

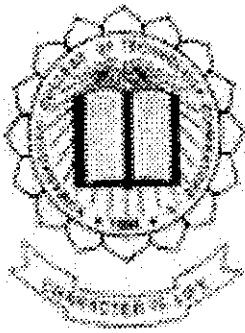
COMPONENT ASSEMBLY FRAMEWORK FOR E-BUSINESS

P-938

PROJECT REPORT

Submitted in partial fulfillment of the requirements for
award of degree

M.Sc.,[Applied Science] Software Engineering



Submitted By
S.SUNDAR RAJAN
9837S0067



UNDER THE GUIDANCE OF,

External Guide

Internal Guide

Mr. Vijay Sargunar

Mr. M.Manigandan,

Project Leader,

Lecturer

TCS

CSE Department

Department Of Computer Science and Engineering

Kumaraguru College Of Technology

(Affiliated to Bharathiar University)

Coimbatore-641006.

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY

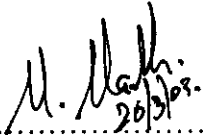
COIMBATORE - 641 006

CERTIFICATE

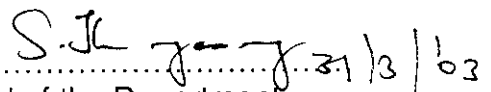
This is to certify that the project entitled
COMPONENT ASSEMBLY FRAMEWORK FOR E-BUSINESS
has been submitted by

S. SUNDAR RAJAN

In partial fulfillment of the award of the degree of
Master of Science in Applied Science – Software Engineering of
Bharathiar University, Coimbatore
during the academic year 2002-2003


20/3/03

Guide


31/3/03

Head of the Department

Certified that the candidate was examined by us in the Project Work Viva Voce Examination held on11/4/2003..... and the University Register Number was **9837S0067**



Internal Examiner



External Examiner

COMPANY CERTIFICATE

PRODUCT DEFINITION

The document is considered valid when it follows the constraints that the DTD lays out for the formatting of the XML data.

Other Concepts to Be Known

Reflection Package

Reflection is the ability of software to analyze itself. This is provided by the

`java.lang.reflect` package and elements in class. It allows you to analyze a software

component and describe its capabilities dynamically, at run time rather than at compile

time. By using reflection, we can determine what methods, constructors and fields a class

supports. It is also possible to retrieve information about constructors, fields and methods

just declared in this class as well as those defined in super classes.

Other ways of using reflection are to dynamically – invoke methods, create new

objects, change values of fields, use arrays etc. The package `java.lang.reflect` has one

interface, called `Member`, which defines methods that allow you to get information about a

field, constructor, or method of a class.

JavaBeans

A Java bean is a reusable platform-neutral software component that can be visually

manipulated in a software development tool.

The JavaBeans standard is a low-level component model tailored to the Java

language. It is low-level because it does not hint at or specify the kind of compound

document or application framework architectures that some other component standards

specify. JavaBeans concentrates purely on the interface a Java software building block

should present, but does not venture into the endless landscape of how such building blocks

can or should be combined to create any type of application. However, it does specify

how two or more beans can communicate information, but without imposing any semantic

rules on the information exchanged.

2. PRODUCT DEFINITION

2.1 OVERVIEW

The aim behind developing this application is to generate a XML file with

necessary input being fed with the help of the tool. This application should enable the user to get a XML file as out put by using the GUI provided by this tool. There should be a provision to add a child XML file to the Existing parent XML file.

The XML editor should also have link to open a Java Class Editor. This Editor

would allow the user the open a JAR file from the system class path or Add a new jar file from any location in the system.

The project aims to give the user an application to generate a XML file with ease and to send Java class files methods as parameters to the XML file so that the intended operation is achieved.

2.2 PROBLEM STATEMENT

The project consists of three applications as follows: -

Café (Component Assembly Framework for E-Business)

Café is a development framework for Java applications based on the Model-View-

Controller (MVC) design paradigm. The focus of café for current working version is on the controller aspect of MVC architecture. Café achieves automation of control delegation by employing a declarative approach to development and integration. The goal behind

developing café is to provide a general infrastructure for developing web applications and to reduce the development efforts by providing the ability to extend the framework in the appropriate places to achieve a framework that better suits developers needs. Café consists of two main modules such as XML Editor and Java class Editor. So the ultimate goal of this work is that the XML editor and JAVA class Editor would be merged together and will



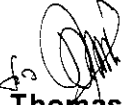
11-Mar-03

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. S. Sundar Rajan**, doing MSc. Software Engineering **Kumaraguru College of Technology, Coimbatore - 6**, has successfully completed a project work titled "**Development of Webdepot**" under the guidance of **Mr. Vijay Sargunar, Asst. Systems Engineer** of our organisation during the period **December 2002 - March 2003**.

His work has been found satisfactory.

Yours faithfully,
For **TATA CONSULTANCY SERVICES**


Thomas Simon
Sr. Manager - HR

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Dr.K.K Padmanabhan, B.Sc. (Eng.), Ph.D., esteemed Principal, Kumaraguru College of Technology for providing me the necessary facilities in the college.

I wish my heartfelt thanks and a deep sense of gratitude to Prof.S.Thangaswamy, B.E (Hon.), Ph.D., The Head of Department Computer Science and Engineering, who motivated me by his valuable ideas and suggestions.

I express my sincere thanks to Mrs. S. Devaki, B.E., M.S, Asst. Professor, Computer Science and Engineering without whose motivation and guidance I would not have been able to embark on a project of this magnitude.

Mr. M. Manikandan, MCA, Lecturer, Computer Science and Engineering has offered me invaluable help and support throughout the semester am thankful to him for coordinating my activities.

I am also grateful to Mr. Salin Reddy, Project Manager and Mr. Vijay Sargunar, Project Leader of Tata Consultancy Services, Sholinganallur, Chennai-119, for allowing me to do projects under this division.

I like to express my indebtedness to Mr. Sudheer Dhurghati for his invaluable guidance through the entire length of my project. My heartfelt thanks of all the employees of the E-Business projects for their suggestions and for making my stay very comfortable.

Last but not the least, I thank my beloved parents, friends, department teaching and non teaching staffs who have been pillar of support from the start, until the completion of the project.

SYNOPSIS

SYNOPSIS

This project entitled “Component Assembly Framework for E-Business” was done at Tata Consultancy Services, Sholinganallur, Chennai-119.

Cafe is a development framework for Java applications based on the Model-View-Controller (MVC) design paradigm. The goal behind developing *café* is to provide a general infrastructure for developing web applications and to reduce the development efforts by providing the ability to extend the framework in the appropriate places to achieve a framework that better suits developers needs. The framework has two main modules based on which the entire functionality depends:

- XML Editor
- Java Class Editor

XML Editor is a simple application that assists in the building of rapid prototypes of XML applications. It allows users to rapidly build and edit small sets of XML data as a test bed during the development of XML-based applications. With XML Editor, we can create XML document prototypes quickly, easily, and in an iterative fashion, using familiar metaphors. XML Editor offers an intuitive and simple user interface that graphically represents the tree structure of XML data. Working with the standard building blocks of XML (Elements, Attributes and Text), users are able to create reproducible data structures that can be easily filled. XML Editor enables to create data sets quickly and easily.

The *Class Browser* is another simple application, which displays all the class files taken from the class path set by the users and get displayed in the graphical form. Along with the class names the method name and access specifiers present in the class is also displayed in the GUI.

CONTENTS

1. INTRODUCTION	1
1.1 TATA CONSULTANCY SERVICES – A PROFILE	1
1.2 ORGANISATION STRUCTURE	3
1.3 INTRODUCTION TO E-BUSINESS	3
PRODUCT DEFINITION	5
2.1 OVERVIEW	5
2.2 PROBLEM STATEMENT	5
2.3 PROCESSING ENVIRONMENT	6
2.3.1 <i>Development Environment</i>	6
2.3.2 <i>Implementation Environment</i>	6
2.3.3 <i>Information processing requirements</i>	7
2.4 USER CHARACTERISTICS	7
3. PROJECT PLAN AND FEASIBILITY STUDY	8
3.1 TEAM STRUCTURE	8
3.2 DEVELOPMENT SCHEDULE	9
3.2.1 <i>Milestones</i>	9
3.2.2 <i>Reviews</i>	9
3.3 DEVELOPMENT TOOLS	10
3.4 MANNER OF DEMONSTRATION	15
3.5 FEASIBILITY ANALYSIS	16
3.5.1 <i>Operational feasibility</i>	16
3.5.2 <i>Technical feasibility</i>	16
3.5.3 <i>Economical feasibility</i>	17
4. SOFTWARE REQUIREMENT SPECIFICATION	18
4.1 SYSTEM STUDY	18
4.1.1 <i>Existing System</i>	18
4.1.2 <i>Proposed System</i>	18
4.1.3 <i>Purpose of this document</i>	19
4.2 PRODUCT OVERVIEW	19
4.3 PROCESS FLOW DIAGRAMS	21
4.4 GENERAL CONSTRAINTS	24
4.5 LIST OF INPUTS	24

8. CONCLUSION AND FUTURE ENHANCEMENTS	53
8.1 CONCLUSION	53
8.2 FUTURE ENHANCEMENTS.....	54
APPENDIX A-TEST CASES.....	55
APPENDIX B-SCREEN SHOTS.....	61
REFERENCES.....	82

INTRODUCTION

1. INTRODUCTION

1.1 TATA CONSULTANCY SERVICES – A PROFILE

TCS is a division of Tata Sons, the holding company of the \$10.4 billion Tata Group, India's best known business conglomerate. Established in 1968, its founding was based on the understanding that the management problems in Indian industry could be resolved through the effective use of information technology. Under the leadership of F C Kohli, we spearheaded the pioneering efforts in creating a globally recognizable brand for the Indian software industry.

Strong linkages with academia, workplace professionalism, and in-house training and learning helped TCS lay the foundation for growing into a world-class organization. We invested heavily in software engineering practices and standards, software quality assurance, software project management, software processes, and research and development in software engineering and technology. As our intellectual horizons enlarged to better fit and address the opportunities of the IT era, we grew in several geographical locations.

For several years now, TCS has been India's largest IT enterprise, as well as Asia's largest independent software and services organization. We are the single largest software services exporters from India, and we now service clients in over 55 countries around the world. With over 100 branches globally, TCS is truly transnational in character and reach. We presently employ over 20,000 consultants and serve hundreds of clients, providing IT and business consulting services to organizations in government, business and industry in India and abroad.

Our service offerings are varied, and straddle many different industries, such as finance and banking, insurance, telecommunications, transportation, retail, manufacturing, pharmaceuticals and utilities. Our clients include small, medium and large companies, and

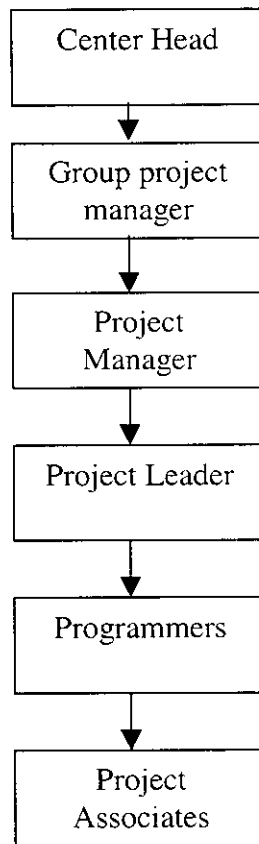
our consulting engagements vary from a few person-months to hundreds of person-years in effort. TCS has grown dramatically during the past decade, doubling revenues every two years for the past six years. And our aim to become a billion-dollar company soon.

Our work has spanned a range of activities, from strategy consulting and system integration services to offshore development center's for some of the most sophisticated software development in the world. Consider these facts about us:

Projects for over 1000 clients in more than 55 countries. Our clients get measurable business results, which is why many Fortune 500 companies prefer us.

- More than 100,000 person years of experience in diverse **business domains** and **technology areas**.
- Knowledge accumulation through **collaborative research** with industry and academia, and **partnerships** with global technology leaders.
- Over 20,000 employees, a range of world-class products, proven offshore development capabilities, and multiple **SEI CMM Level 5 center's**
- TCS believes that IT is a key factor for social change and is committed to several **community development ventures**.
- We have engineered some of the world's most complex applications and next-generation IT infrastructure. We do this by combining our knowledge of business domains with our expertise across various technologies.

1.2 ORGANISATION STRUCTURE



1.3 INTRODUCTION TO E-BUSINESS

E-Business is more than web-enabling existing systems or business processes. It means leveraging the power of Internet technologies for the benefit of all stakeholders. Identifying opportunities for improvement, extending, enhancing or transforming business processes, choosing wisely among available options, architectonic, building, and deploying E-Business solutions rapidly - are all keys to success.

4.6	FUNCTIONAL REQUIREMENTS	25
4.7	PERFORMANCE REQUIREMENTS	26
4.8	EXCEPTION HANDLING.....	27
4.9	PROGRAMMING STANDARDS.....	27
4.10	OTHER NON-FUNCTIONAL ATTRIBUTES.....	28
4.10.1	<i>Reliability</i>	28
4.10.2	<i>Testability</i>	28
4.10.3	<i>Maintainability</i>	28
4.10.4	<i>Portability</i>	28
4.10.5	<i>Reusability</i>	28
4.10.6	<i>Security</i>	28
5.	SYSTEM DESIGN.....	29
5.1	EXTERNAL DESIGN SPECIFICATIONS.....	29
5.1.1	<i>Methodology</i>	29
5.1.2	<i>Data flow diagram</i>	33
5.2	ARCHITECTURAL SPECIFICATION	39
5.3	DETAILED DESIGN SPECIFICATIONS.....	40
5.3.1	<i>Detailed description</i>	40
5.3.2	<i>Pseudocode for each routine</i>	42
6.	CODING AND TESTING.....	47
6.1	CODING STANDARDS.....	47
6.2	TESTING.....	48
6.2.1	<i>Unit Testing</i>	48
6.2.2	<i>Validation Testing</i>	49
6.2.3	<i>Integration Testing</i>	49
7.	PROJECT LEGACY.....	50
7.1	PROJECT DESCRIPTION	50
7.2	INITIAL EXPECTATIONS	50
7.3	CURRENT STATUS OF THE PROJECT	51
7.4	REMAINING AREAS OF CONCERN	51
7.5	TECHNICAL LESSONS LEARNED.....	51
7.6	MANAGERIAL LESSONS LEARNED	52

To create new revenue streams, maximize existing revenue streams, optimize supply chains, foster strong customer relationships, or gain insight from enterprise information; today organizations worldwide are strategizing, building and implementing integrated multi-channel E-Business solutions using the right technologies.

The e-World has created innumerable opportunities for organizations to exploit their core business competencies by transforming their technology infrastructure from business to E-Business. It is imperative that organizations define the right strategy, adopt a viable business model, and choose wisely among the varied technology options.

Our competencies

- TCS comprehensive E-Business consulting offerings help clients strategize architect, build and operate their business in the e-World.
- TCS provides solutions to global and Fortune 500 clients, ranging from customer contact centers, through state-of-the-art B-to-B and B-to-C sites, to enterprise decision support applications.

provide the programmers an easy tool to create, modify and save the XML file with much ease.

XML Editor

The framework for E-biz followed in TCS needs Java class files as input to the XML file. To code the XML file is tedious and special knowledge is required. It has been found in the recent past that this process takes lots of man-hours and errors occur very frequently. Since there is very less number of XML editors available in the market and none suite to the exact requirements of the system, it was proposed to build an XML editor specific to the framework and generic to build any XML file.

Java Class Browser

The Java class files forms the input to the XML file. So it was also found that by building a Java class browser it would lead simplification of the process. So it was decided to develop a Java class browser to get the class files and its methods from the classpath. This class browser provides a GUI that would display the class files and the methods in it, in a tree structure.

2.3 PROCESSING ENVIRONMENT

2.3.1 Development Environment

Language	Java, Java Swings
Designing tool	VisualAge for Java
Additional Packages	DOM Parser and reflect packages
Operating System	Windows 2000

2.3.2 Implementation Environment

Operating System	Any Operating System
Working Environment	Java Virtual Machine

2.3.3 Information processing requirements

The system can be operated on any platform and needs less memory to execute as we are using Java Swings, which is lightweight component. We are provided with the system with following configuration

- Pentium III 850 MHz
- 20 GB HDD
- 128 MB RAM

2.4 USER CHARACTERISTICS

- The user must be an employee of TCS.
- The Employee should be one among the E-Business project.
- The user should have knowledge on XML and Java.
- The user should be able to identify the class file that has to be mapped.

PROJECT PLAN AND FEASIBILITY STUDY

3. PROJECT PLAN AND FEASIBILITY STUDY

3.1 TEAM STRUCTURE

The team structure that we follow is the hierarchical team structure, which limits the number of communication paths in a project while permitting effective communication among team members who need to communicate with one another. The management structure and communication paths in a hierarchical team are given below.

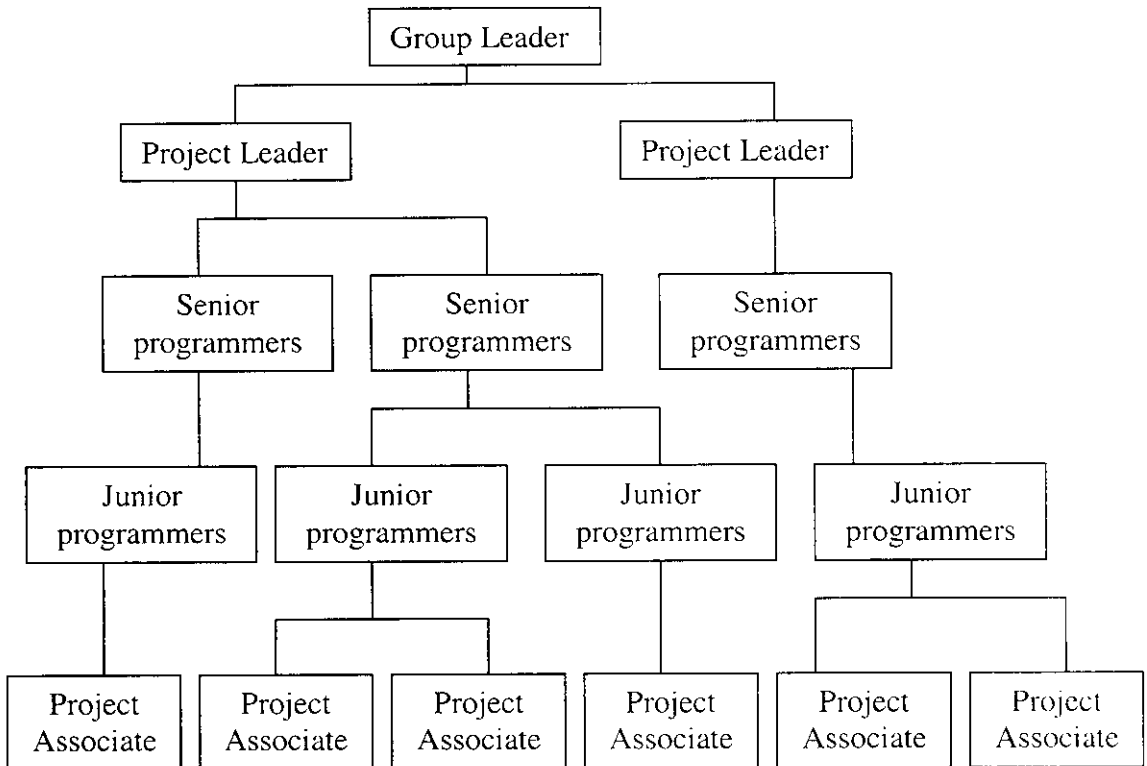


Fig. 3. 1 Hierarchical Team Structure

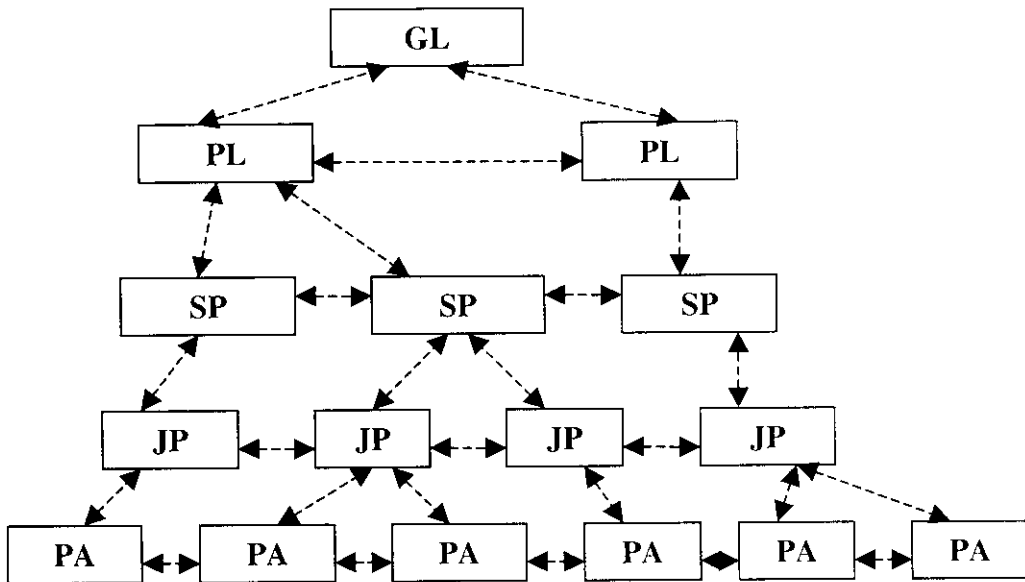


Fig. 4. 2 *Communication Paths*

3.2 DEVELOPMENT SCHEDULE

3.2.1 Milestones

Milestones are being established for each and every module to improve the product visibility. It enhances the development process to become more tangible. It exposes errors, which help in improving the product quality and increases project communication. In our application it has been done sub-module wise.

