forsberg and H.Cotterman, *Visualizing Project Management : A Model for Business and Technical Success*, John Wiley & Sons, New York, NY, 2000.

[7] M.Dorfman and R.H.Thayer, *Standards, Guidelines and Examples of System and Software Requirements Engineering*, IEEE Computer Society Press, Los Alamitor,CA, 1998.

[8] S.G.Eick M.C.Chuah, "Managing software with new visual representations," in *Proceedings of IEEE Symposium on Information Visualization*, 1997, pp. 30–37.

[9] H.D.Mills and P.B.Dyson, "Using metrics to quantify development," *IEEE Software*, vol. 7, no. 2, pp. 15 – 16, 1990.

[10] IEEE, *1058-1-1994,IEEE Standard for Software Project Management Plans*, IEEE, New York, NY, 1994.

[11] IEEE, *P1058.1 R,Planned IEEE Standard for Software Management Plans*, IEEE, New York, NY, 2000.

[12] IEEE/EIA, *12207-0-1997,IEEE Standard for Software Lifecycle Processes*, IEEE, New York, NY, 1997.

[13] B.W.Boehm, "A spiral model of software development and enhancement," *IEEE Computer*, vol. 21, no. 5, pp. 61–72, 1988.

[14] T.DeMarco B.W.Boehm, "Software risk management," *IEEE Software*, vol. 14, no. 3, pp. 17–19, 1997.

[15] "Rational unified process best practices for software development teams," Tech. Rep. TP026B, Rev 11/01, Rational Corporation, 2001, http://www.rational.com/media/whitepapers/rup_bestpractices.pdf.

[16] S.Nageshwaran, "Test effort estimation using use case points," in *Conference, San Francisco, California, USA , May/June 2001, Proceedings*, 2001, vol. 1, pp. 51–60.

[17] "The estimation of effort based on use cases," Tech. Rep. TP171, Rational Software Corporation, 2002, http://www.rational.com/media/whitepapares/finalTP171.pdf.

[18] W.Ching-She, "Software project planning associate: A knowledge based approach for dynamic software project planning and tracking," in *Twenty-Fourth Annual International Computer Software and Applications Conference, Taipei, Taiwan*, October 2000, pp. 25–27.

[19] D.B.Simmons, "A win-win metrics based software management approach," *IEEE Transactions on Engineering Management*, vol. 39, no. 1, pp. 32–41, 1992.

[20] R.L.Chen, "Computerized life cycle advising, monitoring and predicting (clamp)," Ph.D. dissertation, Texas A&M University, Department of Computer Science, 1985.

[21] T.D.Escamilla D.B.Simmons, N.C.Ellis, "Manager associate," *IEEE Transactions on Knowledge an Data Engineering*, vol. 5, no. 3, pp. 426–438, 1993.

[22] "Save time, improve project predictability using rational projectconsole," Tech. Rep. D194A, Rational Software Corporation, 2002, http://www.rational.com/products/projectconsole/D194A_ProjectConsole.pdf.

[23] "Use case management with rational rose and rational requisitepro," Tech. Rep. 418, Rational Software Corporation, 2002, http://www.rational.com/products/whitepapers/418.jsp.

[24] "Microsoft project 2000 enterprise project planning workbook v2," Tech. Rep. 418, Microsoft Corporation, 2000, http://www.microsoft.com/project.

[25] S.G.Eick N.Gershon, S.Card, "Tutorials: Information visualization tutorial," in *Conference on Human Factors in Computer Systems, Pittsburg, PA*, 1999, pp. 51–60.

[26] M.Ankent D.A.Kiem, "Visual data mining and exploration of large scale databases," in *ECML Tutorial, PM08, Pittsburg, PA*, 2001, pp. 20–30.

[27] M.Ward G.Grienstein, "Introduction to data visualization," in *IEEE Visualization Tutorial, Phoenix, AZ*, 1997, pp. 31–33.

[28] D.P.Tegarden, "Business information visualization," *Communications of the AIS*, vol. 1, no. 1, pp. 123–143, 1999.

[29] Y.Yamamoto K.Nakakoji, A.Takoshima, "Cognitive effects of animated visualization in exploratory visual data analysis," in *Fifth International Conference on Information Visualization(IV01), London , England*, 2001, pp. 12–59.

[30] J.Mikael, "Information drill-down using web tools," Tech. Rep. 1, Advanced Visual Systems, 2001, http://www.uniras.dk/info/seminars/Drilldown.htm.

[31] D.A.Kiem, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 1, pp. 138–157, 2002.

[32] W.Ching-She D.B.Simmons, "Plan tracking knowledgebase," in *Proceedings of Twenty-Fourth Annual International Computer Software and Applications Conference, Taipei, Taiwan*, October 25-27,2000, pp. 58–70.

[33] K.Beck, *Extreme Programming Explained : Embrace Change*, Addison and Wesley, Reading , MA, 2000.

[34] G.McLeod and D.Smith, "Managing information technology projects," Tech. Rep. 1, Course Technology, 1996, http://www.coursetechnology.com.

[35] K.Harold, *Project Management: A Systems Approach to Planning, Scheduling and Controlling*, John Wiley and Sons Inc, New York, NY, 1998.

