AN OPEN PRODUCT LIFECYCLE MANAGEMENT SYSTEM

by
Muhammad Anwary
B.Sc. UET,
Lahore, Pakistan
2005

A thesis
presented to

Ryerson University

in partial fulfillment of the
requirements for the degree of

Master of Applied Science
in the Program of
Mechanical Engineering

Toronto, Ontario, Canada, 2005

Author's Declaration

I hereby declare that I am the sole author of this thesis or dissertation.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

Muhammad Anwary

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Muhammad Anwary

Abstract

Muhammad Anwary, MASc. Mechanical Engineering, Ryerson University 2005

This thesis presents a complete set of user requirements and high-level architecture for product lifecycle management (PLM) system for small and medium-sized enterprises (SMEs). Engineering activities such as engineering change management (ECM) and product data management (PDM) are emphasized. The system is designed to be developed in open source environment. Therefore the system is called Open Product Lifecycle Management (OPLM) system.

The thesis begins with a presentation of the motivation for the work and description of products and literature in the areas of PLM, SME and open source. An industry survey is conducted to elicit requirements of OPLM. Engineering change management (ECM) process is described and a modified framework for ECM in OPLM is presented. The proposed model is expected to make ECM faster, reusable and accurate.

Four OPLM subsystems, namely, product data management, engineering change management, process management and business intelligence are defined. For each of the subsystems, subsystem components are identified and defined.

Acknowledgement

I would like to thank Professor Filippo Salustri for being the Principal Investigator of the OPLM project, for sharing his expertise in the field of design and providing perspectives for shaping the thoughts. His guidance and comments have helped to get the work completed and improved its quality.

I would like to thank Materials and Manufacturing Ontario (MMO) and Axiomatic Product Development (APD) for funding the project, organizing workshops and providing resources for building the project portal. Special thanks goes to Ed Panziera of APD and Harvey Pellegrini of MMO for their continuous support over a two-year period of my research.

I would also like to Perennial Survey for hosting the survey free of cost. Special thanks to Dr. Saeed Zolfaghari, Dean Gerl and Tim Quinn for their help and suggestions from time to time.

Finally I would like to thank my wife Farhana Parveen for her patience and care.

Table of Contents

List of Tables

List of Figures

Chapter 6

List of Appendices

Chapter 1:

Introduction to Product Lifecycle Management

1.1 Introduction

This thesis report has been prepared as the requirement for the degree in Master's in Mechanical Engineering from the Department of Industrial and Mechanical Engineering at Ryerson University. The report is based on the project called "Open source Product Lifecycle Management (OPLM)". This project is funded by Materials and Manufacturing of Ontario (MMO) and Axiomatic Product Development (APD) Ltd. through the period from September 2003 to December 2006. Dr. Filippo Salustri is the Principal Investigator of this project. The entire project is a quite broad one that has the goal to develop a complete OPLM solution for Small and Medium Enterprises (SMEs). The report presented here is the first part of the whole project. In this report, a high-level system architecture for OPLM has been laid out. Special focus has been given to Engineering Change Management (ECM). Since the inception of this project, open source principles have been adopted to promote OPLM. APD has provided resources to the creation of an Internet portal that allows collaboration of programmers, designers and other project contributors.

The major focus of this project is to develop a complete Product Lifecycle Management (PLM) solution for SMEs that would greatly enhance their productivity. Special consideration is given to identify possible means for Open Source development of PLM.

1.1.1 Motivation

Product Lifecycle Management (PLM) is the area concerned with the practical control of all stages of a product's life cycle. This control is achieved by considering the environmental, technical, and social contexts of the product's use while it is still in development, and adapting the product as well as its supporting services (manufacturing, distribution, maintenance, reclamation, and other systems) in response.

PLM software provides bridge between management and engineering. New Product Development (NPD), time-to-market, and other long term goals of quality, sustainability, and innovation can be greatly enhanced by using PLM software. However, no existing PLM solution is targeted specifically at SMEs. Existing PLM products are typically monolithic solutions requiring substantial IT infrastructure support (powerful computers, and significant supporting software and manpower for system administration). They also *only* work with certain CAD file formats, and cost more than many SMEs can afford.

Most of the best quality software in the world today is Open-source software. It is free software developed outside the usual commercial framework by groups of volunteers (e.g. Linux, Netscape, MySQL, Apache, etc.) An open-source PLM solution would be ideal for SMEs who do not have the resources to devote to a commercial PLM system and who must support a multitude of data formats as a basic requirement of doing business. However, no such solutions currently exist.

At the same time, SMEs are driving entities of the economy of any country. They could improve their productivity and competitiveness substantially if they had access to PLM tools of the same quality as those available to larger companies with greater resources available. Any such improvements would greatly enhance the engineering capacity of this large sector.

In this research, the term 'productivity' is has been used many times. We define productivity as the amount of output created (in terms of goods produced or services rendered) produced per unit input used. Therefore, productivity can be measured in terms of capital, labor, equipment etc.

The basic business case for SMEs to voluntarily contribute to such a project is one of economy and control. The investment by any one company in an open-source project is inversely proportional to the total number of participating companies; the more companies contribute, the faster the system is developed, the faster it can become an "industry standard", and the cheaper it is for each company. Also, the companies themselves retain control over the system's development – it will perform precisely as the user community needs. SMEs act both as users and developers of the system.

1.1.2 Project Objectives

The development of the Open Product Life Management system (OPLM) is clearly a substantial and long-term project. For this thesis, only the preliminary work will has been done. Namely, to establish the feasibility of the OPLM concept, and lay the groundwork for future work.

The objectives of the thesis include the following.

1. Benchmark existing PLM solutions, and review the research literature on PLM and design methodology.
2. Review the existing literature on open-source software.
3. Develop a detailed requirements specification for the OPLM system. The benchmark exercise (objective #1) constrained by the open-source approach (objective #2) to determine what reasonable targets can be achieved by OPLM.
4. Produce architecture for OPLM, including major software components, interface descriptions, recommended tools (operating system, programming language, etc.) required to fully develop OPLM, and a detailed roadmap for how the system should be developed.

The OPLM system is expected to exhibit great interoperability among different prominent CAD and other data formats. As a proof-of-concept implementation, the OPLM system will be specifically targeted at SMEs who operate mainly in the *automotive sector.*

1.2 Brief History of PLM

The first computer-based graphic system to emerge was US Air Force's SAGE (Semi Automatic Ground Environment) that was used in mid-1950. SAGE was developed my MIT's Lincoln Laboratory. In the following ten years, Lincoln Laboratory, General Motors Research Center and few other industrial initiatives gave meaning to computer-aided design that was called DAC (Design Automated by Computer) and later the term CAD (Computer Aided Design) was used [1].

In the early 1980's, product data management system (PDM) started to emerge as computers became more functional to manufacturing industry. Early PDM systems controlled and managed product data by information authoring tools. As companies realized the need for easy, quick and reliable access to valid data, PDM systems continued to grow.

The first generation PDM systems were becoming more and more effective on the engineering operations, but it was not able to fulfil the need of the whole organization. Among other problems, it had limited data coverage, was difficult to use and available for internal use only.

The vendors improved the functionality by introducing web-enabled interfaces and wider data acceptability to PDM. During early 1990, Enterprise Resource Planning (ERP), Supply Chain Management (SCM) and CRM (Customer Relationship Management) solutions started to appear.

ERP systems are responsible for managing manufacturing, logistics, distribution, inventory, shipping, invoicing and accounting of an organization. By virtue of these services, ERP can effectively handle many business activities like sales, billing, inventory, human resources etc. Compiere and ERP5 are examples of open source ERP software. In commercial arena, SAP and Oracle are among the market leaders in ERP software and services. Some of the major ERP functions are:

1. Financial management and accounting
2. Sales and distribution
3. Materials management
4. Manufacturing
5. Production planning

SCM, often implemented with ERP, manages all related operation in supply chains. Typically, a supply chain is a one-dimensional sequence of processes. SCM usually controls procurement, manufacturing and distribution. i2 Technologies, Manugistics, SAP,, BAAN and IBM are major players in SCM market. Some major components of SCM are [2]:

- Demand planning (forecasting)
- Demand collaboration (collaborative resolution process to determine consensus forecasts)
- Order promising (When can one promise a product to a customer taking account lead times and constraints)
- Strategic network optimization (what plants should serve what markets for what products) (monthly - yearly)
- Production and distribution planning (Coordinate the actual production and distribution plans for a whole enterprise) (daily)
- Production scheduling (For a single location create a feasible production schedule) (minute by minute)
- Transportation planning (For multiple supply, manufacturer, distributor and warehousing points in a network)
- Transportation execution (Enactment of long term plans on a per shipment basis, typically performed by focused organizations called forwarders)

- Tracking and Measuring (An ever increasing aspect of supply chain management designed to highlight potential against the plan and possible process improvements)

CRM systems are focused on customers of an organization. It automates business processes that are directly involved with customers, provides tools for storing, analyzing and reporting customer behavior and ensures that customers can access all related information at any time by means of telephone, fax, email, web etc. OpenCRX, SugarCRM and OhioedgeCRM are few examples of open source CRM software. SAP, Oracle, Microsoft and NetSuite are major players in the commercial market of CRM.

ERP, SCM and CRM systems provided enterprise-wide functionality in terms of integration of data as well as external and extended-enterprise. By the mid-1990's, a visible gap was seen between PDM and ERP/SCM systems. Some of the major problems with these systems were:

1. Each of these systems represents a segment of an organization. In other words, these systems are not fully integrated. Lack of integration often creates inappropriate decisions, high infrastructure and maintenance cost and discrepancy among information.
2. A company may end up with these solutions from three different vendors and none of these systems may talk to each other.
3. Each system is big in term of investment, infrastructure and maintenance. Companies may need only a part of each solution, not the whole. A recent survey shows that 73% companies who own ERP system, do not use it in its full capacity, 40% use it ineffectively and 33% companies has initial setup problems [3].

Late in 1999, a Boston-based research company called AMR Research first coined the term Product Lifecycle Management (PLM) that would eventually provide a complete solution for the companies that was long needed. Since its inception, PLM was designed to integrate all the information of an organization so that a decision-maker has a real-time view of the state of the organization.

1.2.1 Definition of PLM

After a search in the Internet, the following definitions of PLM are gathered from industry's major vendors, consultants and researchers:

1. PLM is a strategic business approach that applies a consistent set of business technology solutions to support collaborative creation, management, dissemination, and use of product definition information across the extended enterprise, from concept to end of life – integrating people, processes, and information. *Tony Affuso, CEO, UGS* [4].

2. PLM is the combination of systems for Create, Collaborate and Control. *William Berutti, Senior VP of Business Development at PTC* [5].

3. PLM includes the definition of the complete product and manufacturing processes, from mechanical and electronic components to software and documentation. The functions are typically independent of whether the data relate to design, prototypes, fabrication, operations or maintenance. *Alan Christman Vice President CIMdata, Inc.* [6].

4. PLM is an extended enterprise solution for manufacturing companies that facilitates the sharing of product data across all company functions, customers and suppliers. *IBM* [7]

5. PLM is a strategic business approach for the effective management and use of corporate intellectual capital. *Datamation Limited* [8].

6. An enterprise initiative that allows you to digitally transform your entire product lifecycle and maximize the business value of your product portfolio. : *EDS* [9].

7. Product Lifecycle Management is the business activity of managing a company's products all the way across their lifecycles in the most effective way. *John Stark* [10].

8. PLM is the integration of business systems to manage a product's life cycle. *University Of Michigan PLM Development Consortium, Arc Advisory Group* [11].

9. PLM includes data authoring, data leveraging, process management, and then lifecycle management—together with tools to retain the knowledge captured for use in the next product cycle. *Bruce Jenkins, Daratech* [12].

1.2.2 Working Definition of PLM

A closer look to the above definitions reveals that PLM has been defined from many defined perspectives. Some of the vendors see PLM as the savior of intellectual capital, other see it as a strategic business approach. Each of these definitions only tells a part of PLM.

PLM needs to be defined properly in order to provide a single and complete viewpoint. John Stark argues that "if you consider that many viewpoints of people who work in the PLM space then contradictions abound. Most obviously, some see PLM mainly in terms of its lifecycle aspects, while for others the issues are those of collaborative product development". The demand for defining PLM is overwhelming [13].

The Internet and publications are researched to collect the viewpoints of PLM among industries and researchers. Common elements of PLM functionality are recognized and a definition of PLM is put together. It is called 'Working Definition of PLM' because this definition may be changed, revised or modified as developers and stakeholders work together in the open source space. The working definition of PLM is as follows:

"PLM is the monitoring, controlling, reporting, and improving all aspects of a product and the activities relating to a product from the identification of the need that the product will address, to the treatment of the product at its end of life. It is meant to integrate and harmonise the many different kinds of information used during product development and engineering."

This definition is designed to give a complete sense of the functionality of PLM. The concept of product lifecycle, business strategic approach and management components is integrated in this definition. This definition is called 'working definition' because this can be changed or modified over time as research continues. However, in this project, the major concern is 'Engineering Side' of PLM, rather than the business side. Detailing business processes, strategy and models are beyond the scope of our current work.

1.3 PLM Current industry perspective:

This section will discuss about the current state of PLM business in the industry. This is important to show that significant investment has been made for PLM and revenues since 2001 has grown rapidly. Despite global depressed economies, PLM market continues to grow at a solid pace Figure 1. In 2003 the total market

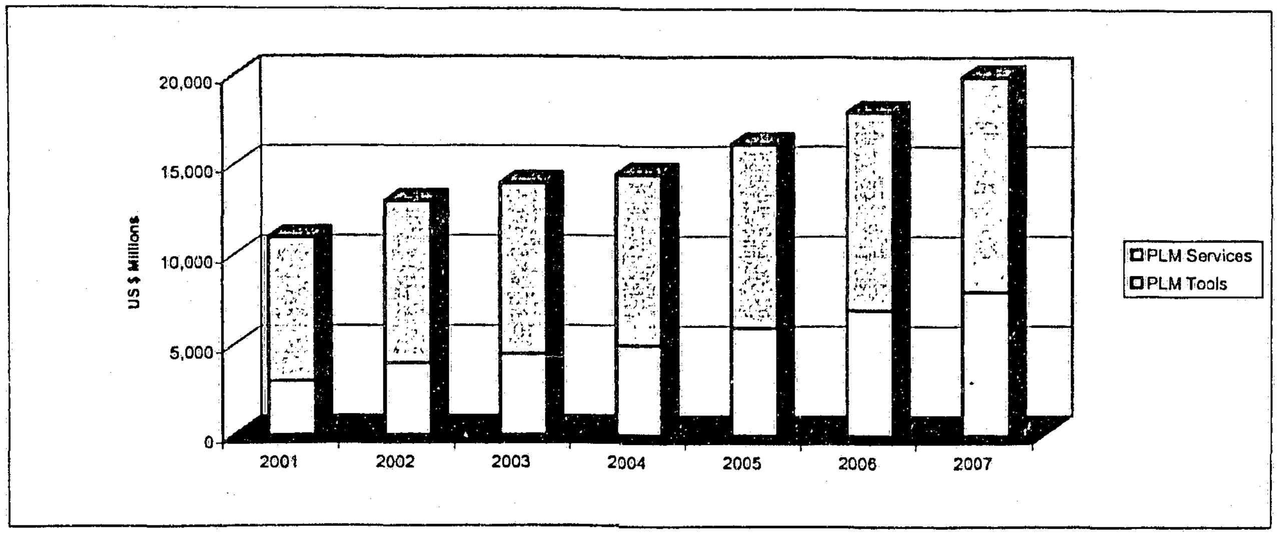


Figure 1: PLM revenues in million US dollars

size of PLM was US$ 8.2 billion. It also experienced a growth of about 5% in 2004. Researchers expect PLM market would grow at an annual rate of 8%

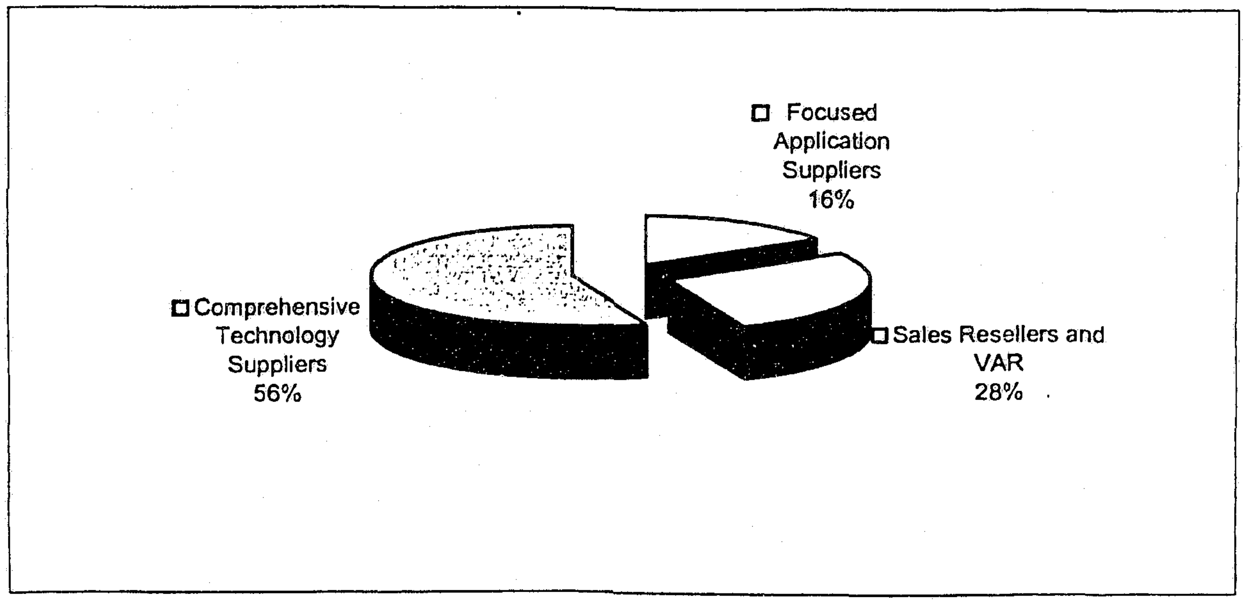


Figure 2: Market segments of PLM solutions

through 2008. In PLM market, comprehensive technology suppliers own 56% of the market Figure 2. Table 1 shows the sales and revenues of leading PLM vendors.

	Company	Retail Sales	Direct Sales	Developer Revenues
1	Dassault/IBM	$2050	$50	$950

2	UGS	$1150	$700	$900
3	PTC	$911	$550	$650
4	SAP PLM	$350	$250	$265
5	Agile	$200	$100	$150
6	MatrixOne	$225	$100	$175

Table 1: PLM software sales and services in year 2004 for leading companies in Million Dollars [14]

PLM has enjoyed a very healthy growth since 2001 when it actually made it presence known to organizations. The promise was clear and within a year or so, researchers were able to point out the immediate benefits of PLM. Larger companies kept on buying and implementing PLM and continues to do so. It is obvious that SMEs could not enjoy the benefits of PLM. If the strength of PLM is coupled with SMEs and bigger enterprises together, the overall market will experience a much higher growth and profitability.

1.4 Background of PLM software:

In order learn more about PLM, existing PLM solutions in the market are studied. There are more than 100 vendors in information technology sector that provide PLM or PLM-related services. Among them a few vendors are selected for review that are market leaders in PLM [15]. A complete discussion of the solutions these companies provide is beyond the scope of this report because each company has many different software suites that make the PLM complete. No research paper has been found at IEEE Xplore that gives a comprehensive outlook of the premium vendors in PLM market. Table 2 shows a short-list of PLM vendors:

No	Vendor Name	Product Name	CAD Package
1	Parametric Technology Corporation (PTC)	Windchill	Pro-Engineer
2	IBM Dassault Systems	Smarteam	CATIA
3	SAP	MySAP PLM	No proprietary CAD package
4	UGS (Unigraphics Solutions Inc.)	Teamcenter	Solid Edge
5	MatrixOne	Matrix10	No proprietary CAD package
6	Agile Corporation	Agile PLM	No proprietary CAD package
7	Arena Solutions	Arena PLM	No Proprietary CAD package

Table 2: List of Major PLM vendors and their products

1.4.1 Windchill

Windchill is the PLM product for Parametric Technology Corporation (PTC). PTC is known in the industry for its CAD package Pro-Engineer. The main features for Windchill are:

- Internet based, requires a web browser and includes embedded 3D visualization of product information
- Interoperates with heterogeneous CAD data
- Real time collaboration
- Reduces information redundancy

Windchill offers the PLM solution that consist of seven packages, here is the description for them:

Name of the package	Major Functions and features
ProjectLink	Collaborative project management on online workspace; CAD integration, visualization and project templates
PDMLink	Controls data and their versions such as CAD files, specifications, analysis results, service records etc.; Visualization
PartsLink	Web-based parts library to help reuse parts and components
DynamicDesignLink	Modify and access Pro-Engineer drawings in real-time
SupplyLink	Connect suppliers, contract manufactures and other external organizations online to facilitate design and production across geographies
Integrations	Control and share product information internally and externally; Communicate between Windchill and CAD tools and other enterprise systems
ProductView	View of a product in its earlier stage for the development team; 2D and 3D interactive visual collaboration and prototyping

Table 3: Windchill Software packages and their major functions

Summary

PTC's Windchill provides a complete spectrum of support for manufacturing companies. Windchill is one of the most popular PLM solutions among the big manufacturing companies. The system provides a comprehensive package of solutions for product development, support and business processes. Due to high costs of implementation, support and infrastructure, Windchill is not suitable for SMEs.

1.4.2 Smarteam

IBM Dassualt provides PLM in a three-pack suite: CATIA, ENOVIA and SMARTEAM

Name of the package	Major Functions and features
CATIA	CAD CAM CAE drawing and visualization tool
ENOVIA	Management and distribution of data/information 3D based collaboration Supports industry-wide applications Scalability management
SMARTEAM	Control, revision management of all types of data Full drawing and hierarchy support Pre-defines business templates Workflow management, process automation

Table 4: Smarteam Software packages and their major functions

Smarteam is IBM/Dassault System's solution. The same company has CATIA, which is a popular CAD modeling tool in automotive and aerospace industry. Smarteam specializes in robust data management, workflow management and process automation. Like Windchill, Smarteam is a costly software that requires significant investment on technology and infrastructure. However, they are offering a customized and web-based service that integrates CATIA and Smarteam. This customized service is designed for SMEs.

1.4.3 mySAP

mySAP is the PLM package for SAP (Systemanalyse und Programmentwicklung "Systems Analysis and Program Development") AG. Main functional features include:

- Data management includes version control, BOM, engineering change management, GIS, SCADA, fleet management, serial number, maintenance etc.
- Project management includes scheduling, budget, cost, revenue, portfolio etc.
- Collaboration includes document handling, large file handling, templates, etc.
- Quality management includes SPC (statistical process control), LIMS (laboratory information management system), manuals and specifications, compliance etc.
- Environment, health, safety and emission management, report generation, MSDS etc.

Summary

SAP is the most popular software vendor in the field of enterprise resource planning (ERP), supply chain management (SCM) and customer relations management (CRM). The PLM solution for SAP is more popular in process manufacturing companies. SAP has integrated a wide variety of applications and tools that other PLM vendors do not offer. Large companies always benefit from enterprise-wide integrations. Again, SAP has limitations for SMEs in terms of size of PLM, cost, infrastructure, resources etc.

1.4.4 Teamcenter

UGS (Unigraphics Solutions Inc.), owned by EDS (Electronic Data Systems) provides PLM service using four tools: NX, e-factory, Solid Edge and Teamcenter.

NX is a portfolio CAD/CAM/CAE tools for product development, solid edge is 3D CAD software, e-factory is a collaboration tool for design/assembly etc. Teamcenter has 8 packages:

Product	Main Functions and Features
Teamcenter Integrator	Gather product information from legacy systems from outside the organization like SRM, SCM, CRM etc.
Teamcenter Requirements	Enable value-chain participants to gather product requirements and use it for product development
Teamcenter Enterprise	Manage product/engineering/process data as a product is developed
Teamcenter Engineering	Multi CAD collaboration and version control
Teamcenter Manufacturing	Synchronize BOM to internal and external use
Teamcenter Project	Collaborative project management
Teamcenter Community	Team collaboration and virtual meetings
Teamcenter Visualization	3D data creation, sharing and management

Table 5: Teamcenter Software packages and their major functions

Main functional features of Teamcenter

- Web-based and portable devices interface
- Data management includes version control, and all organization data
- Supports J2EE (Java 2 Enterprise Edition), .net, SOAP (simple object access protocol), XML (extensible markup language), UDDI (universal description, discovery and integration) etc. open standards

Summary

Teamcenter PLM focuses on full production solution for discrete manufacturing companies. This software uses open standards like J2EE, XML, SOAP etc. This facilitates interoperability between different vendors and programs. Teamcenter has multiple modules to cover concept design, process management, knowledge management, collaboration, systems engineering, manufacturing etc. This is inherently a big package and suitable for large companies.

1.4.5 Matrix10

Matrix10 is Matix One's PLM solution. It includes four packages:

Product	Main Functions and Features
Matrix10 Environment	Share organization data centrally
PLM Modeling Studio	Make process modeling, configure UI
Collaboration suite	Matrix One Teamcentral: information searching, Document Central: manages and shares documents, Program Central: create, share and manage projects globally
Lifecycle applications	Product Central (product information), Specification central, Engineering Central, Sourcing Central and Supplier Central

Table 6: Matrix10 Software packages and their major functions

Summary

Major benefit of Matrix10 PLM platform is 'dynamic modeling studio' that provides flexibility to easily configure business processes and user interfaces supporting large-scale databases. This package complies with open standards like J2EE and XML.

1.4.6 Agile PLM

AGILE PLM Platform include

- Analytics framework: decision support for discrete manufacturing companies
- UI framework: user experience
- Integration framework: system to system integration
- Common services: core infrastructure services

Here is the software that is included with PLM service:

- Product collaboration and product data management: 5-pack suite, product definition, change collaboration, manufacturer management, multiple organization, product interchange

- Engineering collaboration: 3-pack suite, engineering desktop, design collaboration and design analytics
- Product portfolio management: optimize development investment etc.
- Product cost management: 4-pack suite, product sourcing, contract management, analytics framework, product interchange
- Product service and improvement
- Product governance and compliance: managing industry standards and regulations
- Product requirements management
- Product configuration management
- Maintenance, repair and overhaul
- Product catalogue

Summary

Agile PLM provides services by integrating ERP, SCM, CRM and other enterprise-wide applications and it is called 'enterprise product record'. AgilePLM uses multiple modules for different business areas. Engineering changes are well accounted for in this packages. It serves a wide range of industries and leading manufacturers of aerospace, automotive and electronics.

1.4.7 Arena PLM

Arena PLM provides a centralized environment for managing product data, to collaborate in real-time around a single set of information on Internet. This solution is focused on manufacturing companies.

Arena PLM Features

- Item Management: Creates and maintains one central repository for complete item specifications. Also supports revision control and change history.
- BOM Management: Manages different kinds of bill of materials (BOMs).
- File Management: Manage any type of file in the database.

- Supplier Management: Track supplier relationships.
- Manufacturer Management: Keeps track of full specifications of the manufacturer's component, including part numbers, part names and custom attributes.
- Vendor Management: Capture key information on costing, specifications and lead-time.
- Cost Management: Capture and analyze material cost early on in the development cycle.
- Advanced File Management: Comprehensive document management system with full vaulting capabilities.
- Change Management: Formalize the approval process and increase collaboration efficiencies with configurable workflows.
- Supply Collaboration. Share BOM items with suppliers regardless of source relationships.
- Reporting. Create instant reports on items, costing, purchasing and change orders to track product development and new product introduction processes.

Summary

Arena PLM is one of first few PLM providers who started on-demand PLM service using Internet. This strategy was taken to attract SMEs who could not otherwise get PLM benefits from large vendors like SAP, UGS etc. Arena PLM has focused on SMEs by building a PLM that is smaller in size and delivers generic engineering services. For example, Arena PLM offers PDM, ECM, supply collaboration, BOM management etc services in one program. Bigger companies are now developing similar PLM packages to attract SMEs.

1.4.8 Discussion

Investigation on major PLM products is conducted to elicit functionalities of PLM. It has been found that vendors are more concerned about building a robust database management system that can handle CAD/CAM data, manage BOM's, workflows and sometimes, processes. Some of the bigger solutions like mySAP PLM has product portfolio management functions that enables decision-makers make informed decisions based on the product's performance reports. Interoperability among CAD and other vendor-specific applications are getting more commonplace. These findings are used for creating the requirements for OPLM (Chapter 4).

1.5 Open Source Software

1.5.1 Introduction

OPLM project is expected to be developed in open source. Open source principles, attributes, licensing etc will be discussed in the following sections. At the end of the discussion, a model for open source development has been proposed.

Open source is a means of developing and distributing software that ensures software is available for use, modification, and redistribution by anyone. Open source software can generally be downloaded for little or no monetary cost and may be used, shared, borrowed, or changed without restriction.

According to Open Source Initiative (OSI), "open source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code. To be OSI certified, the software must be distributed under a license that guarantees the right to read, redistribute, modify, and use the software freely" [16].