3.2.2 Reviews

The review summary report is prepared on weekly basis to know the progress of each member. Review issues lists, are prepared to identify problem areas within the product, and it is attached with every review summary. As a programmer we do the following reviews.

- Critical Design Review
- Source Code Review
- Acceptance Test Review

3.3 DEVELOPMENT TOOLS

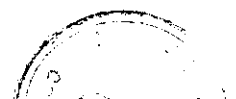
Java

Java is a computer language developed by Sun Microsystems. To quote Sun, “Java is a simple, robust, object-oriented, platform-independent, multi-threaded, dynamic, general-purpose programming environment. It’s best for creating applets and applications for the Internet, Intranets and any other complex, distributed network.” Besides being an elegant language, ideal for platform-independent GUI applications, downloading Java from web pages is one of the best ways to make the World Wide Web truly interactive.

The language is similar in appearance to C++, although memory management has become primarily the responsibility of the language not the programmer. Pointers too are no longer used (they are a major security headache), but the language is powerful enough to make their use unnecessary. Java comes with a large ‘class library’, in the form of ‘packages’, which help especially to make short work of GUIs and networking.

Platform independence is achieved by compiling the language into a machine independent code (“byte code”) which is then run by a platform-specific interpreter (the “virtual machine”). Porting programs between PCs, UNIX machines, Macs, and any other platform is therefore just a matter of copying the byte-code from one machine to the next and running it. The interpreter can also exert control over the program ensuring that it stays within certain security restrictions - essential in the world of distributed computing.

The final advantage of Java is that it costs nothing. Sun appear content to earn money from the language by providing support, training, more advanced tools etc., while making the basic language, compiler and virtual machine free. This has helped to make Java a rapidly accepted and used language.



Java Swings

The Swing package is part of the Java™ Foundation Classes (JFC) in the Java platform. The JFC encompasses a group of features to help people build GUIs; Swing provides all the components from buttons to split panes and tables.

The Swing package was first available as an add-on to JDK 1.1. Prior to the introduction of the Swing package, the Abstract Window Toolkit (AWT) components provided all the UI components in the JDK 1.0 and 1.1 platforms. Although the Java 2 Platform still supports the AWT components, we strongly encourage you to use Swing components instead. You can identify Swing components because their names start with `J`. The AWT button class, for example, is named `Button`, whereas the Swing button class is named `JButton`. In addition, the AWT components are in the `java.awt` package, whereas the Swing components are in the `javax.swing` package.

As a rule, programs should not use "heavyweight" AWT components alongside Swing components. Heavyweight components include all the ready-to-use AWT components, such as `Menu` and `ScrollPane`, and all components that inherit from the AWT `Canvas` and `Panel` classes. When Swing components (and all other "lightweight" components) overlap with heavyweight components, the heavyweight component is always painted on top.

VisualAge for Java

VisualAge for Java is an integrated visual environment that supports the complete cycle of Java program development. VisualAge for Java gives you everything you need to perform the development tasks described below.

Rapid application development

You can use VisualAge for Java's visual programming features to quickly develop Java applets and applications. In the Visual Composition Editor, you point and click to:

- Design the user interface for your program
- Specify the behavior of the user interface elements
- Define the relationship between the user interface and the rest of your program

VisualAge for Java generates the Java code to implement what you design in the Visual Composition Editor. In many cases you can design and run complete programs without writing any Java code.

VisualAge for Java also gives you *SmartGuides* (wizards) to lead you quickly through many tasks such as creating new applets, packages, or classes.

Incremental compilation

The VisualAge for Java Integrated Development Environment (IDE) automatically compiles Java source code into Java byte code. When source code is imported into the workspace (from .Java files) or added from the repository, it is compiled and analyzed with respect to the existing contents of the workspace.

When you change, delete, move, copy, or rename program elements, the affected code is automatically recompiled to flag any problems.

If you introduce an error, the IDE warns you and gives you the option of fixing the problem immediately or of adding the problem to the All Problems page and fixing it later.

Repository-based development

VisualAge for Java has a sophisticated code management system that makes it easy for you to maintain multiple editions of programs. When you want to freeze the state of your code at any point, you can *version* an edition. This marks the particular edition as read-only and enables you to give it a name. This gives you a way to preserve snapshots of significant checkpoints in a development cycle.

Create industrial-strength Java programs

With VisualAge for Java you can develop very robust code. Specifically, you can:

- Build, modify, and use JavaBeans
- Browse your code at the level of project, package, class, or method
- Use the integrated visual debugger to examine and update code while it is running
- Write applications that comply with Sun Microsystems Enterprise JavaBeans (EJB) specifications
- Use the distributed debugger to debug Java applications that are developed outside the IDE

XML

XML (eXtensible Markup Language) is a “meta-language”, that is, a language used to describe another language. XML can be used to describe any form of data in a database. The advantage of a meta-language is that it can be used to define more than one implemented language. The definition of extensible is a system that can be modified by changing or adding features. In the case of XML, the extensibility comes from being able to define the tag set and grammar of a document. HTML, on the other hand, has a definite list of tags to be used that cannot be modified. Since XML doesn't define tag set or grammar, you can define the content of own data as long as you conform to XML's required general form. DTDs and Schemas help define the legal building blocks of an XML document.

XML is a World Wide Web Consortium standard that lets you create your own tags. XML simplifies business-to business transactions on the web. XML's strongest point is its ability to do data interchange. Because different organizations (or even different parts of the same organization) rarely standardize on a single set of tools, it takes a significant amount of work for two groups to communicate. XML makes it easy to send structured data across the web so that nothing gets lost in translation. XML changes the way data moves across networks. XML encapsulates data inside custom tags that carry semantic information about the data.

Data can be organized by using a *DTD* or a *Schema*. By using either of these you can define the elements in an XML document and whether they are empty or include text. You can define attributes and default attribute values for these elements. You can choose what child elements you want, how many, and define the sequence in which they are to appear. The organization of the data is up to you.

DTDs establish the set of constraints for an XML document and define the way a document should be constructed. Within an XML document a DTD can both include markup constraints and refer to an external document with markup constraints. The meaning of the tags must be defined in DTD for the document to be validated by a parser.

The definition of a Java bean is not oriented with software components alone rather it includes the development tool aspect. Java beans are explicitly defined to the “tool aware” in the sense that much of Java beans interface is designed with a modern software developer in mind, manipulating the bean via visual interactions.

XML Parser

For parsing XML and accessing the results, there are two established APIs. First, the *SAX (Simple API for XML)* inter handlers containing methods that are called as the various parts of an XML document are parsed. Second, the *DOM (Document Object Model)* interfaces define a logical tree representing the XML document after parsing. Applications that do not require complex manipulations of the XML structure will find the SAX interfaces very useful. For structural manipulations involving possibly all or most XML tokens, the DOM tree interfaces are useful.

The Document Object Model (DOM) is a set of language-independent interfaces for programmatic access to the logical XML document. We will use the latest Java DOM Interfaces. Interface that encapsulates the structural connections between the XML constructs is called the Node. The Node contains that express structural connections such as `Node#getChildNodes()`, `Node#getNextSibling()`, `Node#getParentNode()`,

The DOM Interfaces also contain separate interfaces for XML’s high-level constructs such as `Element`. Each of these interfaces extends `Node`. For example, there are interfaces for `Element`, `Attribute`, `Comment`, `Text`, and so on.

3.4 MANNER OF DEMONSTRATION

The programmers first do the demonstration of the product in his/her system and later the senior programmer integrates all the sub modules and demonstrates to the team manager as a complete package in the multimedia projector.

3.5 FEASIBILITY ANALYSIS

The feasibility study is very rough analysis of the viability of a project. It is however a highly desirable checkpoint that should be completed before committing to more resources. A feasibility study is conducted to obtain an overview of the problem and to roughly assess whether feasible solutions exist prior to committing substantial resources to a project.

The primary objective of a feasibility study is to assess three types of feasibility.

- Operational Feasibility
- Technical Feasibility
- Economical Feasibility

3.5.1 Operational feasibility

Operational feasibility study is a must, because it ensures that the project implements in the organization works. This feasibility should be high.

The operation feasibility of this project is very high as it automates the code generation and provides good interface, which is easy and friendly for the user to use it.

3.5.2 Technical feasibility

Technical feasibility analysis makes a comparison of the level of technology available and the same is required for the development of the product. The level of technology accounts for factors such as the programming language, the machine environment, the programming practices and the software tools. IT includes the study of function, performance and constraints that may affect the ability to achieve an acceptable system. The following considerations are evaluated with technical feasibility.

Resource availability such as Pentium III processor with 128 MB RAM, software and tools required for the project are available at the organization. Hence it is technical feasible

3.5.3 Economical feasibility

This is the most important aspect that has to be critically evaluated. This includes the feasibility study of cost-benefit analysis. This is an assessment of the economic justification for a computer based system project. The hardware and most of the software are already there in the company. The rest of the software packages needed are freely downloadable. Hence the threat of financial non-feasibility does not exist. It is determined that benefits out beat the cost of implementation and thus the system is considered to be economically feasible.

SOFTWARE REQUIREMENT SPECIFICATION

XML Editor is a simple application that assists in the building of rapid prototypes of XML applications. It allows users to rapidly build and edit small sets of XML data as a test bed during the development of XML-based applications. With XML Editor, we can create XML document prototypes quickly, easily, and in an iterative fashion, using familiar metaphors:

XML Editor offers an intuitive and simple user interface that graphically represents the tree structure of XML data. Working with the standard building blocks of XML (Elements, Attributes and Text), users are able to create reproducible data structures that can be easily filled. XML Editor enables to create data sets quickly and easily.

XML-based data is displayed graphically in the right side of XML Editor using JTree. XML Editor saves time and effort, allowing us to place a greater emphasis on creating the applications than on typing syntax.

Java Class Browser:

The Class Browser is another simple application, which displays all the class files in the graphical form. The class files are listed in the tree, which are got from the class path set by the users.

The Class path contains a directory path, a zip or jar file, a class file and other resources that are irrelevant to the application. If the class path returns a directory structure the sub directory has to be navigated to get the class file. In case of zip or jar files the file has to be zipped to get the class file and all other path, which does not have class files are not displayed.

The selected class file and the path are displayed in the screen along with add and remove options. The class files that are listed in the class Browser are mapped with the appropriate XML nodes presented in the XML Editor.

- XML Editor
- Java Class Editor

framework has two main modules based on which the entire functionality depends they are: files. So these would be the input for the Framework (in its part of operation). The using the XML. Programmers can use the existing class files or can write their own class put in simple terms the Java class files which does most of the operations will be called on the XML. The big advantage is that the programmers could easily follow these rules. To guidelines for usage will be specified in the XML DTD. So Java class files will be passed One important part of this Framework is that the usage of XML. The rules and other Components.

in the Framework, which act together like J2EE, Java Beans, XML, Core Java and host of The Framework has been tested and been implemented. There are lots of Elements project:

TCS decided to form a Framework that will be generic to implement any E-Business conduct Business over the Internet. Since there is a complex process involved in doing so, At TATA, the division of E-Business does projects for the clients who wish to

PRODUCT OVERVIEW

4.2

readers to have knowledge about web applications and Java.

J2EE framework should make the things even clearer. The document also expects the knowledge of the Java Programming Language. Prior working knowledge in any popular engineers actively involved in J2EE development projects. Several sections assume This document is intended for software programmers, developers and technical

4.1.3 Purpose of this document

minimizing the errors and development time.

So the proposed system aims at getting a XML file with maximum ease and the methods and send it to the text field in the XML file.

class files as arguments to the XML file. So one can simply click on the application and get The very idea of doing this is because the TCS framework for E-Biz passes Java along with the methods in it with access modifiers and arguments in a tree format.

system class path or the user can select one, and can open the same display the class files

4. SOFTWARE REQUIREMENT SPECIFICATION

4.1 SYSTEM STUDY

4.1.1 Existing System

In the recent past the E-Business group of TCS have found out that the most vital part of their work (i.e.) passing of Java class files as input to the XML file is taking up lots of time. Further more they have experienced lots of errors while incorporating the Java class file in Xml. So, to avoid wastage of time and error in code a new system has been proposed.

4.1.2 Proposed System

The proposed system is to build a GUI based tool to generate the XML file. This tool would contain two main modules viz. XML editor and Java class Editor. The XML editor is the one where the user could use to generate a XML file. The system would provide an easy interface to manipulate the XML file building blocks such as the attributes, text, comment etc, just by clicking on the appropriate menu or tool bar.

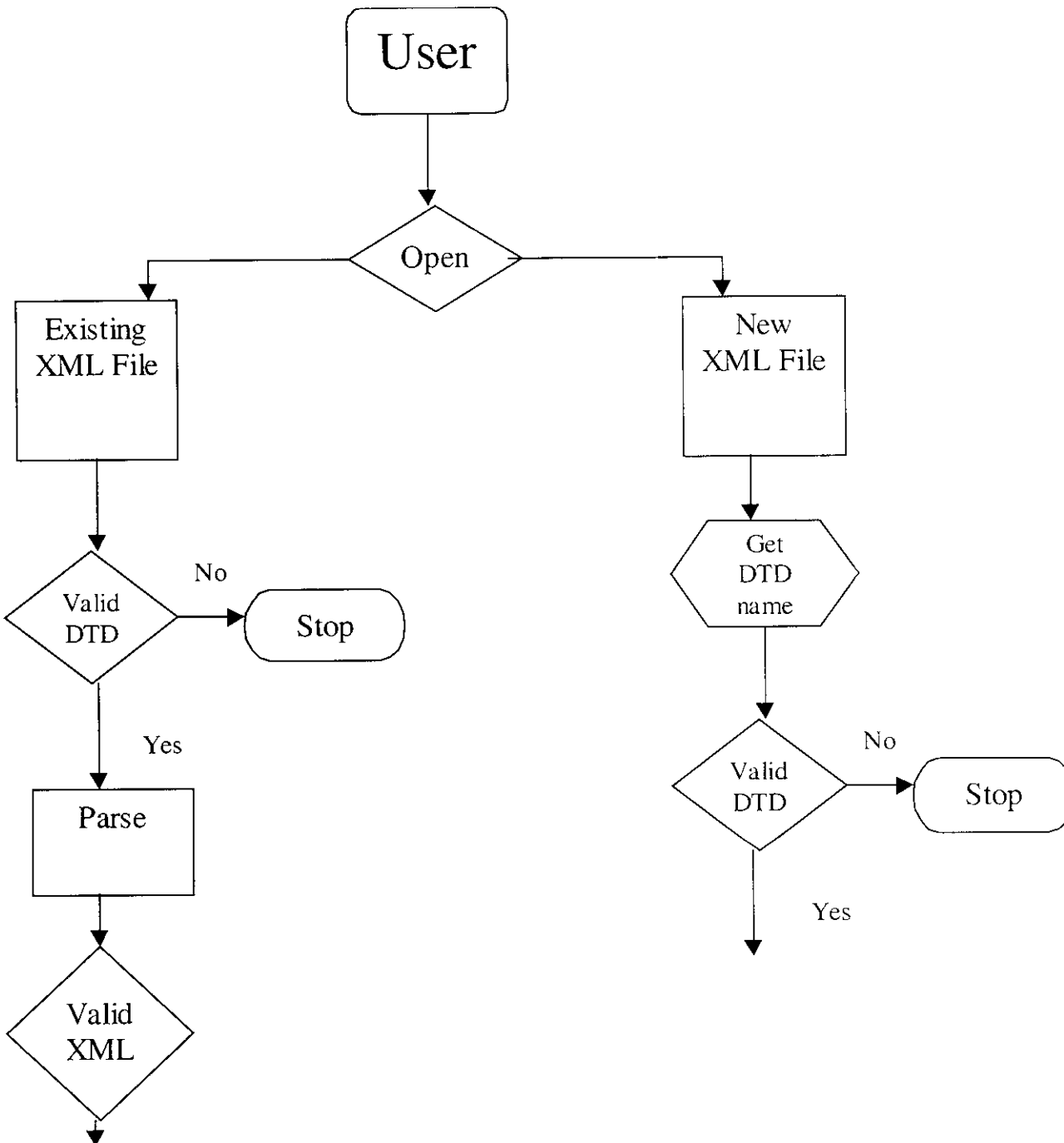
The proposed system will be able to open an existing XML file, parse the same and display it in a Jtree. This file could be modified and saved. The system also allows creating a new XML file and saving the same.

The display is to be made in a simple format so that the user could be able to understand how the XML file is built and can visualize its usage. The user might be adding the same kind of elements to XML file. So, to enable the user to get this work done the system would provide a method by which a child XML file could be added to the parent XML file. So, this serves the purpose of adding a template, which could save lots of time and also reduce errors considerably.

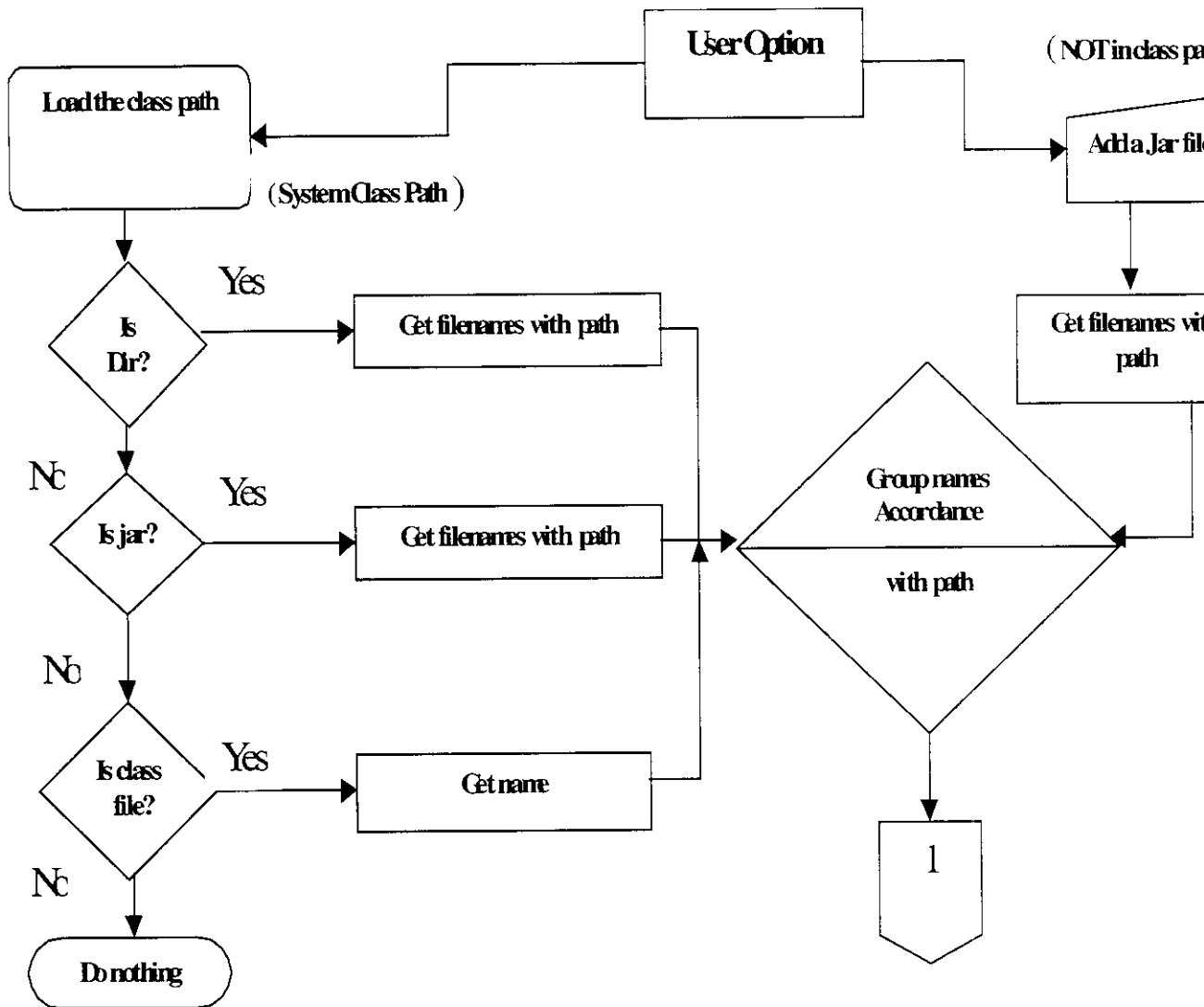
The system will have also a JAVA CLASS EDITOR. This is another important addition to the XML editor. This Java class Editor will be able to load a jar file form the

4.3 PROCESS FLOW DIAGRAMS

XML Editor:



Java Class Browser:



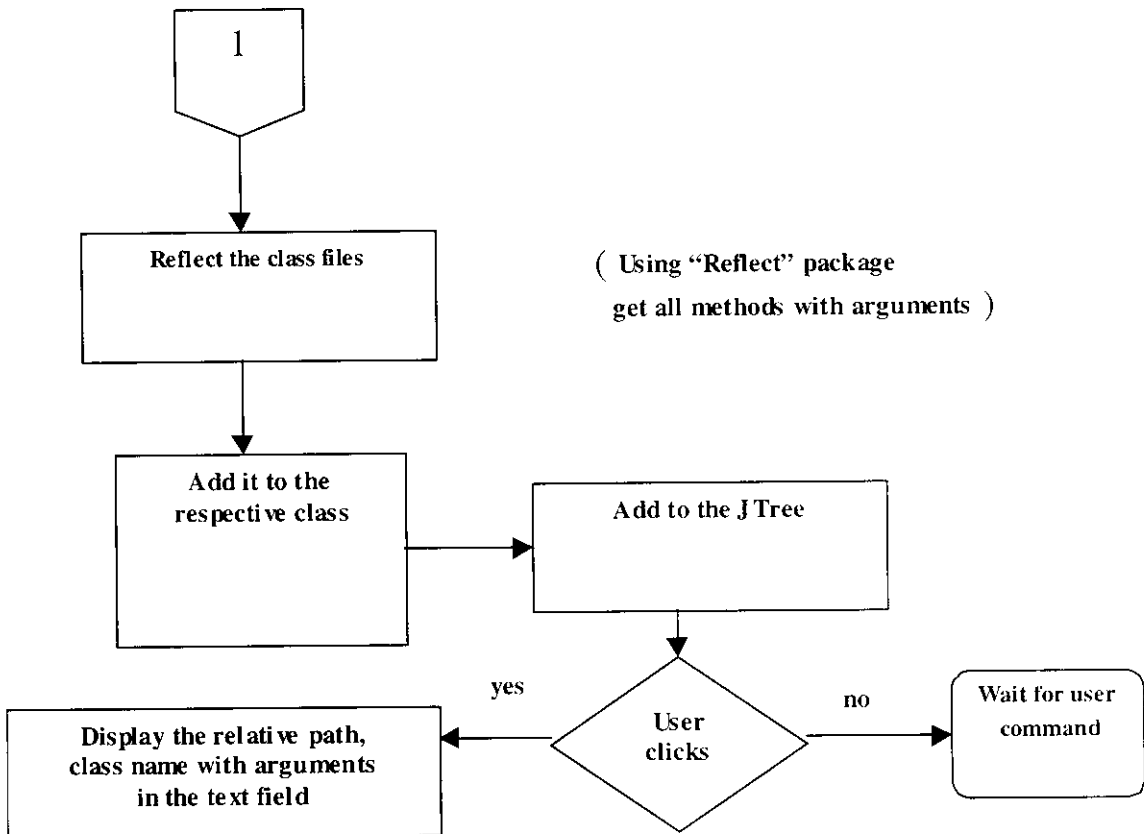


Fig. 4. 2 Process Flow Diagram for Java Class Browser

4.4 GENERAL CONSTRAINTS

- The product has to be flexible, so that other project team, which implements TCS Framework, can also use.
- It product should be platform independent.
- Coding standards has to be followed through out the framework development.