[36] D.K.Anton, *Practical Project Management*, Random House Business Division, New York, NY, 1987.

[37] J.Heizer and B.Render, *Principles of Operations Management*, Prentice Hall International, Edison, NJ, second edition, 1997.

[38] P.Bennet B.Lientz and K.P.Rea, *Project Management for the 21st Century*, Academic Press, San Diego, CA, 1995.

[39] *MIL-HDBK-881, Work Breakdown Structure*, Military Standard, Washington, DC, 1998.

[40] D.B.Simmons, "Visualization tool for managing projects," Tech. Rep. 1, Software Process Improvement Laboratory,TAMU, 1999, http://csl16.cs.tamu.edu.

[41] W.W.Royce, "Managing the development of large software systems," *IEEE Computer*, vol. 7, no. 1, pp. 138–157, 1970.

[42] *SPC-93098-CMC,Version 01.00.06,Process Engineering with the Evolutionary Spiral Process Model*, Software Productivity Consortium, Maryland, VA, 1994.

[43] W.E.Royce, "Trws ada process model for incremental development of large software systems," in *Proceedings of International Conference for Software Engineering*, 1990, pp. 2–11.

[44] J.W.Bailey T.P.Frazier, "The cost and benefits of domain oriented software reuse: Evidence from the starts demonstration projects," Tech. Rep. P-3191, Institute for Defence Analysis, Washington, DC, 1996.

[45] F.Martin, *Refactoring: Improving the Design of Existing Code*, Addison and Wesley Co., Reading , MA, third edition, 1999.

[46] A.O.Ramirez, "Three-tier architecture," *Linux Journal*, vol. 1, no. 75, 2000.

[47] P.Fraternali, "Tools and approaches for developing data intensive web applications: A survey," *ACM Computing Surveys*, vol. 31, no. 3, 1999.

[48] "Statistical analysis software," Tech. Rep. 1, SAS Corporation, 2002, http://www.sas.com/products/index.html.

[49] "Ods document reference," Tech. Rep. 1, SAS Corporation, 2003, http://support.sas.com/rnd/base/topics/odsdocument/rf.html.

[50] "Microsoft project 2000 database format," Tech. Rep. 1, Project MVP, 2000, http://www.mvps.org/project/projdb.htm.

[51] "Integrated object model," Tech. Rep. 1, SAS Corporation, 2003, http://support.sas.com/rnd/itech/doc/dist-obj/iom.html.

[52] "Macromedia flash mx," Tech. Rep. 1, Macromedia Corporation, 2001, http://www.macromedia.com/software/flash/index.html.

[53] "Overview of the jni," Tech. Rep. 1, Sun Microsystems Inc., 1999, http://java.sun.com.docs/books/tutorial/native1.1/concepts/index.html.

[54] "Automation through the microsoft project 2000 object model," Tech. Rep. 1, Microsoft Corporation, 2000, http://msdn.microsoft.com/library/default.asp?note=/library/en-us/dnproj2002/html/pro02aut.asp.

[55] D.Adler, "The jacob project: A java-com bridge," Tech. Rep. 1, 1999, http://danadler.com/jacob.

[56] "Java database connectivity," Tech. Rep. 1, Sun Microsystems Inc., 1999, http://java.sun.com.docs/books/tutorial/jdbc.

[57] G.Hamilton S.White, M.Fischer and M.Hapner, *JDBC API Tutorial and Reference,Universal Data Access for the Java Platform*, Addison Wesley Longman, Inc., Reading, MA, second edition, 1999.

[58] "Microsoft odbc," Tech. Rep. 1, Microsoft Corporation, 1997, http://www.microsoft.com/data/odbc/default.htm.

[59] J.Gosling and H.McGilton, "Java language environment," Tech. Rep. 1, Sun Microsystems Inc., 1996, http://java.sun.com/docs/white/langenv/Intro.doc.html.

[60] "W3c remommendation for html," Tech. Rep. 1, World Wide Web Consortium, 1999, http://www.w3.org/TR/html4.

[61] "Java server pages," Tech. Rep. 1, Sun Microsystems Inc., 2000, http://java.sun.com/products/jsp.

[62] "Using primavera," Tech. Rep. 1, Primavera Inc., 2000, http://www.primavera.com.

[63] "Using the metrics center," Tech. Rep. 1, Metrics Center, 2000, http://www.marinres.com/pg51doc/user/user22.pdf.

[64] "Cost estimating software process tool," Tech. Rep. 1, Cost Xpert Inc., 2000, http://www.costxpert.com.

VITA

Prabhu Anand Inbarajan received his Bachelor of Engineering in Electrical and Electronics Engineering from PSG College of Technology, Coimbatore, India in May 1999. He worked as a Software Engineer in Cognizant Technology Solutions, Chennai, India from June 1999 to July 2001. He joined the graduate program in Computer Engineering at Texas A&M University in August 2001. His main research interest is in the area of Software Process Improvement and Designing and Architecture of Enterprise Systems. His present address is New No.4, Priya's Madhuram,F-2, Ranganathan St., Ganesh Nagar, Velachery, Chennai - 600039, India . His e-mail id is iprabhua@chn.cognizant.com.

The typist for this thesis was Prabhu Anand Inbarajan.