A piece of software is called open source when many developers contribute to the development of a given system and the source code is available for free distribution. Table 7 shows a list of prominent open source software. There are few general attributes of open source software [16]:

- Free Redistribution: Open source license do not restrict distribution, usage or selling the software. Moreover, the software can be used as a component of a software that is an aggregate containing programs from other open or non-open source entity. One cannot charge any royalty or any other fee for selling open source software.
- Source Code: Open source software is distributed with the source code or at the least, repository of the source code is advertised that can be obtained without charge. One can modify or add to the source code provided the source code obtained from open source is referred in proper manner.
- Derived Works: Open source also allows modifications and derived works.
- Integrity of The Author's Source Code: The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
- No Discrimination Against Persons or Groups: The license must not discriminate against any person or group of persons.
- No Discrimination Against Fields of Endeavor: The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
- Distribution of License: The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- License Must Not Be Specific to a Product: The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution

and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

- License Must Not Restrict Other Software: The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
- License Must Be Technology-Neutral: No provision of the license may be predicated on any individual technology or style of interface.

	Category	Software	Website	Core Team Members	No. of Contributors
1	CAD	BRL-CAD	http://sourceforge.net/projects/brlcad/	13	
2	CAM	IMPACT	http://impact.sourceforge.net/	11	
3	Mathematics	GAP		47	
4	Scientific data analysis	Gwyddion	http://gwyddion.net	4	
5	Database management System	MySQL	http://www.mysql.com/		More than 100
6	Document editing	OpenOffice	http://www.openoffice.org/		500
7	Groupware	PHProjek	http://www.phprojekt.com/	Not available	
8	Operating System	Linux	http://www.linux.org/	Not available	
9	Programming language	GCC	http://gcc.gnu.org/		More than 368
		Perl	http://www.perl.org/	Not available	
		PHP	http://www.php.net/	Not available	
		Python	http://www.python.org/		64
		Ruby	http://www.ruby-lang.org/en/	Not available	
10	Web browser	Mozilla	http://www.mozilla.org/foundation/	Not available	
11	Web server	Apache	http://apache.org/		More than 300
12	Enterprise resource	Compiere	http://www.compiere.org/		43

Table 7: Information on Major Open source projects

1.5.2 Open source software attributes:

1. Generic characterization: Open source community is largely oriented to produce applications for the same community. Among the licenses, GPL is most popular (refer to Figure 4 for licensing types). For programming language, C is the most popular one followed by C++ and Perl.

	Use free of charge	Source code modifiable	Source code must be open in derived products	Combination with proprietary elements not possible
Freeware	X			
Berkeley Software Distribution (BSD)	X	X		
Lesser General Public License (LGPL)	X	X	X	
General Public License (GPL)	X	X	X	X

Table 8: License categories in open source software [8]

		Technical Property Disclosure of the	
		Source code	Binary code
Economic attribute Distribution is	Free of charge	Open source software Examples: Linux, Apache	Freeware, public domain. Examples: Adobe Acrobat Reader, Pegasus mail
	Subject to changes	Commercial Open source software. Examples: Open-source software distributions	Shareware, Commercial software. Example: Windows, MacOS

Table 9: Cost and distribution information on open source software [17]

2. Developer community: According to the survey conducted by [18], 49% of OS projects have only one person developing an application. And 6% of the projects have more than 20 developers. However, research suggests that there is no correlation between size of project and active developers to the life-span of a project. Moreover, there is no relationship (Figure 3) between the

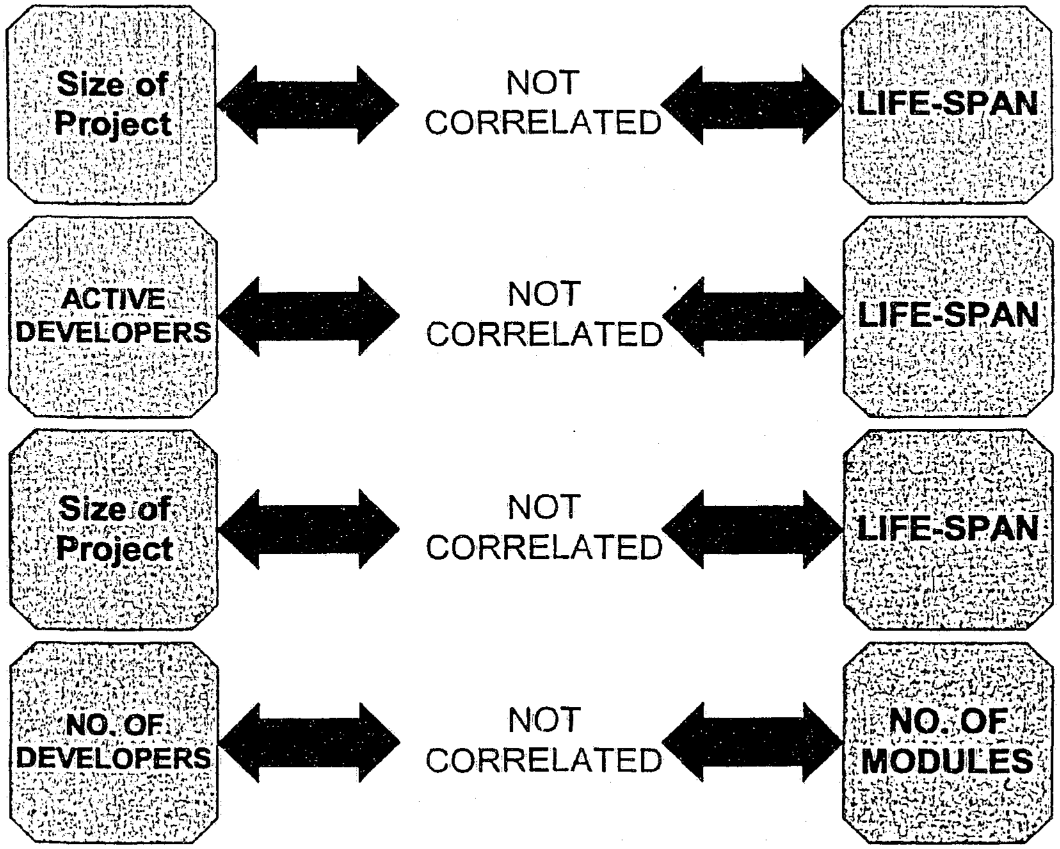


Figure 3: Correlation of attributes in open source software

number of developers to the number of modules of a project.

3. Core team: Core team is defined as a project that has ten or more developers. 16% of OS project have a core team [18]. Projects with a bigger core team are more successful and considered larger than others.
4. Evolution: Evolution of OS projects is determined as vitality, activity and popularity.

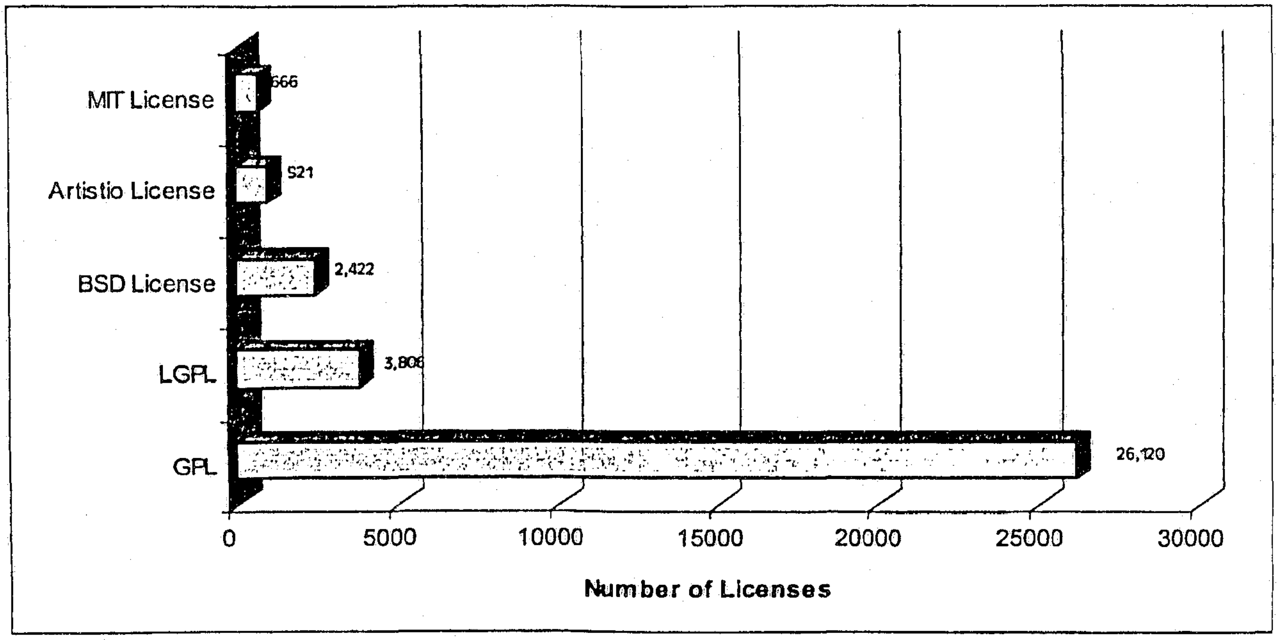


Figure 4: Number of open source licenses in different categories

1.5.3 Modified Open Source Development Model

Open source software development has evolved with time. Few innovative models of open source development have emerged to meet the needs of developers, companies as well as software users. Even companies like Microsoft who dominate proprietary software market are promoting projects that would be built in open source. Open source projects are no longer non-business decisions.

Due to the shift of trend in open source development from the traditional framework, the author has proposed a new model to develop PLM in open source (Figure 5). This model is new and theoretically workable.

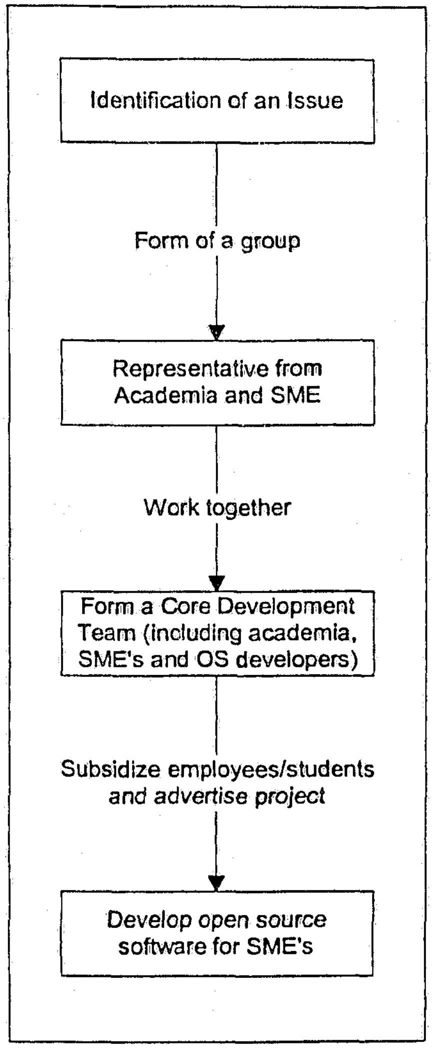

Identification of an Issue
Form of a group
Representative from Academia and SME
Work together
Form a Core Development Team (including academia, SME's and OS developers)
Subsidize employees/students and advertise project
Develop open source software for SME's

Figure 5: Proposed open source software development model for SMEs

Following are the features and advantages of this new open source model:

1. In a setting for development of software exclusively for the use of SMEs the author proposes that the software be built in joint collaboration with academia and developers from SMEs
2. The academia and representatives from SMEs join together to discuss an issue that can be resolved using open source methods.
3. The academia prepares the groundwork for the project such as requirements elicitation, systems development, prototype building etc.

4. The representative builds a pool among affected SMEs and makes a team of developers and contributors to the project that is called the core team.
5. The core team is subsidized by SMEs. The subsidy can be worked out in two ways: direct compensation for the hours worked for the project or proposal of higher incentives within the organization.
6. The core team has two sources of motivation: one for organization and the other for the recognition in the developers community and general open source motivational factors.
7. As the project grows, developers outside the core team may join to the effort and contribute significantly.

1.5.4 Limitations

The proposed open source development model has following limitations:

- The motivation for participation is not the same as traditional open source environment.
- This model is more structured and controlled as opposed to traditional open source development. Structure and control are imposed by SMEs and academia while the project is initiated and developed. Such conditions may hinder voluntary contributors and developers.

1.5.5 OPLM in open source

In recent years, enterprise solutions software has gained great popularity in open source. For example, Compiere has been a very successful open source ERP solution that is now used by small and medium companies. There are many open source groups who are collaborating to develop solutions like PDM and PLM. The survey conducted by this research (available at Appendix A) shows that 99% of the participants want to be a part of this open source initiative. Therefore, developing OPLM in open source environment is feasible.

1.6 Small and Medium Sized Enterprises

1.6.1 Introduction

OPLM is primarily designed for small and medium sized enterprises (SMEs). This section discusses about SMEs, their attributes, economic profile, sector-wise distribution of employment. Overall productivity of Canada is also discussed, and compared with that of other nations. The reason for this research is to show that a solution such as OPLM is required for SMEs for the greater benefit of the economy.

1.6.2 Definition of SME

Challenges of today's small and medium sized enterprises (SMEs) are diverse. Due to the pressure of market and competition, SMEs are required to respond to the drivers of market-based economy in very short periods. Information technology (IT) has advanced very rapidly in the past years. By integrating solutions provided by IT, an SME can maximize its profitability; ensure prolonged maintenance of good market share. However, the latest IT solutions are often very costly that requires unjustifiable amount of investment and working capital for SMEs. A sectorwise distribution of employment and relationship to export SMEs has been shown in Figure 6 and 7 respectively.

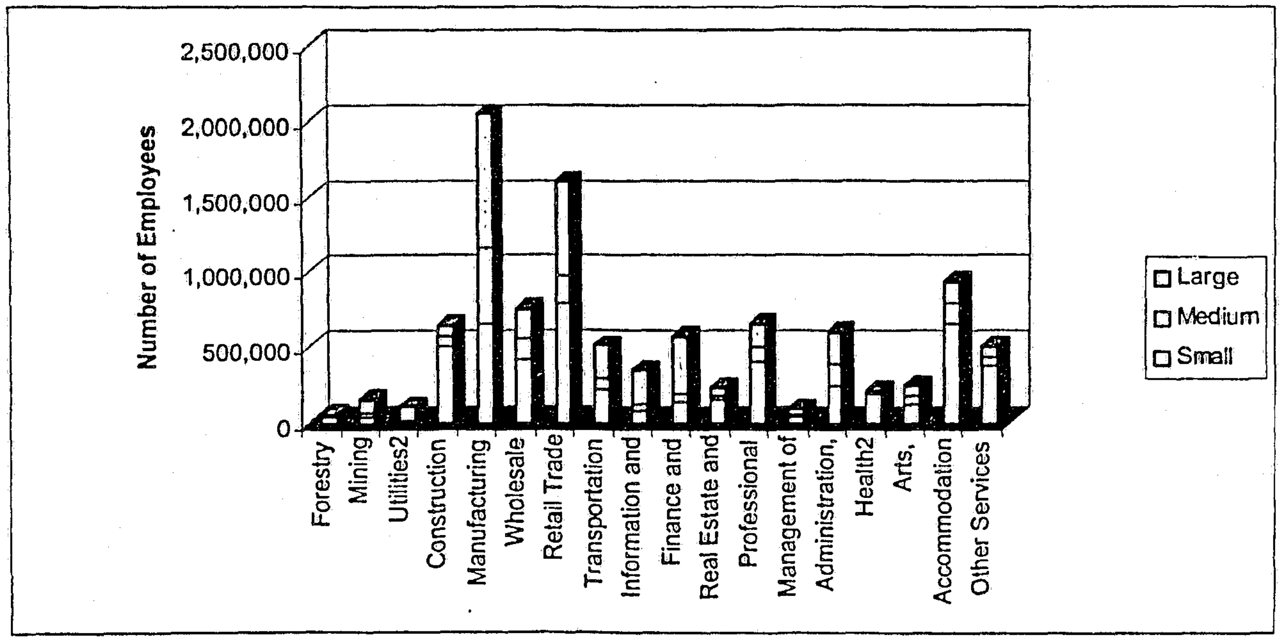


Figure 6: Sector-wise employment distribution in SMEs [19]

SMEs have multiple business sectors e.g., design, manufacturing, services, support, consumer products etc. Any product or service has its own lifecycle in the perspective of the company. For an auto-part manufacturing company that makes shock-towers, the lifecycle would start from receiving CAD drawing from the Original Equipment Manufacturer (OEM) and end at final delivery of the product. While a company that manufactures bicycles, the end of life would include disposal of the product.

In order for the SMEs to be productive and profitable, a complete view of the lifecycle of the product is necessary. Market constraints like shorter time-to-market, high quality, low price, continuous improvement etc. driven the SMEs to focus on the complete lifecycle of a product. Product Lifecycle Management (PLM) software systems is thus created in order for the companies to take full control of their products.

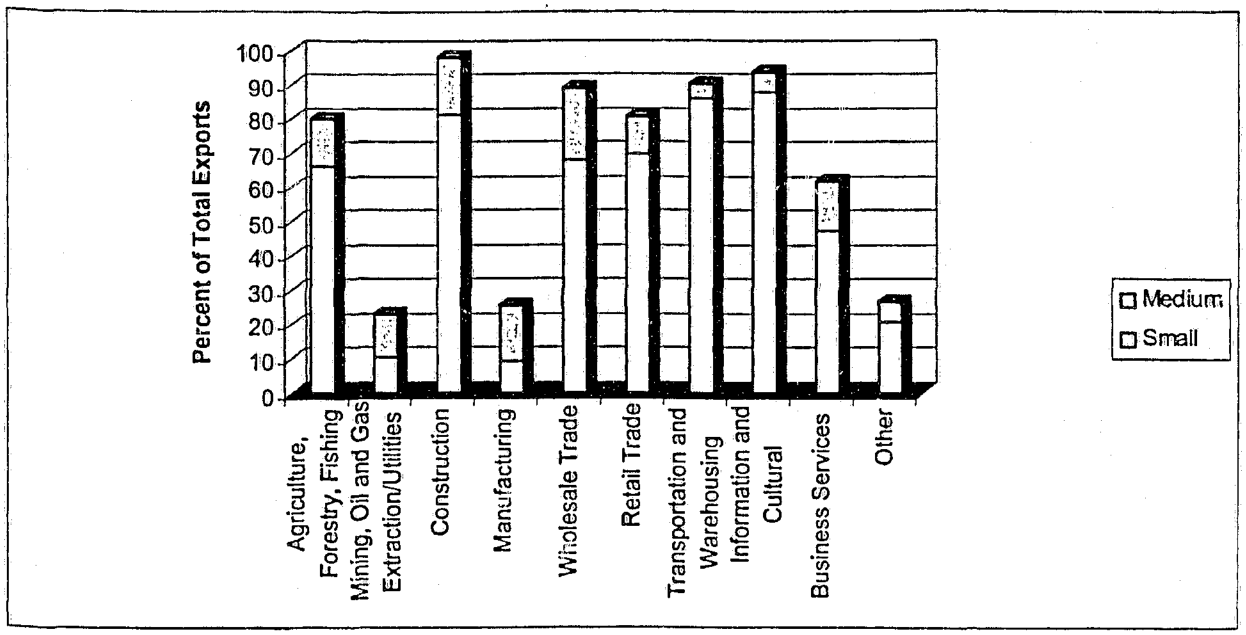


Figure 7: Sector-wise distribution of percent of total export of SMEs in Canada [19]

Manufacturing sector plays very important role in the gross domestic product (GDP) in Canada. Figure 8 shows how GDP (of individual province and the country) is related to the contribution from the manufacturing sector. It is obvious that overall GDP contribution has decreased from that of 2002 from 2003.

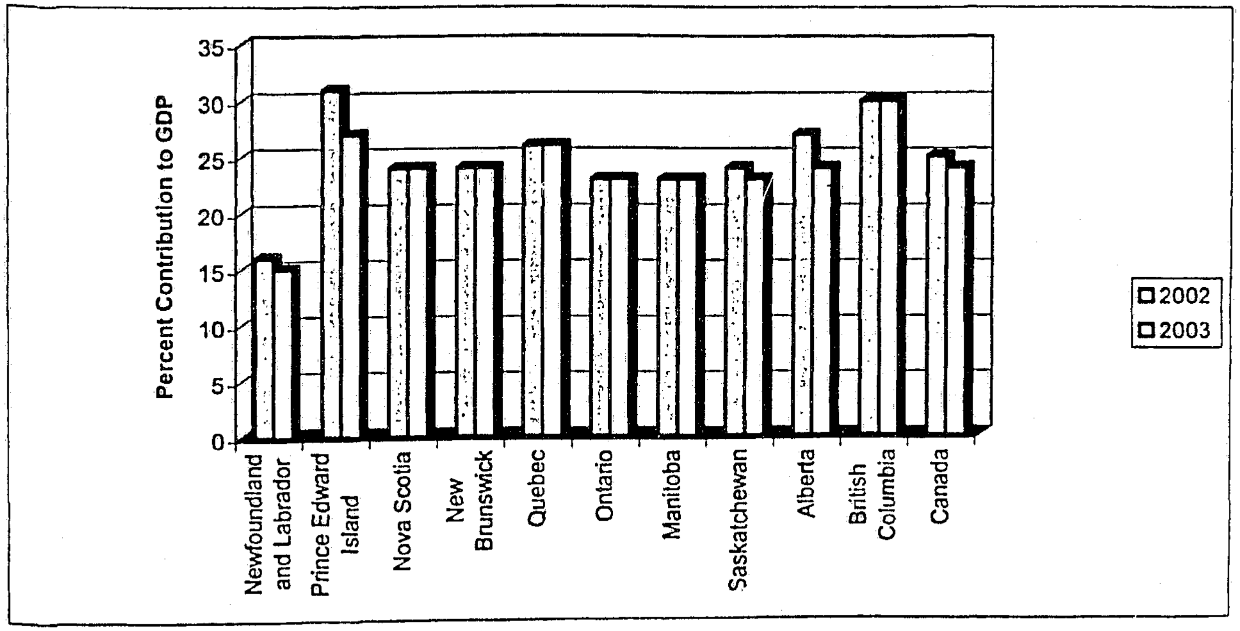


Figure 8: Contribution to GDP (provincial and national) from the manufacturing sector in Canada [19]

Another comparison is worth noticing that is showed in Figure 9. World's leading industrial nations (called G7 countries) are United States, Japan, United Kingdom, Germany, Italy and France are compared with Canada in relation to following factors [19]:

- Growth in industrial production;
- Growth in manufactured exports;
- Change in manufacturers' selling prices;
- The rate of improvement in labour productivity;
- Change in unit labour costs;
- After-tax profit margins;
- Industrial investment in machinery and equipment as a percent of GDP;
- Industrial investment in research and development as a percent of GDP;
- Industrial investment in skills training as a percent of payroll costs; and
- The rate of new product commercialization, measured in terms of the value of patents filed internationally as a percent of GDP.

These factors are benchmarked against G7 leaders and an overall score of 100% is given for perfect case.

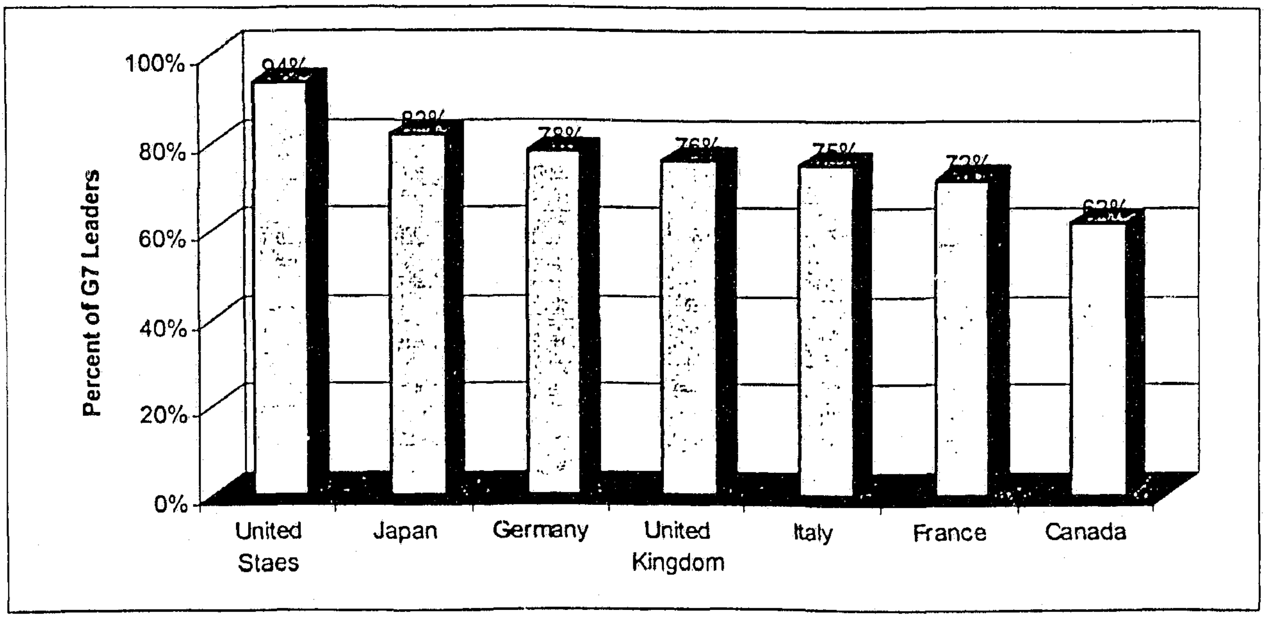


Figure 9: Competitive industrial rating of G7 countries [19]

It has been found that Canada is 62% and USA is 94% of the perfect case. This illustrates that Canada is falling back as an industrial nation.

It can therefore be deduced from the previous information that manufacturing sector plays very important role in Canada's economy but Canada's performance as an industrial nation is poor compared to other G7 countries.

Using PLM in industries effectively can create significant growth in productivity, control in price and market competition, and increase profit margin.

Large enterprises incorporated PLM systems as soon as they were available in the market. PLM solutions are very costly, they require significant IT infrastructures and they are very much vendor-dependent systems. SMEs are yet to implement such systems. That is why a gap of productivity has been created between large and medium sized enterprises.

In the software industry, Open Source concept has been proven to be an extraordinary success. Products like Linux, Apache, Compeire etc. have been built collaboratively by developers around the world and distributed at no cost to the users. PLM software can be built using open source concepts by which, SMEs would be immensely benefited. The Open Source PLM would be built using most modern technologies by expert developers around the world, reflecting the real needs of SMEs.

1.6.3 Summary

It has been found that SMEs play significant role in the overall economy. It is also obvious that Canada has the least productivity among the G7 countries. OPLM is designed to increase productivity, decrease time-to-market, enhance new product development time and maintain market competitiveness. Manufacturing sector, which is the backbone of Canada's SMEs, would become the primary users of OPLM. Other sectors would also be benefited by OPLM. As OPLM would essentially be free, designed and developed by SMEs, it is expected that OPLM would change the current trend of SMEs to a highly productive and prosperous sector. The role of PLM software in the industrial sector of G7 nations is not known. However, PLM can be used to enhance productivity in Canadian manufacturing sector for sure.

1.7 Discussion

From the discussion made in sections 5 and 6, it is clear that SMEs need to increase productivity for sustainable profitability. SMEs lack use of information technology tools that contribute to running business very effectively. Financial constraints do not allow them to use systems like PLM. They need a solution that is cheap to buy and implement. Only open source solutions can offer such services. Moreover, SMEs need a PLM solution because PLM can fulfill their business needs by collaborating, managing and controlling information both internally and externally. Therefore, it can be stated that SMEs need a PLM solution developed in open source environment.

During the life of this research, two workshops were conducted at one APD facility where representatives from SMEs were invited. This project was also presented using posters at two occasions (Open Source Conference at University of Toronto 2004 and MMO Interact 2004). Such initiatives have been proved to be very successful because of OPLM necessarily involves SMEs, open source community and academia.

Chapter 2

Methodology

2.1 Introduction

In order to develop OPLM system, the first step is to create a set of user requirements and then define the architecture of the system. A 'systems design' approach is therefore adopted in this research. Systems design is a recursive process. This means that design is not done continuously. Rather, design is done at intervals. Design process starts with a set of user requirement. At every stage, design is followed by some requirement specifications.

Let us assume that R_i is Requirements and d_i is design. Then:

$$R_{i+1} = A(R_i + d_i)$$

$$d_{i+1} = D(R_i + d_i)$$

Where A and D are design analysis and design process, respectively [20].

In the first step, requirements are collected. These requirements are from the perspective of the user. The user is only concerned with input and output of the system. This is called 'Level-zero Requirements'. The convention for a requirement has been set as [Requirement][no. of level]/[requirement no.]. For example, R0/5 means fifth requirement at level zero.

Once the level zero requirements are collected, we make some design. This design is called Level one design. This design is basically performed to identify the sub-systems. The general model of architecture is straight forward (Figure 6), which has a user interface, input, process/control function, output and maintenance/test function. Based on the simplest model, we try to find out what the subsystems are (based on the requirements at level zero).

Once the subsystems are identified, a flow diagram is established. The flow diagram shows how a process or an activity is linked with the subsystems.

Level-one requirements start once the level-one requirements (Figure 10). At this stage (as well as the subsequent stages), requirements are more specific to the system itself. Requirements of the subsystems are derived from the technical point of view that considers the user's requirements at level-zero. After this stage, level-two design is done to design sub-subsystems.