4.5 LIST OF INPUTS

- The Input to the XML Editor is a valid XML file with or without DTD in it.

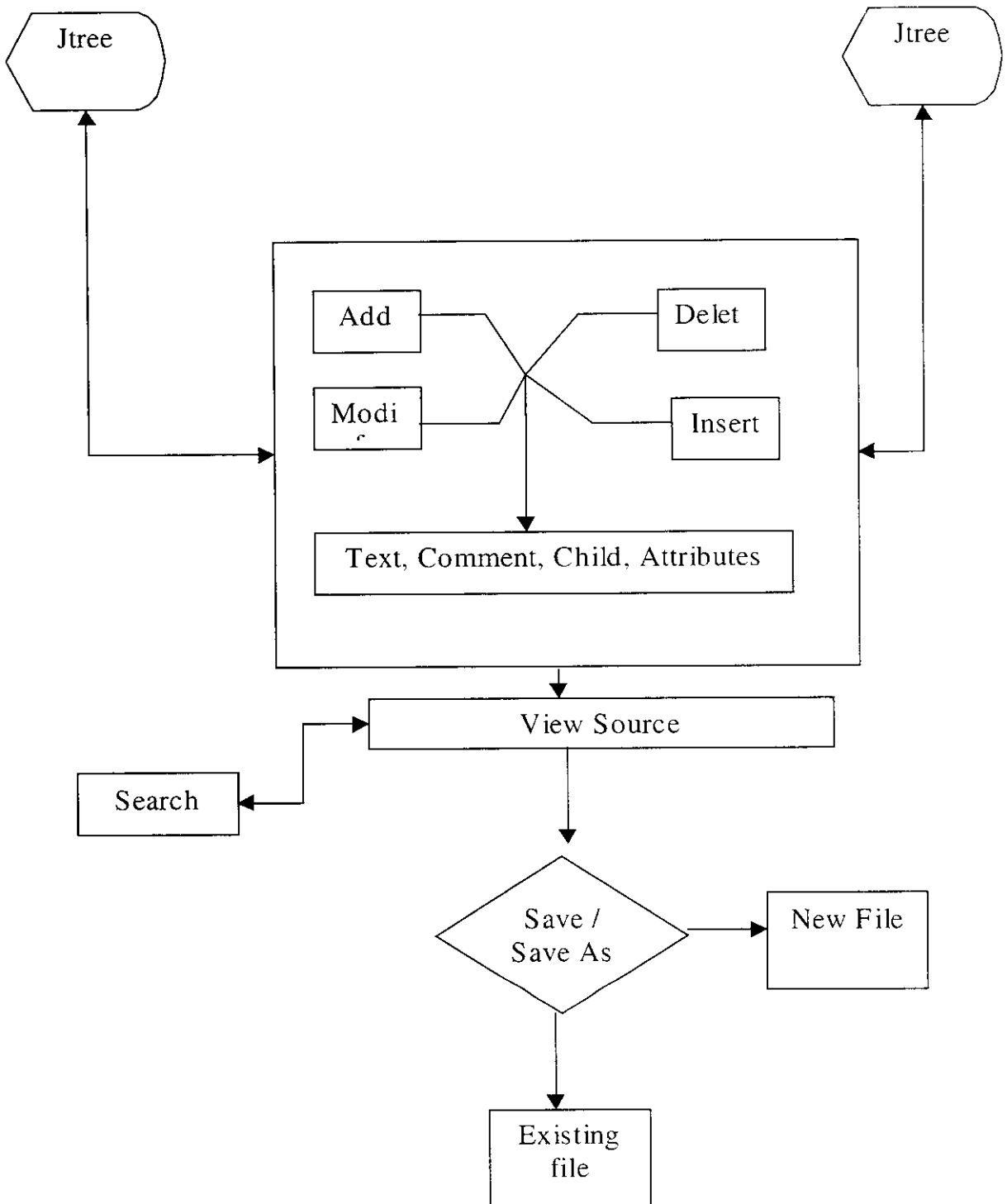


Fig. 4. 1 Process Flow Diagram for XML Editor

- The input to the Java Class Browser is the list of class files fetched from the class path.
- The input to the café is both the output of XML Editor and Java class Editor

4.6 FUNCTIONAL REQUIREMENTS

The functions of the entire framework depend on the functions of XML Editor and Class Browser.

XML Editor:

- It should open a valid XML file.
- The nodes of the XML file has to be displayed in the tree format.
- The building blocks of the XML file such as attribute name attribute value, child elements, text and comment has to be displayed in a table.
- Standard menu options has to be provided such as new, open, save, cut, copy, paste, etc.
- Apart form the standard menu, options has to be provided for viewing the complete source file, move up and move down of the child nodes, clearing the comment and text nodes etc.
- A template has to be provided for making the user input easier.
- Validation has to be done to check whether it matches with the DTD.
- Separate icons have to be added for the nodes in the tree.
- In case of any modifications in the tree structure such as adding a node or deleting a node has to be reflected in the XML file.
- Searching of a tree node is done based on the building blocks of XML file. If the search node is found then the tree has to be expanded and the respective node has to be highlighted.

- Searching of a text in the view source is also done and the respective text is highlighted.
- Adding of a new XML file in the current file, as a child element has to be provided.
- Separate pop up menu is provided for the view source to cut copy and paste the text.
- Option for creating a new XML file has to be provided.
- Help menu for the new user has to be provided just like other software products.
- Validation for a XML file has to be done to check whether the XML file follows the exact DTD specified in case if DTD is included in the XML file.

Java Class Editor

- The class Editor has to fetch the files from the class path.
- The directory structure has to be displayed in the tree format.
- The zip or jar file has to be zipped and the contents are displayed.
- Locations of the selected class file have to be displayed in the screen.
- Add and remove options are provided to add a classpath and remove a classpath
- The class files has to be reflected using reflect package and the methods name has to be displayed in the tree format below the respective class file.
- Various icons are added to differentiate the different access modifiers of a class file.

Café

- In café both the XML Editor and Java Class Editor has to be integrated.
- An option has to be provided for mapping the required Java file with the exact XML nodes.
- On saving the mapped document an XML file has to be generated.

4.7 PERFORMANCE REQUIREMENTS

The system would maintain a consistent output frequency. The system response time would be as per normal standards taking into consideration the Network speed. To

increase the processing time, review of code is done to reduce the number of loops in the application and to maintain Java coding standards. The product needs some fine tune works to be carried out before it is being delivered.

4.8 EXCEPTION HANDLING

Exception Handling is a powerful and general-purpose system for error handling and recovery. The errors occur during the loading and processing of a file. We handle all the exception by using a general error page in all the files, which holds the reference to the uncaught exception that caused the error page to invoke. To handle the errors and exceptions in the code the following statements are used.

- *Try* statement
- *Catch* statement
- *Throw* statement

Some of the exceptions, which are handles, are as follows: -

- Input/Output Exception.
- Classnot found Exception
- NullPointerException
- ArrayIndexOutOfBounds Exception

4.9 PROGRAMMING STANDARDS

- The entire program should be properly indented.
- Descriptive names of all variables, function names, constants and other identifiers are used.
- To improve code reusability functions are used often.
- Comment statements are added before each code, which involves complex logic.

4.10 OTHER NON-FUNCTIONAL ATTRIBUTES

4.10.1 Realiability

Java Swings is a lightweight component with some high features so the usage of this for developing the GUI provides reliability to the product. The Java swings takes the same shapes in other platform also, so the look and feel is same when working in other platform. The reliability of the product lies upon the hardware and memory constraints too.

4.10.2 Testability

The application is developed in a modular way, so the testability is high. Internal code reviews and functional test will be conducted on each module. The functional points and test cases for each module is prepared and documented. Unit testing is done for each module and an integration testing is done for the entire system. After integration testing the functionality test is carried out for the entire system. The validation test is done on each field based on the requirements.

4.10.3 Maintainability

Since each phase of development is properly documented the maintainability of the system is high.

4.10.4 Portability

The application is running on JVM. So it is portable to any platform provided the JVM is installed in that platform too.

4.10.5 Reusability

All the programs are written as a collection of packages. So they can be re used in similar applications without changing the code. Internally all the operations are put in a function to avoid rework.

4.10.6 Security

The product would be accessible only to authorized users who are employees of TCS. Since we are using Java Swings for developing the product only the byte codes are visible to the user.

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 EXTERNAL DESIGN SPECIFICATIONS

5.1.1 Methodology

The methodology used for the design is Rapid Application Development

Rapid Application Design

A software development process that allows usable systems to be built in as little as 60-90 days, often with some compromises.

Principles behind the Definition

- In certain situations, a usable 80% solution can be produced in 20% of the time that would have been required to produce a total solution.
- In certain situations, the business requirements for a system can be fully satisfied even if some of its operational requirements are not satisfied.
- In certain situations, the acceptability of a system can be assessed against the agreed minimum useful set of requirements rather than all requirements.

Problems Addressed By RAD

- With conventional methods, there is a long delay before the customer gets to see any results.
- With conventional methods, development can take so long that the customer's business has fundamentally changed by the time the system is ready for use.
- With conventional methods, there is nothing until 100% of the process is finished, then 100% of the software is delivered.

Bad Reasons For Using RAD

- To prevent cost overruns (RAD needs a team already disciplined in cost management).
- To prevent runaway schedules (RAD needs a team already disciplined in time management).

Good Reasons For Using RAD

- To coverage early toward a design acceptable to the customer and feasible for the developers
- To limit a project's exposure to the forces of change to save development time, possibly at the expense of economy or product quality.

RAD Uses Hybrid Teams

- Team should consist of about 4 people, including both developers and full time users of the system plus anyone else who has a stake in the requirements.
- Developer chosen for RAD teams should be multi-talented "renaissance" people who are analysts, designers and programmers all rolled into one.

RAD Uses Specialized Tools That Support

- Visual development
- Creation of fake prototypes (pure simulations)
- Creation of working prototypes
- Multiple languages
- Team scheduling
- Teamwork and collaboration
- Use of reusable components
- Use of standard APIs
- Version control (because lots of versions will be generated)
- RAD uses "TIMEBOXING" secondary features are dropped as necessary to stay on

RAD Tends To Work When

- Major use can be made of preexisting class libraries (APIs)
- Performance is not critical
- Product distribution will be narrow (in-house or vertical market)
- Project scope (macro-schedule) is constrained.
- Reliability is not critical
- System can be split into several independent modules
- The project has strong micro-schedule constraints.
- The required technology is more than a year old.

RAD Tends To Fail When

- Application must inter operate with existing programs
- Few plug in components are available
- Optimal performance is required
- Product distribution will be wide (horizontal or mass market)
- RAD becomes QADAD (Quick And Dirty Application Development)
- Technical risks are high due to use of “bleeding” edge technology
- The product is mission or life critical
- The system cannot be modularized (defeats parallelism).

Advantages Of RAD

- Buying may save money compared to building
- Development conducted at a higher level of abstraction(because RAD tools operate at that level)
- Early visibility (because of prototyping)
- Greater flexibility (because developers can redesign almost at will)
- Greatly reduced manual coding (because of wizards, code generators, code reuse)

- Increased user involvement's(because they are represented on the team at all times)
- Possibly fewer defects(because CASE tools may generate much of the code)
- Possibly reduced cost(because time is money, also because of reuse)

Disadvantage Of RAD

- Buying may not save money compared to building
- Cost of integrated toolkit and hardware to run it
- Harder to gauge progress(because there are not classic milestones)
- Loss of scientific precision(because no formal methods are used)
- May accidentally empower a return to the uncontrolled practices of the early days of software development
- More defects (because of the “code-like-hell” syndrome)
- Prototype may not scale up, a B-I-G problem
- Requirements may not coverage (because the interests of customers and developers may diverge from one iteration to the next)
- Unwanted features(through reuse of existing components)
- Successful efforts difficult to repeat (no two projects evolve the same way).

5.1.2 Data flow diagram

Café:

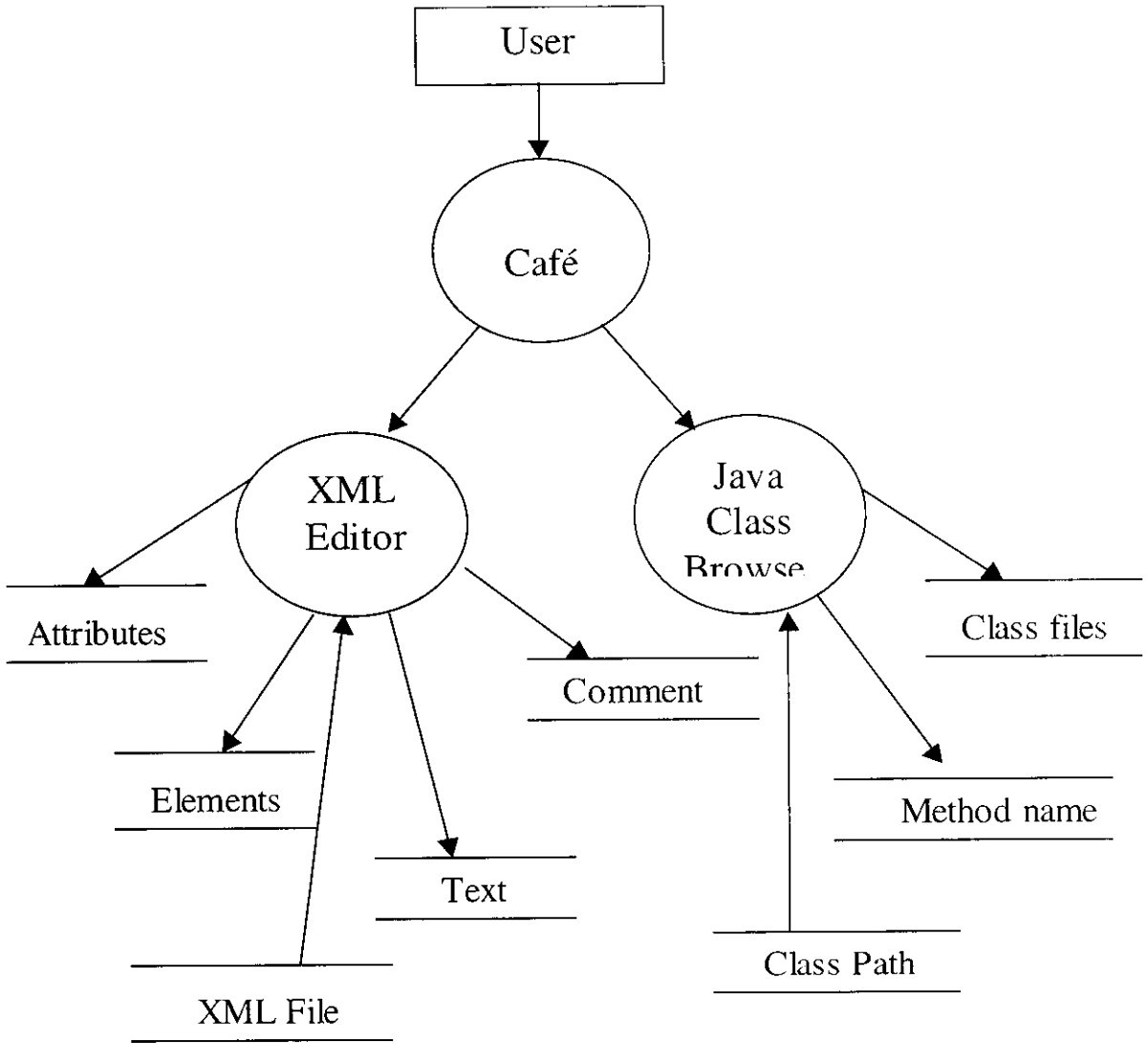


Fig. 5. 1 Data Flow Diagram for café

XML Editor:

Level 0:

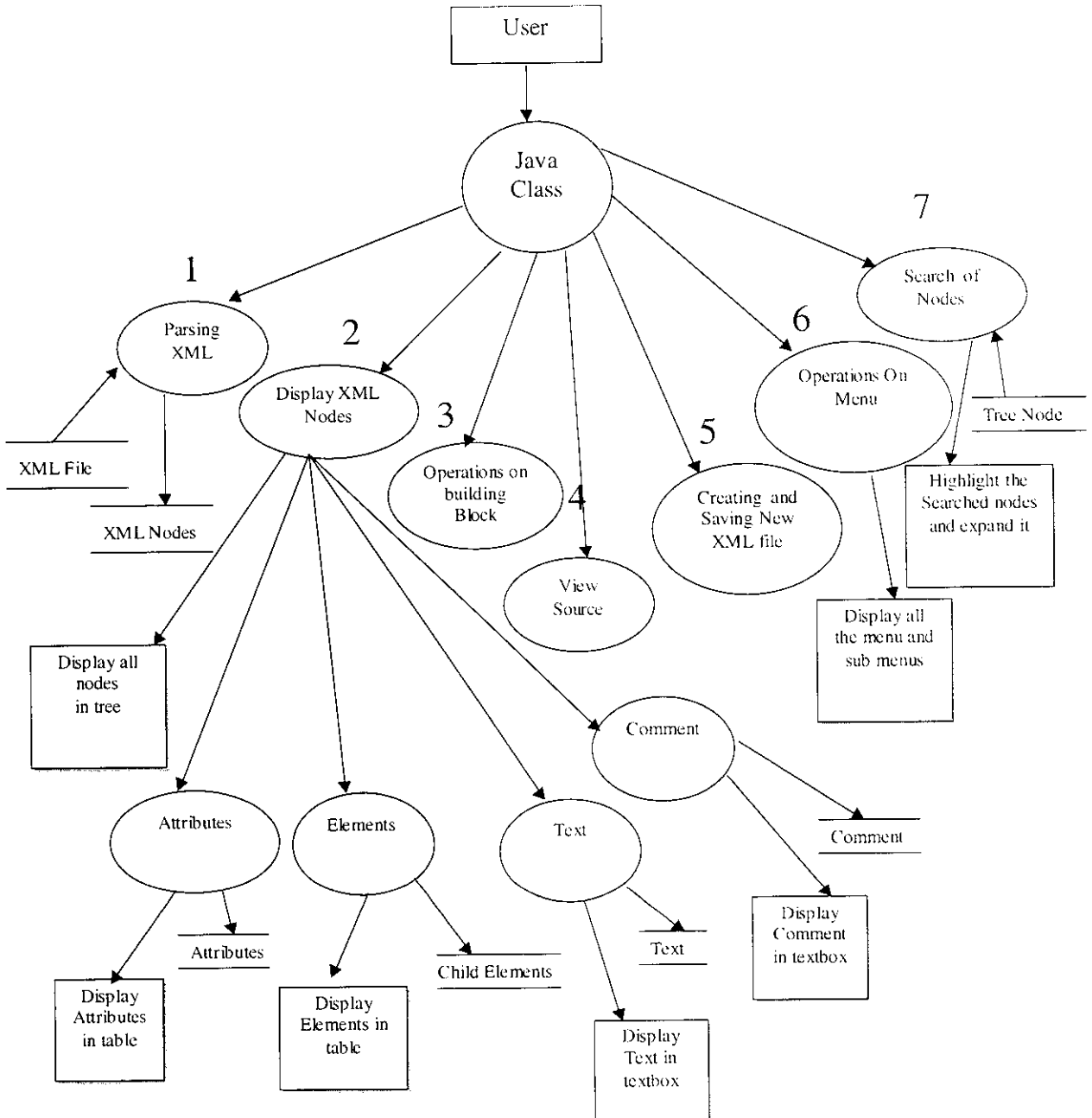


Fig. 5. 2 Data Flow Diagram for XML Editor level 0

Level 1:

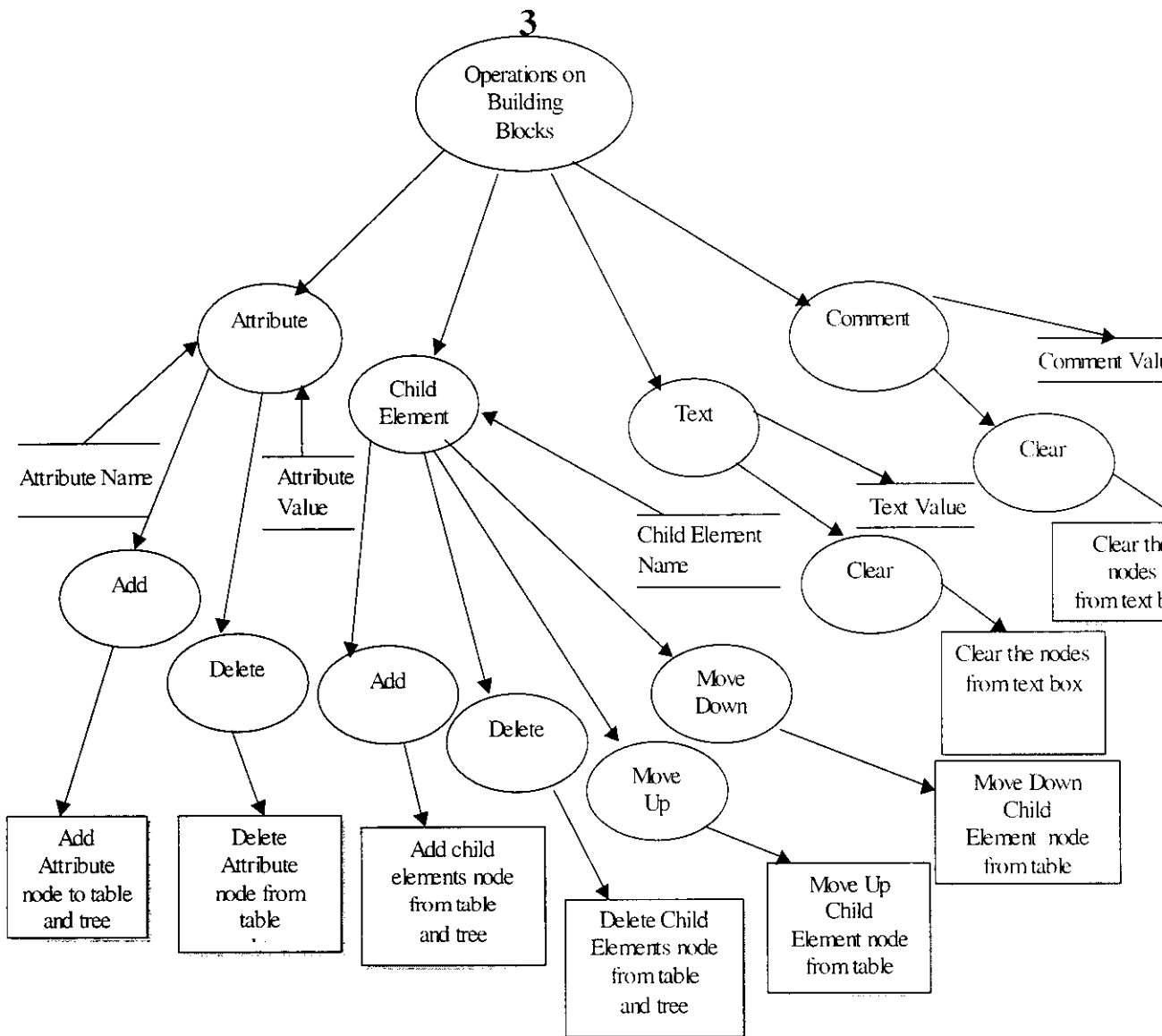


Fig. 5. 3 Data Flow Diagram for XML Editor level 1

Level 1:

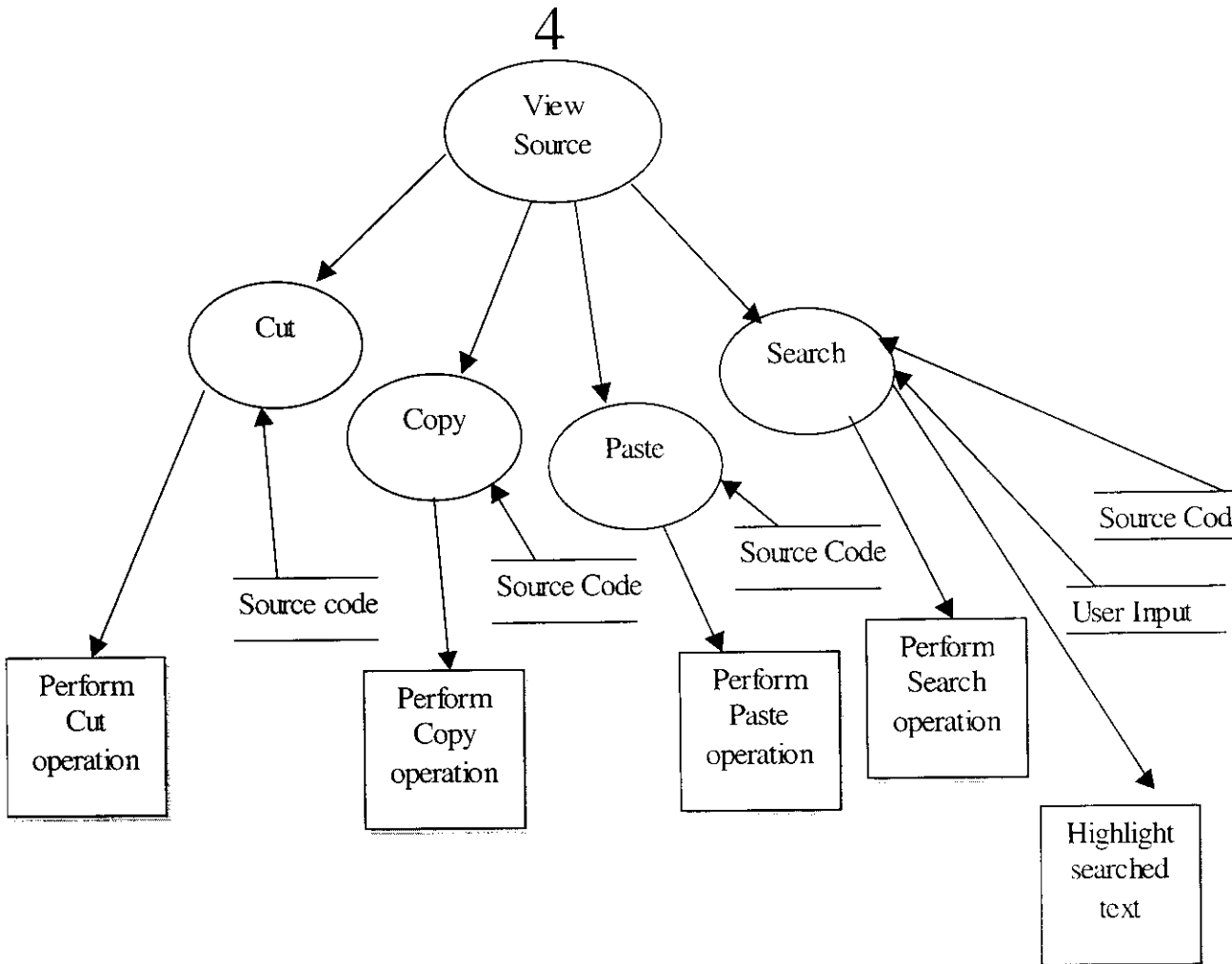


Fig. 5. 4 Data Flow Diagram for XML Editor level 1

Java Class Browser:

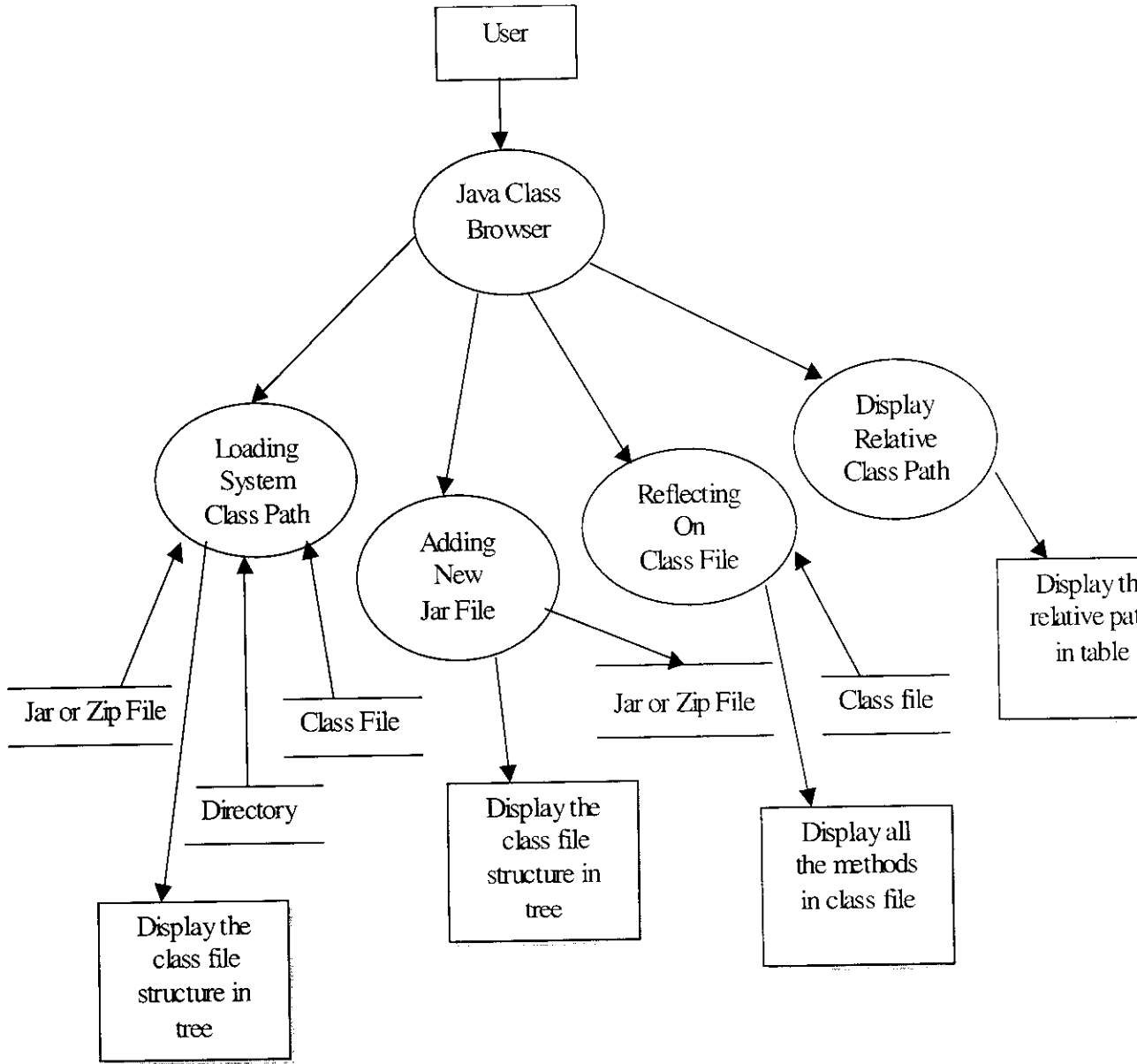


Fig. 5. 6 Data Flow Diagram for Java Class Browser level 0

5.2 ARCHITECTURAL SPECIFICATION

Café from e-Business group of Tata Consultancy Services, Chennai, India, is a development framework for Java applications based on the Model-View-Controller (MVC) design paradigm.

MVC Architecture

In Model-View-Controller (MVC) architecture, the data are represented by the Model, and the View by the visual component. The Controller is the communication between the Model and View objects. The idea of this MVC pattern is to keep the data in a *model* class, display the data in a *view* class and vary the data and view using a *controller* class.

An enterprise application or a simple web application that can contain a mixture of data access code, business logic code and presentation layer code. Maintaining such applications can be difficult as the interdependencies between the various components cause strong serious ripple effects whenever there is a change made to the code. In other words this is a situation of high coupling of components. The high coupling not only creates difficulty in reusing the existing code but sometimes it can make reusability of the code almost impossible costing huge maintenance and re-implementation efforts. The Model-View-Controller design pattern solves these problems by decoupling data access, business logic, and data presentation and user interaction.

MVC provides a host of design benefits. It separates design concerns (data persistence and behavior, presentation and control), decreasing code duplication, centralizing control and making the application more easily modifiable. MVC also helps developers with different skill sets to focus on their core skills and collaborate through clearly defined interfaces. Often MVC functionality is captured in a framework that is reused by different applications.

Level 1:

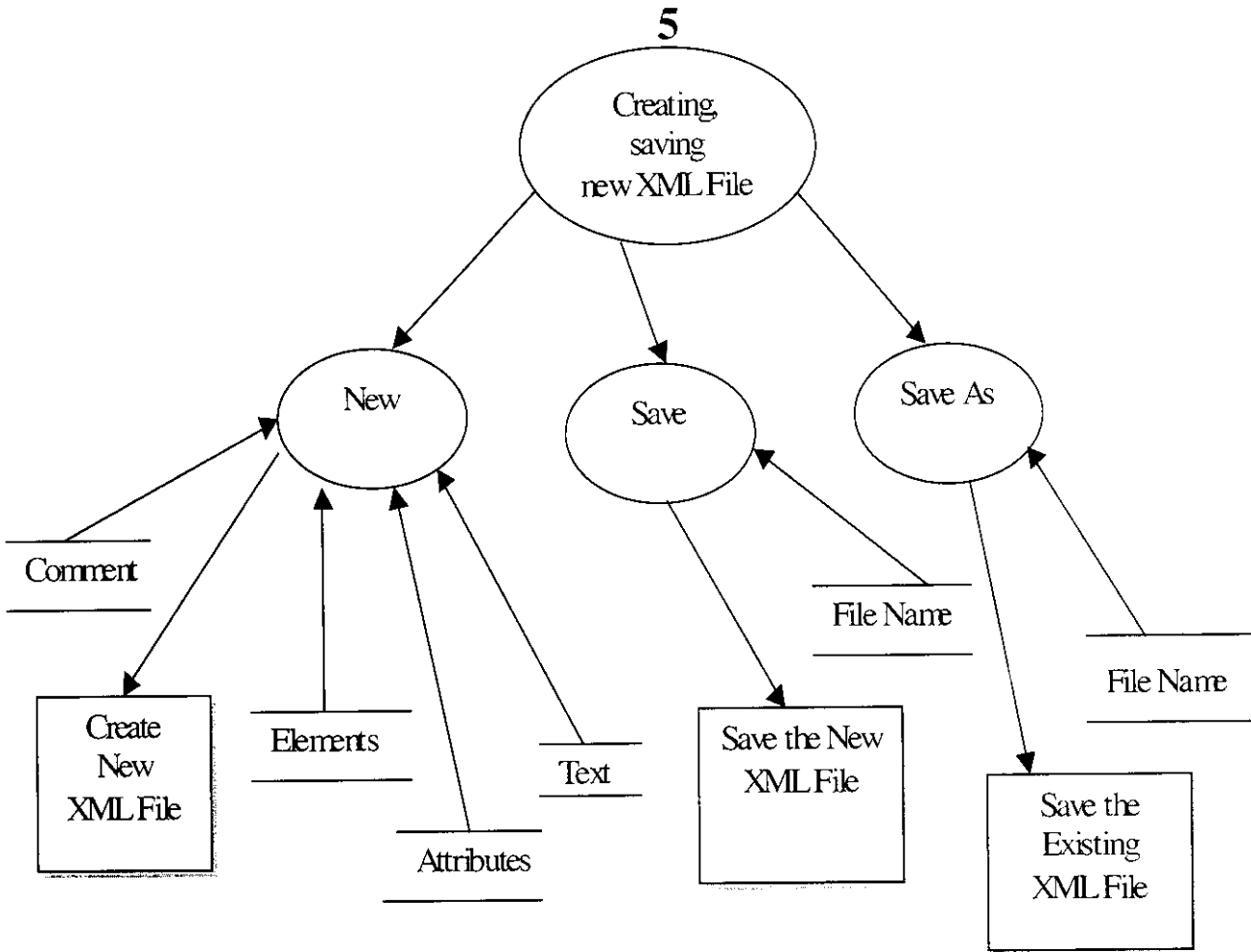


Fig. 5. 5 Data Flow Diagram for XML Editor level 1

5.3 DETAILED DESIGN SPECIFICATIONS

5.3.1 Detailed description

Café

The Café is the Component Assembly Framework for E-Business that has to be developed for the employers in TCS working in E-Business projects. Currently, the Java class files which does most of the operations is incorporated in-between the XML nodes and the XML file is generated, which does the operations. This product is developed to automate the XML code generation and also to provide a good user interface for selecting the class file and to include it in the XML file. The café has two modules, which does all the proposed functionality's, they are as follows.

- XML Editor
- Java Class Browser

XML Editor

The XML Editor acts as an individual application as well as a part of café. The XML editor would be a tool that would allow a user to create, edit, modify and save an XML file. The XML Editor consists of the following modules: -

Parsing XML: -

In the user interface the user has to choose a XML file and open it. Using DOM parser the XML file will be parsed and the file is checked for a valid file or not, in case if not a valid file error message is displayed.

Display XML Nodes: -

From the parsed XML file the node type has to be determined and the building blocks of XML such as attributes, elements, text, comment etc are displayed in the tree structure using Jtree component. The name, value of each node based on the type is categorized and displayed either in table or textbox using respective components.

Operations on Building Blocks of XML: -

Buttons are provided to add & delete the attributes, child elements. The Move Up and Move Down of nodes is performed on clicking the respective button. In case if the user

selects to add an attribute or child element a separate row is added at the end of the table and the user can type the needs for it to get reflected in the Jtree. Similarly for Move Up or Move Down the selected row in the table moves correspondingly according to the selection made. For all the operations performed in the right hand side, it has to be reflected in the Jtree.

View source: -

The source code for the XML file is displayed in the JeditorPane. Only when the user selects the root node from Jtree he/she is displayed with entire source is displayed otherwise the source corresponding to the particular node alone is displayed. Other operations such as cut, copy and paste are also performed on selecting the text.

Creating and Saving New XML file:

A new XML file can be created by adding the required nodes and it can be saved as XML. Before saving, the validation for the file is checked. New Save dialog is displayed if the user selects saveAs menu option or if he/she saves the file for first time.

Operations on Menu:

Various links to operations such as cut, copy, and paste of Jtree nodes, validation of DTD, help menu for new users and insertion of attributes, elements, text, comment nodes are provided in the menu.

Search of nodes in Jtree and view source: -

The user can select what type of XML node has to be searched and respective parent node will be expanded and searched node will be highlighted. In case of search in view source the text that has to be searched is entered and it highlights the searched text.

Java Class Browser

Java Class Browser lists down the collection of class files fetching from the classpath which is set by the user for their respective project. This application consists of following modules they are: -

Loading System Class Path

The class path is got from the system. The Class path contains a directory path, a zip or jar file, a class file and other resources that are irrelevant to the application. If the

class path returns a directory structure the sub directory has to be navigated to get the class file. In case of zip or jar files the file has to be zipped to get the class file and all other path, which does not have class files are not displayed. The valid contents are displayed in the Jtree.

Adding New Jar file

A New jar file can also be added in the classpath in case if the classpath is not in the system. All the operations similar to the previous modules are performed to load it in the Jtree.

Reflecting on Class File

Using reflect package the methods with the arguments along with the access modifiers are got and add these contents to the Jtree. For each access modifiers various icons are loaded to differentiate it from the others.

Display relative class path

When a user clicks on the Tree Node get the relative path and display it in the Text Field. The last five-selected path is displayed in the table just to know the visited paths.

5.3.2 Pseudocode for each routine

XML Editor

Parsing XML: -

1. Pass the XML file name along with its path to the DOM object.
2. Create a object of class Document
3. Use these to parse the XML file.
4. Identify the various types of nodes and the values associated with it.
5. Call the routine for display of XML tree node

Display XML Nodes: -

1. Start
2. With the given type of the node, get the values associated with them.
3. Add this node to the Tree
4. Display the node type in it.
5. Display the values in the nodes in the appropriate place.
6. Render the icons with the associated type.

Operations on Building Blocks of XML: -

1. Start
2. To add child nodes, Attributes and its values the add routine is called.
3. Place the cursor and add a new row and focus it for user to type in the entry.
4. This adds the new value that is being typed by user.
5. The tree node is also added by one of its built in methods
6. The move up and move down of nodes is performed if the respective button is clicked.
7. Delete is activated by the mouse key pressed in the delete icon
8. This deletes the row from the display and also the associated row.

View source: -

1. Start
2. View source is triggered by mouse click even on the view source tab.
3. Current node being clicked is read and the entire contents are displayed on the view source
4. When the root is visited the entire document is shown on the source as it visits all the nodes beneath the root element
5. Cut, Copy, Paste of Text is allowed and is done using the text area cut, copy, and paste routine in Java.
6. Search triggers a algorithm where it searches for user requested text

Creating and Saving New XML file: -

1. Start
2. Create new XML removes all the children in the root if they exists any
3. Saving the XML is done by visiting the root node
4. Calling the view source routine
5. Convert the stringBuffer to a String
6. Open the file in Save
7. Ask for a new file in case of save as
8. Write this string into the file using the output stream
9. Save the file
10. Close the file.

Operations on Menu:

1. Start
2. Validating a DTD is done by setting the validation true in the SAX parser object
3. Cut routine stores the values using deep clone method and removes that node
4. Copy call the deep clone method and keep alive the current node
5. When paste option is selected the copied node or cut node get pasted as a child node to the current selected node.
6. The insert in an object will allow the user to add in between the nodes, the user can insert comment, text, child elements etc.
7. Help menu provides the guidelines for the new users.

Search of nodes in Jtree and view source: -

1. Start
2. Node search can be done on the user-selected node.
3. The search criteria are mentioned by selecting the appropriate checkbox.
4. The search algorithm is called and the criteria are passed on to the algorithm.
5. The results are stored in a linked list

6. The linked list contains the tree path of the node found matching to the requirements
7. The linked list is traversed and the tree is expanded and nodes are highlighted and shown.
8. For view source similar linked list is created and results are traversed.

Java Class Browser

Loading system class path: -

1. Start
2. Use Java system.getProperty and get the system class path
3. Use string tokenizer and separate the path and jar files.
4. Put both in different array lists.
5. Read the First array list and get the jar files, class files and separated them with accordance with their paths.
6. Pass these values to the Adding New jar file Module.
7. Get the name and pass them to the module of adding a Jtree.

Adding New Jar File

1. Start
2. Display a dialog box to open a file
3. Get the file name and send it to function which opens a jar file.
4. Open the jar file using the ZIP class methods in Java
5. If a file ends with a “.class” consider it to a valid class file
6. Add these above mentioned names to a array list
7. Add the names with their path
8. Send this array list to build tree routine

Build tree routine

8.1 : Start

8.2 : Read the values from the array list.

8.3 : Add it to the Jtree node with accordance with their path

8.4 : Call reflecting class file.

8.5 : Call the overridden method defaulttreecellrender.

Defaulttreecellrender routine

8.5.1 : Start

8.5.2 : Override the DefaultTreeCellRenderer by using *super*

8.5.3 : Use the setIcon method to set appropriate icons for class, methods, package, public, static, private, protected.

Reflecting the Class Files

1. Start
2. Use the reflect package to get the access modifiers, methods and arguments
3. Add these information as the child node of the respective class tree node
4. Return its parent node to the build tree operation.

Display relative class path in the Text Field

1. Start
2. Get the node where the mouse has clicked
3. Get the nodes path
4. Convert it to a path by replacing all “/” by “.” and add this to the text field of the XML Editor.

CODING AND TESTING

6. CODING AND TESTING

Coding converts the design model into executable domain. Java swings is the ideal choice for implementation since it provides the GUI for the design model with the power of JAVA. This makes this project module easy to extend or improved for future purpose.

6.1 CODING STANDARDS

Begin each file with a comment including the file name and/or related identifying information.

The template of a file header as follows:

```
/******
```

```
*CREATED BY:
```

```
*
```

```
*NAME DATE PURPOSE:
```

```
*
```

```
*Project:
```

```
*Operating Environment:
```

```
*Description:
```

```
*Provide the overview of the approach and the underlying concepts.
```

```
*IMPORT PACKAGES
```

```
*****/
```

- Descriptive names of all variables, function names, constants and other identifiers are used.
- Single letter identifiers are only for the counter in loops.

- Variable names are started with lower case.
- Multi-word identifiers are internally capitalized
- Brackets begin and end on a new line and are exist even for one-line statements
- Import statements must be fully qualified for clarity.
- In-line comments are used to explain complicated sections of code, such as loops.
- Two blank spaces are used to separate logical sections of code within a method.
- All the classes begin with a capital letter and all packages are started with lower case.
- Methods begin with a lower case letter.
- Class data members and member functions are prefixed with m opening and closing braces are aligned to match conditional keyword (e.g., if, else, do, while, for) and enclosed statements are indented one more level.

6.2 TESTING

Testing of software extends throughout the coding phase and it represents the ultimate review of specification, design and coding. A series of test cases are created with an intention of testing the software. Based on the way the software reacts to these tests, we can decide whether the product that has been built is robust or not.

6.2.1 Unit Testing

Each of the modules developed was tested independently. The following were considered during the unit test:

- The module interface is tested to ensure that information properly flows in and out of the unit under test.
- The local data structure is examined to ensure that data stored temporarily maintains integrity throughout the unit's lifetime.
- Boundary conditions are tested.

- All independent paths through the control structure are exercised to ensure that all statements in the module have been executed at least once.
- All error-handling paths are tested.

6.2.2 Validation Testing

It is said that validation is successful when the software functions in a manner that can be reasonably expected by the customer. This type of testing is very important, because it is the only way to check whether the requirements given by the user have been completely fulfilled. However, it does not depend on the user inputs to perform the validation testing. Validation testing is done for all user interfaces modules. All the validation codes are tested before they are incorporated with the modules or subsystem.

6.2.3 Integration Testing

Once the modules are tested individually under the unit testing strategy, it is necessary to put all these modules together-interfacing. It is here that the data can be lost across the interface, one module can have an inadvertent, adverse effect on another. Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit-testing modules and build a program structure that has been dictated by design.

Top-Down Integration

This is an incremental approach to the construction of the program structure modules are integrated by moving downward through the hierarchy, beginning with the main control module. Modules subordinate to the main control module are incorporated into the structure in either a depth-first manner or breath-first manner.

The system developed has been put to this method of testing as well. In the system the modules are incorporated into the structure using the depth-first search manner. This integrates all the modules on a major control path of the structure. This verifies the major control or decision points early in the test process.

PROJECT LEGACY

7. PROJECT LEGACY

7.1 PROJECT DESCRIPTION

The project is done for the E-Business project team, which is a part of Tata Consultancy Services, Chennai-119. The project is to build a GUI for the proposed framework and to develop a tool, which is used to include the class files in the XML node. The tool has to generate the XML code by itself and save as XML file. The project covers the development of three tools:

- Café
- XML Editor
- Java Class Browser

7.2 INITIAL EXPECTATIONS

Café:

- Developing a tool that has to be completed within the limited time.
- The look and feel of the product has to be really good.
- The low-level design of the module was expected to be submitted before starting the coding.
- Integration of XML Editor and Java Class Browser.

XML Editor:

- Developing it as an individual product as well as a part of the café.
- Manage the design part of the whole application.
- The usage of Java Swings to the full extent and to explore all the components.
- Completing within the given time.

Java Class Browser:

- Developing it as an individual product as well as a part of the café.
- The look and feel of the product has to be really good.
- Providing additional features by comparative study.

7.3

CURRENT STATUS OF THE PROJECT

- The café is yet to get released which is in the integration part and yet to complete the testing phase.
- The XML Editor has been completed and its being used only by few teams members in order to see the performance of the product.
- The Java Class Browser has also been completed and released for limited people only.

7.4

REMAINING AREAS OF CONCERN

- Based on the feedback from the users, additional features have to be included.
- The integration part of the two applications may lead to problems.
- To make the product look still better by adding gifs and images.

7.5

TECHNICAL LESSONS LEARNED

- Working with Java Swings, Java, XML, etc.
- Coding according to the Java Standards.
- Study on providing security to the FTP client and server.
- Exception Handling.
- Developing a product to suit the requirement.
- Designing the application.

7.6 **MANAGERIAL LESSONS LEARNED**

- The flow of the software company.
- Conducting a seminar with more interaction.
- The value of sharing knowledge with other team members.
- How to move along with higher officials?
- How the Project leader leads the team in an efficient way?
- How to extract work from the team members?
- Working under pressure competing with date of delivery of product.

CONCLUSION AND
FUTURE
ENHANCEMENTS

8. CONCLUSION AND FUTURE ENHANCEMENTS

8.1 CONCLUSION

The TCS Framework tool for e-business has been built, and it's under testing phase. The XML Editor and Java Class Browser is completed and deployed in TCS E-Biz division. The very intention of making this application has been successfully achieved. The cumbersome work of creating a XML file and adding Java classes as input has been made easy for programmers. The errors that earlier occurred while writing an XML file has been brought down to the least.

The GUI has been devised in such a way that it is easy for programmers with less knowledge of XML can work at ease. By using this application the time required for building a XML file is at the minimum.

A very efficient cut, copy, paste operation has been built, as it is often required by the user, most of the time it is the same kind of elements that is repeated in the application.

Special functions such as Validating DTD has been found to be of much use to the programmers as they don't have to worry about validating the XML document as to whether it conforms to the requirements.

On the whole it can be said that this application has been of much use to the programmers and that the basic object of maximizing efficiency, minimizing errors and ease of use has been achieved.

3.2 FUTURE ENHANCEMENTS

The system has been developed flexible enough to support future enhancement. It has been designed such that it accepts the modifications without affecting the presently developed systems functionally to a maximum extent.

This application can be further enhanced in the context of: -

- By giving an option of adding a child XML file so that, the user can create a template like XML and can simply keep adding the template and filling in the required details, this template addition can add more ease in work for the programmers.
- A separate input form for the above mentioned child XML file can be provided as they could be of any type and the info provided in this form can be very useful in the context relevant data can be filled in directly instead of using the standard interface.
- The Java class browser, is merely a display tool and can send text to the XML Editor as input. Since in E-BIZ division most of the work is done in Java, so the Java files used can be compiled and the output of those can be seen in the application itself. This would eradicate and doubts in the minds of the programmers when they are about to give the Java class files as input to the XML editor.

APPENDIX A- TEST CASES

APPENDIX A-TEST CASES

Test Cases are used for checking the functionality and defect identification. The various Test Cases used in modules are:

Test Case for XML Editor

Module Name: Parsing XML

S. No	Functional Point	Passed
1.	On click of the File Menu, a popup menu has to appear with appropriate submenu.	✓
2.	The open dialog has to appear when the respective menu is selected.	✓
3.	The Xml File is selected and parsed to DOM Parser.	✓
4.	If the file is not valid then error message is displayed.	✓
5.	On opening the application the entire table values, test values has to be null, i.e. nothing has to be displayed.	✓
6.	Only "RootElement" text should appear in Jtree before opening XML file.	✓

Module Name: Display XML nodes

S. No	Functional Point	Passed
1.	The XML nodes are displayed in the tree structure.	✓
2.	The building blocks of XML such as attributes, element, text, comment has to be displayed in the table or text box.	✓
3.	The current node being clicked by the user has to be displayed in the text box.	✓
4.	Based on the click event in tree the attribute, child element, text and comment have to get changed in the table, if it exists in that node.	✓

Module Name: Operations on Building Blocks of XML

S. No	Functional Point	Passed
1.	On click of the attribute table '+' button new row has to be added at the end of existing row.	<input checked="" type="checkbox"/>
2.	On click of apply button the typed information in the last row of table has to be added to the tree.	<input checked="" type="checkbox"/>
3.	On click of the '-' button the selected row has to be removed from the table as well as from the tree.	<input checked="" type="checkbox"/>
4.	Before clicking '+' button the apply button has to be disabled	<input checked="" type="checkbox"/>
5.	After clicking '+' button, it has to be disabled and apply button has to be enabled	<input checked="" type="checkbox"/>
6.	On click of move up arrow or move down in the child elements the selected node is moved up and moved down respectively, reflecting the changes both in table and tree.	<input checked="" type="checkbox"/>
7.	On changing the selected element name and 'ok' button is clicked then the name is changed in both textbox and tree node.	<input checked="" type="checkbox"/>
8.	The clear button for text and comment has to clear the contents of the application.	<input checked="" type="checkbox"/>
9.	On change of value in text and comment and if ok is clicked then the value is changed in tree node as well as text box	<input checked="" type="checkbox"/>
10	On click of cancel button the application has to get exit.	<input checked="" type="checkbox"/>

Module Name: View Source

S. No	Functional Point	Passed
1.	The entire source code has to be displayed while the root node is selected.	✓
2.	On selection of any node in tree corresponding code alone has to be displayed.	✓
3.	On right click of view source popup menu with respective sub menu options has to appear	✓
4.	The cut, copy and paste have to perform their respective action on selection of any text.	✓
5.	The popup menu should not get displayed when no text is selected.	✓

Module Name: Creating and Saving New XML file

S. No	Functional Point	Passed
1.	When 'new' menu is clicked from 'file' menu then the existing tree has to be cleared.	✓
2.	The values in the table and text box has to cleared when 'new' is selected	✓
3.	Save dialog has to appear only when the user saves the XML for the first time.	✓
4.	SaveAs dialog has to appear whenever the user clicks the saveAs menu.	✓
5.	The view source contents has to get cleared when new option is clicked.	✓

Module Name: Search of nodes in Jtree and view source

S. No	Functional Point	Passed
1.	On click of the search menu from Edit a new dialog has to be opened	✓
2.	If the textbox in the search dialog box is null then the next button has to be disabled.	✓
3.	Until anyone of the checkbox is checked the next button should not be enabled.	✓
4.	If the node that are to be searched is found then it is highlighted in the tree.	✓
5.	The tree has to be expanded until it finds all the nodes that have to be search is completed.	✓
6.	On click of new search the value in textbox has to be cleared.	✓
7.	On click of view source search, separate dialog box has to be opened.	✓
8.	If searched item is found in view source, it has to highlight and traversed until all the text in view source is visited.	✓

Test Case for Java Class Browser**Module Name: Loading System Class Path**

S. No	Functional Point	Passed
1.	The class path has to be displayed in the tree structure if it is found to be valid	✓
2.	The class file should not display the common directory name; instead only the class name is displayed.	✓

Module Name: operations in menu

S. No	Functional Point	Passed
1.	On click of 'exit' from 'file' menu the application has to get closed.	<input checked="" type="checkbox"/>
2.	On click of 'cut' from 'edit' menu the current selected tree node has to be cut.	<input checked="" type="checkbox"/>
3.	On click of 'paste' from 'edit' menu the nodes has to get pasted as a child to the current selected tree node.	<input checked="" type="checkbox"/>
4.	On click of 'copy' from the 'edit' menu the nodes has to be copied.	<input checked="" type="checkbox"/>
5.	On click of 'Attributes' from the 'insert menu the cursor has to get pointed to the last new empty node.	<input checked="" type="checkbox"/>
6.	When 'DTD' is clicked which is in the validation menu then appropriate message has to be displayed based on the valid check.	<input checked="" type="checkbox"/>
7.	When 'toolbar' in view menu is clicked the toolbar is hidden and on another click the toolbar is made visible.	<input checked="" type="checkbox"/>
8.	The help menu opens the separate dialog box with information on the product.	<input checked="" type="checkbox"/>

Module Name: Adding New Jar file

S. No	Functional Point	Passed
1.	On click of the '+' Button the open dialog box has to be opened.	<input checked="" type="checkbox"/>
2.	Only the jar file, class file can be included, other than this error message has to be displayed.	<input checked="" type="checkbox"/>
3.	The relative path of the selected file is displayed in the table.	<input checked="" type="checkbox"/>
4.	The move up and move down button has to move the relative path one step above and vise versa.	<input checked="" type="checkbox"/>

Module Name: Reflecting the Class File

S. No	Functional Point	Passed
1.	The method name is added along the Jtree below the respective class file.	<input checked="" type="checkbox"/>
2.	The access specifiers have to be included before the method name.	<input checked="" type="checkbox"/>
3.	Different icons have to be included for different access specifiers.	<input checked="" type="checkbox"/>

Module Name: Displaying relative class path

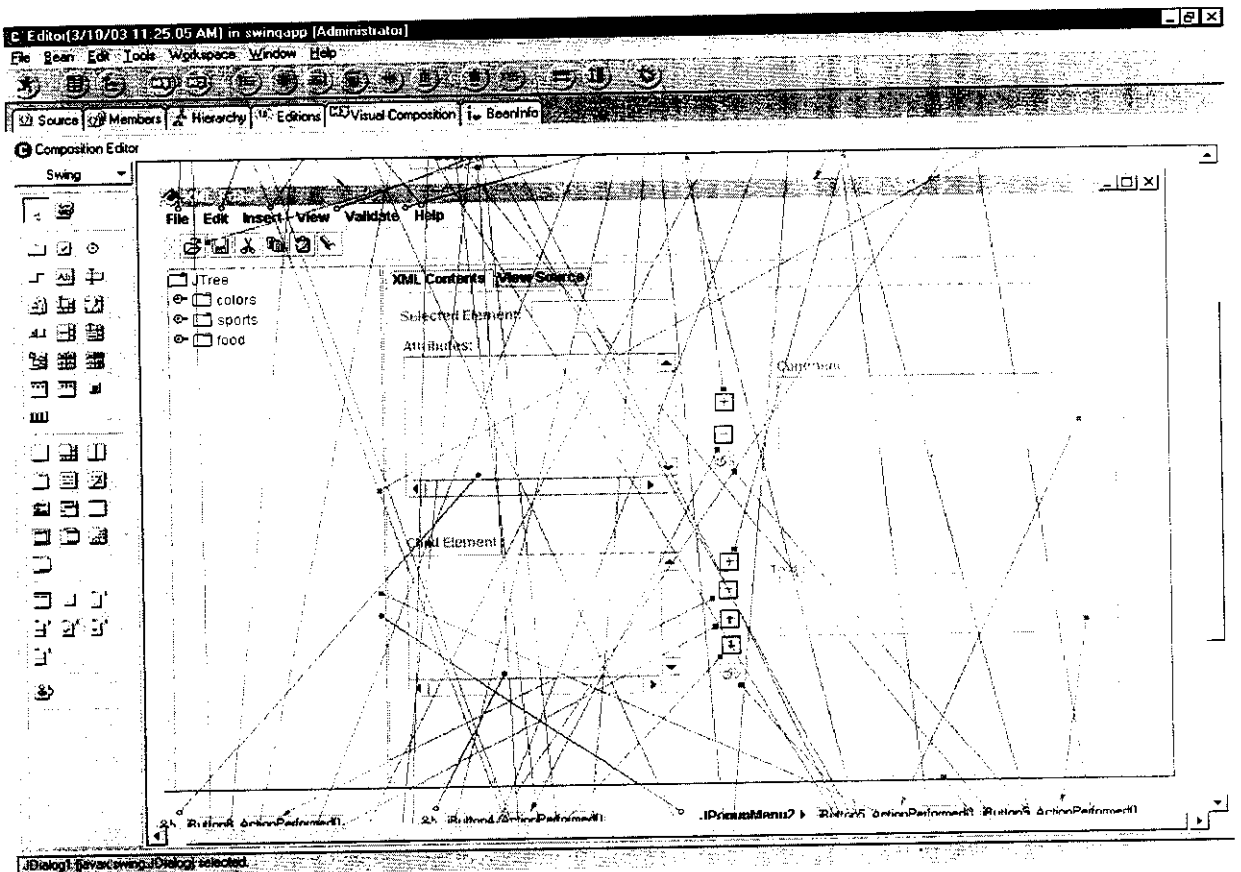
S. No	Functional Point	Passed
1.	On click of '-' button the relative path from the table is removed.	<input checked="" type="checkbox"/>
2.	The selected class file relative path is added appended in the table.	<input checked="" type="checkbox"/>

APPENDIX B- SCREEN SHOTS

APPENDIX B-SCREEN SHOTS

Screen Description:

The XML Editor during the development phase under visualAge for Java is shown. The lines indicate the various functions written on each component.