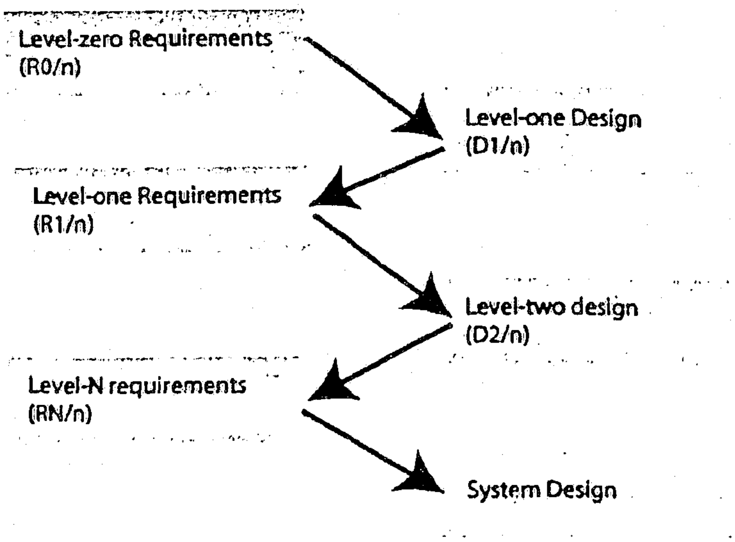


Figure 10: Recursive design process

The whole process is like opening black boxes. A black box is taken at each stage and open it to find multiple boxes in it at each level.

Traceability is very important in this process. It is often required to go back and fix an error here and there and then be able to follow subsequent links to make required changes. 'Some design' at every stage enables one to trace the evolution of changes.

2.1.1 Systems development lifecycle (SDLC)

SDLC is often called as 'software development life cycle' or 'software life cycle'. According to IEEE Standard Glossary, SDLC is the period of time that starts when a software product is conceived and ends when the product is no longer available for use. Typically, SDLC includes seven phases: requirements phase, design phase, implementation phase, test phase, installation and check-out phase, operation and maintenance phase and finally, retirement phase.

A variety of lifecycle models are used to develop a system or software. Major methodologies for SDLC are discussed in the following section:

1. Code and Fix: This is the simplest approach of product construction without specification or attempt of design. Coding, compilation, testing and fixing errors are repeated until the solution is good enough. This is a 'trial and error' approach, which works fine for small software projects but unacceptable for large or complex systems [21]
2. Waterfall model: This is one of the first formalized SDLC models (Figure

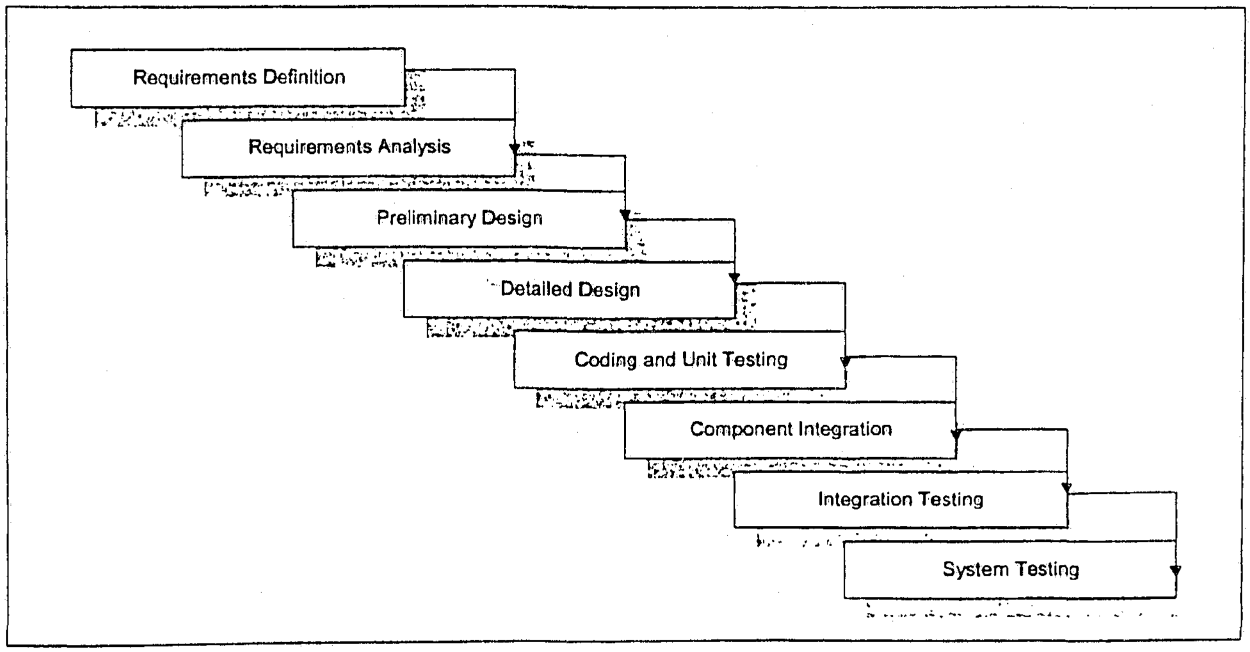


Figure 11: Waterfall Model for software development

11). In this model, development proceeds in a stepwise manner from requirements through design, implementation, testing, and finally operation. Figure 11 shows waterfall model.

Waterfall model has following advantages:

- Good progress tracking due to clear development stages.
- Milestones and deliverables can be clearly identified.
- Project Management and control is facilitated by the need to complete each stage before moving to the next.

Despite its simplicity, this model has few limitations. They are

- It is difficult to define all requirements at the beginning of a project
- Waterfall model has limited adaptability to change
- A working version of the system is not produced until late in the project's life
- Not suitable for large projects

3. Incremental model: The incremental approach attempts to combine the waterfall sequence with some of the advantages of prototyping. This approach is favored by many object-oriented practitioners. It basically divides the overall project into a number of increments. Then it applies the waterfall model to each increment. The system is put into production when the first increment is delivered. As time passes additional increments are completed and added to the working system (Figure 12).

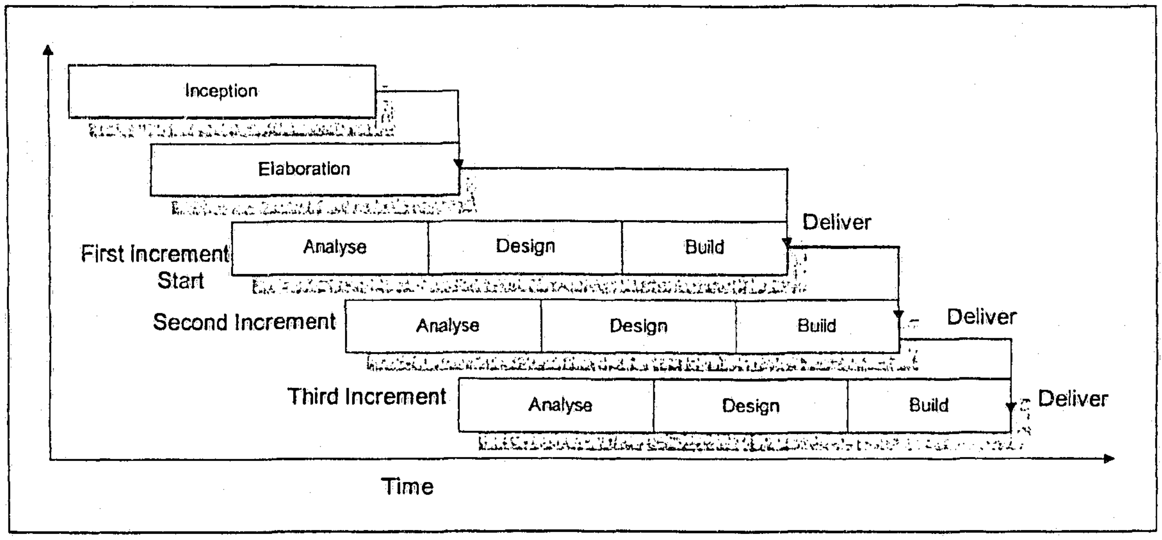


Figure 12: Incremental model for software development [22]

Incremental Phases

- Inception: During the inception phase, the purpose, business rationale, and scope of the project is determined. This is similar to the feasibility analysis that is done in other life cycles.
- Elaboration: During the elaboration phase, more detailed requirements are collected, high-level analysis is performed, and a general architecture is determined. This phase divides the requirements into increments that can be built separately. Each increment consists of a subset of high-level use cases that capture the user's requirements.
- Construction: The construction phase builds increments of the system. Each increment is developed using a waterfall approach. This includes detailed analysis and design for the use cases in the increment and coding and testing of the event processors that implement the sequence of events defined by the use cases. The result is production quality software that satisfies a subset of the requirements and is delivered to the end users. Work on different increments may be done in parallel.
- Transition: The transition phase (not shown in the figure) is the last phase in the project. This may include such things a performance tuning and rollout to all users.

2.1.2 Summary:

Apart from the above three SDLC models, there are many other system development models such as spiral model, build and fix model, rapid application development model etc. Most of these applications are highly methodological and applicable for commercial software development. As OPLM would be developed in an open source environment, it is necessary to define the open source software development model. However, research shows that there is no established method for open source software development. In the next section, the author will propose a model for SDLC that can be used in open source software development.

2.2 Adaptive model

For the development of OPLM, the 'adaptive model' is implemented to depict the lifecycle of OPLM system. In this model, a survey is conducted in the beginning. Along with the survey, a literature review is also conducted. The Literature review consists of existing product research, research of publications by academia and industry and reports from other sources on the domain of the system being developed. As the survey and literature review are complete, requirements for the system start to develop. As the requirements are listed, a little bit of design is done. Depending on the state of design, coding (programming) is done for part (or whole) of it and it is tested for bugs. Feedback is taken from these steps as the process continues to a major revision. Once this cycle stops (at n-level of test) (Figure 13). At that point, the product is ready for rollout.

It can be noted that the versions presented in each cycle can be seen as a 'ball' that rolls to the next step as it accumulates more information. The term 'version' inside the balls are intended to show the 'internal' versions of the system. The final product that is ready for the market to be rolled out is called 'release'.

Questionnaire and results of the survey conducted for OPLM is available at Chapter 4 and Appendix A. Literature review (industry perspectives, existing product review and review of research publications) is presented in chapters 1, 3 and 5. A detailed set of user requirements are given in chapter 4. High-level design of OPLM is presented in chapter 6.

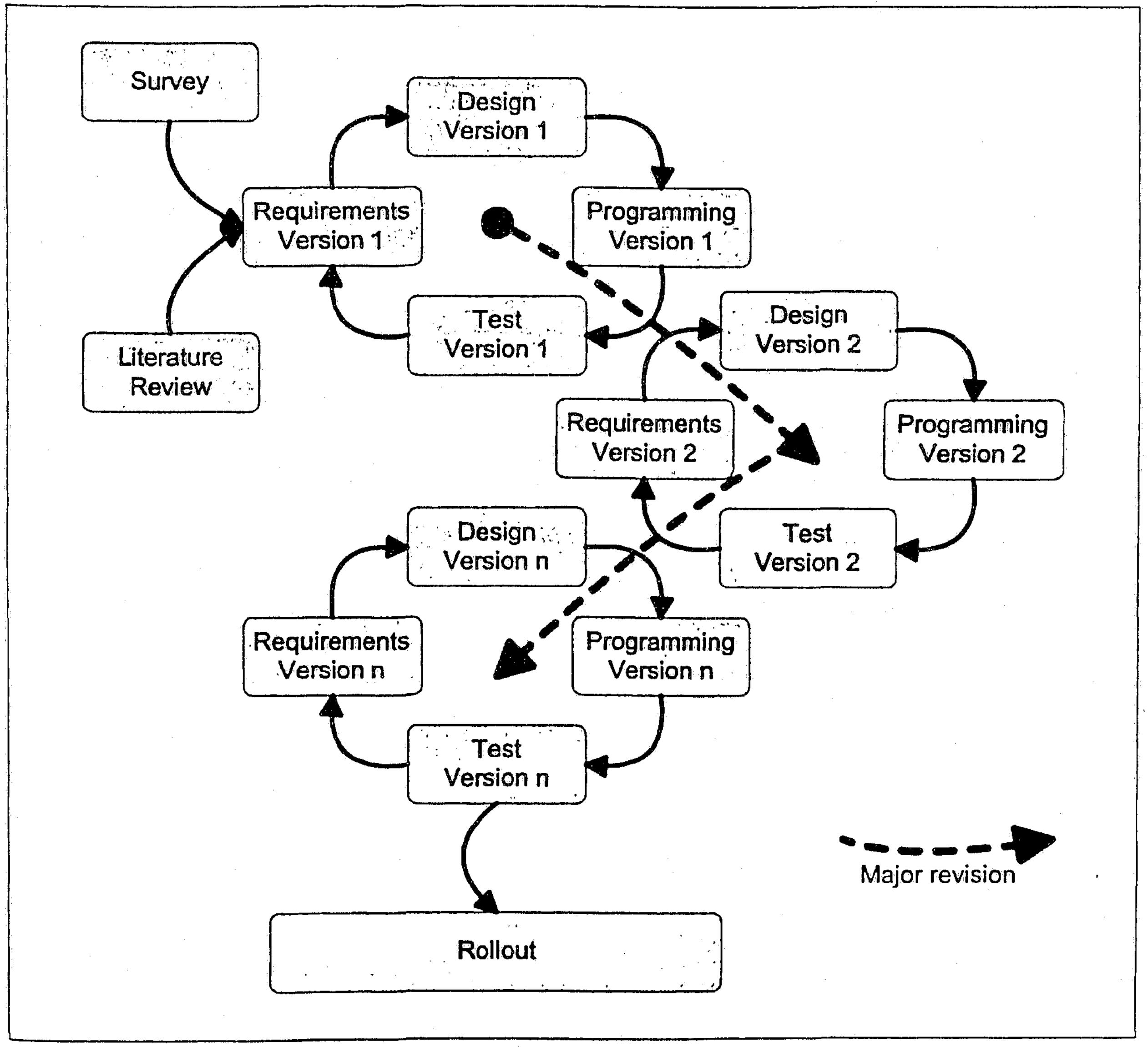


Figure 13: Adaptive systems design methodology proposed for open source software development

2.3 Discussion

This proposed model is designed for open source software development that would follow our proposed open source model discussed in the previous chapter. In this model, requirements, design, programming and testing continues as cycle.

Open source developers and contributors often have multiple roles (requirements engineer, design engineer, programmer, test engineer etc.) and their responsibilities may overlap. This model is expected to work in such setting. It is imperative that requirements versions in each 'major revision' may be the same while other products (such as design, program or test version) may be different.

The Adaptive Model proposed has certain advantages. First, it is suitable for open source development and consistent with the open source model proposed by us. It has been found that open source development does not follow traditional ways of software building, rather, developers in a dispersed environment work loosely in the beginning of a project and later, as things try to take shape, productivity grows [18]. The adaptive model would sustain the nature of open source development because it captures development cycles till the final version of testing. In traditional software development, testing takes 25-30% time of the development cycle time of the product. In adaptive model, testing is performed simultaneously with other functions, thus reducing the time for roll out of the product. This hypothetical model is expected to work efficiently as OPLM continues to be developed.

Chapter 3

Literature review

3.1 PLM research

Only a few research papers in product lifecycle management has been published in the recent years. A search conducted at IEEE Xplore or ScholarsPortal for 'product lifecycle management' returns thirteen results. Most of the work in PLM has been done either from the perspective of environmental lifecycle management or radio frequency identification (RFID) technologies. In the first group, it is argued that the product design and environmental management should be integrated. In each step of a product's lifecycle, that is, product design, process design, production, distribution and service, eco-friendly inputs are provided. The best one can do is utilize the 'product design' phase and consider environmental issues. The concept of eco-PLM is that the product is designed using environmental guidelines and then it is controlled throughout its life.

Issues relating to disposal of products have been the focus of PLM using RFID technology. A great deal of product information is lost as a product passes through its lifecycle from production, to retail, to the consumer, to disposal, remanufacturing or resale [23]. RFID (Radio Frequency IDentification) is an electronic barcode where semiconductor chips are used to store data that can be broadcast via radio waves (30 meters) to the reader eliminating the need of current bar-code reading technology that requires line-of-sight. This paper [23] argues that 'products are becoming increasingly self-managing' as RFID can be used to track product disposal, remanufacture or resale. A research from Hitachi [24] conducted a study on used PC's and their disposal. Information held in an

RFID tag can be up to 2 kilobytes. It has been shown that retailers and manufacturers can optimize their recycling process by using RFID tags.

A PLM solution based on Java 2 Enterprise Edition (J2EE) and stage-gate technology is proposed by [25]. The basic framework layer has client, functional model, basic management, basic framework and J2EE platform. Client layer defines roles and interface to applications, and the environment for making the application run. Function module layer gathers structure management, project management, workflow management etc. Basic management layer is the core module which performs similar duties of PDM, such as persons, data, processes etc. This paper presents a unique way of 'conversion of states' (responsible for determining maturity process of a product and transferring data from one lifecycle to another). Those are 'submit, promote, demote, deny, drop, reassign and reproject' (Figure 14). The set of requirements put forward in this paper is limited because this paper does not discuss the issues of engineering change management, business process management and most importantly, business intelligence.

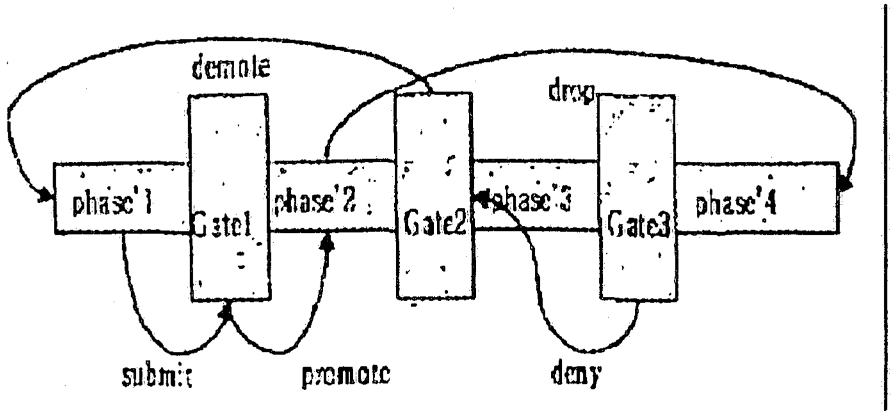


Figure 14: Basic actions of the conversion of the stage states [14]

Due to economic and market factors, today's companies are forced to globalize. As management of such disperse locations has given rise to collaborative engineering using the Internet. Especially CAD/CAE collaboration is required to be seamlessly occurring during the product design process over the Internet [26]. The type of such collaboration can be synchronous as well as asynchronous. Using open standard protocols like TCP/IP, HTTP and FTP, [26] proposes a framework for workflow driven collaborative session management. In this process, a 'planner' defines product design process. Designers and problem solvers use the 'product design workflow system' to design, simulate and evaluate product design. A 'coordination system' takes control of the Internet collaboration.

Similar perspectives are presented by [27] where distributed engineering design can be accomplished by using 'virtual enterprises'. Some of the common characteristics of virtual enterprises are distributed execution of virtual business processes across organizations barriers, autonomous execution of processes within administrative domains, access control and authorization of remote business process requests, and dynamic selection of partners and automated negotiations [27]. The model presented by this paper has four domains: customer, virtual enterprise (VE) representative, VE candidate/partner, virtual marketplace. Customer is one who subscribes to the services, VE representative is where end users log on and services are deployed, VE candidate/partner is the domain that registers its service offerings in the marketplace and negotiates with other partners and finally, virtual marketplace is the domain that provides registration and selection of services for VE candidate partners. Some agents like service type agent, service offer agent and service offer retrieval agent are used to manage service types, service offers and service offer retrieval respectively.

3.2 Product Data Management

Product data management systems are employed to control documents and data files emerging from the creative and collaborative stages of product design, e.g., CAD/CAM data. Workflow management systems allow managers to coordinate and schedule the activities of an organization to optimize the flow of information and operations between resources of organization. Therefore in manufacturing systems design engineers typically use a PDM and production managers use Production Planning system or/and workflow management software [28]. Product breakdown structure and assembly breakdown structure are used to control BOM information. [28] proposes a model that integrates workflow with PDM. In order to accomplish the integration, a common definition of data model is created. This common data model allows links between PDM and workflow in a seamless way. The data model is called CRISTAL. Using references from Object Management Group (OMG), CRISTAL is integrated with Object Oriented database management system for full functionality. This system is focused on integration of workflow management and does not include issues like creation of business objects, integration with STEP and change management functions. A few other papers [29], [30], [31] provides solutions for workflow management and PDM integration as well as manufacturing process analysis and optimization.

In general, management of design data has been a focus of researchers. As companies deal with increasing number of files, increasing complexity of design processes, high rate of update in design data, challenges of robust data management system is now more demanding. [32] proposes a model for managing design data management. Five dimensions of data are version, views, hierarchy, status and variants. Using one or more dimensions, any mix of design data can be represented. This model is very useful for complex data structures that include assemblies, change processes, data exchange and related issues. For example, using a two-dimension case of views and status, one can define the quality control elements into data flow model. [32] makes a pair wise comparison for two-dimensional combinations. Using more dimensional combination, one can represent fairly complex data structures.

Modern approaches to engineering, design and manufacturing facilitate design flow with PDM systems. PDM systems serve as the central repository for all design, manufacturing and BOM data that include CAD models, engineering data and other documentation. Figure 15 shows a schematic how PDM can support different departments in an organization.

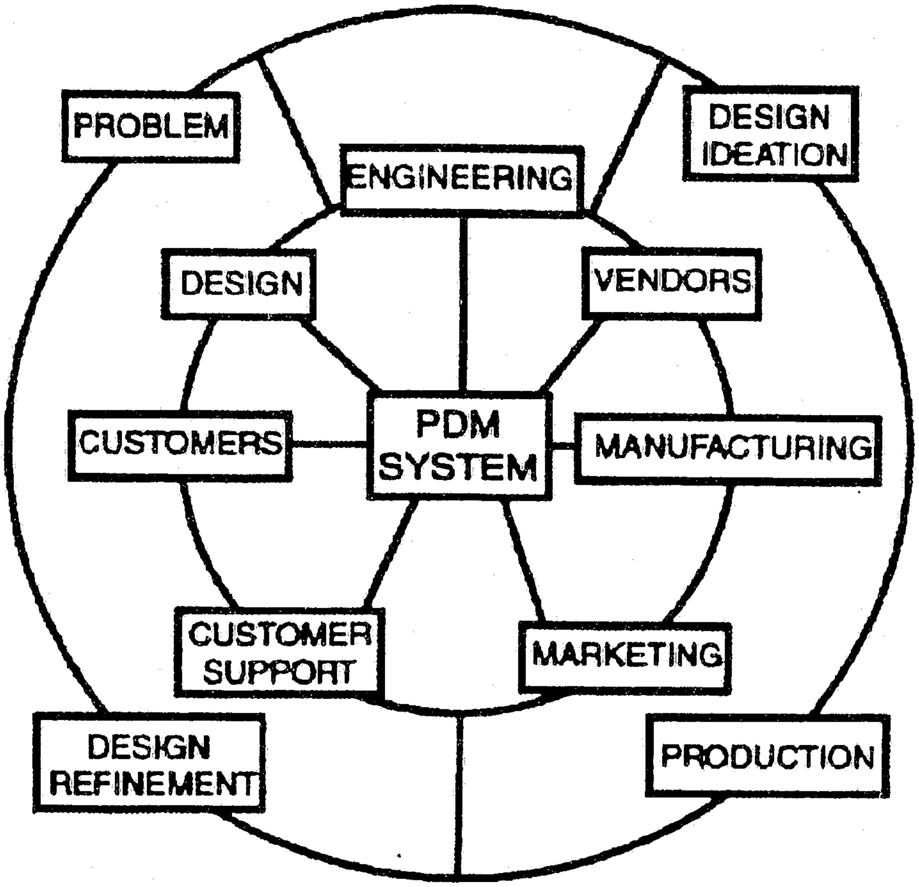


Figure 15: Interaction of units and functions of organization in a PDM system [33]

Two features of PDM will be discussed now: data management and process management.

3.2.1 Data Management

The objective of data management is to help engineers find the information they need. Traditional systems are good at storing components and assembly drawings. But often engineers need comprehensive records for the size, weight, etc. of parts that the traditional PDM systems do not capture. Ideally, PDM systems should be able to manage attribute and documentary product data, as well as relationships between them.

One of the fundamental capabilities of a PDM system is to be able to classify information. Similar types of information should be grouped in each class.

3.2.1.1 Classification of Components

Components are entered in the database in accordance with the business need of a company. Therefore, PLM would have to be configured for each company separately. Classes can be grouped together under convenient broad headings that allow all working stock of components to be organized in an easily traceable hierarchical network structure. Each part can be given its own set of attributes. Additionally, some systems allow registering that certain components are available with specific 'optional' attributes. This can be invaluable in controlling BOM.

3.2.1.2 Classification of Documents

Documents relating to components and assemblies can be classified similarly; for example, classes might be 'drawings', '3D models', 'Technical publications', 'Spread Sheet Files', etc. Each document can have its own set of attributes - part, number, author, date entered. At the same time, relationships between documents and the components themselves can be maintained. So, for example, a dossier for a specific 'bearing assembly' could be extracted, containing 2D drawings, solid models, and FEA files.

3.2.1.3 Product Structure

The third way product data can be accessed is by product structure. For any selected product, the relationship between its component assemblies and between the parts that make up these assemblies should be maintained. This would mean that one could open a complete Bill of Materials, including documents and parts, either for the entire product or selected assemblies. One distinct advantage is the ability to hold not just the physical relationships between parts in an assembly but also other kinds of structures; for instance, manufacturing, financial, maintenance or document relationships. So, it is possible for specialist team members to see the product structured from their point of view.

3.2.1.4 Querying the Data

It is required to 'get at' the components and assembly data by a variety of routes. One can move up and down a classification tree; pick way through a product structure; simply call-up the data by searching for it by name or part number, or search for groups of data by specifying an attribute or combination of attributes. For example, one could ask to see all stainless steel rivets with anodized shanks less than 10mm long. This type of function is performed in relational databases.

3.2.1.5 Process Management

Data management is basically a passive procedure. Process management, on the other hand, is about controlling the way people create and modify data - active procedures.

Process management systems normally have three broad functions:

- They manage what happens to the data when someone works on it. ('Work Management'.)
- They manage the flow of data between people. ('Workflow Management'.)
- They keep track of all the events and movements that happen in functions 1 and 2 during the history of a project. ('Work History Management'.)

PDM systems vary widely in how they perform these functions. The following is a broad overview.

3.2.1.6 Work Management

Work management can be defined as management of data when it is being accessed, modified or added. It is sometimes synonymous with ‘revision control’. PDM systems offer a solution by acting as the engineer's working environment, meticulously capturing all new and changed data as it is generated, maintaining a record of which version it is, recalling it on demand and effectively keeping track of the engineer's every move. A solid model, for example, may go through hundreds of design changes during the course of development, each involving far-reaching modifications to the underlying engineering data.

In order to carry out a design modification, more than just the original design and the Engineering Change Order (ECO) is required. Many documents, files and forms may need to be referred to and other members of the design team involved, too. Current PDM systems cope with this requirement with varying degrees of success. One approach is that which emulates paper-based processes by using what are known as 'user packets'. The packet allows the engineer to manage and modify several different master documents simultaneously as well as providing various supporting documents for reference.

3.2.1.7 Workflow Management

Workflow management controls and monitors the flow of work among participants. During the development of a product many thousands of parts may need to be designed. For each part, files need to be created, modified, viewed, checked and approved by many different people, perhaps several times over. What's more, each part will call for different development techniques and different types of data - solid models for some, circuit diagrams for others, FEA data for others.

Most PDM systems allow the project leader to control the progress of the project via 'states' using pre-determined 'triggers' and a routing list that may vary according to what type of organization or development project is involved. The way systems differ is in how much flexibility they permit within the framework discipline. The most rigid systems are based on procedures. Every individual or group of individuals is made to represent a state in a procedure - 'Initiated', 'Submitted', 'Checked', 'Approved', 'Released'; a file or record cannot be moved from one individual or group to the next without changing states. Some systems make it possible to give the task an identity of its own, separate from the people working on it.

3.2.1.8 Work History Management

Work history management is an extension of 'revision history'. It not only captures the revisions of data but also the revisions of a process. Product data management systems should not just keep comprehensive database records of the current state of the project, they should also record the states the project has been through. This means that they are a potentially valuable source of audit trial data. The ability to perform regular process audits is a fundamental requirement for conformance to international quality management standards. But project history management is also important to allow to 'back-track' to specific points in a project's development where a problem arose, or from which you may wish to now start a new line of development.

With specific development milestones, the system records are important. Some systems record only the changes in ownership of documents. So ownership can be traced at a specific point in time, but not the modification itself. Others have the ability to record changes but may do so as a series of 'snap-shots' taken only when a file changes 'state'. This can leave large gaps in workflow history as a user may have been making modifications to a design for several weeks without any change to its state. Some systems provide an historic record that is like a 'moving picture' by allowing you to record changes to any system-defined level you choose - for example, every time a modified file is saved.

3.3 Introduction to STEP

Data management in an engineering environment poses the issue of non-compatibility between CAD applications from different vendors that are complex to solve. The International Standardization Organization (ISO) has outlined how data could have an internationally acceptable format that would ensure interoperability. ISO 10303 is the standard for STEP and it describes CAD and PDM related information for representation of data. STEP is written in EXPRESS. This section will discuss how STEP works towards the solution of data management in the lifecycle point of view.

STEP, the Standard for the Exchange of Product Model Data, is a comprehensive ISO standard (ISO 10303) that describes how to represent and exchange digital product information. STEP consists of different Application Protocol (AP)'s. The architecture of STEP is shown in Figure 16.