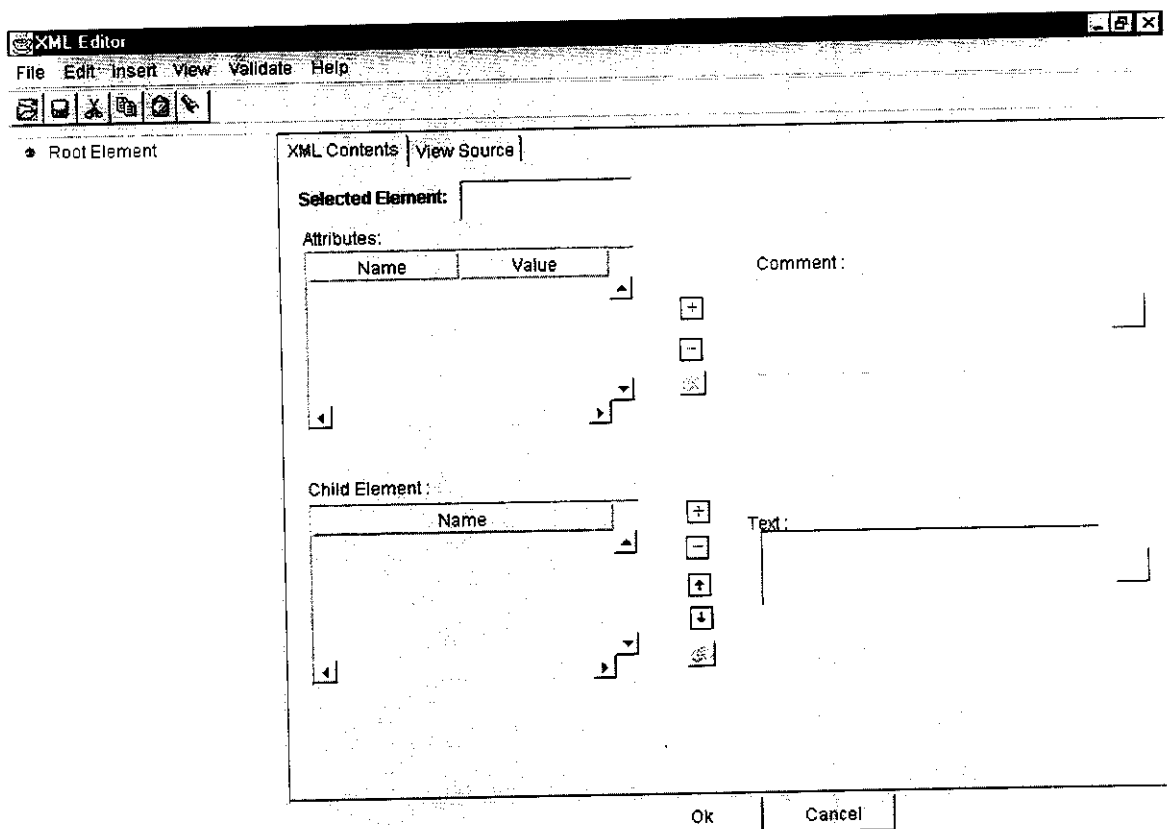


Screen Shots. B. 1

Working Environment of VisualAge for Java

Screen Description:

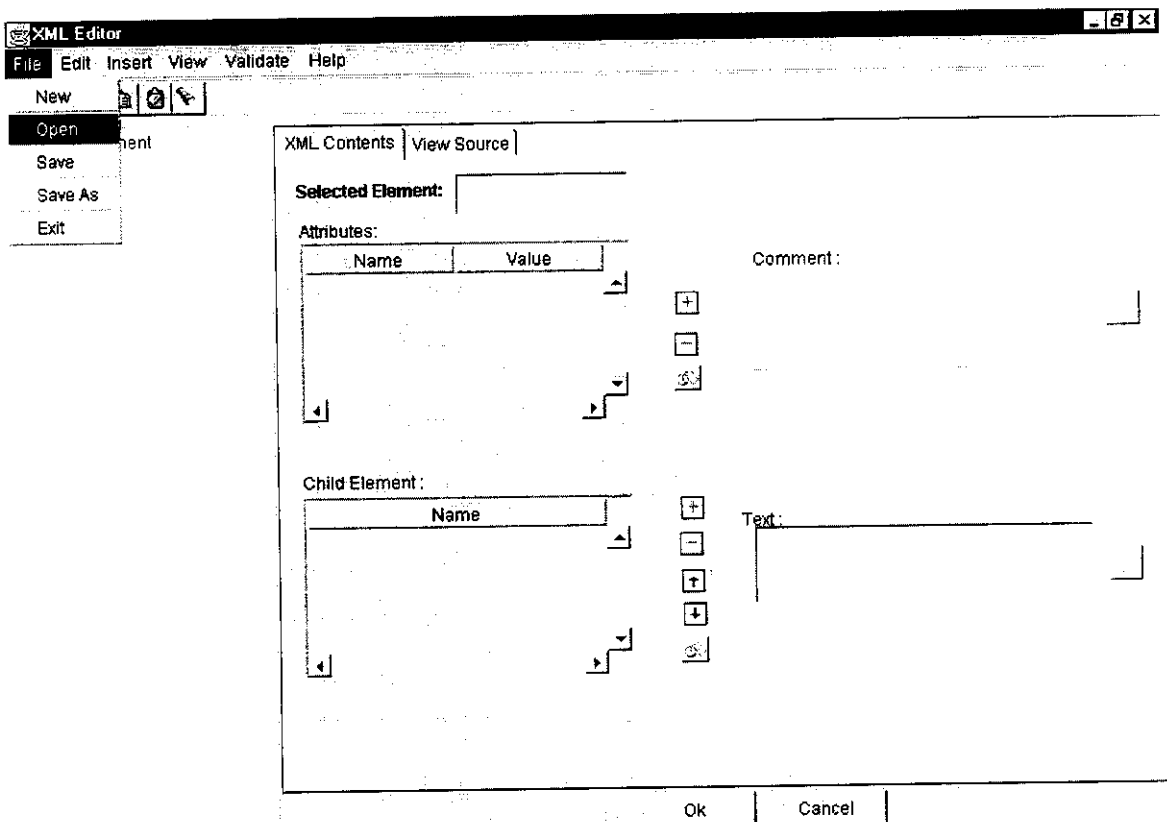
This screen appears when the application is opened. This is the opening screen of the XML editor. The Java class browser is an internal frame not shown.

**Screen Shots. B. 2**

Opening Page for XML Editor

Screen Description:

When the user need to open a XML file, this would open an OPEN dialogue box where the user can browse the system and open a XML file.

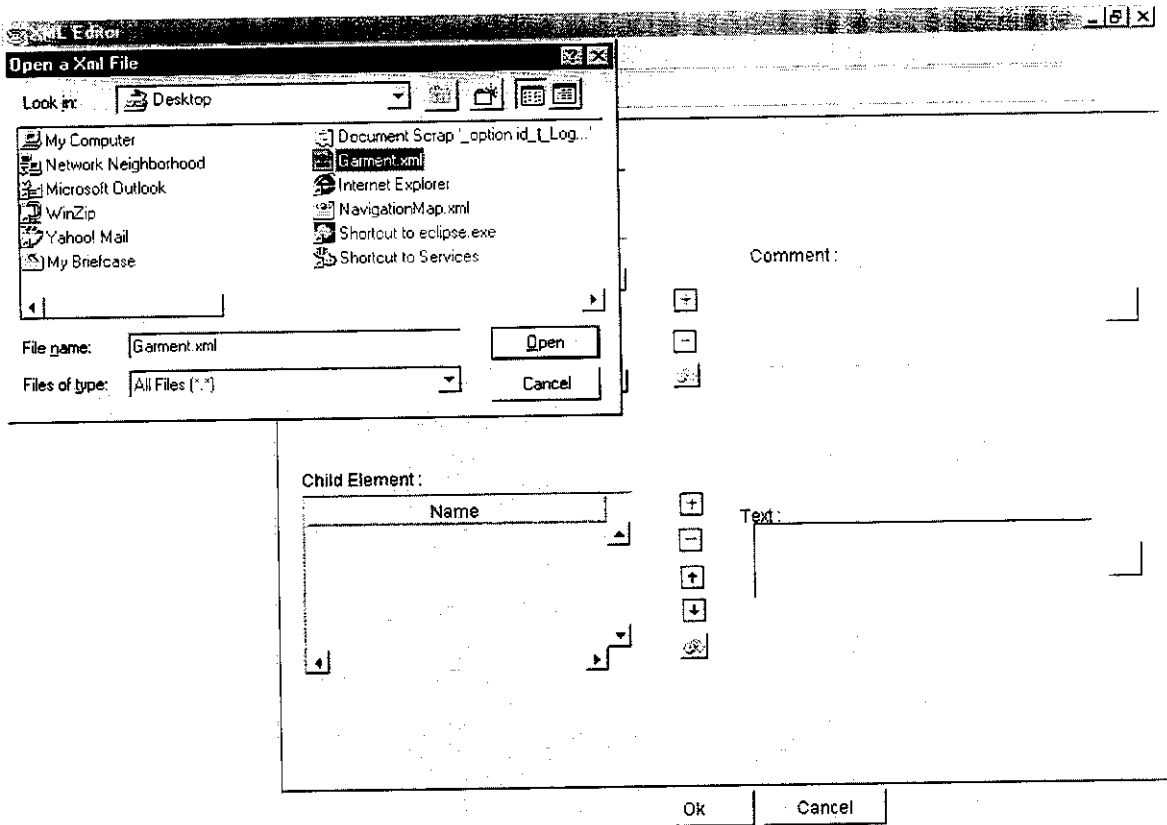


Screen Shots. B. 3

Open Sub-menu is selected

Screen Description:

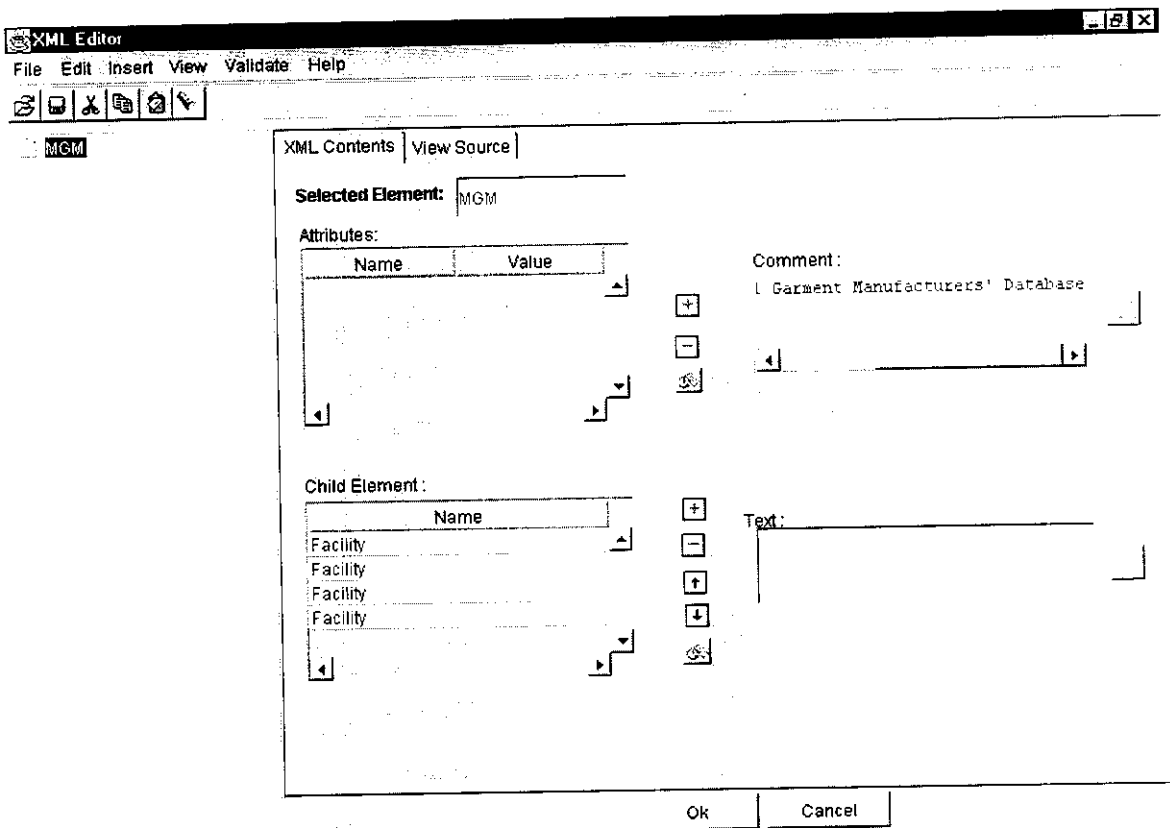
After the user selects the XML file as shown in the shot, this document (XML) file is parsed by DOM for the next stage of displaying it in the Jtree.



Screen Shots. B. 4
Open dialog for opening a file

Screen Description:

The XML was found valid and then it was parsed and now it is shown in a perfect hierarchy of a Jtree. Also the other information such as the comment, Child elements are displayed in the appropriate box.

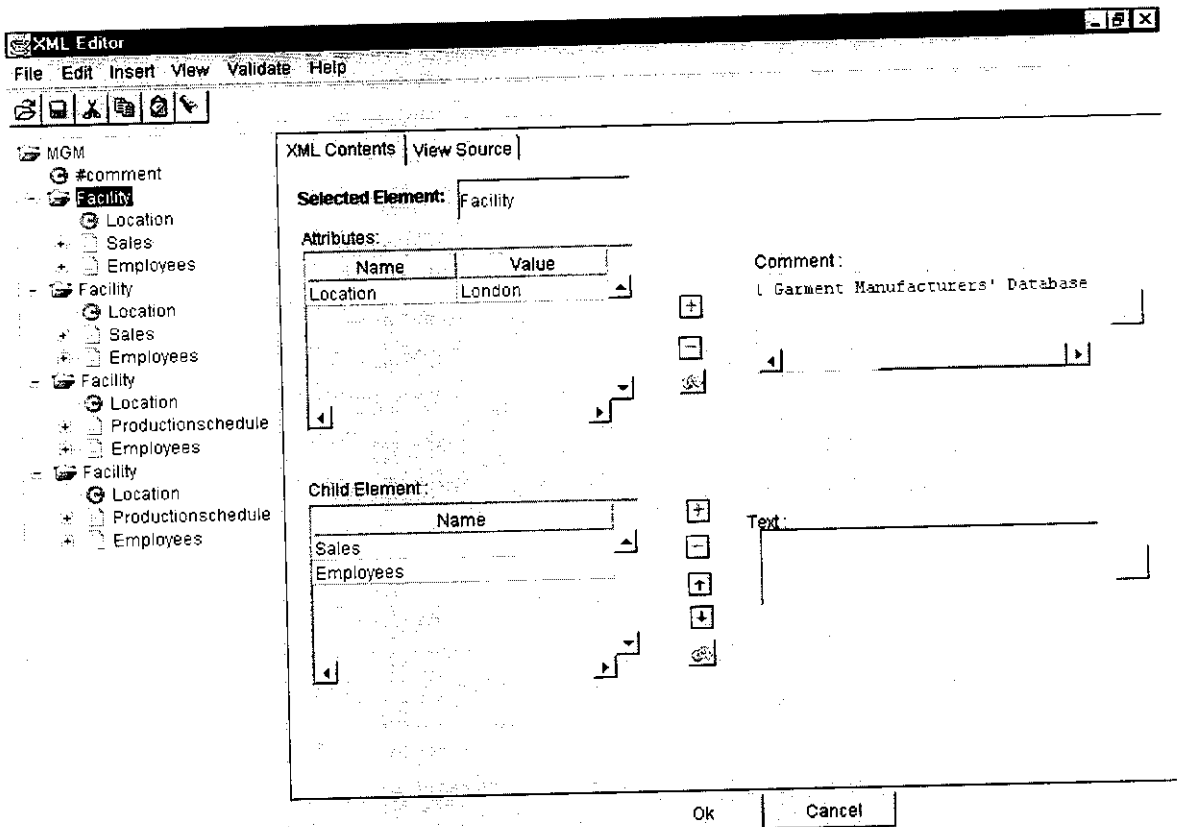


Screen Shots. B. 5

The parsed XML File displayed as Tree node

Screen Description:

We could view the contents of the XML file by expanding the tree and the values associated with it are shown in the appropriate places. So here we can see the actual XML building blocks at work.

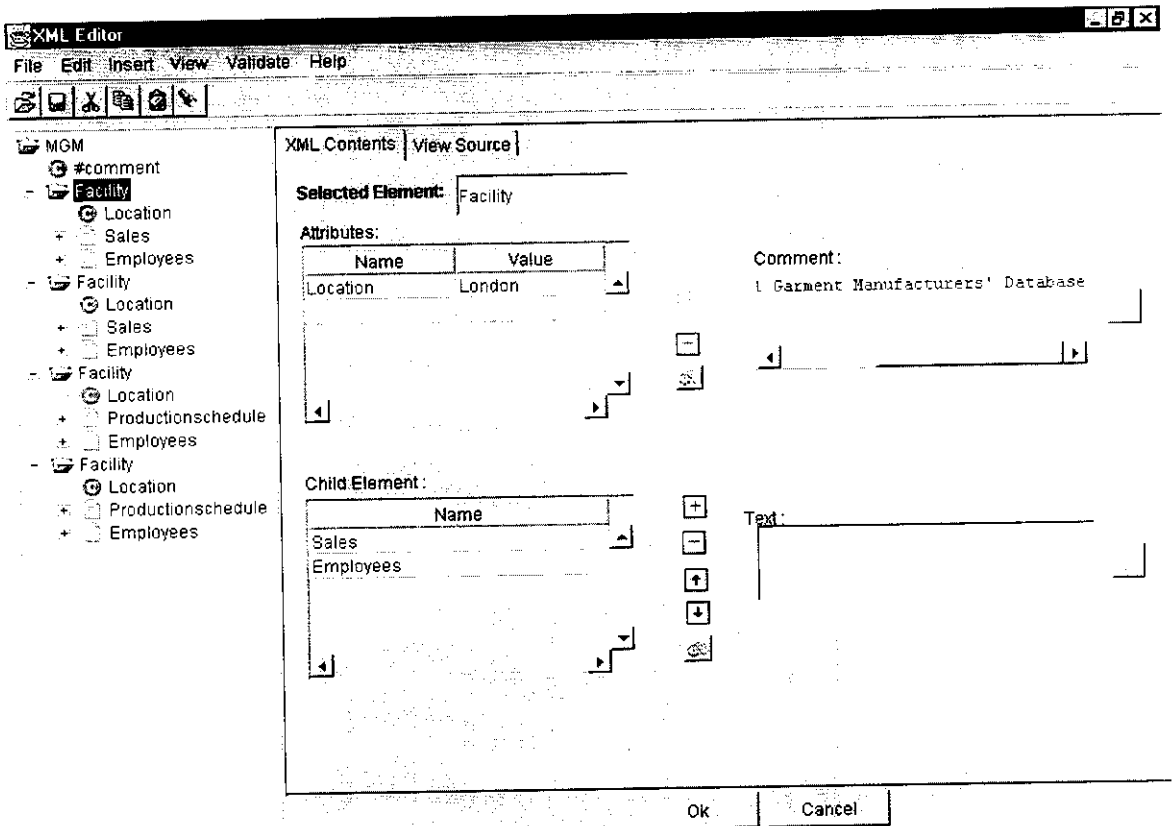


Screen Shots. B. 6

For selected Node Various Building Blocks are displayed

Screen Description:

This screen shows how easy is to add attributes to the XML file, just clicking on a “ADD” icon a new attribute could be added easily.

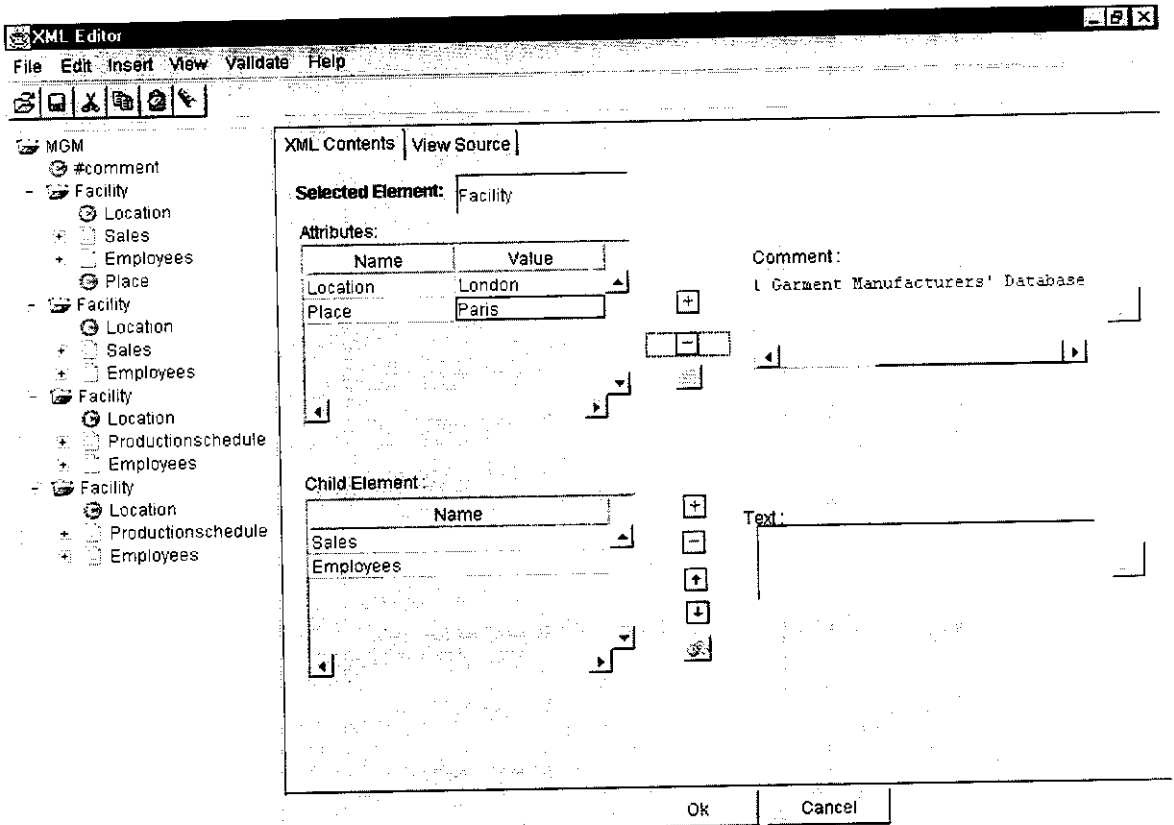


Screen Shots. B. 7

Adding new nodes

Screen Description:

Here we could see an attribute being added and it is reflected on the tree and this will lead it to generation of code necessary for the XML file automatically. So it is quite easy to access and modify the XML document.

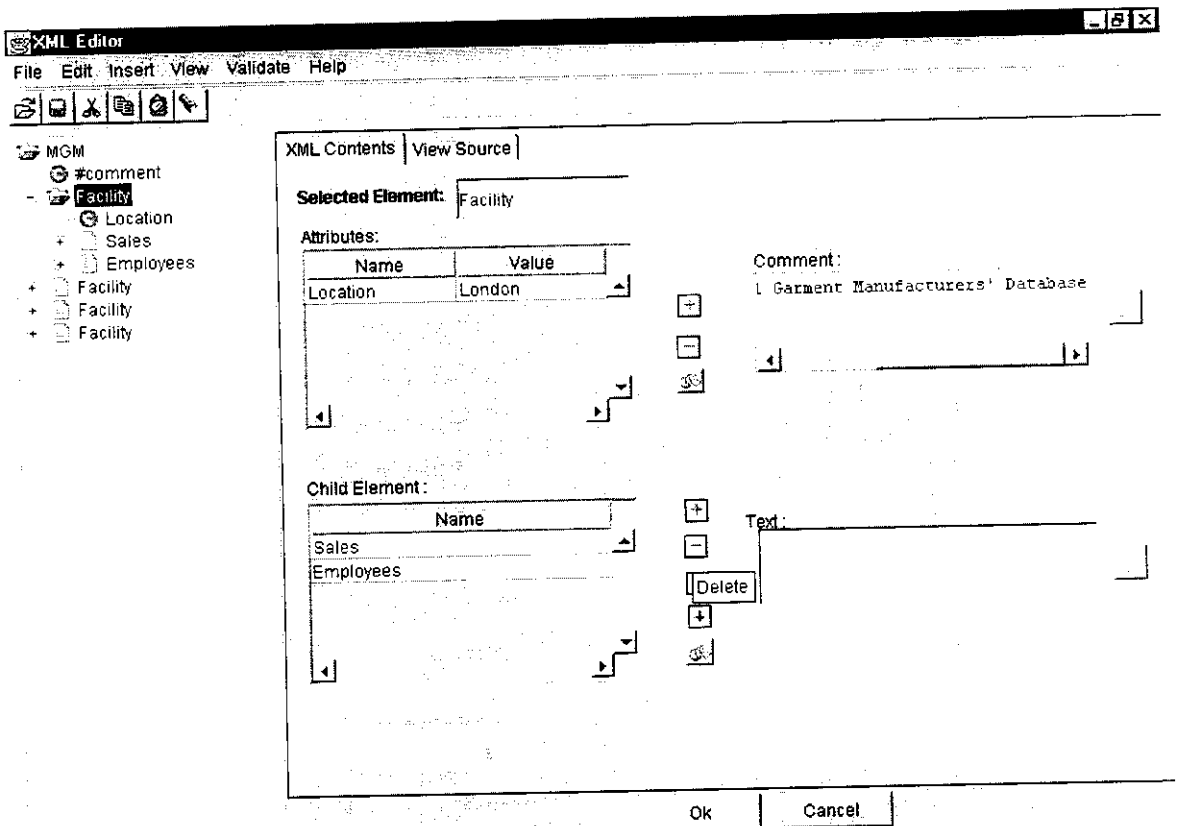


Screen Shots. B. 8

The added Node Gets reflected in tree

Screen Description:

As any application contains a deletion of data here we could see how easy it is to remove a node in our terms a XML data tags by just clicking on the remove icon. This would get reflected on the tree and the XML coding also, So the user can see immediately see how the XML file would look.

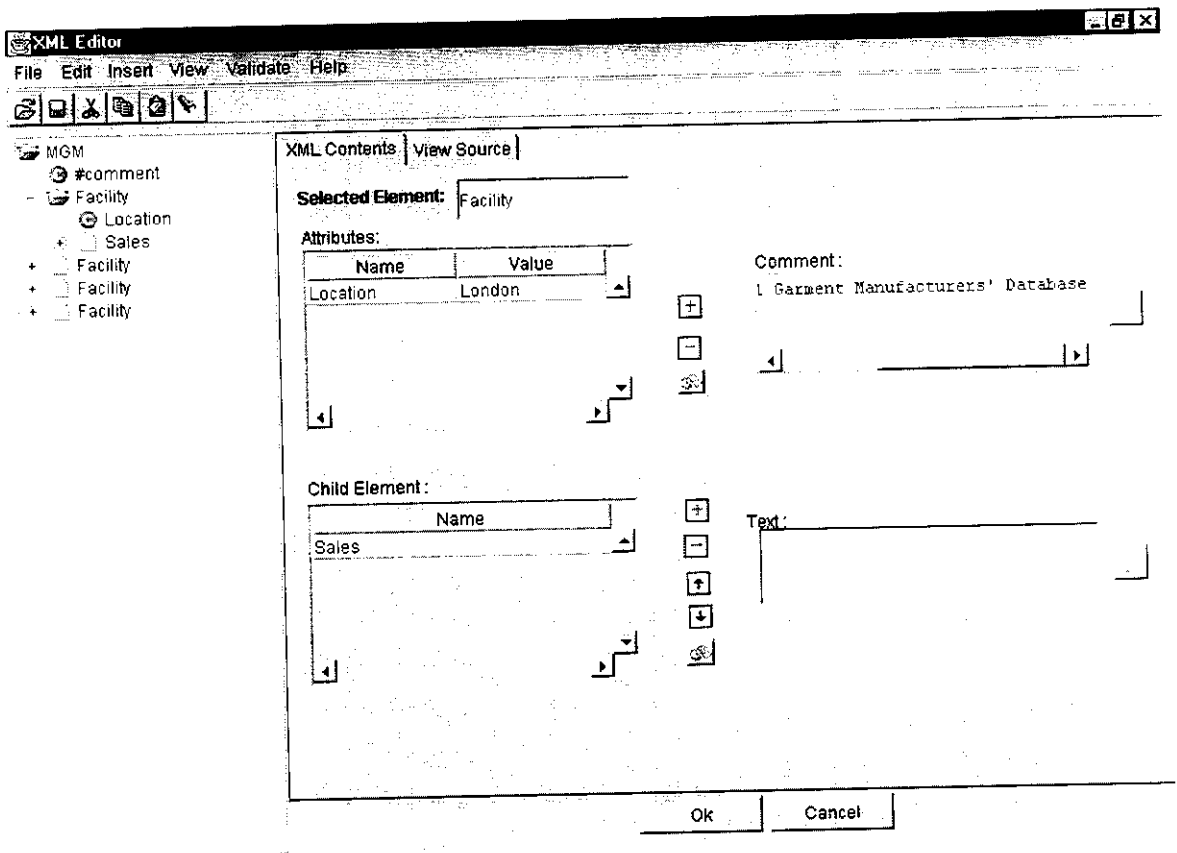


Screen Shots. B. 9

Delete the selected node

Screen Description:

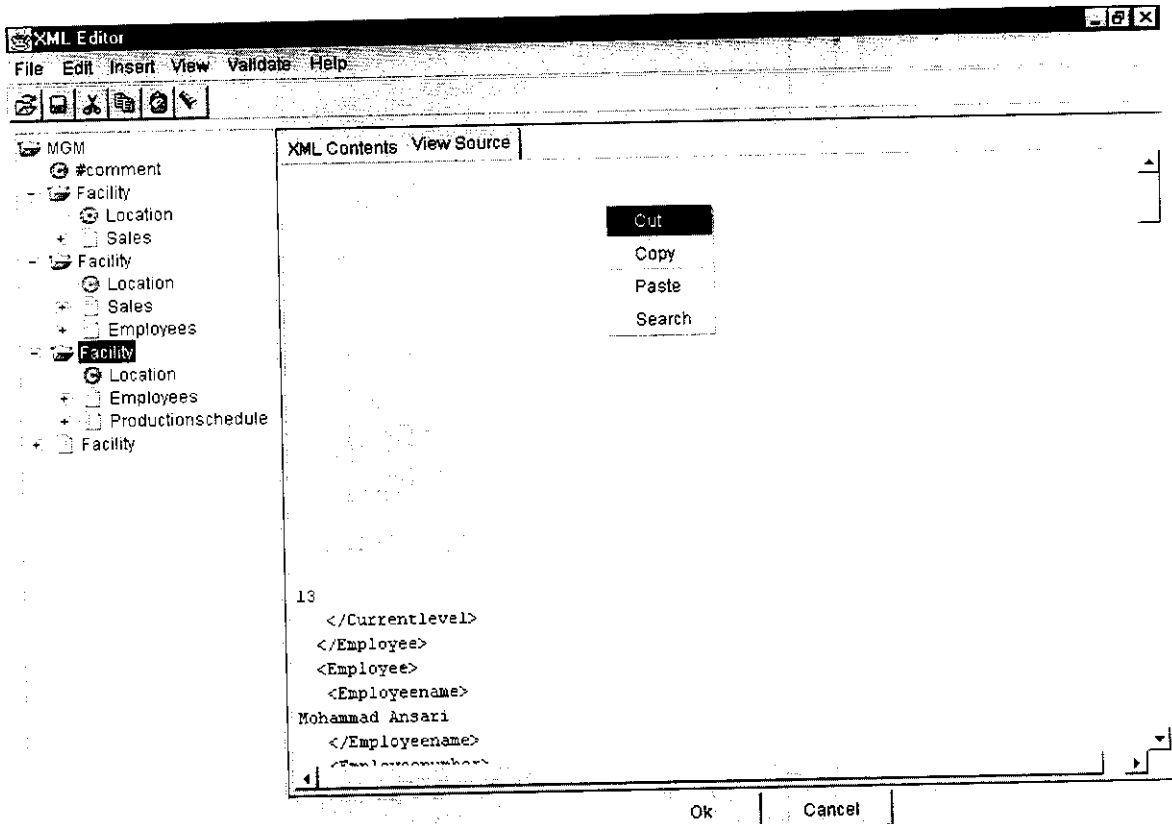
Now we could see the employee element is removed and it's not present in the tree and that shows how easy it is to work with XML files than manually removing it.

**Screen Shots. B. 10**

The Node is deleted from tree

Screen Description:

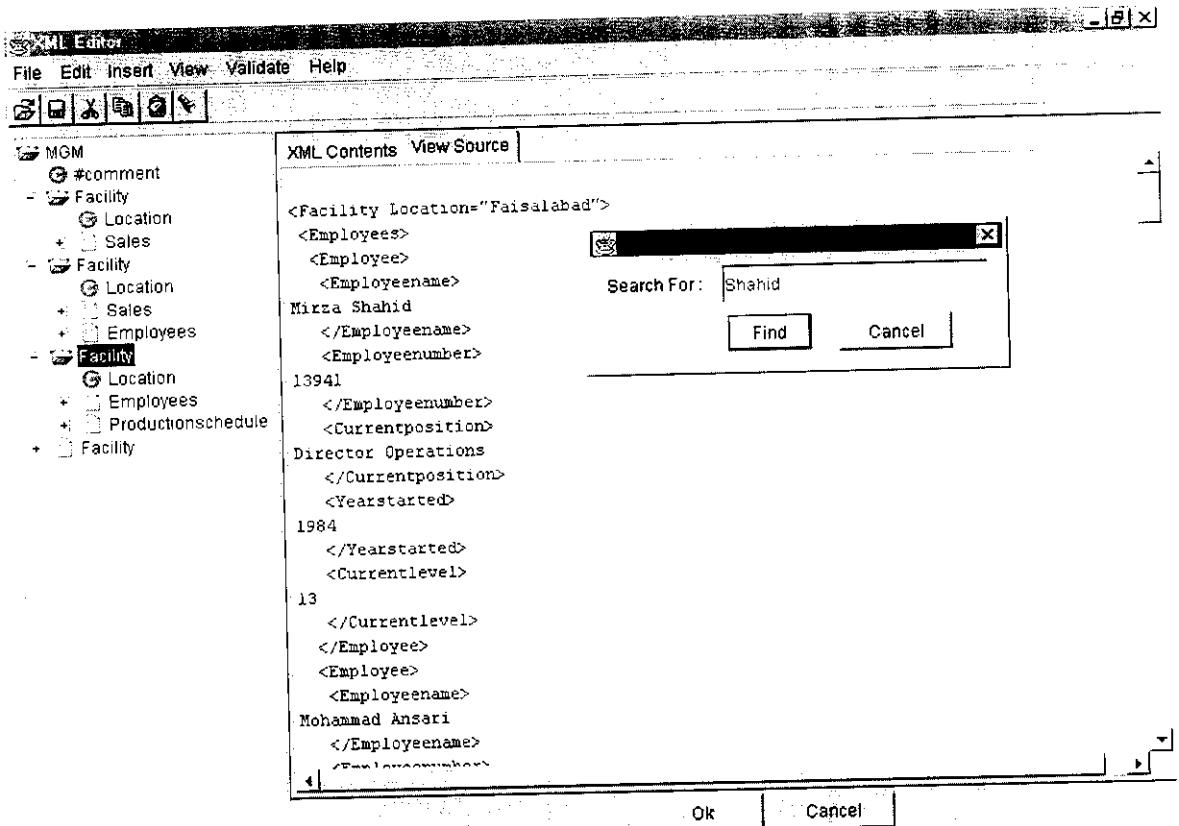
As XML file is coded by this tool, if the user wishes to use this coding somewhere else or use it repetitively the usual Cut, Copy, Paste is provided for user convenience.

**Screen Shots. B. 11**

The view source menu option

Screen Description:

The users would always like to look into data if they don't know where it is, so a search operation is given to search for any data in the XML file. This search is a fast and easy way to look into data.

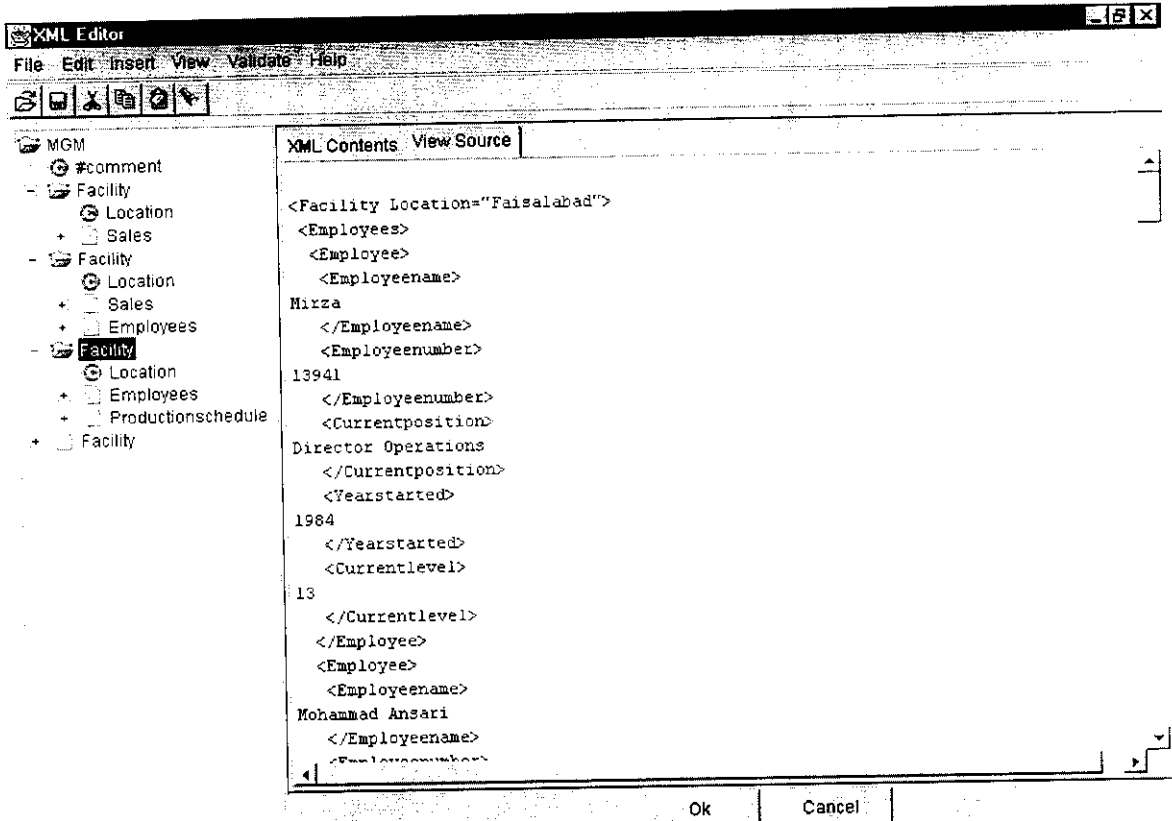


Screen Shots. B.12

The search dialog for View source

Screen Description:

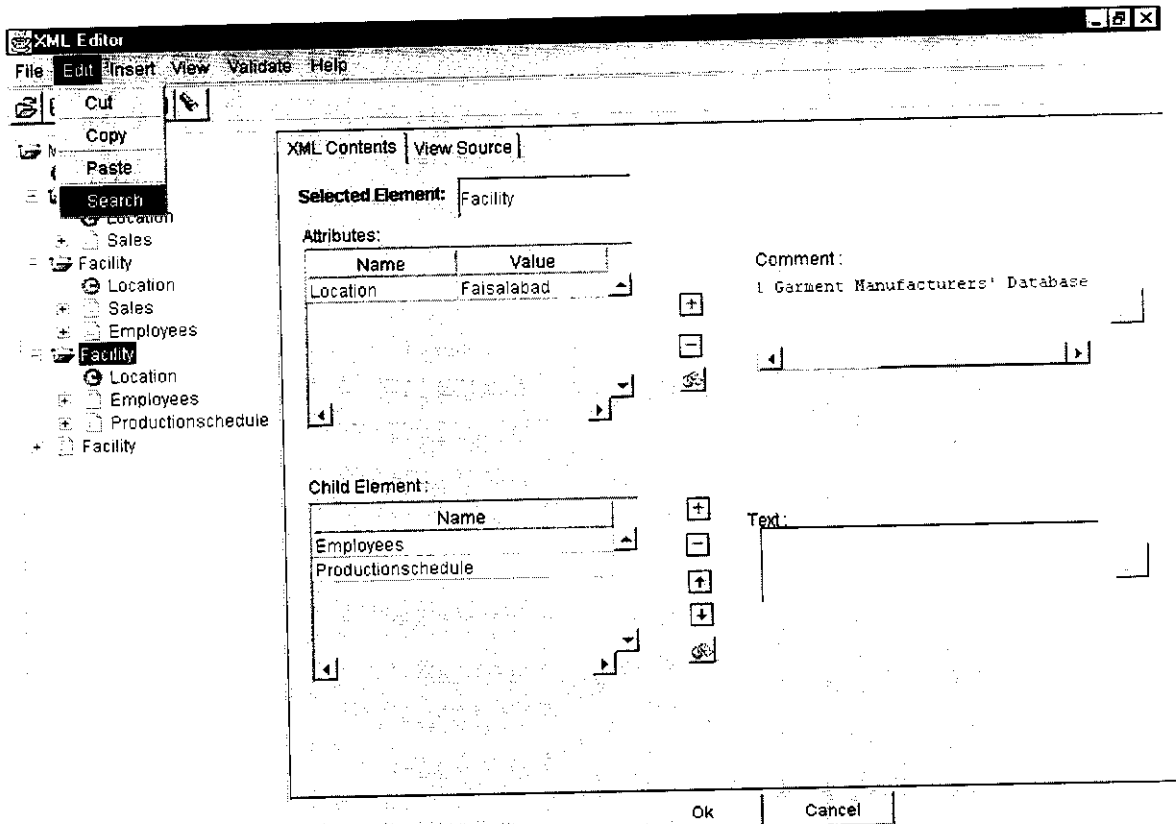
We could see the searched text being highlighted in the view source area.

**Screen Shots. B. 13**

Search text is highlighted

Screen Description:

If the user wishes to know where the data is located in the Tree, then this powerful search program will look into whatever type of XML data like text, comment, attribute value etc. Then it can be searched for the suitable nodes mating the user criteria will be shown by expanding the tree and highlighting the node.