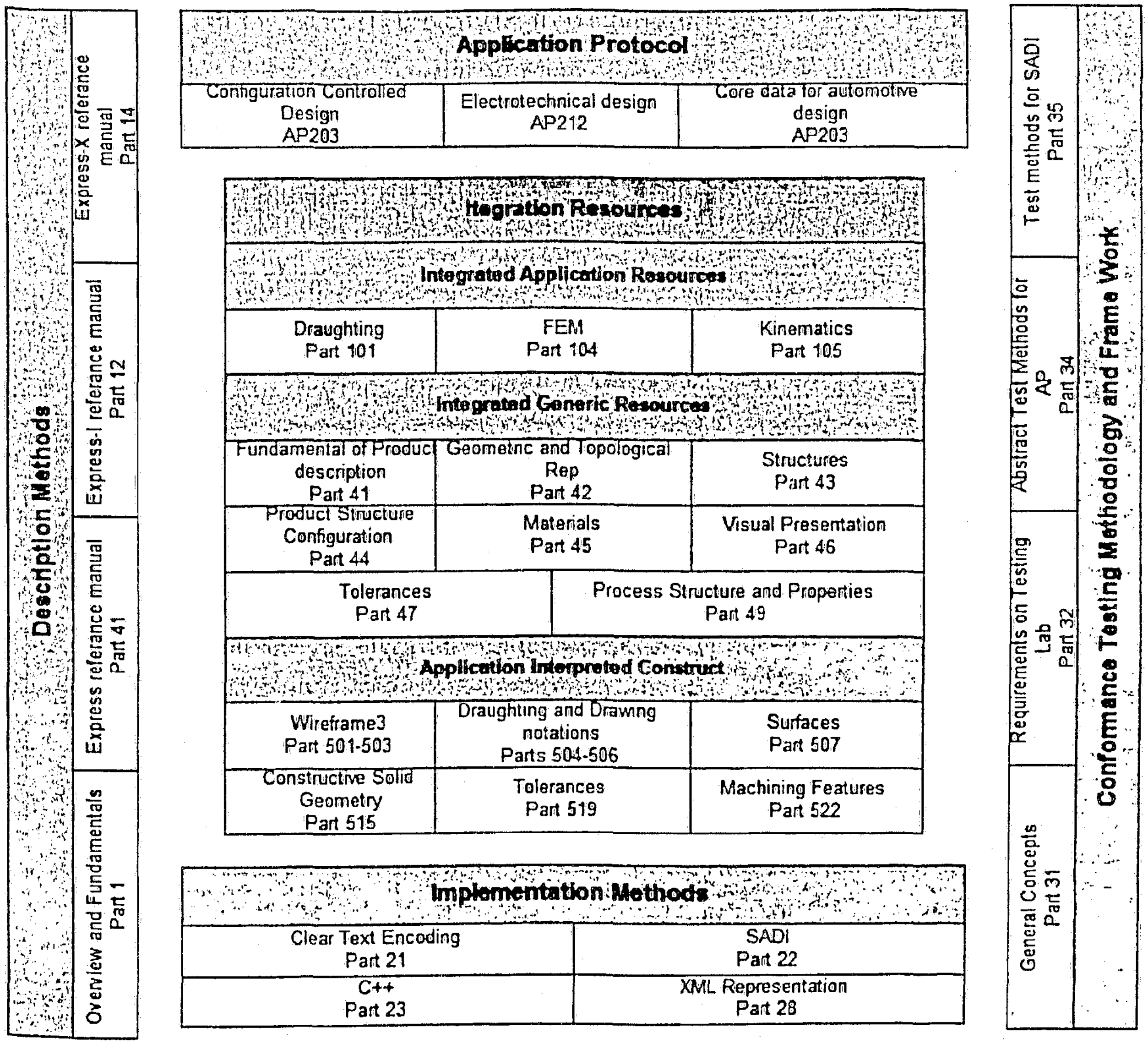
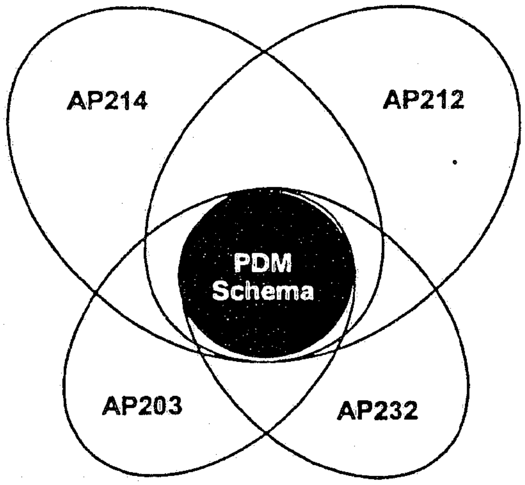

Application Protocol
Configuration Controlled Design AP203
Electrotechnical design AP212
Core data for automotive design AP203
Itegration Resources
Integrated Application Resources
Draughting Part 101
FEM Part 104
Kinematics Part 105
Integrated Generic Resources
Fundamental of Product description Part 41
Geometric and Topological Rep Part 42
Structures Part 43
Product Structure Configuration Part 44
Materials Part 45
Visual Presentation Part 46
Tolerances Part 47
Process Structure and Properties Part 49
Application Interpreted Construct
Wireframe3 Part 501-503
Draughting and Drawing notations Parts 504-506
Surfaces Part 507
Constructive Solid Geometry Part 515
Tolerances Part 519
Machining Features Part 522
Implementation Methods
Clear Text Encoding Part 21
SADI Part 22
C++ Part 23
XML Representation Part 28
Description Methods
Express-X referance manual Part 14
Express-I referance manual Part 12
Express referance manual Part 41
Overview and Fundamentals Part 1
Conformance Testing Methodology and Frame Work
Test methods for SADI Part 35
Abstract Test Methods for AP Part 34
Requirements on Testing Lab Part 32
General Concepts Part 31

Figure 16: Architecture of STEP showing major resources and framework [34].

3.3.1 STEP Schema for PDM

The STEP PDM Schema is a core set of entities in STEP that support the mapping of concepts for Product Data Management (PDM). The PDM Schema is the result of a cooperative development process between ProSTEP AG and PDES, Inc. ProSTEP is a Europe based company works to solve the integration problems faced by the manufacturing industry. It has 220 members from 17 countries. PDES Inc., based in North America also works together with industry leaders, government and acidic members to advance the development of STEP technology. The PDM Schema has been established to promote interoperability between STEP APs in the area of product data management.

A PDM system would essentially handle the lifecycle data of a product (Figure 17). From this point of view, management activities can be grouped into six major functions: engineering drawing management, material specification management, product structure management, production schedule management, technical document management and engineering change management [35].



☞ **CommonPDM data Schema generated and maintained by PDES, Inc. and ProSTEP**

☞ **Real Subset ofPDM relevant STEP APs (AP203, 212, 214, 232)**

☞ **Fulfills nearly all requirements for PDM data exchange**

Main functionality for parts and documents:

- identification
- versioning
- structures incl. transformations
- approvals and authorization
- project, work order, work request
- effectivities
- classification and properties

Figure 17: Positioning and Contents of the PDM Schema [35]

The PDM system should be an integrated one and have different technologies need to be applied to ensure that. It should be capable of handling multiple databases; use an international data representation format, client/server architecture and user-friendly interface (Figure 18). It is also desirable that it can be implemented on Internet using a simple web browser.

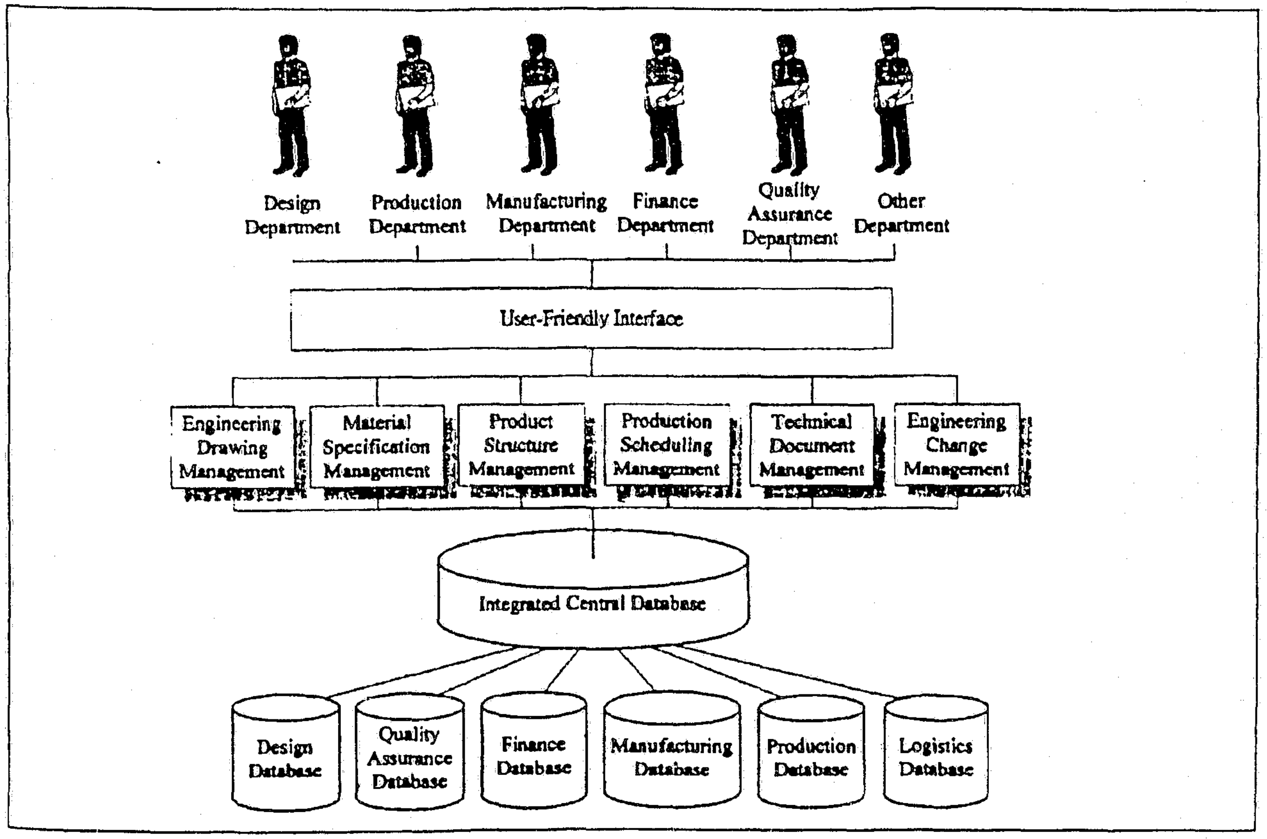

Design Department
Production Department
Manufacturing Department
Finance Department
Quality Assurance Department
Other Department
User-Friendly Interface
Engineering Drawing Management
Material Specification Management
Product Structure Management
Production Scheduling Management
Technical Document Management
Engineering Change Management
Integrated Central Database
Design Database
Quality Assurance Database
Finance Database
Manufacturing Database
Production Database
Logistics Database

Figure 18: The schematic concept of a PDM system [36]

Product development is complex and needs to be integrated. Figure 18 shows the basic architecture of a PDM system that would be interoperable and ensure proper exchange of heterogeneous data.

Figure 19 shows the implementation architecture for STEP-based PDM.

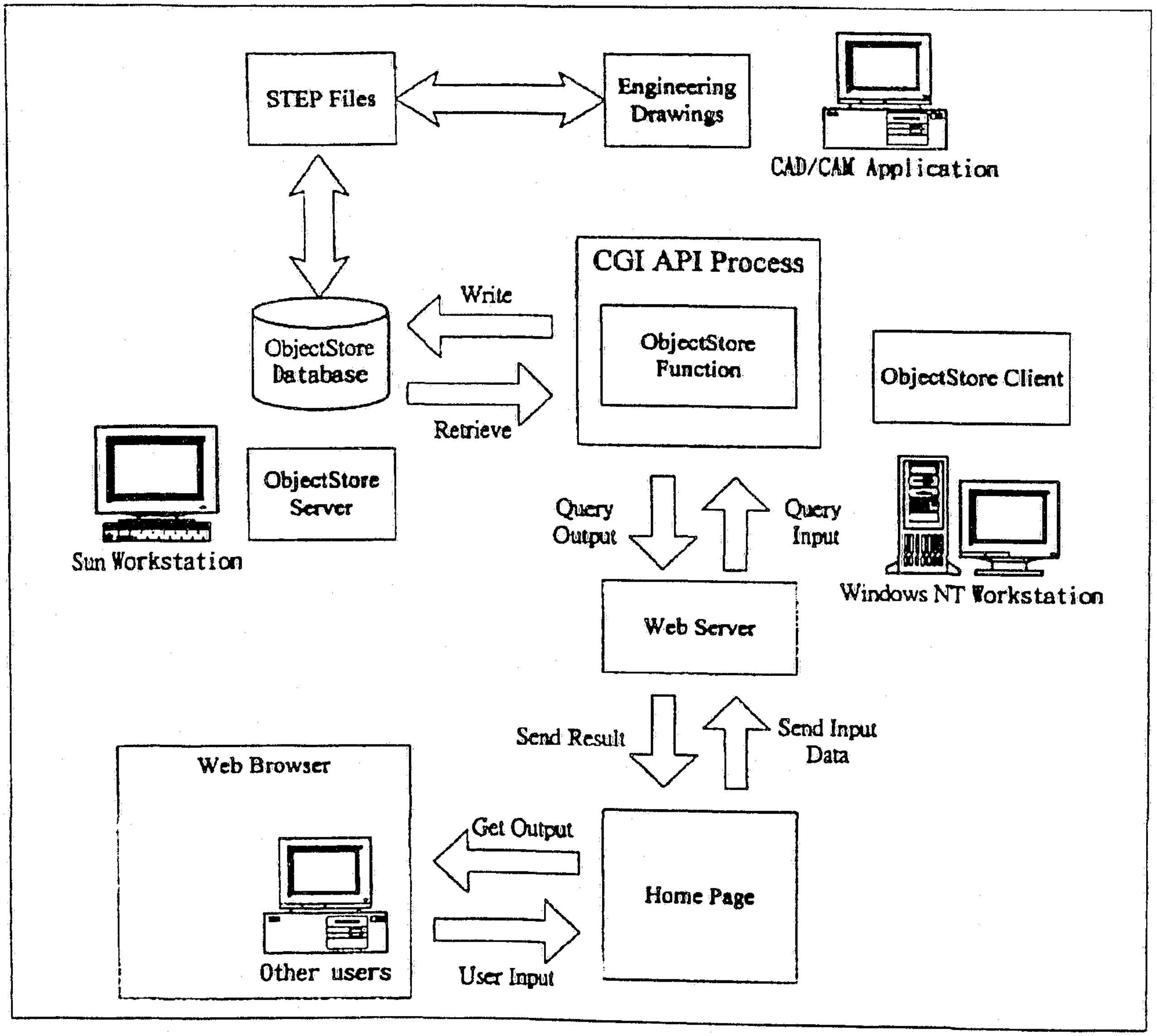


Figure 19: Implantation of STEP in PDM systems [36]

3.4 Business Process Modeling

A business process can be defined as 'a network of actions performed in the context of one or more organizational roles in pursuit of some goal' [37]. Another definition of business process can be 'the combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result' [38]. In general, a business process has five elements [39]:

1. A business process has its customers
2. A business process is composed of activities

3. These activities are aimed at creating value for customers
4. These activities are operated by actors which may be humans or machines
5. A business process may involve several organizational units

In the last decade, information technology has undergone rapid advancement. E-business has dramatically changed the context in which decision-making takes place. While fundamental human elements in the organization remains largely unaffected, e-business places new demands and constraints on the decision maker to provide better service to the customer [40]. Decisions are required to be faster, more accurate and effective although amount of information is huge as compared to the same as a decade ago. This pressure has created the need to study Business Processes.

Business process modeling enables a common understanding and analysis of a business process. A process model can provide a comprehensive understanding of a process [38]. Building a model for the business processes in an organization facilitates improved understanding organizational processes. The model can also be monitored, analyzed and controlled. Creating an enterprise model can be instructive in itself by revealing anomalies, inconsistencies, inefficiencies and opportunities for improvement [41].

3.4.1 Business Process Modeling Techniques

Different techniques and tools are used to model business processes. In order to choose the right technique, the purpose of the business model needs to be defined. For example, a model built for representing a process would be different than the model built to control the process.

Table 21 provides a comparison among different business process modeling tools. Flow charts are easy to create and use but the boundaries of processes are vague. Flow charts are good for high-level representation of a process. Data

Flow Diagram (DFD) is used to show the flow of raw data to its destination. Action diagram, which is a special case of DFD, shows material as well data flow. Role-Activity Diagram (RAD) focuses on roles and responsibilities to represent a process. RID (Role Interaction Diagram) is a graph of process that is a combination of RAD and Object oriented model. Activities are show vertically and roles are shown horizontally at top. RID is more effective in showing coordinated and interrelated activities. Gantt charts are simple graphs (Table 10) to represent activities on a time scale what give good control although it is not good for analysis of a process.

TASK	DURATION	WHO	WEEK					
			1	2	3	4	5	6
RESEARCH NEEDS:	1 week		■					
REGULATIONS:								
Research	1 week		■					
Compliance	1 week		■					
ENERGY CONSERVATION:								
Research	2 weeks		■	■				
Apply to prototype	1 week							
INTERIOR DESIGN:	3 weeks				■	■	■	
EXTERIOR DESIGN:	3 weeks				■	■	■	
PROTOTYPE:	1 week							■
COST ANALYSIS:	6 weeks	ALL	■	■	■	■	■	■

Table 10: Example of a Gantt chart

Integrated Definition of Function Modeling (IDEF) is derived from Structured Analysis and Design Technique (SADT) [42]. IDEF has different parts; most important of them are IDEF0, IDEF1, IDEF1X, IDEF2, IDEF3, IDEF4 and IDEF5. Figure 20 shows a basic syntax for IDEF0 model.

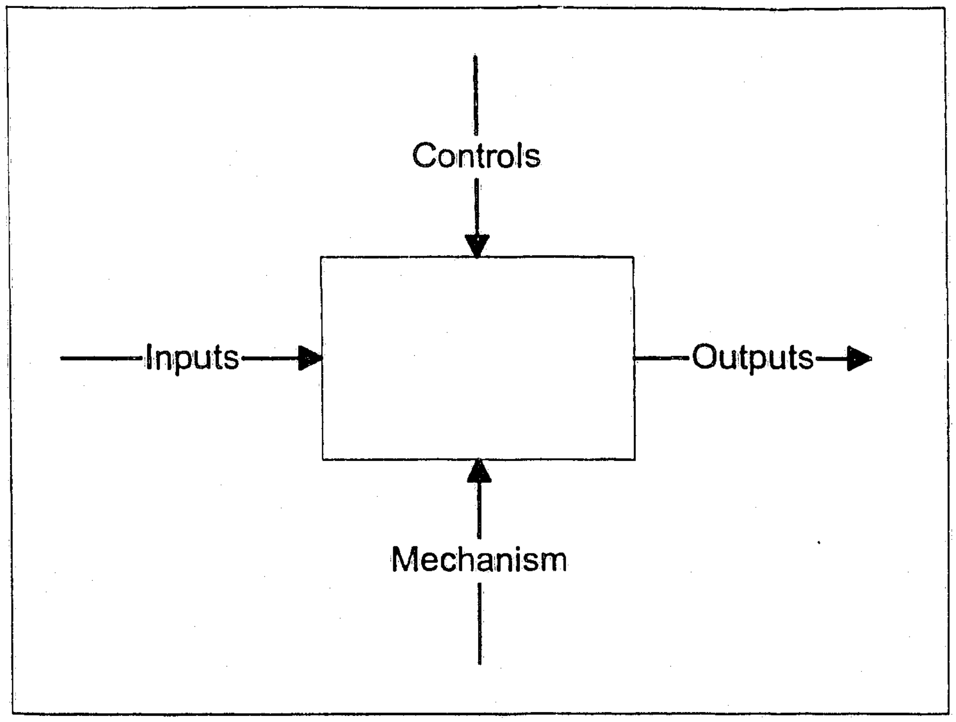


Figure 20: Basic syntax for IDEF0 [42]

IDEF0 is the most popular standard for process modeling. It has a hierarchical structure that facilitates quick mapping at high level. Other parts of IDEF deal with information modeling, data modeling, simulation modeling, process description capturing, object modeling and ontology modeling. Object oriented methods use UML for process modeling. UML has nine types of diagrams (class, sequence, activity, statechart, object, collaboration, use-case, component and deployment diagrams). UML covers a conceptual and concrete element that creates a stronghold for business process representation.

3.5 Business Rules Modeling

When an enterprise conducts its business, it follows certain policies, procedures and constraints. These are called business rules [28]. Business rules are generally statements that guide the behavior or information of the organization. They are usually decisions on how a business must be conducted. Figure 21 shows the relationship between enterprise objectives and operational system with business rules. They actually determine how the business should be run.

The objectives of the enterprise determine what the rules would be. Here are few reasons why business rules should be modeled [44]:

1. Business rules enable an analysis from a global perspective to see if the operating principles are sufficiently complete and consistent
2. They make possible for an analysis of how well the rules achieve the enterprise objectives
3. They ensure that the developed or acquired applications and systems are in conformity with the operating principles of the enterprise
4. The developed systems are better capable of quickly changing in response to modified operating principles, without major systems efforts.

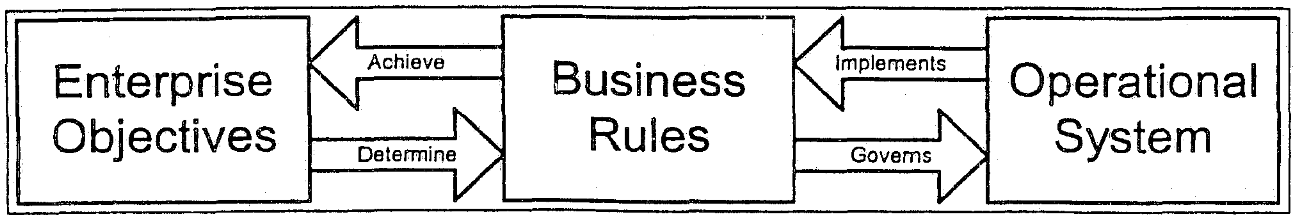


Figure 21: Relationship of business rules in an enterprise [44]

Business rules are often hard-coded in software applications. This creates a lingering problem for the organization because rules always change and it is impossible to adapt to the changing rules by changing codes. The solution to this problem is to create a business rules repository that would control the operations and systems of the organization.

Business rule are required to have the following attributes:

1. Atomic: A business rule must talk about a single thought and be complete by itself
2. Non-procedural: Business rules must not describe a procedure, it will simple make a declaration of the rule
3. Non-conflicting: Two rules must not create conflict with each other.
4. Non-repetitive: Similar information must be repeated among rules

Technique	Description	Attributes	Characteristics	Strengths and Weakness			
				User perspective		Modeler perspective	
				Strength	Weakness	Strength	Weakness
Flow Chart	Graphic Representation	Flow of actions	Not sub-layers Great details No overview	Communication Ability	Can be too large	Flexibility quick, Simple	No method available Different notations
DFD	Descriptive Diagrams for structured analysis	Flow of data	Explains logical level sub-layers	Easy to Understand	Only flow of data is shown	Easy to verify and draw	
RAD	Graphic view object state Transition Diagrams	Flow of individual Roles	Detailed view Degree of Empowerment No overview	Supports Communication Intuitive to read	Not possible to be decomposed	Include business objects	Different Notations
RID	Matrix representation of processes for co-ordination of Activities	Flows of activities And roles	Inputs to and outputs from are not modeled Performers are Included	Intuitive to Understand	Important Information is not included	Rigid notation Complex processes can be displayed	Difficult to edit an existing diagram Hard to construct
Gantt Chart	Matrix Representation	Flow of activities And duration	Relate Activities to time	Easy overview Representation and control of Performance	Not aid for analysis or Design	Simple	No clear representation of Dependencies
IDEF0	Structural Graphical representation, text and glossary	Flows of activities, Inputs, outputs, control and mechanisms	Based on SADT Sub-layers The most Popular	Shows inputs, outputs, control and mechanisms Overview and Details	Trend to be Interpreted only as a sequence of activities Roles are not represented	Strict rules Possible to build a software Quick mapping	
Object Oriented Methods	Describe a system with different type of objects	Object's structure and behavior	Three concepts: objects, classes and messages There are many Modeling techniques	Enactable model to control and monitor processes	Model are excessively large and detailed Fragmented Information	Internal consistency across design, analysis and programming Possible to build	Need lot of data Time consuming when modeling Complexity

Table 11: Prominent business process modeling techniques

3.6 Summary

Our literature survey has been focused on product lifecycle management system. As expressed in the beginning of the chapter, there is only a few publications available in PLM field. However, there have been considerable amount of research present for product development, product data management systems, STEP, business process modeling, business intelligence, business rules etc. The author reviewed ideas and concepts of multiple areas and compiled a set of user-requirements with the help of these concepts, ideas and guidelines. The architecture of OPLM using these reviews is presented thereafter.

Chapter 4

Requirements Analysis

4.1 Introduction

User requirements of OPLM system are outlined in this chapter. The set of user requirements presented in this chapter has been gathered using three major sources, namely existing product study, study of research papers and using the results of the survey. There has been no published work that describes a complete set of PLM user-requirements.

4.1.1 Survey on Product Lifecycle Management (The complete survey, including results and analysis are included in the appendix)

A survey was conducted in January and February 2005 and distributed it to companies using two sources: Materials and Manufacturing Ontario (MMO)'s monthly bulletin and '2PLM' publication, a web-based PLM information bulletin. An Internet link to the survey is available at Professor F. Salustri's website [45]. By itself, the survey was appealing to companies with design and manufacturing operations. It was expected that the respondents were mainly the SMEs in Ontario. A total of 20 responses were gathered. Major findings of the survey were:

- PLM software could benefit directly to daily engineering activities of a company
- Design related issues are more significant than manufacturing issues
- Conceptual design is the area that requires highest improvement
- There is very little automation of Engineering Change Management and Revision Control in the companies surveyed
- Robust data management facilities are absent among the companies surveyed
- Multiple CAD format and data exchange is a major issue

- High costs and lack of control are the two major barriers for purchasing PLM solutions
- Overwhelming majority of the companies want to participate in building an Open source PLM software

4.1.2 Discussion

The survey had a few limitations. Firstly, there was no mechanism used to track information of a respondent, such as name, name of the employer, location of employer or respondent, age, profession, expertise, interests etc. No demographic information on the respondents could be collected. This was because of the nature of survey. A 'free' survey software was used that offered limited services. Secondly, the number of respondents was very low as compared to the target population. Yet, the number of the respondents can be well justified if the nature of this survey is considered. Thirdly, Some of the questions required a generalized answer rather than a specific answer. This was intentionally done in order to attract more respondents.

There are few very important aspects of the respondents that emerge from the survey. When it comes to gain complete control of a product's lifecycle, companies exhibit lack of software resources and automation. The reason is not that there are no solutions to their problems, rather, it is related to cost, size of infrastructure, control and vendor dependency. The survey clearly indicates the necessity of an Open source PLM that would enable companies to get hold of technology and tools that would highly contribute to their growth of productivity and profitability.

4.2 Requirements capture and analysis

In the first chapter, a perspective of existing PLM products and their attributes are presented. A number of research papers are consulted; an overview of those has been presented in the literature review chapter. In the light of all these sources, following user-requirements for OPLM are presented hereafter.

In the beginning of the requirements list, an outline a list of 'general requirements of OPLM' are presented (Table 12).

No	General Requirement
1	PLM must be built on platform that support all prevailing OS
2	PLM must be web-enabled, distributed access
3	Database must be open source compatible
4	PLM must minimize upfront software procurement costs
5	The GUI of PLM must be intuitive and easy to use
6	PLM must be able to show real-time data and most current information
7	PLM must integrate all information related to each and every product in an organization
8	PLM must automate engineering change management activities
9	PLM must be flexible and tunable to the need of an organization
10	PLM must facilitate the decision making process of an organization by providing required information and analysis
11	PLM must ensure data-interoperability between different vendor-software using STEP and PIF
12	PLM must be able to create, measure and control business processes (OPLM is focused on Engineering side of PLM. Business process creation and engineering is itself a vast area and beyond the scope of this thesis).

Table 12: Set of high-level user requirements of OPLM system

In order to outline the requirements, three categories are used, namely, 'product', 'lifecycle', and 'management'. In each the phases of a product's life, these are further categorized. The categories for product are 'structure', 'engineering change' and visualization (Table 13); for lifecycle phase, 'product development', 'first sale', 'growth' and 'maturity and decline' (Table 15) and finally for management requirements, they are 'monitoring', 'controlling', 'reporting' and 'improving' (Table 16).

Product Structure	Engineering change	Visualization
BOM Overview	Electronic engineering change orders (ECOs)	Navigator Functions
Define Standard Operations	Electronic manufacturing change orders (MCOs)	Log On for the First Time
Define a Routing	Permanent change history	Toolbar, Menus, and Hot Keys
Define a Bill of Material	Electronic engineering change requests (ECRs)	Search for Data
Copy Bills of Material	ECO, MCO & ECR file attachments	The List of Values
Nested BOMs, Flattened BOMs, Redline BOMs, Sourcing BOMs	Deviations	Place Documents on the Navigator
BOM import & export	Engineering Mass Changes	Pictorial views
Item master import & export	Assembly subscriptions & automated notifications	Printable views
Item numbers	Version control	"Thumbnail" image attachments
Item descriptions	Retention of all historical versions of every item	Visibility control
File attachments	Change history	2D/3D visualization
BOM exports in PDX format	Visibility into future changes	Visual CAD modification
Manufacturing BOMs	Item duplication with history	
Multi-level, before-and-after redline BOMs	Redline AML view	
Purchasing BOMs	Automated requests for review	
PDX import & export	Visibility into reasons for change	
Customizable export formats	Visibility into change originator	

Table 13: Set of requirements for product issues

4.2.1 Bill of Materials (BoM)

Bill of Materials is an integral part of product information. It is used to define the relationship between a manufactured item and its components and materials, as well as the sequence of operations required to produce the item, facilitating an accurate expected cost against which to track actual production costs [44]. Manufacturing companies rely on BoM heavily to maintain the product structure. It has been identified from our analysis that a robust BoM system is required in OPLM. BoM is invariable used by manufacturing companies to maintain product information, assembly/manufacturing information, cost, materials, parts, machines etc. An accurate, updated and most recent version of BoM is what

SMEs need. Therefore, a set of user-requirements is compiled for BoM as shown in Table 14.

Requirements	Descriptions/formats
BOM import & export	CAD packages usually have the capability to export BOM (called engineering BOM) as a spreadsheet in cvs format. So, BOM import can be readable by OPLM. Exporting BOM to other systems should be even easier for OPLM because it will use similar formats.
BOM comparisons	BOM of different products or different revisions of a product are often required. OPLM must be able to compare BOM's and display the results.
Prototype & production cost roll-ups	BOM Cost Rollup is done when an item changes and the costing is done ,eflecting the change. Cost roll up is also used to calculate and update the Fixed Cost (standard cost) of single or multi-level manufactured items.
Percentage cost contribution by item	This can be done using a reference cost vs. the item cost
BOM redlining/annotation	Redline/annotation is a tool that present in OPLM
BOM & item sharing with suppliers & partners	BOM is often shared with outside companies (less critical). More critical part is to read supplier BOM that has different part numbers.
BOM exports in PDX format	BOM that is contained in OPLM can be exported in PDX format
Manufacturing BOMs	Engineering BOM is tranformed to manufacturing BOM by the production department. This BOM has information like matrials, inventory, labor cost, machine cost, machine operation schedule etc.
Multi-level, before-and-after redline BOMs	BOM must show the breakdown of assemblies in many levels. Usually 10 levels are good for showing a product tree
Purchasing BOMs	BOM prepared by the purchasing department using the engineering BOM as well as the manufacturing BOM. This contains material inventory related to parts, order time, inventory level etc.
Synchronization of BOMs (including quantities, units of measure & effective dates)	All BOMs must be synchronized and no information should be disputable. All information must be correct.
CSV import & export (items, BOMs, AML, AVL, quotes & purchasing history)	CVS is an open format, existing CAD/ERP/MRP usually export BOMs in CVS format. Similar to microsoft excel.

Table 14: Set of requirements for BOM issues

Lifecycle			
Product Development	First sale	Growth	Maturity &Decline
Feasibility study	Sales management	Product support information	Maintenance document
Requirements specification	Price management	Maintenance information	Market research
Concept design	Order management	Feedback	Feedback to design department
Systems design	Customer management	Routing feedback to product development and manufacturing	Disposal management
Product architecture	Price information	Inventory control	Recycling
Design document	Customer information	- Material disposition	
Manufacturing document		Sales management	
Equipment information		Price management	
Plant layout		Order management	
Plant capacity		Customer management	
Purchasing		Price information	
Item costing		Customer information	
Item subscriptions			
Supplier contact information			
Supplier-input quotations			
Simulation for factory layout, ergonomics, material routing etc.			

Table 15: Set of requirements for lifecycle issues

Management			
Monitoring	Controlling	Reporting	Improving
Quality management	Prototype & production cost roll-ups	Visibility into supplier items used in BOMs	Associating a single file to any number of other objects like items, suppliers, etc.
BOM comparisons, redlines & other reports	Customized access by supplier	Electronic sign-off	Integration with PDM/ERP/MRP systems
Ability to view logged-in users & suppliers with access	Ability to hide costs from suppliers		Integration to FEM, CFD etc. packages
Configurable supplier access and submission with auto-routing	Version control	Support for any number of issues raised per request	Direct integration to multiple CAD/PDM systems like Pro-E, CATIA, Solidworks etc.
User-defined approval boards	BOM & item sharing with suppliers & partners		
User-defined approval requirements by role	*Partner participation in ECOs*	*Reporting on affected items & assemblies*	*User-defined import & export templates*
Sign-off status visibility	Visibility control	Quote history	Downloadable data archives
Automated logging of recent activity	Approved Manufacturer List (AML)	*Search by affected items & assemblies*	
Logging of all file related activity	*Multi-stage approval routing (manual or automatic)*	*Purchase history*	
Approved vendors & potential vendors	External partner participation	Report runs stored in history	
Electronic manufacturing change orders (MCOs)	Configurable workflow for evaluating and resolving issues or (change) requests	Powerful search across all request attributes	
Electronic engineering change orders (ECOs)	One designated administrator for each request	Configuration of report *templates*	
Assembly subscriptions & automated notifications	Support for customizable request, deferral and resolution codes	Report sharing across different roles	
Authentication via electronic signatures for recommendation/response and administrative actions	Locking and viewing the complete history of a *file*	Support for run-time setting of reporting variables	
Automatic notifications based on per-user participation level	File check in / check out		
	Multiple output formats		
	Maintaining consistent linking between a new edition of a file and the item working version		
	Synchronization of item master list (including material disposition & effective dates)		
	Synchronization of BOMs (including quantities, reference designators, units of measure & effective dates)		
	Synchronization of sourcing information (including approved manufacturers and vendors & supplier part numbers)		
	Synchronization of target, standard & current costs		

Table 16: Set of requirements for management issues

4.3 Summary

The set of user requirements presented in this chapter are suitable for SMEs. Multiple resources (existing product review, literature review and survey) of information have been used to compile the requirements. Consideration was given to the complete lifecycle of a product and aspects related to management of a product. Ancillary applications like simulation, CAD modeling, engineering analysis are not a part of OPLM package, yet many vendors provide these tools with PLM to facilitate operations of a company. Our goal is to satisfy the needs of SMEs. The requirements presented in this chapter are believed to be a true representation of the requirements of SMEs. However, the requirements have not been validated with representatives from SMEs or not been presented in a research paper. As our methodology for OPLM design showed (in chapter 2), requirements gathering and design are performed simultaneously. So, once the project is being developed at open source environment, it is expected that the requirements will undergo modifications to suit the constraints of open source developers. What we present here is a guideline for the developers to make the development of OPLM faster. As programmers and representatives from SMEs work together in OPLM project, the user-requirements presented here is likely to change.

Chapter 5

Engineering Change Management

5.1 Introduction

In the course of a lifecycle of a product, it goes through many cycles of changes. Product design process may include design problem acquisition, requirements engineering, concept design and evaluation, systems design and evaluation [45]. Later the product is manufactured, marketed, supported and scraped/recycled. In each step of this lifecycle, changes occur. Change is an integral part of product development. In this chapter we will discuss version control, ECM models, and finally propose our model for engineering change management.

5.2 Version Control

Version control can be defined as 'Keeping the history of changes to a set of files or directories'. A repository keeps track of each and every change(s) made to a file or directory in such a way that it can generate its previous state in time. Version control is an aspect of documentation control wherein changes to documents are identified by incrementing an associated number or letter code, termed the "revision level", or simply "revision". It has been a standard practice in the maintenance of engineering drawings for as long as the generation of such drawings has been formalized. A simple form of revision control, for example, has the initial issue of a drawing assigned the revision level "A". When the first change is made, the revision level is changed to "B" and so on [2].

In OPLM, version control is required for product data management. Version control falls into engineering change management. In this section, we have discussed about the features and advantages of version control. We have also performed a comparative analysis of available version control tools. This comparative study gives us the clue to get a suitable VC tool for OPLM. It can be mentioned here that the best VC tools are developed by open source developers. Some of the VC tools undergo continuous improvement and modifications to suit the needs of developers. A list of open source VC tools are provided at the end of this section for reference.

5.2.1 Features of version control software

1. Unreserved vs. Exclusive check-out model: Exclusive check-out gives sole right to the person who first check a file out and no one can access the file until the file is checked back in. On the other hand, unreserved check-out allows any user of a file to check it out at any point of time and then check it back in. there is no over-writing done, rather the changes are merged automatically if multiple users made changes to it.
2. Atomic commit: The change made to a single file changes the whole repository. Though it is time-consuimng , it allows to preserve the state of the repository at a point in time.
3. Merging files: When more than one developer checks in the same file, merging is done. This operation is automatic and if there is a conflict, then manual intervention is required.
4. Distributed repository: This means that a developer can copy the entire repository, make changes and then check in the repository. On the other hand, if the system is not distributed, only the access-holders are allowed to make changes to the repository.
5. Labels or tags: Labels or tags are used to mark two or more files in a tree. This is very useful when the development reached milestones or when an important build is made.
6. Renaming: If a file or directory is moved, renamed or copied elsewhere, the system allows its version history to carry forward.

5.2.2 Advantages of version control

1. Reversion: If a change to a file proves to be useless, the original file should be restored.

2. Tracking of changes: One can see the chronological change made to a file
3. Bug control: Bugs discovered at a later version can be eliminated by restoring the older version and apply changes to it

5.2.3 Available open source VC tools

- Source Code Control System (SCCS): Developed by Bell Labs in 1980 for UNIX. This is not open source, however, it has got a clone named CSSC (Compatibly Stupid Source Control) that is under development..
- Revision Control System (RCS): Started in 1982/83 at Purdue University. Most used revision control system for Unix world. It has functionality like SCCS but has a better interface
- CVS (Concurrent Versions System): Started in 1990's. Industry standard, client-server model. It allows multiple user check-in and check-out. It automatically merges changes and need manual help when there is a conflict.
- Aegis: This is an active project, GPL licensed. This transaction-based software is mainly used for source-code control.
- Subversion: New open source version control software designed to be the successor of CVS.
- BitKeeper: Proprietary version control tool, used by Linus Torvalds for LINUX kernel sources

	Aegis	*Arch*	*BitKeeper*	*CVS*	*OpenCM*	*RCS*	*Subversion*
Atomic commits	Yes	Yes	Yes	No	Yes	No	Yes
Renaming	Yes	Yes	Partial	No	Yes	Not-known	Yes
Copies	No	No	Yes	No	No	Not-known	Yes
Local replication of remote repository	Yes	Yes	Yes	No	No	No	Indirectly
Change-communication to multiple repository	Yes	Yes	Yes	No	No	Not-known	Yes
Bundling of relevant atomic changes in a group	Yes	Yes	Yes	No	Yes	Not-known	Partial
Line-by-line history for most recent changes (annotation)	Yes	Yes	Yes	Yes	Not known	Not-known	Partial
Check-out or check-in a single directory vs. the whole repository	Repository	Repository	Repository	Directory	Repository	Directory	Directory
Tracking uncommitted changes in a working copy	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Commit message for per file or per set basis	Unknown	Per Change	Both	Per Change	Unknown	Not known	Not present
Access to Online Documentation for users	Medium	Medium	Very Good	Excellent	Good	Medium	Very good
Deployment issue and dependencies	Medium	Excellent	Good	Excellent	Excellent	Good	Complicated
Networking issues	Poor, Only NFS	Excellent	Good, HTTP support	Client/server and SSH	Good, proprietary protocol		Excellent, HTTP, SSH, etc
Portability to different operating systems, computer architectures etc.	Good for UNIX, not Windows	Better for UNIX	Excellent for UNIX and Windows	Excellent: UNIX, Windows, MacOS	Good for UNIX	UNIX and Win95 NT	Excellent: UNIX, Windows, MacOS
Web Interface for browsing tree and versions of a file	Yes	Work-in progress	Yes	Yes	No	Yes	Yes
Availability of GUI's for different systems	tkaegis	ViewARCh(web-bsaed) and tlator	Several GUI's	WinCVS, Cervicia, ViewCVS (web-based), TortoiseCVS	Not present	Not known	RapidSVN ViewCVS, ToroitseSVN, Jsvn (Java)
License	GNU GPL	GNU GPL	Proprietary	GNU GPL	GNU GPL	GNU GPL	Apache/BSD

Table 17: Information and comparison of open source version control tools [46]

- OpenCM: This project was also designed to replace CVS and has many excellent features that make version control still more effective
- Arch: Arch is well known for distributed repositories and advanced merging and renaming capabilities
- Monotone: Known for distributed version control and its ability to maintain local transactional database
- SVK: This is a version control system written in Perl. It supports distributed control and uses Subversion library
- Vesta: Configuration management tool, it supports very large software projects.

5.2.4 Discussion:

For OPLM, a robust version control system is required. There are few issues pertinent to OPLM for version control such as web accessibility, multiple check-in/check-out, networkability, support for multiple operating system, documentation etc. It is obvious from the comparative analysis of the VC tools that Subversion is a good choice for OPLM VC. After looking into more details in Subversion, it has been found that Subversion provides excellent support for all of VC issues. That is why incorporation of Subversion in OPLM has been proposed. Right now, Subversion is available as only a 'command prompt' version control tool. For our use, it is proposed that it should be wrapped with a robust user interface so that all the required output can be generated for the user with little knowledge in Subversion. The output requirements for version control would be version history, customizing options for atomic or multiple check-in/check-out, assigning different files or formats for implementing different check-in options etc. Subversion development is one of the most active open source projects that is always adding new features to it. It is expected that the Subversion community would provide support for incorporating it in OPLM.

5.3 Introduction to Engineering Changes

Engineering Change (EC) control the process of building an item. Each stage of development undergoes some evaluation that gives rise to changes. EC assures that product is designed and manufactured properly. In manufacturing companies, Engineering Change Notices (ECN) plays a big role for change coordination among departments. ECN is initiated and passed for approval to concerned department. Later the change is implemented.

Here is a list of problems that occur with engineering changes:

1. Lengthy cycle time for review, revision and approval
2. Limited or no access to the most current and released controlled documents
3. Slow distribution of new and revised document
4. Costly and time-consuming manual tracking
5. Manual and unorganized electronic systems are difficult to keep compliant
6. Too many unnecessary changes caused by cross-functional misunderstandings
7. Too many well-intentioned changes made in later stages without consideration of the negative impact to the product's functionality, performance, quality, manufacturability, cost or schedule.
8. Change process may be bypassed in order to meet production demands that may result in valuable design history
9. Absence of a formal change management process creates lost information that results in loss of valuable time for tracking down and determining the history and impact of change

A manufacturer may be forced to shift focus and divert valuable resources away from areas of more critical need, which ultimately leads to wasted labor and missed delivery dates

5.3.1 Principles of EC

Two useful strategies are most used in case of change management:

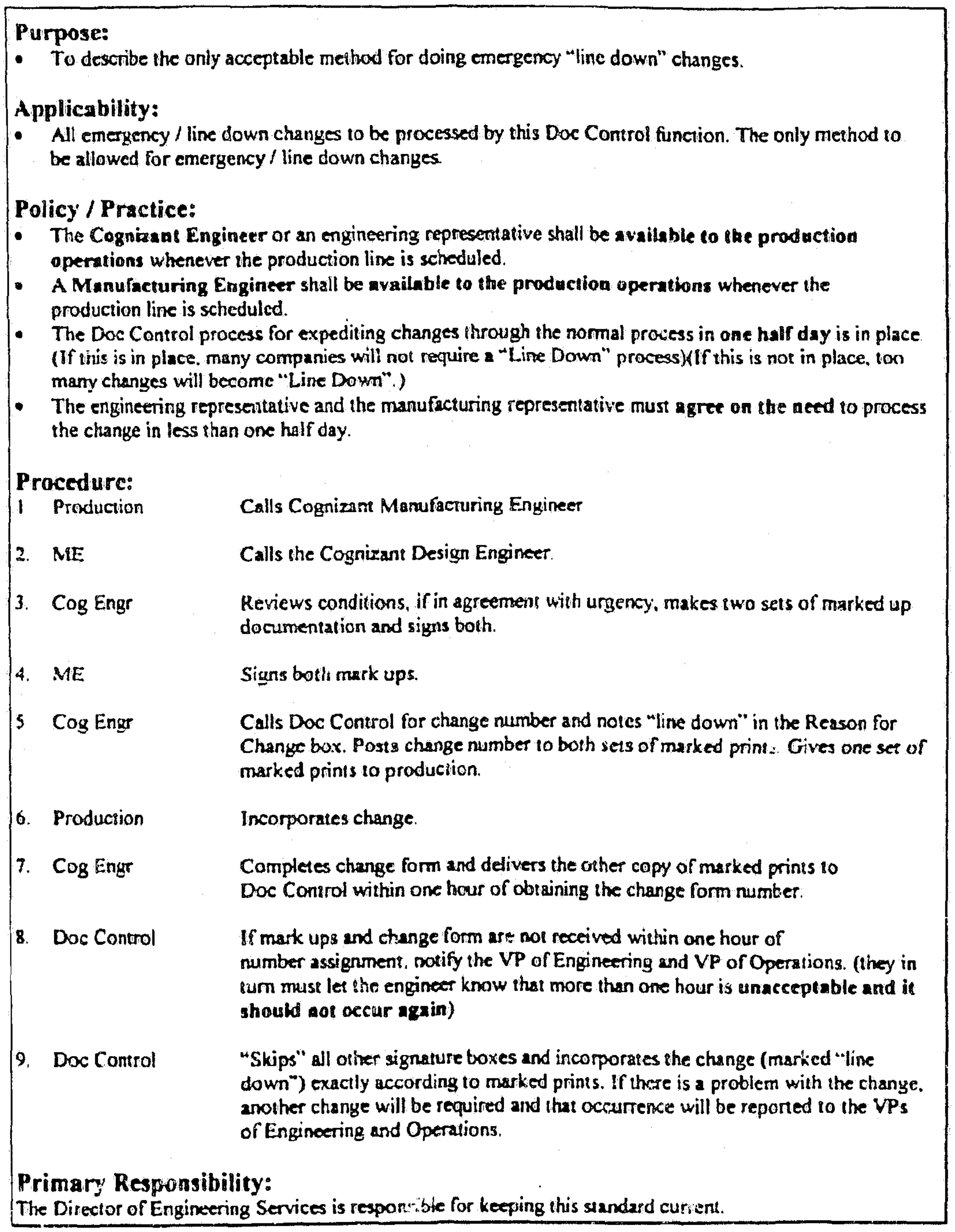
1. Minimize number of changes: A company can never avoid making changes. Change is a basic phenomenon of a business. In product development, manufacturing and maintenance related operations, engineering change occurs in product, plant and support information. A poorly developed product is likely to have more changes as it is designed, manufactured and supported. But a product that is developed using robust change management is likely to incur fewer changes. Changes are costly and often time consuming. So, the first principle of change management is to 'minimize changes'. There are few ways that companies can minimize making changes. Spending more engineering time during the first release of a product can significantly minimize changes in later stages. Concurrent engineering can also reduce number of changes as different interest groups for a product provide their input in the early stages of development, designers can incorporate attributes that would minimize changes later.
2. Make change as early as possible in the product lifecycle: Making changes in the early stages of product lifecycle is desirable. Changes during the design stage are easier to make and they are less costly. Making a change when the product is in product is costly in terms of man-hours, complexity, change cycles etc.

5.3.2 Cost of change:

Product design and development is a dynamic process. Typically, a product is developed based on a set of requirements. The concept design for the required product undergoes many changes in order to refine the design. The final design goes to the manufacturing department who creates the product using the information from the final design. Changes during concept design and subsequent steps are almost inherent to product development and may not be a source of costs (costs related to 'changes'). Monolithic product development scenarios where a relatively simple product is developed in a single company with in-house design and manufacturing support may find 'changes' easily manageable and less costly. But if a product is fairly complex, 'engineering changes' can be very costly. There has been very limited research on the cost of changes. An average change may cost $1,500 for a company [47]. This cost, however, is not the cost of actual change, rather the 'administrative cost' of a change. Administrative cost per change is calculated by dividing the budget for 'configuration management' department and some similar units by number of engineering changes handled by these units for the budgetary period. Industrial surveys have shown that EC is a serious problem in manufacturing [48]. Bonzak reported cost for administering EC in companies he surveyed (that includes small firms to fortune 500 companies) are US$ 3.4 million to US$ 7.7 million per year. Maul reported that costs related to ECs could be as much as 10% of a company's annual turnover [48]. It takes an average of 40 days to discover an EC, 40 days to process and approve and 40 days to implement it. EC activities also consume 33-50% of engineering capacity, 20-50% of tool costs and 20-40% of die development costs [47].

5.3.3 Models of ECM

There are many models proposed by researchers for change management or engineering change management [49]. Companies use customized models for their own change management. Figure 22 presents a model of a 'fast' EC process [50].



Purpose:

- To describe the only acceptable method for doing emergency "line down" changes.

Applicability:

- All emergency / line down changes to be processed by this Doc Control function. The only method to be allowed for emergency / line down changes.

Policy / Practice:

- The **Cognizant Engineer** or an engineering representative shall be **available to the production operations** whenever the production line is scheduled.
- A **Manufacturing Engineer** shall be **available to the production operations** whenever the production line is scheduled.
- The Doc Control process for expediting changes through the normal process in **one half day** is in place (If this is in place, many companies will not require a "Line Down" process)(If this is not in place, too many changes will become "Line Down".)
- The engineering representative and the manufacturing representative must **agree on the need** to process the change in less than one half day.

Procedure:

1	Production	Calls Cognizant Manufacturing Engineer
2.	ME	Calls the Cognizant Design Engineer.
3.	Cog Engr	Reviews conditions, if in agreement with urgency, makes two sets of marked up documentation and signs both.
4.	ME	Signs both mark ups.
5	Cog Engr	Calls Doc Control for change number and notes "line down" in the Reason for Change box. Posts change number to both sets of marked prints. Gives one set of marked prints to production.
6.	Production	Incorporates change.
7.	Cog Engr	Completes change form and delivers the other copy of marked prints to Doc Control within one hour of obtaining the change form number.
8.	Doc Control	If mark ups and change form are not received within one hour of number assignment, notify the VP of Engineering and VP of Operations. (they in turn must let the engineer know that more than one hour is **unacceptable and it should not occur again**)
9.	Doc Control	"Skips" all other signature boxes and incorporates the change (marked "line down") exactly according to marked prints. If there is a problem with the change, another change will be required and that occurrence will be reported to the VPs of Engineering and Operations.

Primary Responsibility:

The Director of Engineering Services is responsible for keeping this standard current.

Figure 22: An example of a 'fast' change process [47]

Figure 23 is another example of change management process. This model has been taken from [48]. This EC model depicts a detailed change process in

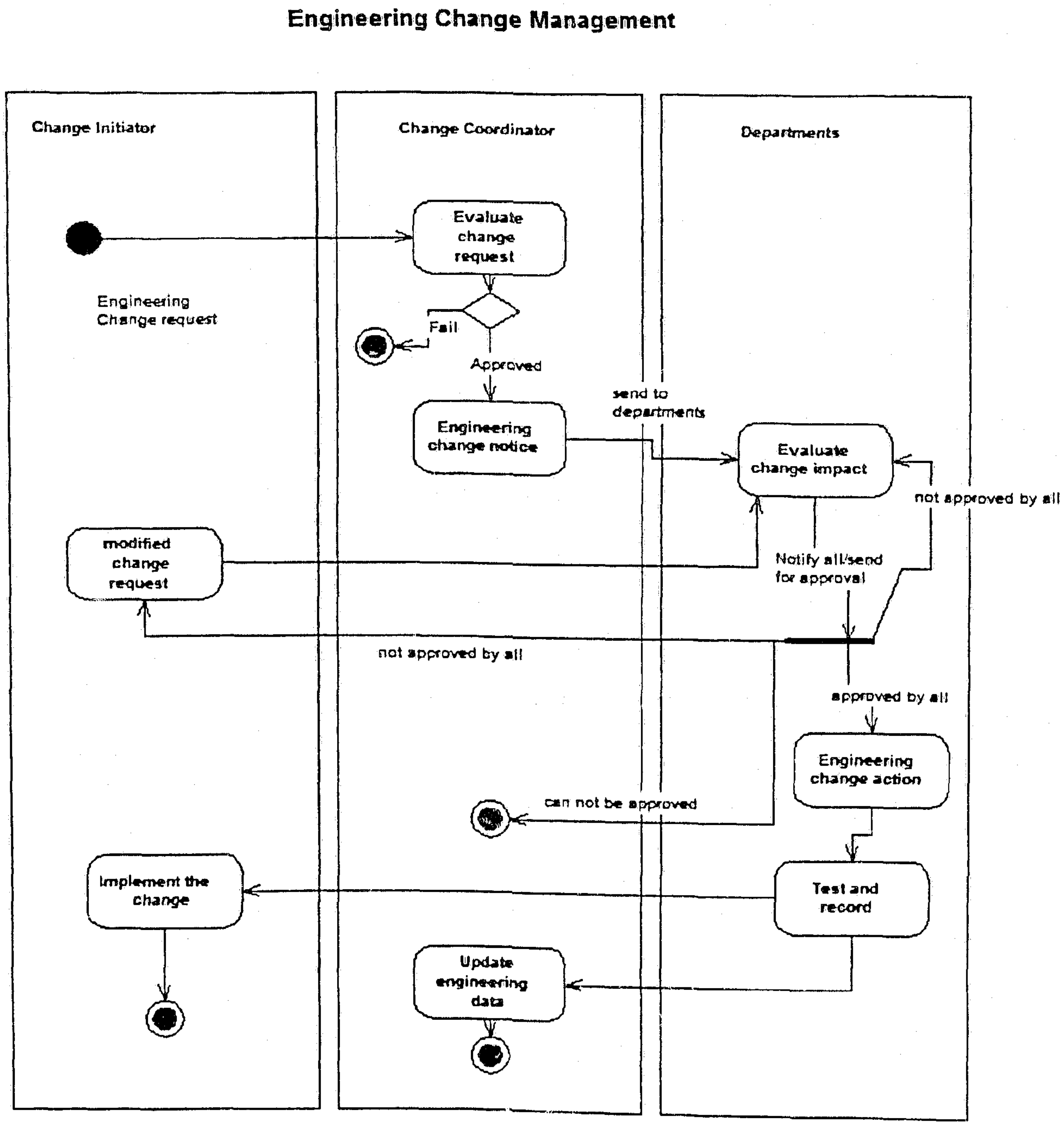


Figure 23: Activity diagram of ECM process adopted from [48]

an organization. The change initiator starts the change process by creating an Engineering Change Request (ECR). The ECR is submitted to Change Coordinator for evaluation. The Change Coordinator issues an Engineering Change Notice (ECN) that is sent to the department(s) that is considered to be affected by the change. The department(s) may make clarifications to the change initiator. Once approved by the departments, Engineering Change Action takes place. The change initiator implements change. The Change Coordinator updates engineering data so that it can be used in future. If the departments disapprove of the change, the change process stops.

Figure 24 is another example of EC process. In this case, a more practical version of change management [51] has been considered. This method is used with Windchill PLM package. When a change is identified, it is then investigated; cost and feasibility of the change are evaluated. For change disposal, two different approaches are taken: Low-impact and high-impact change.

In low-impact changes, the decision for the change can be taken by the line supervisor or line managers. In case of high-impact changes, a Change Implementation body id formed who evaluate the change, find a solution and approve it. Once approved by the line manager or the change implementation board, the change is implemented. Later an audit of the change is done for future reference.