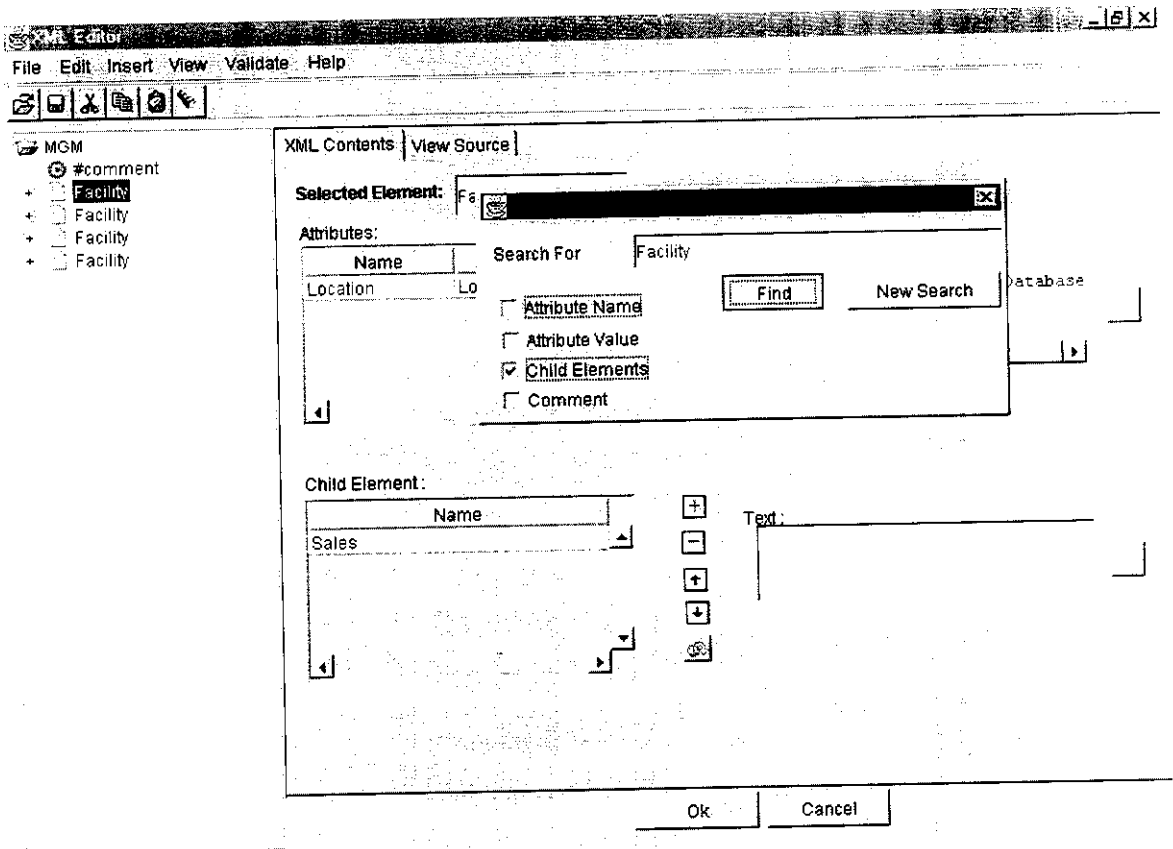


Screen Shots. B. 14

The search menu for XML Editor

Screen Description:

Here we see the user giving the criteria as to search for node that has “Faculty” and should only be of type Child Element. So precise results can be got from this powerful search tool.

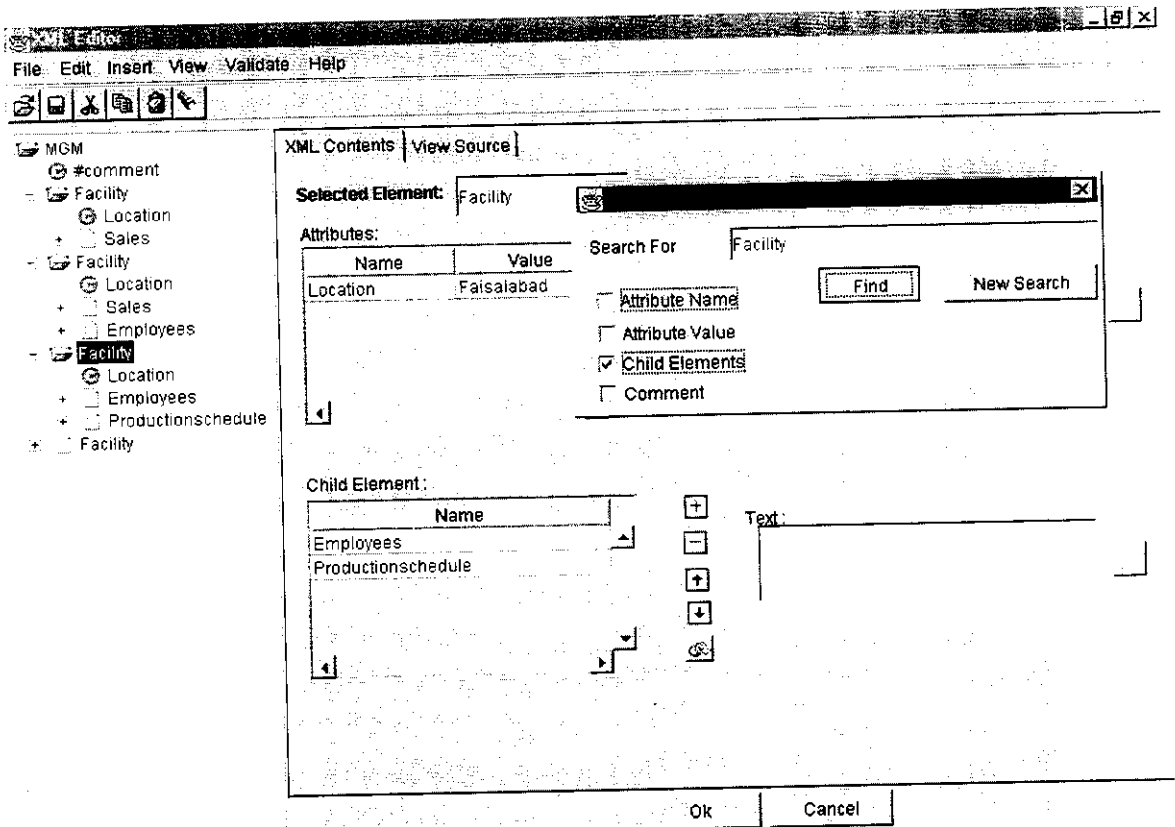


Screen Shots. B. 15

The search dialog for XML Editor

Screen Description:

As mentioned earlier the 3 matching node of they faculty is being expanded in the tree and the node being highlighted.

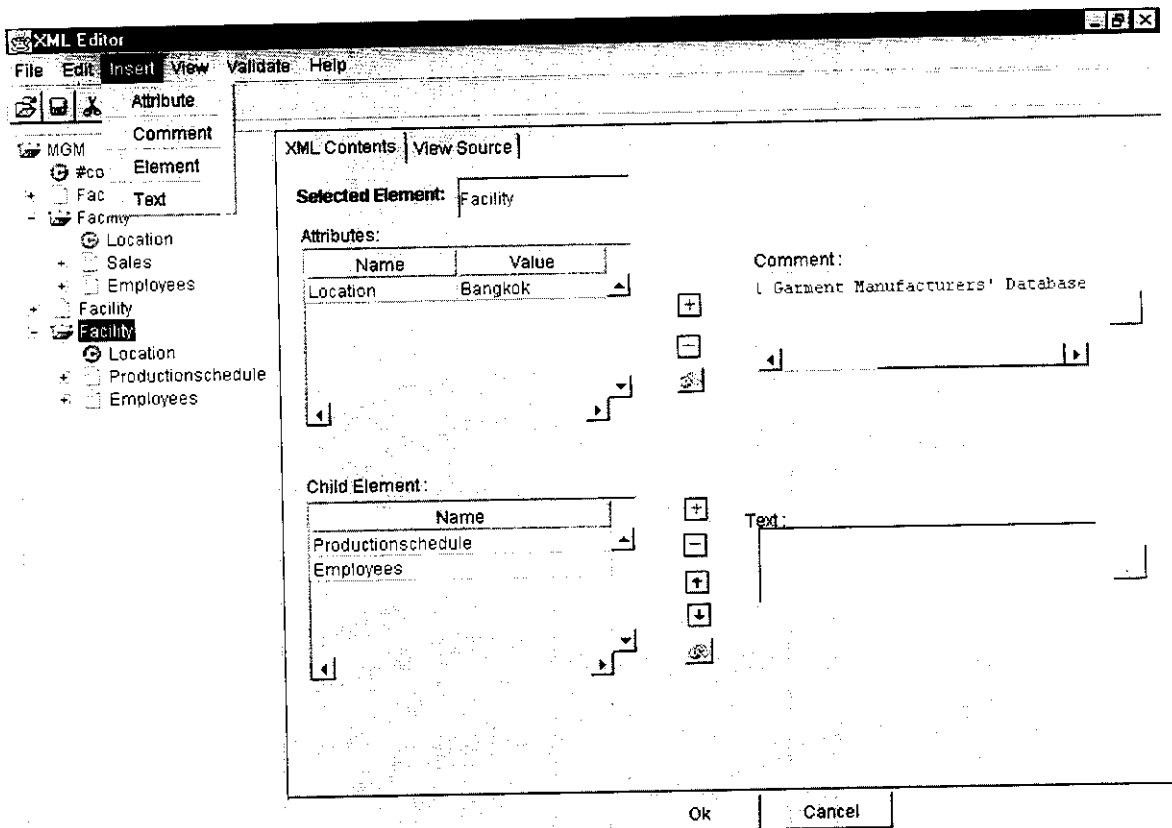


Screen Shots. B. 16

Search result reflecting in tree

Screen Description:

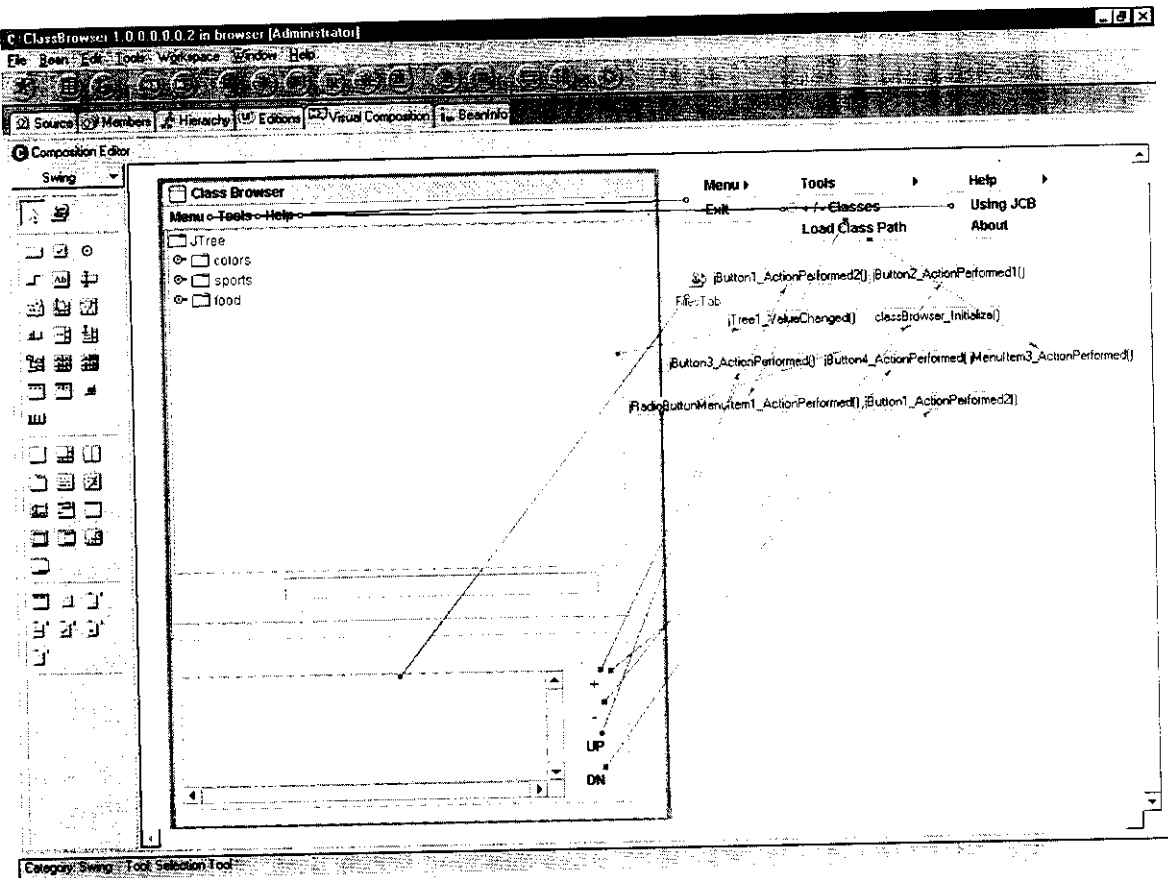
The very idea of this application is to insert data easily with less work. So this screen shows how a user could add just of any kind of XML node by clicking on the insert in the menu. Clicking on the tree could do this and the node things can be inserted.

**Screen Shots. B. 17**

Insert menu for XML Editor

Screen Description:

Working environment of Java class browser.

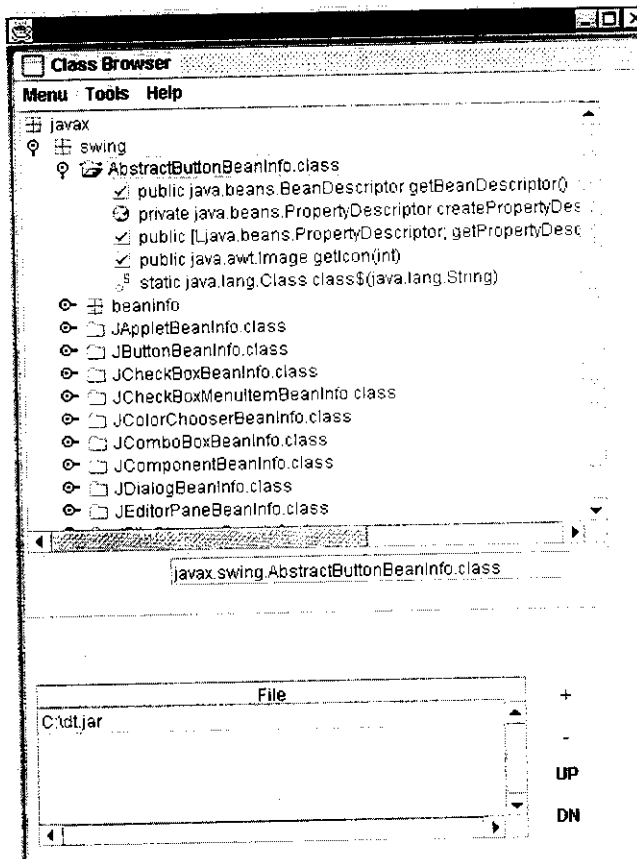


Screen Shots. B. 18

Working Environment for Java Class Browser

Screen Description:

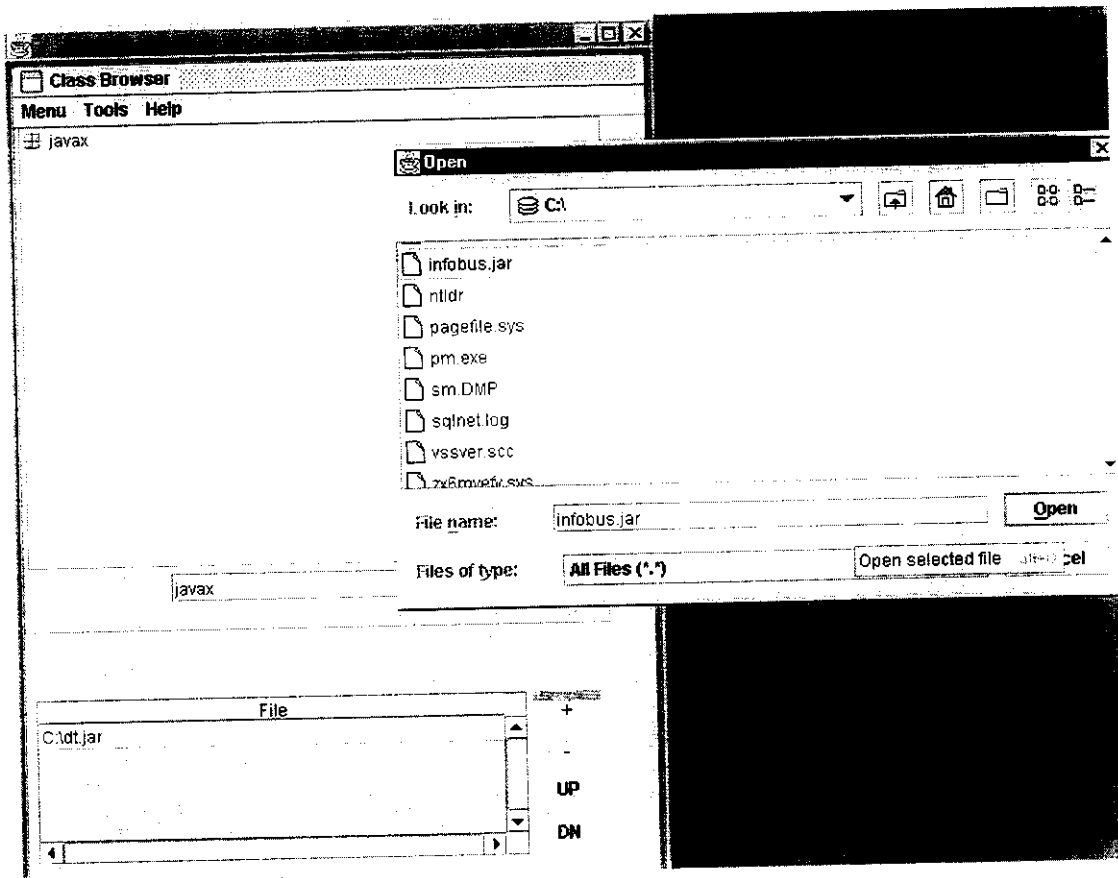
The Java class browser could load the jars and classes in the System class path. Its understood that the environment variables set in the working of the project these jar files are used, so on just a click of a button the entire list could be got with the methods available in it.

**Screen Shots. B. 19**

The Java Class Browser with all the class files

Screen Description:

This application gives a flexibility to add jars manually. So we get an open dialog box and we could select the file to be open.

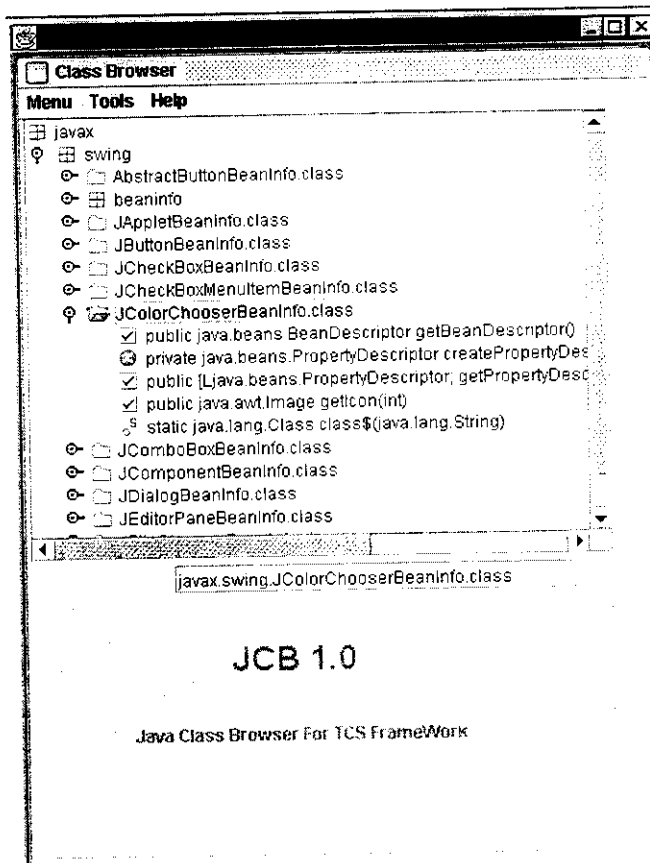


Screen Shots. B. 20

Java Class Browser with open dialog

Screen Description:

Here we see the entire work of class browser, that necessary class files open and shown with icons matching the package, class, methods, public, private static etc. So it's easy for user to identify and differentiate the things open in the screen.

**Screen Shots. B. 21**

Java Class Browser after New jar file is added

REFERENCES

REFERENCES

Books:

- The Complete Reference, Java 2, Patrick Naughton, Tata McGraw-Hill, 1999
- Java 2 Complete, Denise Santoro, BPB Publications, 1999
- Java, NIIT Press
- Software Engineering Concepts, Richard Fairley, Tata McGraw-Hill, 1999
- Software Engineering Principles, Roger Pressman
- Concepts of Java, SSI Press

Websites:

www.java.sun.com

www.caucho.com

www.w3.org

www.jguru.com

www.software.ibm.com/ad/vajava

www.javafaq.nu