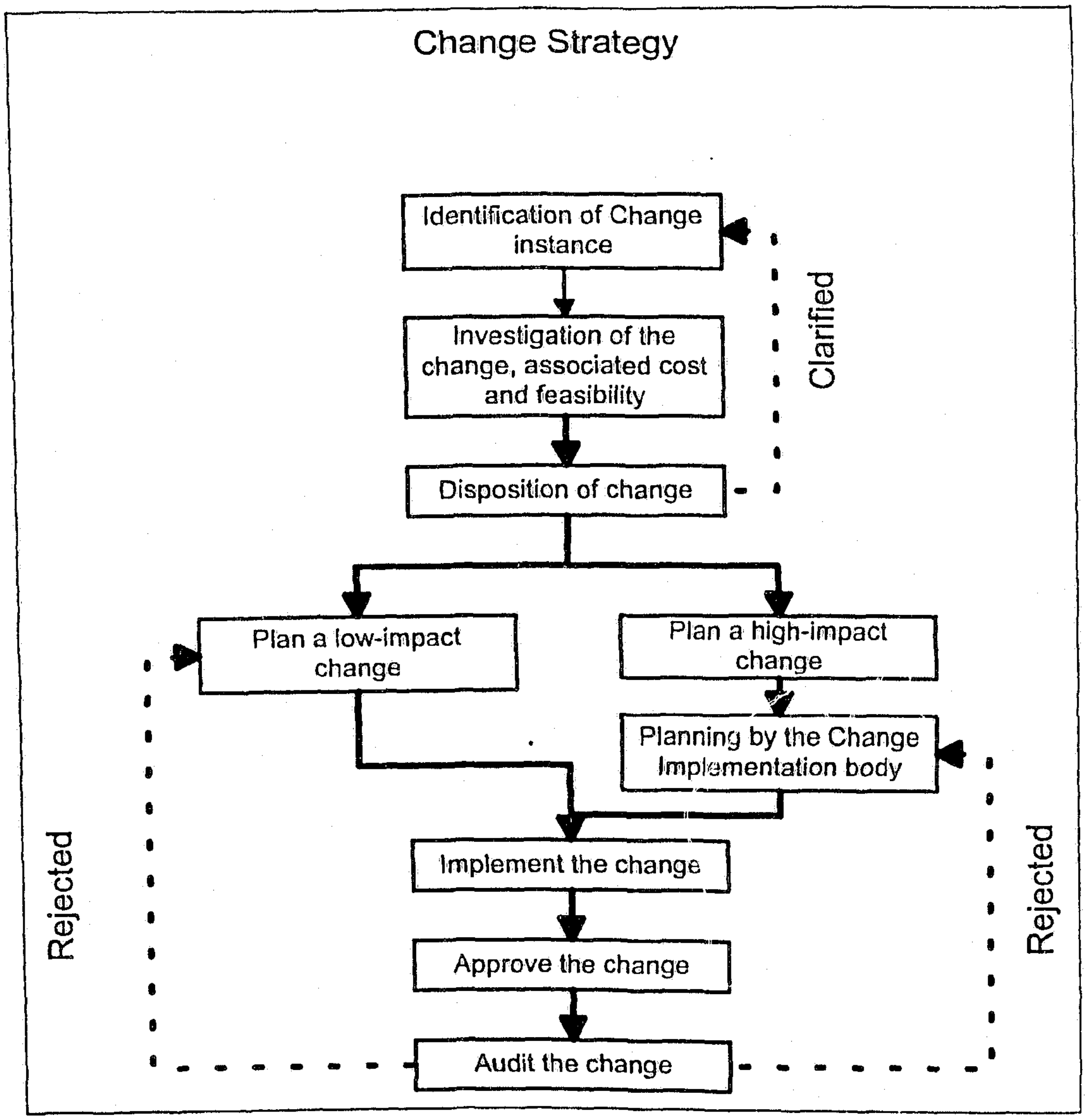

Change Strategy
Identification of Change instance
Investigation of the change, associated cost and feasibility
Disposition of change
Clarified
Plan a low-impact change
Plan a high-impact change
Planning by the Change Implementation body
Implement the change
Approve the change
Audit the change
Rejected
Rejected

Figure 24: Nine steps of change strategy adopted from [51]

EC models discussed above has the following limitations:

1. Involvement of manual steps for documentation, communication, approval etc. are often time consuming and error prone.
2. Labeling a change as 'fast', high-impact, low-impact is often subjective decision and may not reflect actual scope of change. Any change that is mishandled may give rise to subsequent changes.
3. These models do not show the flow of solution of proposed change. Absence of the track-record of solutions is not desirable because such int ɔrmation can be reusable in future changes in similar situations.

4. These models does not have provision for reuse of change information

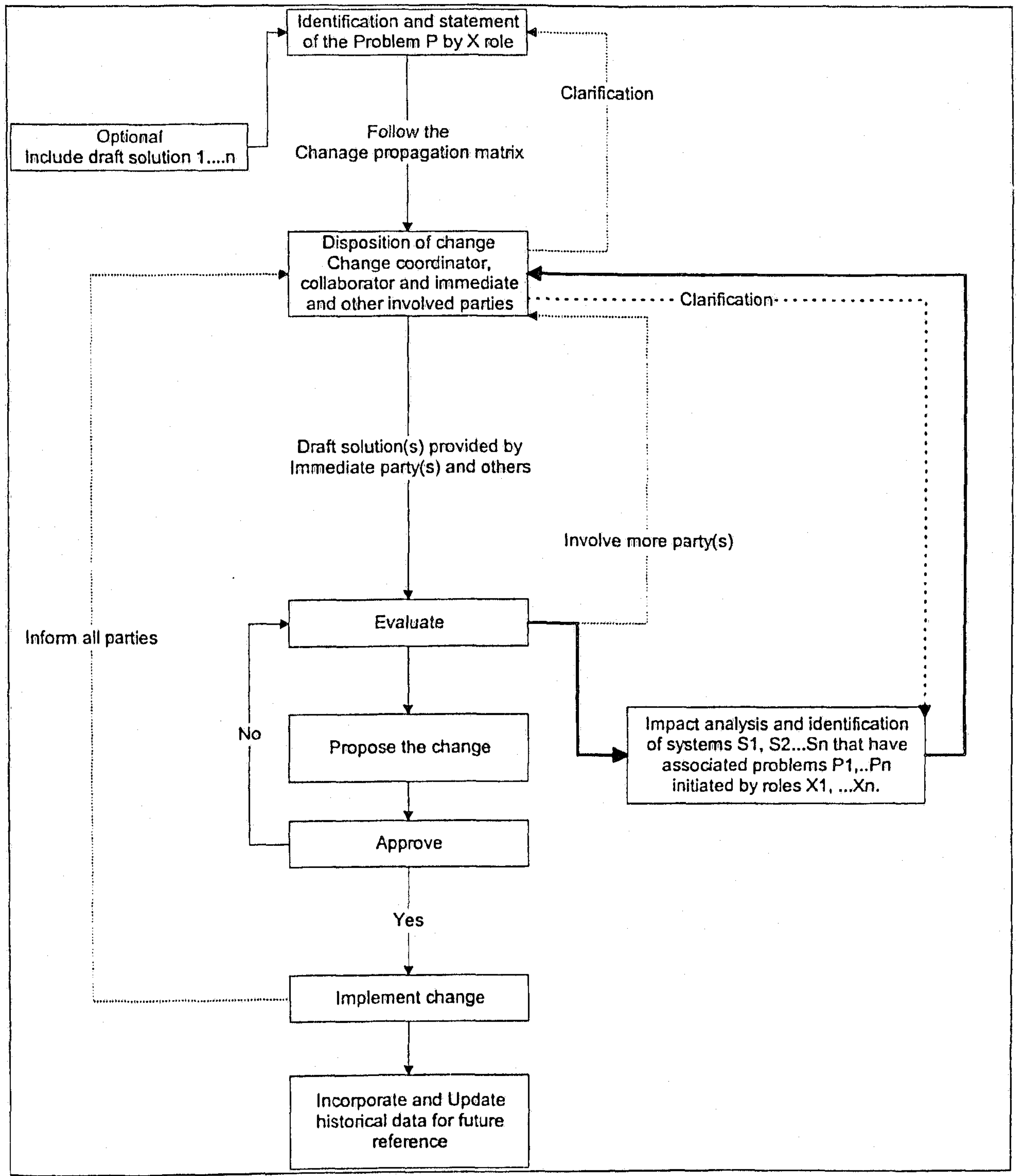

Identification and statement of the Problem P by X role
Clarification
Optional
Include draft solution 1....n
Follow the
Chanage propagation matrix
Disposition of change
Change coordinator, collaborator and immediate and other involved parties
Clarification
Draft solution(s) provided by
Immediate party(s) and others
Involve more party(s)
Inform all parties
Evaluate
No
Propose the change
Impact analysis and identification of systems S1, S2...Sn that have associated problems P1,..Pn initiated by roles X1, ...Xn.
Approve
Yes
Implement change
Incorporate and Update historical data for future reference

Figure 25: Proposed Modified ECM process

5.3.4 The Modified ECM Model:

It is apparent that companies should have an EC process that can be used by all and moreover, that EC provides a model where companies can benefit from changes. We are proposing an ECM model (Figure 25) that overcomes the limitations of currently available change management procedures. Following are the steps of the modified model:

1. Identification of problem: A problem in a system is identified by a role in the organization. We denote the problem by P and the role by X. X may or may not include a 'draft solution' to the problem P.
2. Disposition of the Change: The change is disposed to the Change Coordinator, collaborator and all other related parties. The members of related parties are determined by using the 'Change Propagation Matrix' 9discussed in the following section). Change propagation matrix is a similar method of identifying relationships to systems. We have followed the principles of System Identifying Matrix [52] to create change propagation matrix. Table 18 shows an example of this. The members of the change team may ask for clarifications on the problem from role X.
3. Evaluation: The problem P is evaluated at this step. All the drafts of the solutions provided by related parties are evaluated at this stage. In some cases, newer party(s) can be included if such is deemed necessary by the Change Coordinator. At the same time, an Impact Analysis is done to identify what other system(s) is affected by the change.
4. Propose the change: The change is proposed to the team at this step. This proposal contains all related change proposals that arise from the initial change proposal.
5. Approve the change: The change proposal is sent for approval from all affected parties. If not approved, the proposal is sent again for evaluation.
6. Implement the change: Upon approval, all change initiators implement the change. This process creates a loop when further changes are required

that might arise from an initial change. As all parties are notified about the change, further investigation may give way for new change proposals.

7. Incorporate and update information: The change process and occurrence for all change proposals are documented and updated. All the relationships, solutions and interaction information are updated. This information is utilized in future.

5.3.5 Change Propagation Matrix:

Change propagation matrix (CPM) is a tabular form of product information to show the relationship among parts, subassemblies or assemblies. This matrix is similar to the concept of System Identification Matrix discussed in [20]. We use CPM for implementing our Modified ECM Model. Table 18 shows an example of a CPM for a pencil assembly. A pencil has a body, upper cap, lower cap and a core assembly. The core assembly has a head, head hold, barrel and a spring.

			1	2	3	4	5	6	7
	Core Assembly								
1		Head		X	X				
2		Barrel	X		X	X	X		
3		Head hold	X	X		X			X
4		Spring			X		X	X	
5	Pencil body			X				X	X
6	Upper cap			X			X		
7	Lower cap			X		X	X		

Table 18: Example of 'change propagation matrix' of a pencil assembly

Each part is compared with other parts and if there is a relationship between two parts, a 'X' is placed in the box. When a change proposal is created for 'lower cap', the change coordinator identifies that 'spring' and 'pencil body' may be affected by the change. So, persons responsible for 'spring', and 'pencil body' are notified for the change.

5.3.6 Advantages of the Modified ECM Model:

1. Our model proposes a faster and more effective ECM: Change process has two challenges: it must be fast and accurate or effective. The CPM proposed here is created in the early stage of a product's lifecycle. Very limited overhead is used for creating the CPM. The CPM is continuously updated as a product is developed. This ensures that our model would be more effective in handling changes.
2. Our model considers change not as an isolated one, rather a complete one. The SIM table is employed to identify all affected party or parties. At a later stage of the ECM, all ECs related to the initial one are identified. Such association makes the change related database rich and can used for future ECM. When a change in initiated, all parties that are likely to be affected are involved. If any change is made to the product other than the initial change proposal, that is also incorporated in the change. In this way, we gather the intelligence of a particular change relative to its effect on the product as a whole.
3. Our model can be fully automated: The role of the change coordinator can be automated. The process of sending messages, approval of solutions and storage of all related information can be automated too. In some process, there is too much human overhead involved that cannot otherwise be automated and more so, those models are error prone.
4. Our model keeps a track record of the solution of change. Such information can be used in future changes that can make the change process faster.
5. Even though this process may involve more work in the early days of implementation, we expect that as the change data grows and reusability becomes frequent, EC time and cost will be significantly reduced.

5.3.7 Limitations of the Modified ECM Model:

1. This is a hypothetical model. Therefore, it needs to be tested for validation.

2. The Modified ECM Model automates EC process. Implementing this process manually can be more time-consuming.
3. During the implementation of this model, it is expected that engineers will find this process more time-consuming than usual. This can be frustrating.
4. This model may require large-scale organizational changes and business process modifications. This may create negative effect on implementation of this model.

5.4 Discussions

ECM plays very important role in product development. The parameter based ECM concept becomes cumbersome as the number of parts of a product increases. In parameter based ECM, a matrix is created for showing the relationships between each parameter. Another matrix is created to identify the relationships of the parameters to each part. When a change proposal is created, first the parameter relationship table is consulted and later roles are involved using the parameter-parts table. Number of parameters of a product is always more than number of individual parts. That is why, parameter based ECM is more time consuming. ECM model proposed in this research overcomes such drawbacks. This model takes a lifecycle view of ECs. Retaining a history of each EC, its relatioship to other ECs (parts/assemblies etc.) creates valuable information for engineers in product development, manufacturing and support.

Chapter 6

OPLM System Architecture

6.1 Introduction

Three topics are discussed in this chapter: identification of major OPLM subsystems using System Identification Matrix (SIM), OPLM Subsystem Visualization and OPLM system architecture. Each of these topics is necessary to building the architecture of OPLM. SIM is a process to identify the subsystems and their relationship to major Functior,al Requirement (FR)'s. Visualization exercises are performed in order to visually display the relationships and the boundaries of each subsystem and FR. Both SIM and visualization techniques are done in trial and error basis.

OPLM system architecture presents the major divisions of OPLM based on our current research. Product Data Management (PDM) architecture, Engineering Change Management architecture, Process Management architecture, Business Intelligence architecture are discussed. Finally, OPLM architecture has been laid out. In presentation of architectural views, only high-level components are shown. The interfaces are not clearly defined at this stage because of the limitation of information of each subsystem. However, the relationships that can be helpful to define the interfaces are identified.

6.2 OPLM Subsystems

PLM consists of multiple sub-systems. An exercise has been carried out to identify the subsystems and their association with major functions of OPLM. This process starts with building a matrix of the major functional requirements (FR) and their relationship to major subsystems. FRs and subsystems are identified using the results from the requirements (chapter 4) and literature review (chapter 3). Each subsystem in tables (19), (20) and (21) are defined in Appendix B. Rational for each check-mark in the tables are also given in Appendix B.

Each FR and sub-system is ranked and later they are re-distributed in the order of ranking. The final matrix provides a view of the FRs and subsystems that are arranged in the order of preference. In Table 19, we have first laid out the FRs in rows and subsystems in columns. If an FR has relationship with a subsystem, a check-mark is given in the box.

	Functional Requirements	UI subsystem	API subsystem	Security Subsystem	Revision control	Internet access	Information storage	Search subsystem	Information associativity	
1	System must access information	X	X	X		X	X			5
2	System must display product information that is graphical, textual, organized, structured and dynamic	X	X							2
3	The system must interface with PDM, CRM, ERP etc. to access product information		X	X					X	3
4	The system must maintain relationship in product information		X		X		X		X	4
6	The system must be able to access Internet			X		X				2
7	The system must be secure in terms of access control, usage, external interference			X		X				2
8	The system must track versions resulting from changes				X					1
9	The system must be able to interface with email/wiki/video/teleconference etc.			X			X			2
10	The system must require little training	X								1
12	The system must notify changes to the users		X							1
13	The system must be universally searchable			X		X		X		3
14	The system must be configurable at runtime	X	X	X			X			4
15	The system must support API's for new data types	X	X							2
		5	6	7	2	4	4	1	2	

Table 19: OPLM subsystem identification in a SIM table (exercise 1)

The next step of the exercise is to arrange the FR's in order of preference. In Table 20, the FR 'system must access information' is placed on top because this

FR has the most number of subsystems associated. Subsequent FR's are arranged accordingly.

	Functional Requirements	UI subsystem	API subsystem	Security Subsystem	Revision control	Internet access	Information storage	Search subsystem	Information associativity	
1	System must access information	X	X	X		X	X			5
4	The system must maintain relationship in product information		X		X		X		X	4
14	The system must be configurable at runtime	X	X	X			X			4
3	The system must interface with PDM, CRM, ERP etc. to access product information		X	X					X	3
13	The system must be universally searchable			X		X		X		3
2	System must display product information that is graphical, textual, organized, structured and dynamic	X	X							2
6	The system must be able to access Internet			X		X				2
7	The system must be secure in terms of access control, usage, external interference			X		X				2
9	The system must be able to interface with email/wiki/video/teleconference etc.			X			X			2
15	The system must support API's for new data types	X	X							2
10	The system must require little training	X								1
8	The system must track versions resulting from changes				X					1
12	The system must notify changes to the users		X							1
		5	6	7	2	4	4	1	2	

Table 20: OPLM subsystem identification in a SIM table (exercise 2)

In the final step, similar operation is performed on subsystems Table 21. It turns out that the 'security subsystem' is the most important subsystem for OPLM.

	Functional Requirements	Security Subsystem	API subsystem	UI subsystem	Internet access	Information storage	Revision control	Information associativity	Search subsystem	
1	System must access information	X	X	X	X	X				5
4	The system must maintain relationship in product information		X			X	X	X		4
14	The system must be configurable at runtime	X	X	X		X				4
3	The system must interface with PDM, CRM, ERP etc. to access product information	X	X					X		3
13	The system must be universally searchable	X			X				X	3
2	System must display product information that is graphical, textual, organized, structured and dynamic		X	X						2
6	The system must be able to access Internet	X			X					2
7	The system must be secure in terms of access control, usage, external interference	X			X					2
9	The system must be able to interface with email/wiki/video/teleconference etc.	X				X				2
15	The system must support API's for new data types		X	X						2
10	The system must require little training			X						1
8	The system must track versions resulting from changes						X			1
12	The system must notify changes to the users		X							1
		7	6	5	4	4	2	2	1	

Table 21: OPLM subsystem identification in a SIM table (exercise 3)

6.2.1 Discussions

The SIM exercise provides a very useful piece of information on the subsystems. As OPLM is designed for development using computer programs, SIM would play a very important role for developers. Developers can easily identify the subsystems, interfaces and boundaries using the SIM table. In addition to representation of subsystem, this exercise is helpful in visualization of OPLM system. The final table of the exercise (Table 21) has a greater density of cross-marks near the top-left area and less density in bottom-right area. Developers can identify any discrepancy immediately if such visual effect is presented.

6.3 OPLM Visualization Exercise:

OPLM visualization exercise is performed to present OPLM subsystems and FR's visually. The visual representation helps developers identify the complete system and its components and boundaries very easily.

Figure 26 is the first step of this exercise. Basic product template has been taken from [53]. We positioned the subsystems of OPLM on this template. OPLM project has a fundamental goal: to be developed by open source developers.

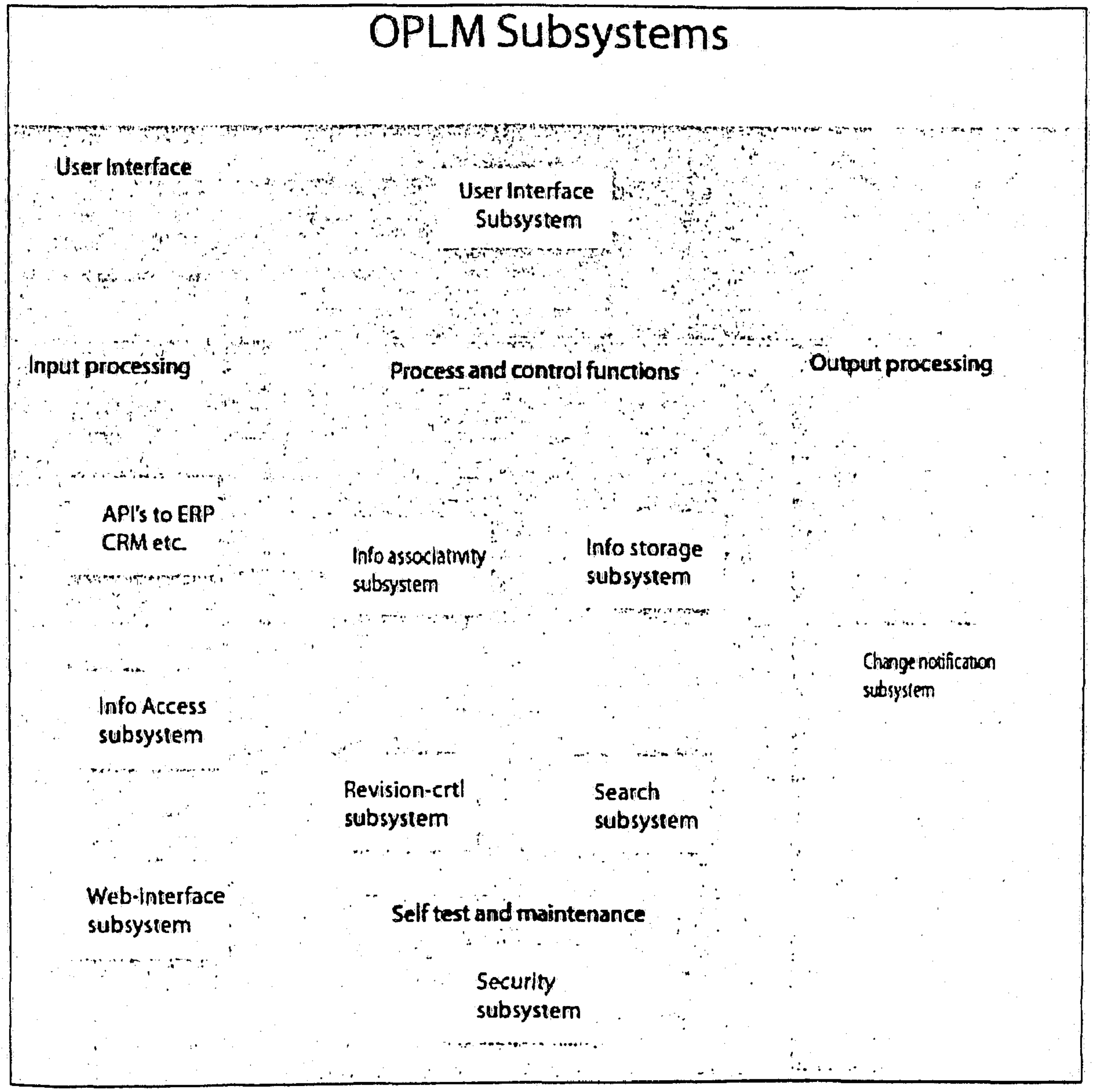


Figure 26: Basic OPLM architecture, using the product architecture template [53]

For this reason, the architecture has been presented in a form that is easily understood by developers. Clear Visualization is therefore necessary for the developers to understand the design. Figure 26 shows the basic 'Product Architecture' for OPLM. The intention of this figure is to show the sub-systems of OPLM and their interfaces.

This exercise is performed to determine the interaction of the sub-systems of OPLM. The visualization of the interaction is intended to be clean and easily recognizable using multiple colors. The visualization still needs to be refined. Adobe Illustrator has been used to produce this drawing.

Figure 27 shows a bipartite graph with functional requirements on one side and OPLM subsystems on the other. Connecting lines are drawn to represent relationships. The resulting graph becomes almost illegible because of the high number of connecting lines. The goal of this exercise is to create a visualization that would produce clear and legible visualization of OPLM subsystems and FR's. Therefore, the bipartite graph is discarded and further methods for visualization are explored.

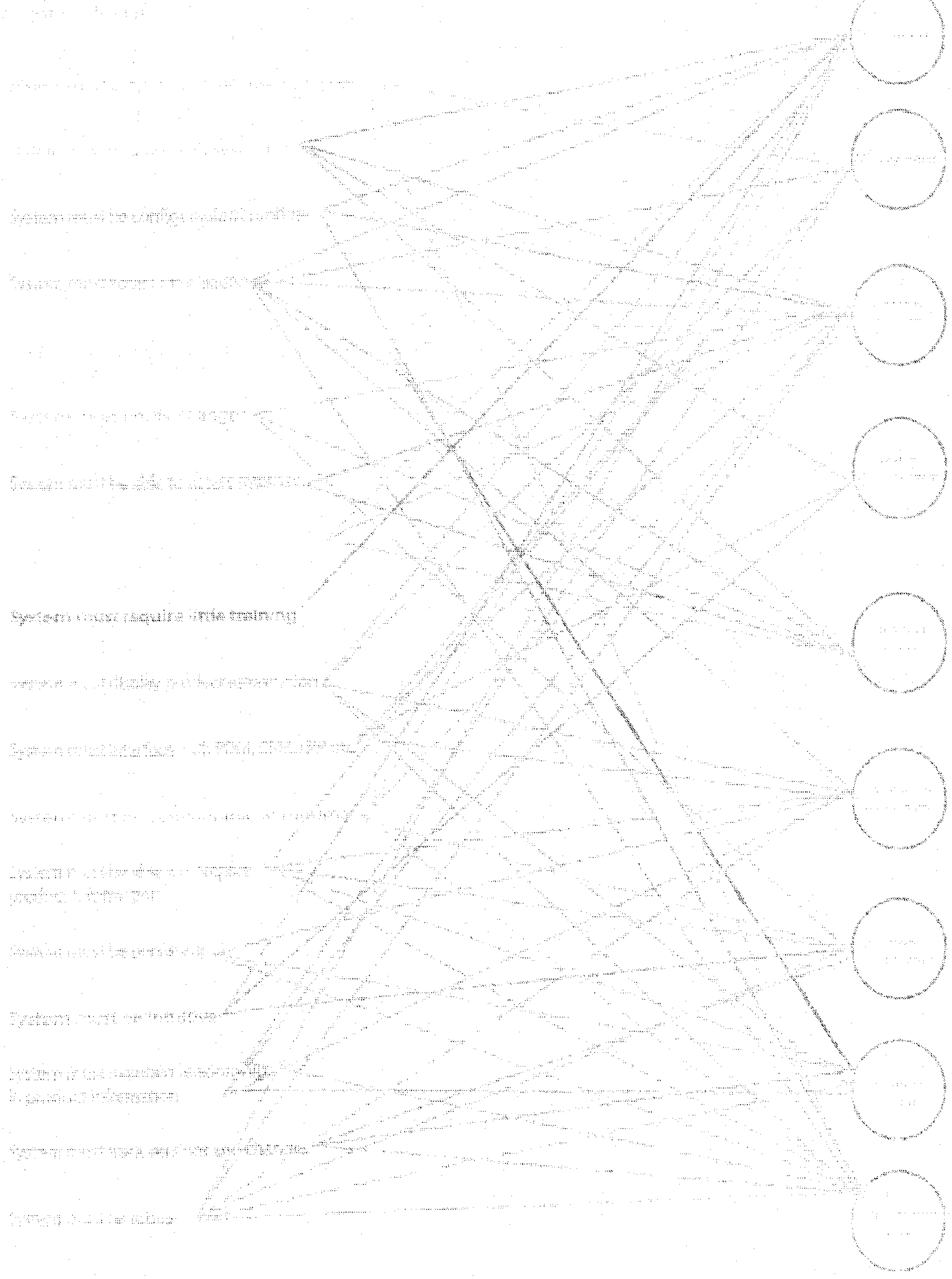


Figure 27 OPLM subsystems Visualization exercise 1 (bipartite graph)

Another version of subsystems and major functional requirements diagram are shown in Figure 28. The previous attempt of visualization has undergone a further refinement. In this case, multiple colors, shapes and transparency techniques are used for achieving a clearer view of the subsystems. The figure presented here is a final product of 'trial and error' process of similar graphs.

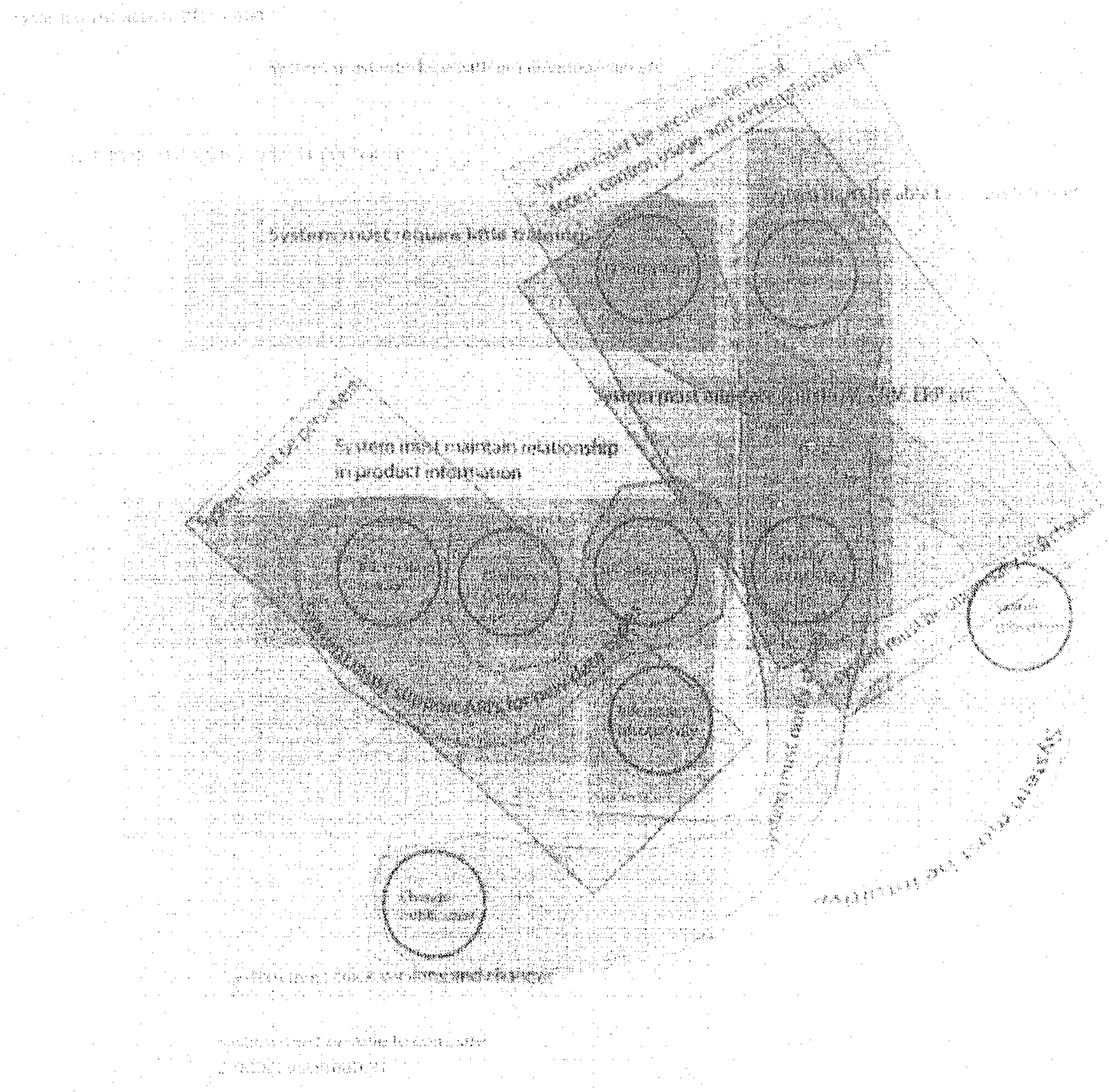

System must require little training
System must maintain relationship in product information

Figure 28. CPLM subsystems Visualization exercise 2

It has been recognized that the diagram needs further improvement in order to get a clearer view of the relationships. If is obviously difficult for one to grasp the overall picture of the FR's and sub-systems.

There are no known conventions for these kinds of visualizations. Open source projects succeed because of good communications. Therefore, it was important to investigate for new and interesting ways to visualize information. Versions of this exercise have been provided to show the iterations that would finally create a means for better visual communication.

Different software tools are used to create better visualization of OPLM subsystems. SmartDraw (a graphic software tool) is used to create Figure 29.

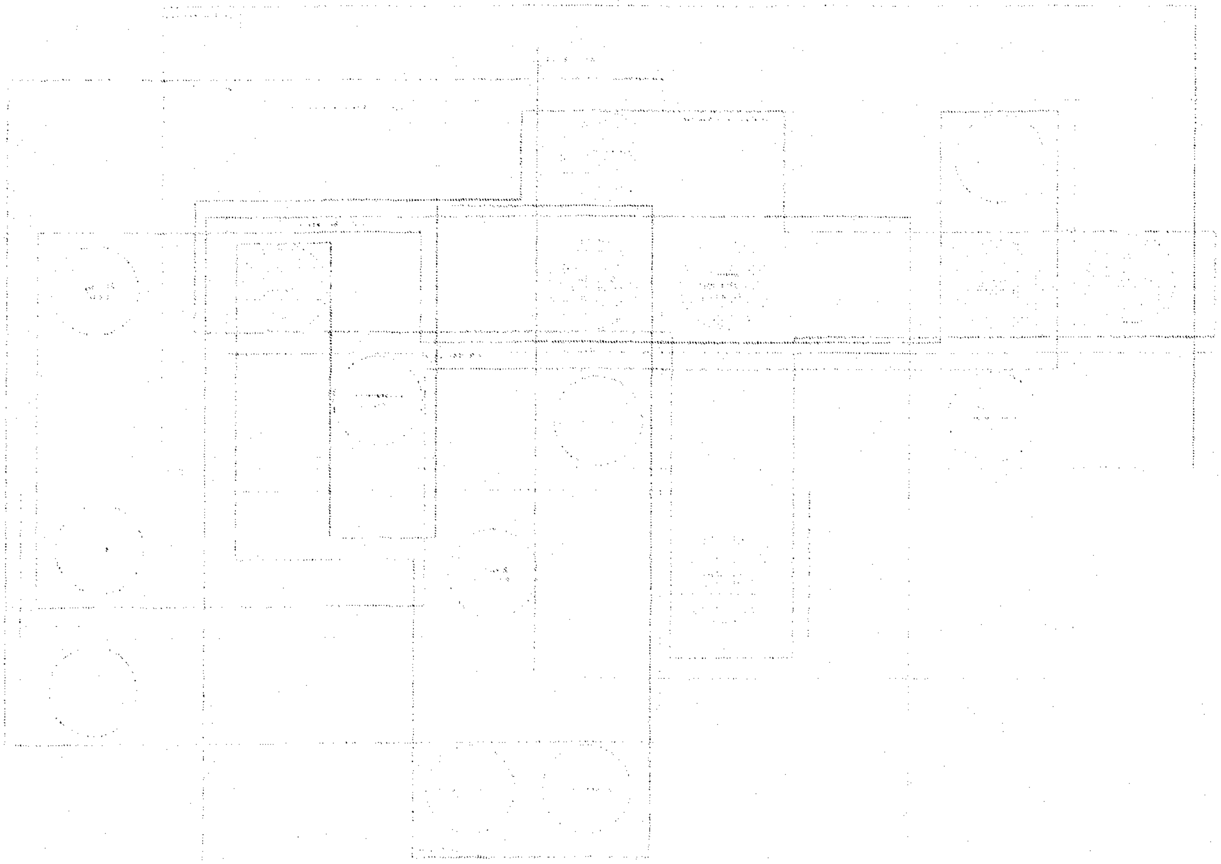


Figure 29: OPLM subsystems Visualization exercise 3

The red circles represent the subsystems that are more important over others. As a result of this exercise, it has been realized that the number of subsystems as well as number of major functional requirements needs to be reduced to produce a better view.

In the final step Figure 30, the number of subsystems is reduced from 17 to 12 and number of major FR's from 13 to 8. This is reached through step-by-step refinement of the systems. It can be seen that this diagram is created using a trial and error method. Each box and circle has a specific location to be visually effective to the developer. This location is selected using trail and error methods. In the final version of the exercise, the shape of each box is rectangular. Such clear view is expected to help developers catch the subsystems and FR's that are related.

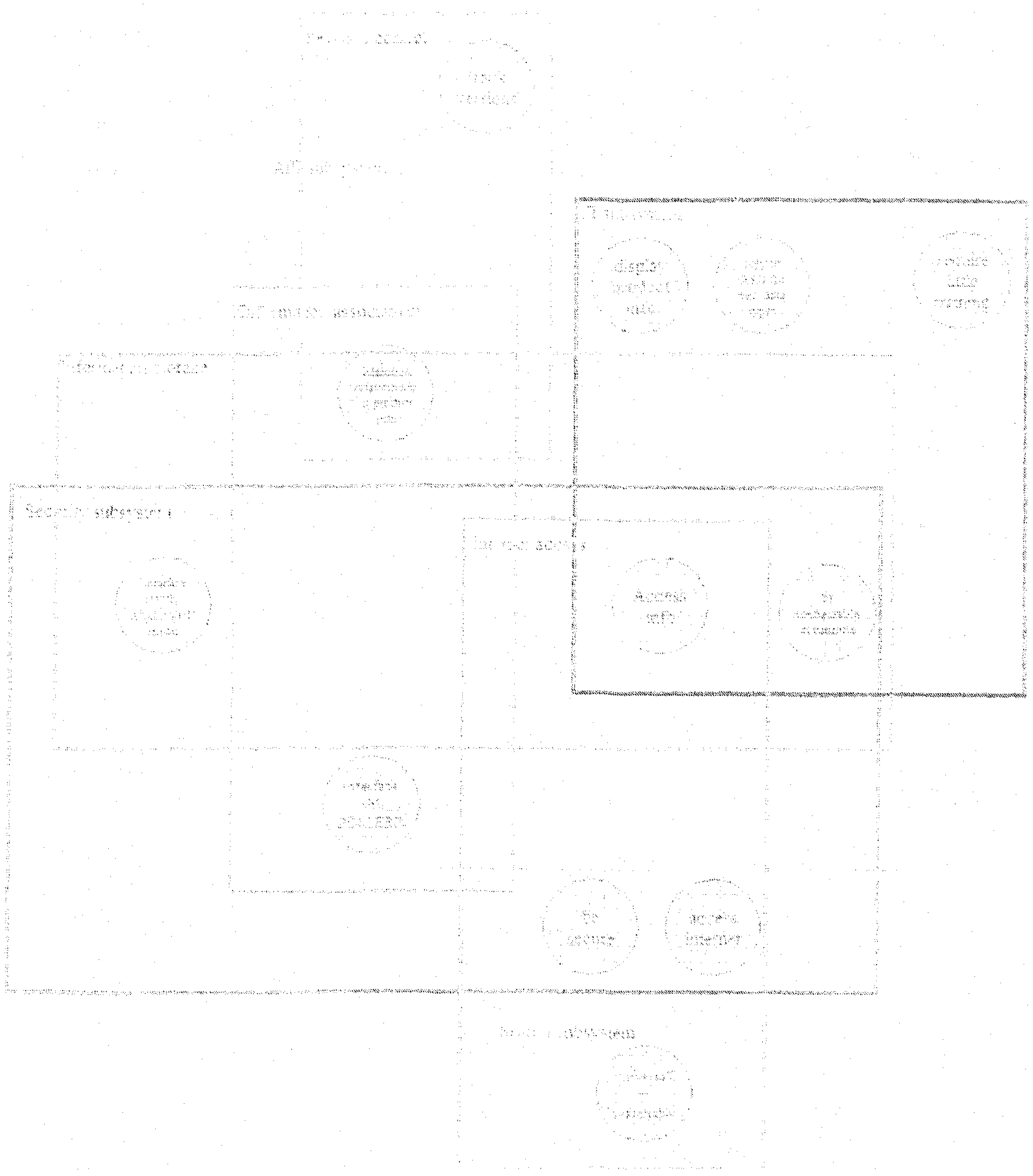


Figure 30: OPLM subsystems Visualization exercise 4

6.3.1 Discussions:

Visualization of system components and their association with major functional requirements help developers and system designers get a clear idea about the boundary of the system. The reason for conducting visualization exercise so vigorously is that OPLM is an open source project. When a developer joins the OPLM community, it is necessary to convey an overall picture of OPLM that would not be very time consuming, yet, be able to communicate the major functional requirements and boundaries.

6.4 OPLM Systems Architecture

The objective of thesis is to create the systems design of OPLM. So far various aspects of OPLM, e.g., background analysis, requirements analysis and presented a model for Engineering Change Management have been discussed in this section. OPLM systems architecture is presented. Using Unified Modeling Language (UML) notations and CmapTools, the basic workflow of OPLM has been laid out. Then the architecture for each major subsystem of OPLM is presented along with a combined architecture for OPLM.

6.4.1 OPLM Preliminary Model:

Figures 31 (a), (b) and (c) show the user's view of OPLM. In this scenario, a user first logs in to OPLM using a username and password. The system identifies the user's role, team, project, access rights and other affiliations. Users cans chose

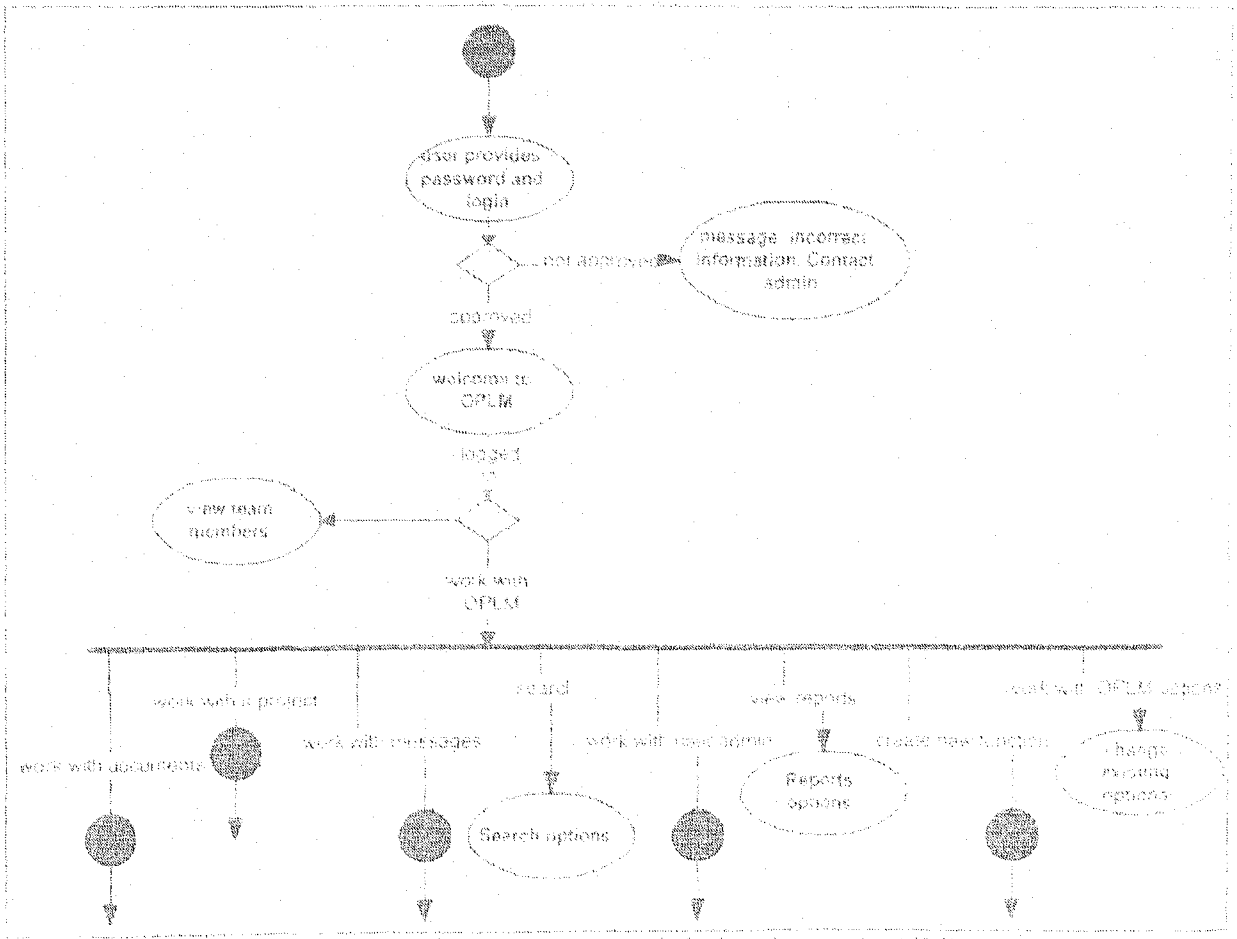


Figure 31 (a): User interaction diagram for OPLM shown in UML notations.

to work with a PLM functional area or administrative area depending on the access allowed. A user is a pre-configured entity in OPLM that can have many attributes. Once the user takes upon a work, OPLM keeps track of versions of documents, automated notifications, updates, etc.

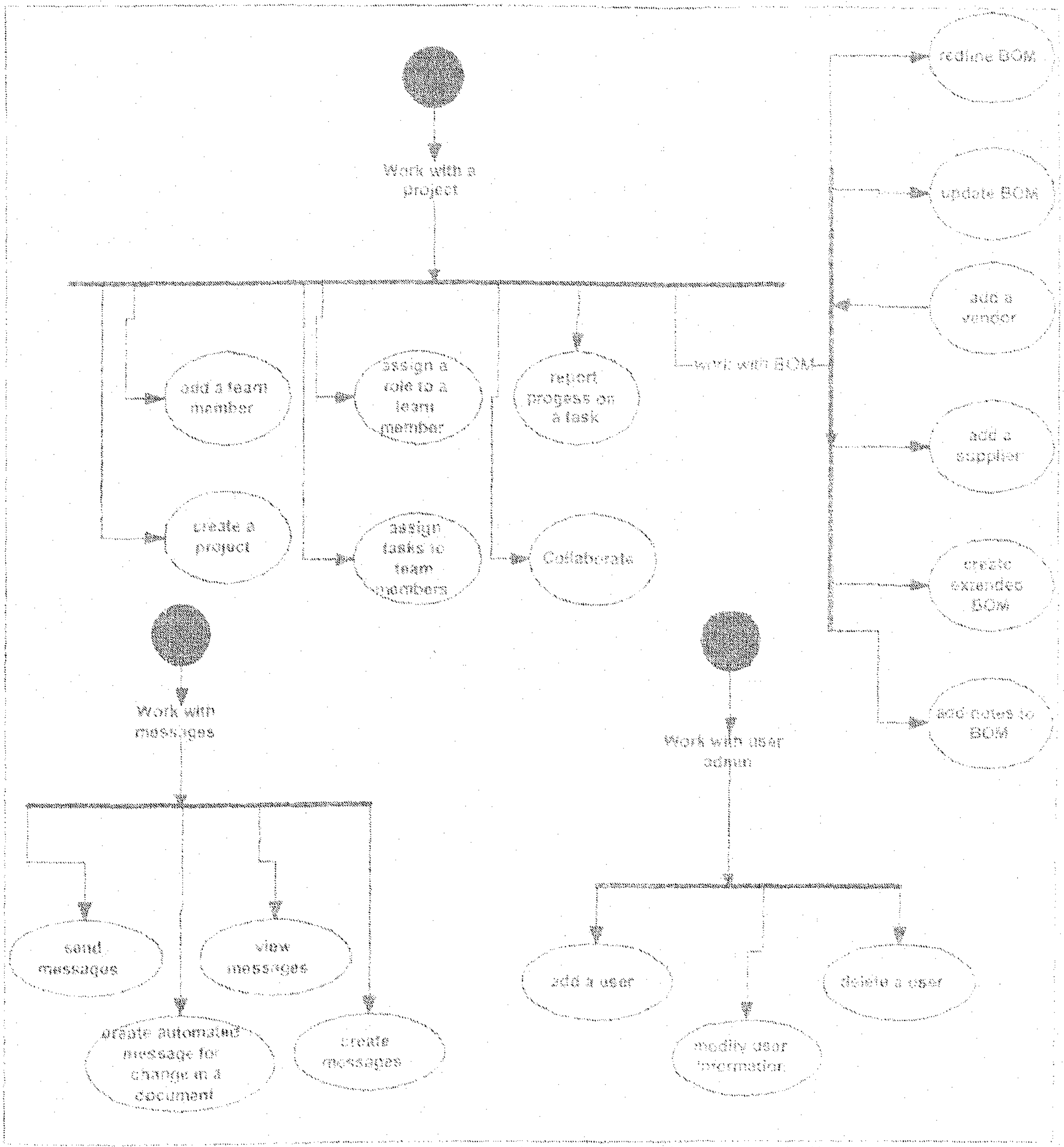


Figure 31 (b): User interaction diagram for OPLM shown in UML notations.

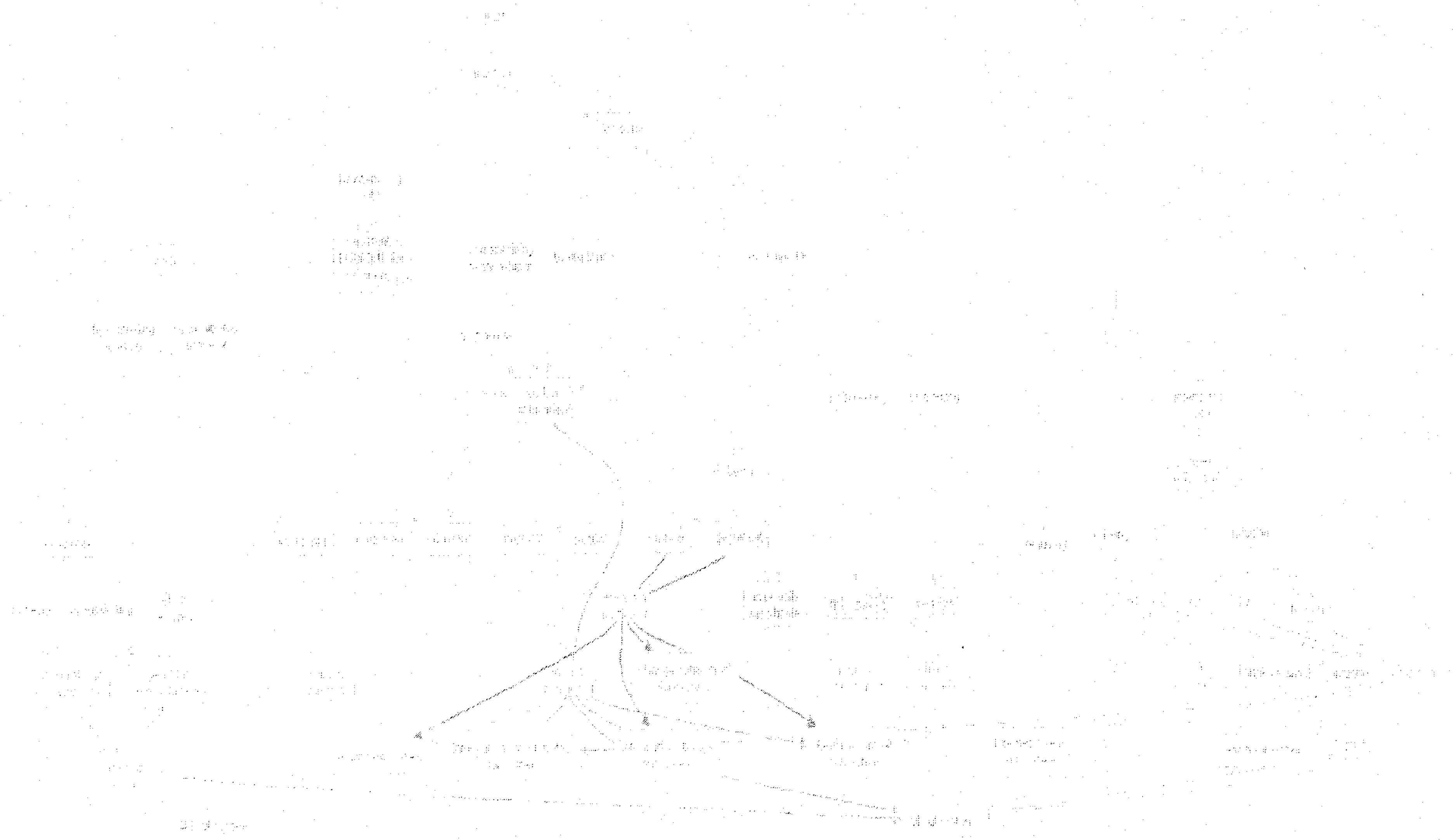


Figure 32 :OPLM concept map depicting major functions and relationship to subsystems

Figure 32 shows a systems point of view of OPLM. This is a 'concept map' of OPLM. Four major attributes of PLM are given in the first layer. In the second layer it has been shown how these attributes work. In the third layer, subsystems are tied to attributes. This concept map is further extended in the following sections to create the system architecture.

6.5 Product Data Management (PDM) Model for OPLM

PDM related topics have been discussed in details in chapter 2 (literature review). PDM is used to manage all documents including CAD/CAM/CAE documents. In the proposed PDM architecture, some additional functionality such as 'product breakdown structure' and 'workflow breakdown structure' are included to ensure that requirements are fully met.

Here is the description of the PDM model Figure 33:

1. Product breakdown structure: This function actually represents the BOM. Manufacturing companies rely on product breakdown structure heavily because of the continuous use of it across the organization by various departments. A real time and most recent and correct product structure require being present in order to avoid design and manufacturing faults.
2. Workflow breakdown structure: Workflow breakdown structure is one of the most important functions of PDM. A workflow includes a group of activities and their connections, conditions of starting and ending processes, and particular descriptions such as executors, correlative applications, and data needed or created for each activity. Therefore, workflow breakdown structure can help enterprises to achieve their goals more effectively.

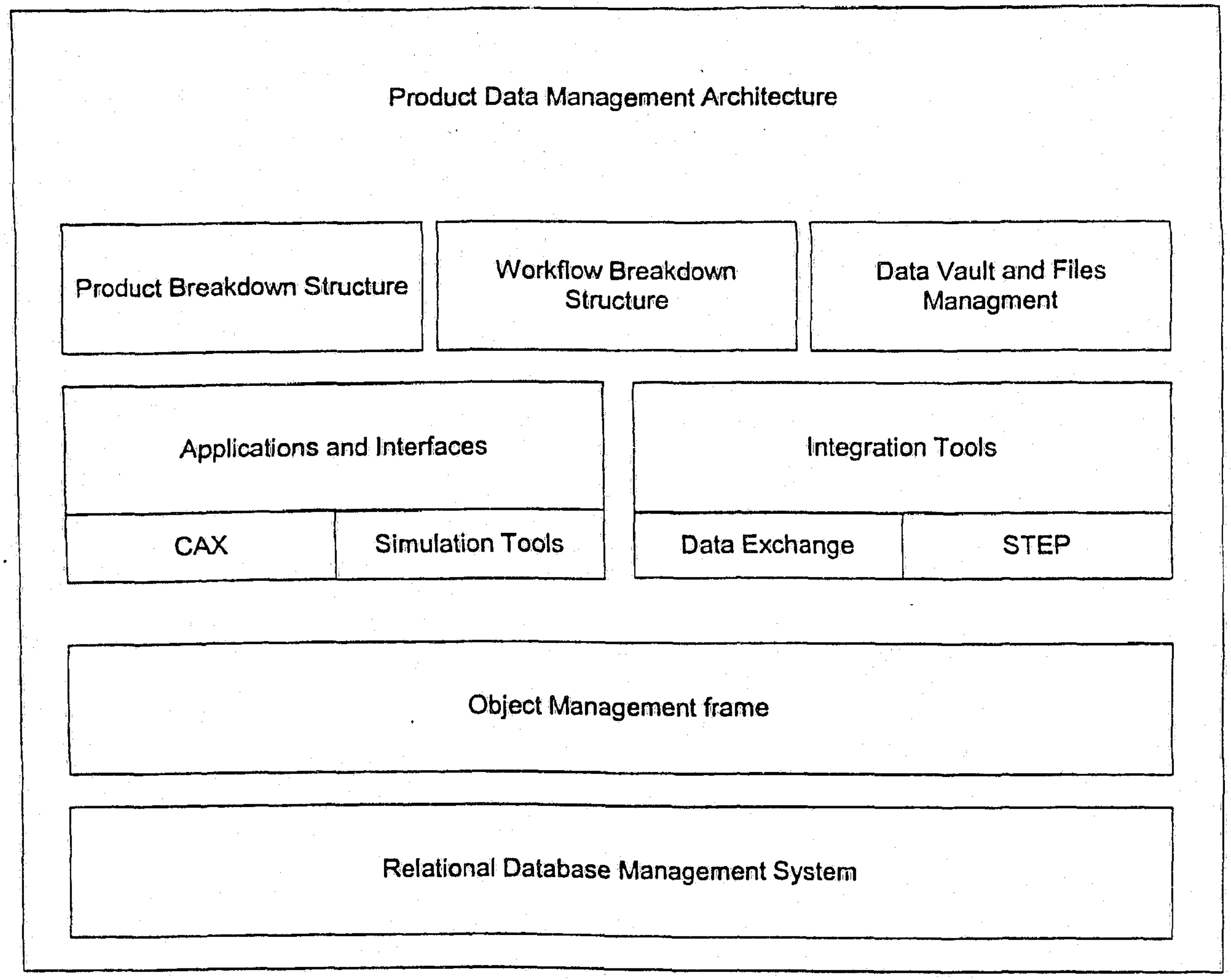

Product Data Management Architecture
Product Breakdown Structure
Workflow Breakdown Structure
Data Vault and Files Managment
Applications and Interfaces
CAX
Simulation Tools
Integration Tools
Data Exchange
STEP
Object Management frame
Relational Database Management System

Figure 33: PDM architecture showing system components integration tools

3. Data vault and files management: A data vault is used to as a repository to control all kinds of product information. Data are stored in two ways, namely, product data (like CAD, CAE data) and meta-data that is data related to PDM control information that supports functions performed by PDM. Data vault, in other words, provides secure storage and retrieval of product definition information. Files management function is facilitated by meta-data (tracking changes, release levels, approval/authorizations etc.).
4. Applications and Interfaces: At the second layer of PDM, applications and interfaces are controlled. An organization uses multiple applications from multiple vendors and performs operations, creates and changes product information. All these applications must be integrated into PDM for seamless control of data.

5. Integration tools: In order to ensure portability and exchangeability of CAD and other data, multiple integration tools are required. STEP can translate CAD, CAE and other engineering data into a vendor-neutral form. An SME that cannot use multiple CAD formats would use integration tools to translate data into the format they use with the help of STEP and then perform required operation on the product information.
6. Object management frame: Object management frame essentially works on the concept of 'business objects'. Business objects can be defined as objects in a computer program that abstract the entities in the domain that the program is written to represent. For example, an order entry program needs to work with concepts such as 'orders', 'invoices' etc. A business object may represent each of these. A business object server holds the definition of all the objects of a business. The advantage of using a business object server is that it helps developers create a simple model of a business with basic workflows and configure the underlying schema (types, attributes, relationships etc). The object management frame in OPLM thus makes the PDM object oriented. Using this would create an insulation of schema and business logic from relational database management system (RDBMS). Users would not be exposed to any SQL (structured query language) rather a 'business object query language' will deal with deal with business objects.
7. Relational Database Management System: Theoretical concept of RDBMS can be found at [2]. Based on E.F. Codd's theory [54], RDBMS was created where these systems presented data to the user as relations (a presentation in tabular form) and provided relational operators to manipulate the data in the tabular form. In existing businesses, databases use RDBMS almost everywhere. For OPLM we have proposed MySQL, an open source multithreaded, multi-user SQL RDBMS that has an estimated five million installations.

6.6 Engineering Change Management (ECM) Model for OPLM

When ECM model is incorporated to PLM, following components perform the change tasks (Figure 34):

1. Change monitor agents: Change monitor agents (CMA) handle initial requests for changes. They identify the role of change initiator and make necessary changes to the change repository. CMA is responsible for updating the ECR data.
2. Change processing agents: These agents are responsible for processing the change requests. There are two types of change processing agents:
    a. Change coordinator agent: Responsibilities of Change coordinator agents (CCA) are accepting the ECR from CMA, reading the Change Propagation Matrix and sending message to all affected parties, updating the change repository, keeping track of message-interaction among the affected parties.
    b. Change approval agent: Once CCA receives draft solutions for the change, change approval agent (CAA) submits the solutions for approval. It then takes the final solution and sends it back to the change initiator for change implementation.
3. Version Management: Version management of all documents is rigorously performed in OPLM. Any change made to the document is made reversible by using version management. It has been decided that Subversion (an open source version management tool available at [55]) is most suitable for OPLM. This version management tool would be wrapped with a more user friendly UI that would provide customizing options to a user. Some of the primary requirements of such interface would be able to provide version history management options and check-in/check-out options. Version history can be managed in several ways, for example, minor versions are documented as x.1, x.2 etc. and major versions as (x+1).1, (x+2).1 etc. Check-in/check-out options are more important for CAD documents. It would be made optional if certain format of documents use atomic check-in/check-out methods or multiple

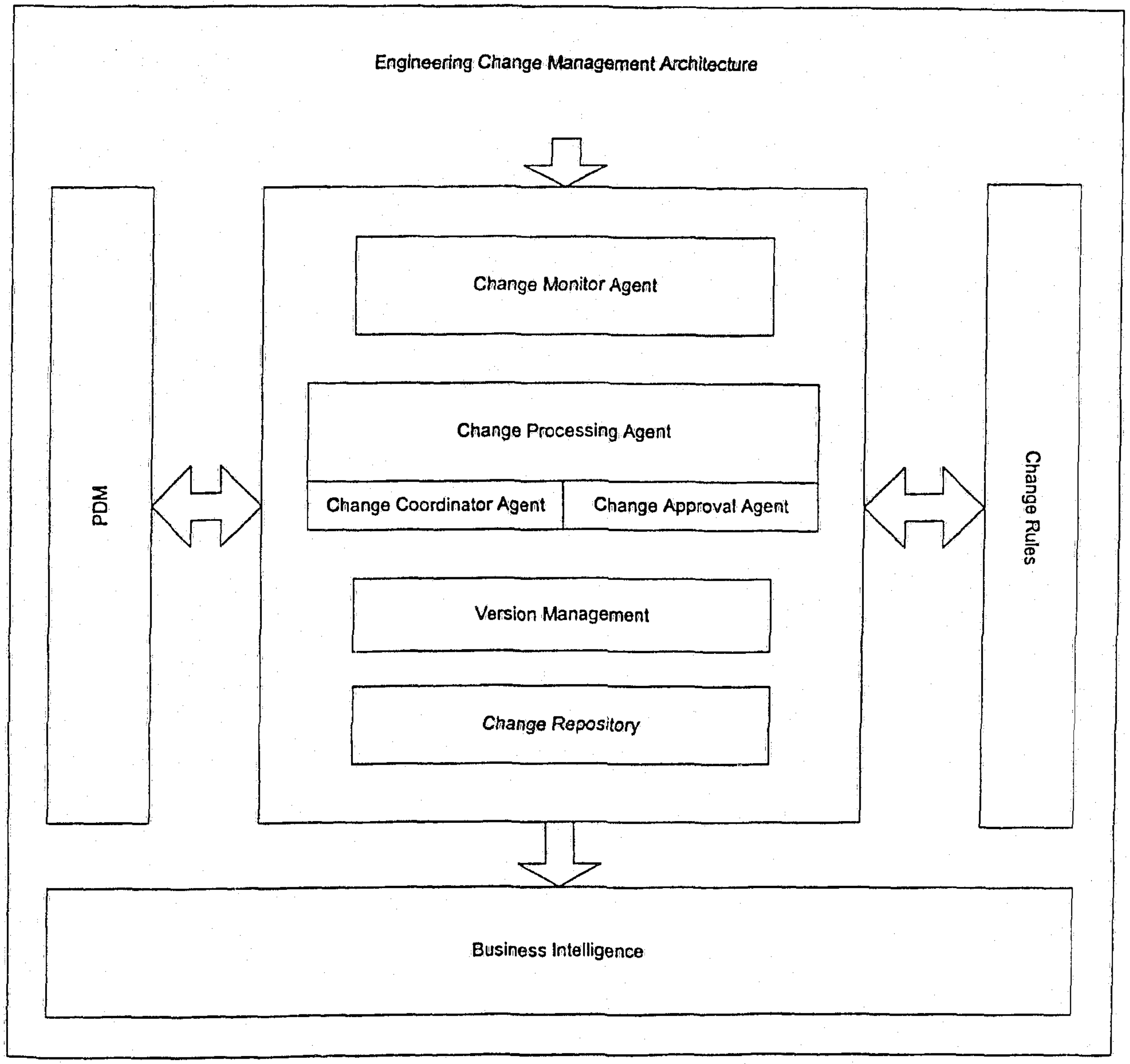


Figure 34: OPLM ECM subsystem showing major components and relationship to other subsystems

4. check-in/check-out methods.
5. Change Repository: Change repository is the storage area where all change related information is held. This repository is used to generate real time change data. For example, this can retrieve information like number of ECR's issued for a given product, processing time for each stage of

change etc. This repository is also used to extract intelligence from the change process.

6. Change rules: Change rules are set for the change agents. Each change agent operates with certain customizable rules. Change rules are part of company strategy that may change from time to time. These rules are implemented for change automation as well as manual change activities.

6.7 Process Management Model for OPLM

It has been recognized that OPLM must be able to create, monitor and control all

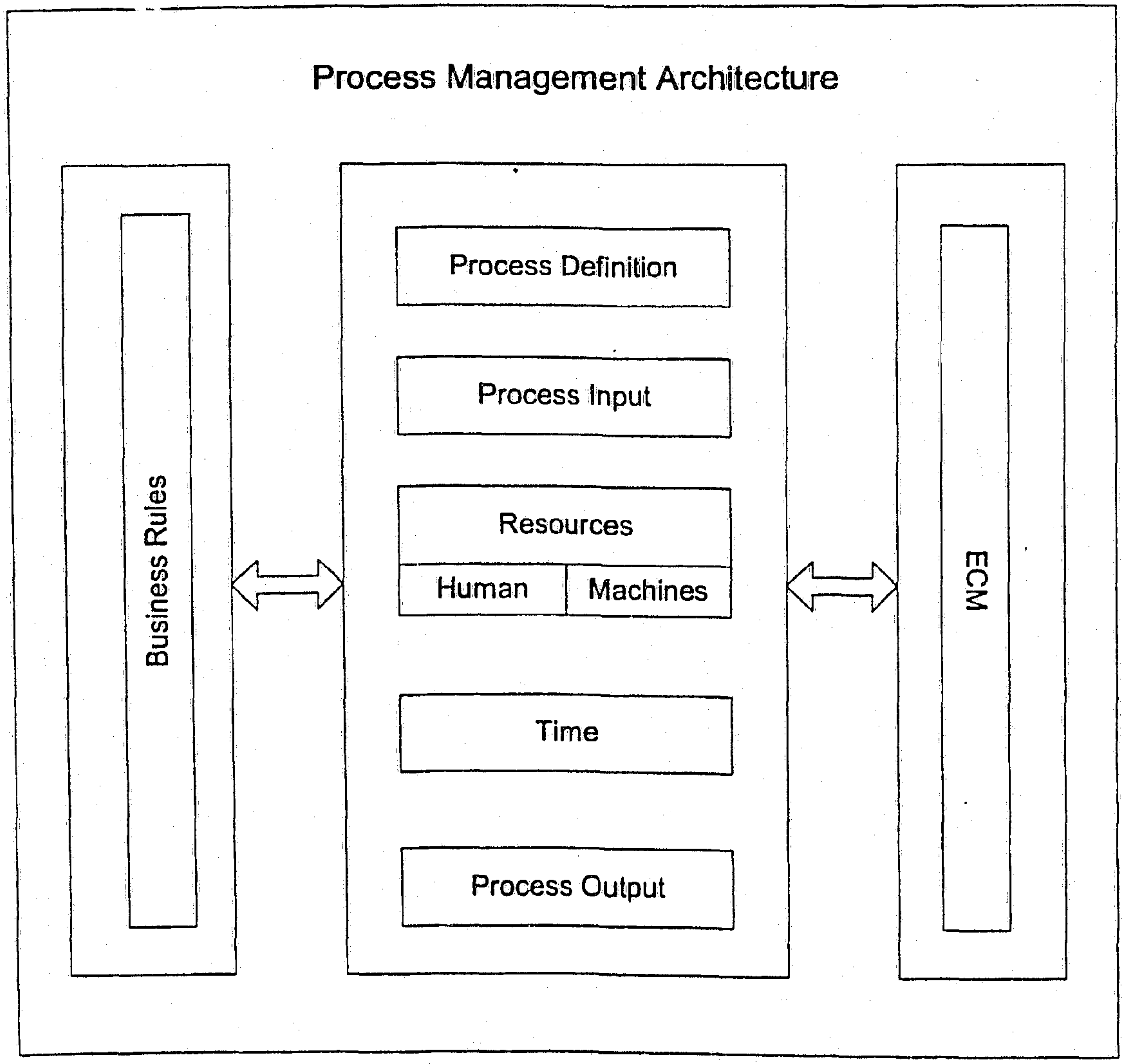


Figure 35: OPLM process management subsystem showing major components and relationship to ECM and BI

possible processes in an organization. It could be the design process, manufacturing process etc. OPLM process management architecture consists of the following components (Figure 35):

1. Process definition: The process must be precisely defined with distinct boundaries. Defining processes creates reusability.
2. Process input: The input of the process must be defined
3. Resources: The process must define what resources and tools it would use. Resources can be human or machine.
4. Time: Each process has a 'time' component. Time records the process duration and deviation from designed limits.
5. Process output: At the end of a process, data, information or a product is modified from the initial state. This final state at the end of a process is the process output.

6.8 Introduction to Business Intelligence (BI)

Business intelligence (BI) can be defined as a set of concepts, methods and processes to improve business decisions using information from multiple sources and applying experience and assumptions to develop as accurate understanding of business dynamics [56]. The area of business intelligence research started since 1980 mainly by military and government intelligence agencies. BI encompasses software like extraction, transformation and loading (ETL), data warehousing, multidimensional/online analytical processing (OLAP), data mining and visualization. Technically, BI is a very broad field that covers all business data, business analyses, knowledge management, decision support system etc [57].

6.8.1 OPLM BI Model

OPLM BI model has the following components (Figure 36):

1. Business Objects: Business objects can be defined as reusable objects or components that can make an abstraction of the entities of a domain. For

example, a customer, an order, a purchase receipt. Each object represent the information needed by users. Business objects in OPLM are the basic functional units. In order to gather business intelligence, business objects are constantly monitored against business rules.

2. Decision Support System (DSS): These are defined as computerized information systems that support decision making activities. There are numerous models for DSS. There are a few open source projects that has developed distributed DSS, for example, Dicodess (Distributed Cooperative Decision Support System) [58], GeNIe (Graphical Network Interface) and SMILE (Structural Modeling, Interface and Learning Engine). We propose to incorporate Decodess as OPLM DSS tool.

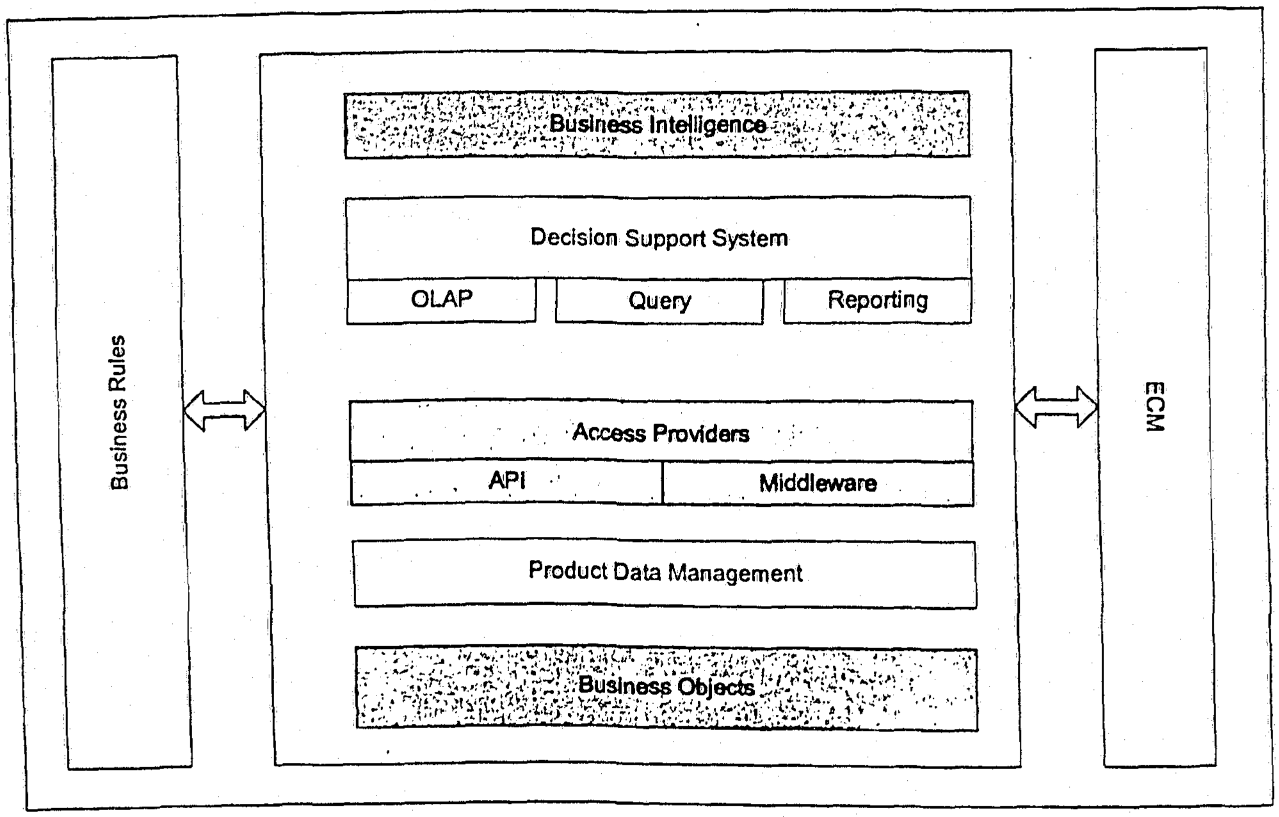


Figure 36: OPLM business intelligence subsystem showing major components and relationship to business rules and ECM

   a. OnLine Analytical Processing (OLAP): OLAP is a decision making tool that manipulates multidimensional data from data warehouse.

Major component of OLAP is an OLAP server. Mondrian is the best-known open source OLAP server that supports RDBMS such as mySQL. Therefore we propose to incorporate Mondrian for OLAP server.

b. Query and Reporting: In OPLM, query can be defined as a specification that needs to be calculated from the database. We have already proposed to use MySQL for OPLM database use. This is a robust application that supports Sequential Query Language (SQL). The idea of SQL was first proposed by E.F. Codd [54] in his 1970 paper. Later SQL was adopted by ISO and ANSI. MySQL uses its own query language for 'query' purposes. However, there are many customized open source query tools that can be used with MySQL. For example ObjectVisualizer, Dbxtra, Navicat, dbQwikReport etc. Among these, ObjectVisualizer is most useful for our purposes. It supports reporting and query functions. A reporting tool can be defined as a tool that creates reports against the data stored in a database. The best part of ObjectVisualizer is that it provides an interface to create queries, reports and charts that are extracted from the objects directly without the need of SQL.

3. Business Intelligence: BI can be gathered by integration of DSS, OLAP, queries and reporting tools. All these tools provide information about how data is created, used, stored and trended. BI is the most significant component of OPLM. One of major issues with BI is 'data quality'. High data quality means that the data is fit for the intended use in business operations, decision-making and planning. There are different attributes that 'data' can have in order to be of 'high quality', for example, accuracy, correctness, completeness, relevance and currency [56].

6.9 Overall OPLM Architecture

OPLM subsystems have been discussed so far. A complete OPLM architecture is now shown in Figure 37. As defined in beginning of this thesis report, OPLM would monitor, control, report, and improve all aspects of a product. In addition, it would manage the activities relating to a product from the identification of the need that the product will address, to the treatment of the product at its end of life. OPLM architecture satisfies the definition of PLM.

Firstly, the process management ensures that all the processes involved in the business are defined, modeled and monitored. The processes include manufacturing, design, support and other processes. Along with the processes, resources necessary to execute the processes are also monitored and controlled.

Secondly, the ECM subsystem performs all required change management activities. ECM not only controls the engineering changes, but it also controls, monitors and reports all other changes that occur in processes, PDM and business rules.

Thirdly, the PDM subsystem manages all organization data. In OPLM, data is represented in the form of business objects.

Fourthly, business rules are modeled and it ensures that all processes, ECM and PDM activities are performed according to the rule. Any discrepancy in the rule is monitored and reported.

Finally, the business intelligence provides with the knowledge and decision-making abilities.

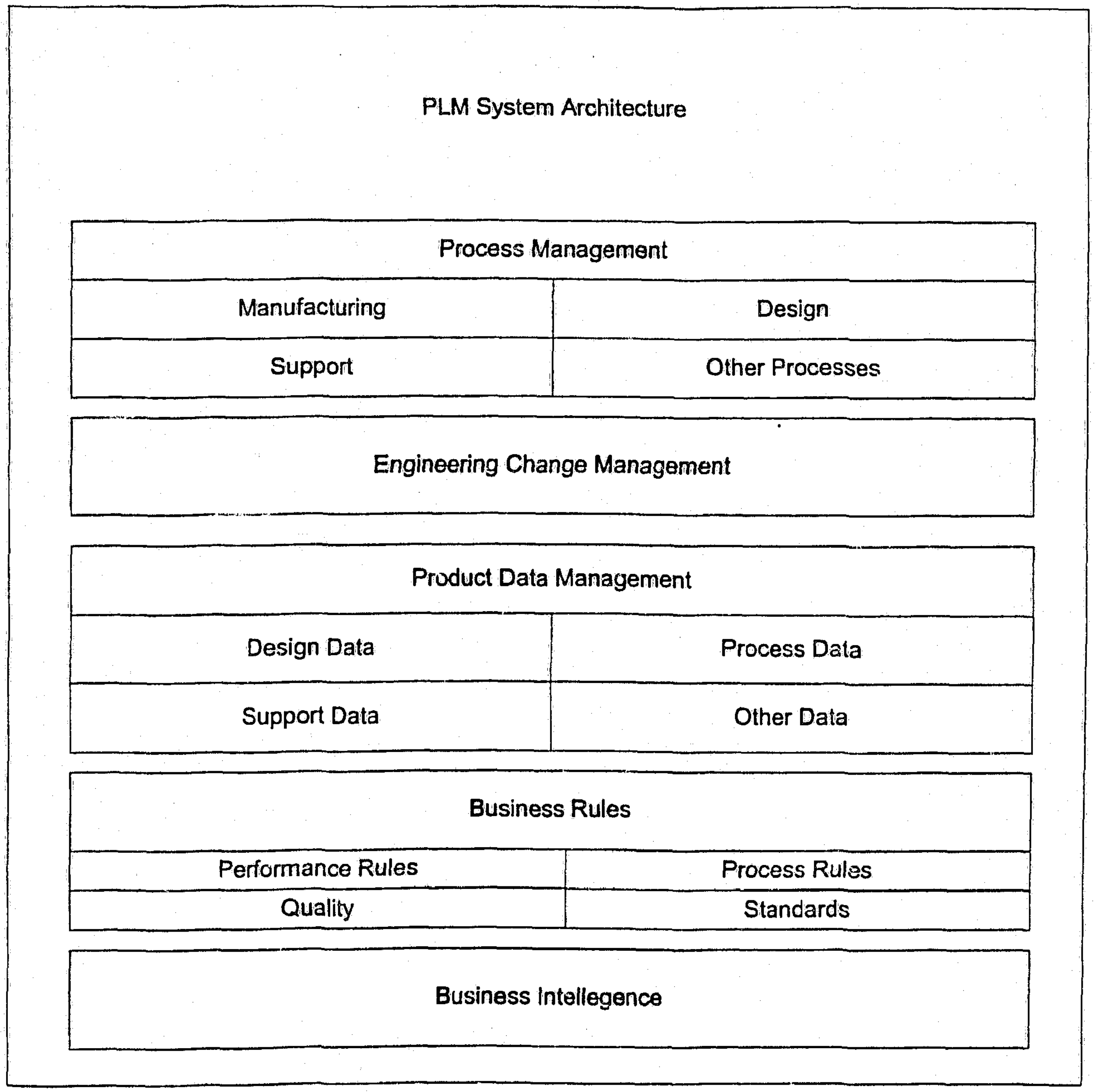


Figure 37: OPLM systems architecture showing all major subsystems

It has been pointed out in this research that 'engineering side' of the PLM is the focus of this work. Models of process management and business intelligence are closely related to OPLM. Process management model gathers all information from operations and business processes while business intelligence presents decision-making information to the user. This attribute of OPLM makes it a complete tool that SMEs can use for their product lifecycle management requirements.

Thus, OPLM integrates and harmonises the many different kinds of information used during product development and engineering.

Chapter 7

Conclusions

7.1 Conclusions

Product lifecycle management system is capable of increasing productivity, time-to-market and control over products. Larger enterprises took the benefits out of PLM. Due to financial and infrastructure constraints, SMEs are left out from this technology. Proposed OPLM system is expected to meet the demands of SMEs and contribute to their business immensely. This research has defined a set of user requirements and laid out high-level systems architecture for OPLM. The engineering-side of PLM is our focus; the business-side of PLM is beyond our scope of work. Open source methodology plays a great role in this research as OPLM is expected to be developed using open source means. A model for open source development has been proposed in this research that integrates developers and contributors from SMEs and academia. We also proposed a method for systems design called 'adaptive model' that would be suitable for open source development. We also conducted a detailed study of engineering change management principles, models and systems. A model is then proposed for engineering change management in SMEs. This model has been designed to reduce change management time, reuse of process and information and accurate product development.

7.2 Future Research

The research presented in this document can be described as the first step towards development of OPLM. Following are the desired future research scope for OPLM:

1. Open Source Development Model: The model presented by author needs to be validated in an open source setting. Despite obvious benefits (integration of academia and SMEs, subsidized developers, knowledge transfer in academia and SMEs etc.) of our model, it has some issues like motivation factors, control mechanism etc. that needs to addressed and validated in an actual setting.
2. Adaptive Systems Design Model: This model assumes that developers and contributors are in an open source setting that is compatible to the open source model proposed in chapter 1. Any change in open source model is likely to affect the adaptive system design model. Therefore, validation of the proposed models is crucial for this project.
3. Modified Engineering Change Management Model: Modified ECM model is proposed for SMEs. This model needs to be tested in actual change management setting. Initial results may show 'slower' change processing time because of limited information of previous change process. An alternative model can be designed that would make the change process faster when first implemented. This model may be more effectively deployed as change information (collected from an alternative model) is gathered by the system for generating automated responses.
4. Visualization: Visualization exercises performed in this research are mostly for the use of developers. Another area of visualization is the user interface (UI). Representation of information to the user in the most meaningful manner is a primary requirement of OPLM. OPLM requires the UI to be simple and easy-to-use. Using the preliminary concepts presented here in visualization exercise, future researchers can elaborate on UI and work on a generic model for visualization.

5. Coding, Testing and Implementing OPLM: Coding and testing are the immediate next step to the current research. A prototype OPLM should be built shortly using the guidelines provided in this research. The prototype can be tested in SMEs. Then at the final step, complete OPLM software can be made available to SMEs. Once SMEs across industries use OPLM, it can become a de facto standard. At that point, bigger companies would provide resources in building a complete integrated OPLM that can be used throughout the industry irrespective of size of organization.
6. A detailed, structured survey: In spite of many useful and interesting results, the survey conducted in this research was limited in terms of demographic information of users. It is desirable that a survey can be conducted in future equipped with better resources so that the missing information can be collected and analyzed.
7. Business process management (BPM): One of the major components of OPLM is business process management. A detailed BPM study and integration to OPLM is beyond the scope of this research. Experts and researchers from information technology management and business disciplines can contribute to such endeavor.

Appendix A

Results of a Survey of Product Lifecycle Management Issues

M. Anwary and F.A. Salustri
Ryerson University

Summary

During late 2004 and early 2005, Ryerson University conducted an informal and preliminary online survey of industries who practice *product lifecycle management* (PLM). The purposes of the survey were to understand:

1. the key issues in regards to the IT needs of PLM;
2. the impact of PLM on core operational concerns;
3. the extent of exposure to legacy PLM software; and
4. the expected level of participation in a new project to develop an opensource PLM system.

Manufacturing industries have been facing many challenges trying to maintain their competitive edge. Many companies are utilizing technology as a means for competing globally and decreasing costs of operations. This study was developed to better understand what types of technology is being used by this industry as well as what this industry has planned for the short term.

Analysis of the survey results indicate that potential users of PLM software find cost, control, infrastructure requirements, and "feature bloat" to all be serious issues that are not currently addressed by available PLM systems. It was also found that nearly every survey respondent would welcome an opensource PLM solution.

It also seems to be the case that a more detailed and formally structured survey is indicated, to discover more precisely what industry needs and expects of PLM systems.

Survey Implementation

The survey was created using a free online survey tool and hosted at http://perennialsurvey.com/s.cgi?11776

Survey questions were based on an intensive review of the existent literature on PLM and some of the existent PLM software packages.

The survey was advertised in three ways:

1. The survey was announced (in November 2004 and February 2005) using the monthly bulletin **MMO Matters**, published by Materials and Manufacturing Ontario. This bulletin is circulated to the members of MMO. MMO has approximately 2000 members including industry and academia who work in the manufacturing industries in Ontario.
2. A bi-weekly web-based publication called **2PLM**. Notification of the survey was circulated on January 31, 2005 issue to an international audience (http://www.johnstark.com/2PLM178.html).
3. A link to the survey was placed on Salustri's home page (http://deed.ryerson.ca/~fil).

About the Respondents

By itself, the survey appeals to companies that have design and manufacturing operations. We expected that all the respondents were from our target industries. As this survey was a free one, with no obligation to participate in it, the respondents are expected to be willing to be share their views generously.

No demographic information about participants was collected.

The number of respondents was low (about 20 responses) compared to the target population. Yet, the number of the respondents can be well justified considering the relatively informal nature of this preliminary survey.

Some of the questions required a generalized answer rather than a specific answer. This was intentionally done in order to attract more respondents, but it also affected the capacity to draw reasonable conclusions from the responses.

Major Findings

The following summarize the major findings of the survey.

- Companies believe that daily engineering activities can benefit from PLM software.
- Design related issues are more significant than manufacturing issues.
- Conceptual design is the area that requires highest improvement.
- There is relatively little automation of Engineering Change Management and Revision Control in the companies surveyed.
- Robust data management facilities are absent among the companies surveyed.
- Managing multiple CAD formats and data exchange is a major issue.
- High costs and lack of control are the two major barriers for purchasing PLM solutions.
- The overwhelming majority of the companies want to participate in building an opensource PLM solution.

Detailed Results

For the sake of this survey, *product lifecycle management* is defined as:

> PLM is monitoring, controlling, reporting, and improving all aspects of a product and the activities relating to the product from the identification of the need that the product will address, to the treatment of the product at its end of life. It is meant to integrate and harmonize the many different kinds of information used during product development and engineering.

Question 1

> *Do you believe PLM software as defined at the beginning of this survey could be of direct benefit to your daily engineering activities?*

There was overwhelmingly high response about the contribution of PLM software to engineering activities. This shows that companies need a PLM solution.

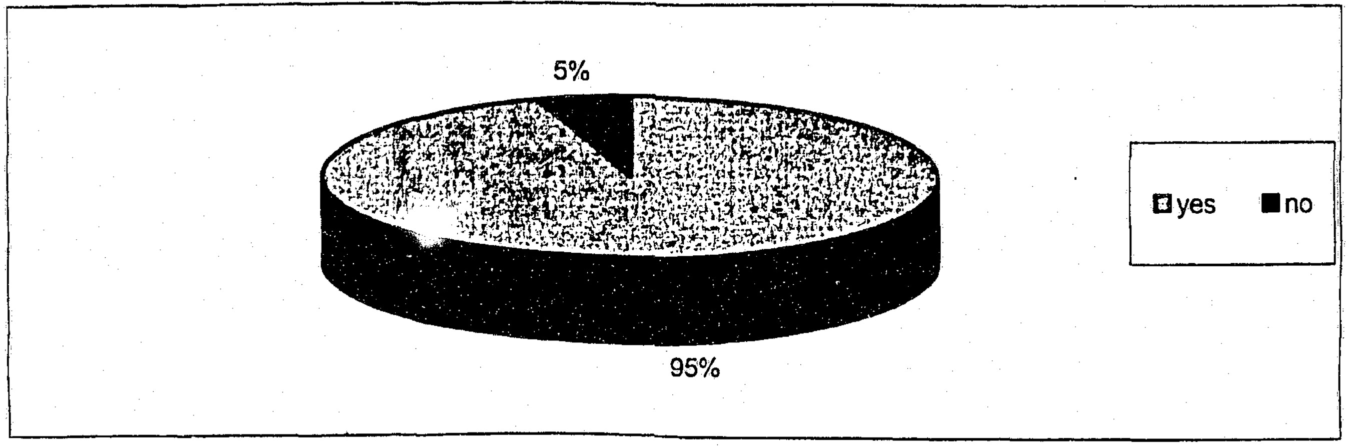


Figure 1: Response shows the awareness of benefit of PLM

Question 2-5

> *Which activities do you presently control with software tools?*
> *Which activities do you wish you had software to manage?*
> *Which activities do you manage and integrate via paper and/or verbal communications?*
> *For which activities would like to have improved software integration and communication?*

The answers to these questions represent how companies use software tools in PLM-related tasks. From the results (below), we can draw three conclusions:

- Most respondents (90%) use CAD software. Many respondents (65%)use software for product documentation. The ubiquity of tools like MS Word and Excel has likely contributed to this.
- Companies want significant improvement of tools that enable conceptual design, product testing, documentation and CAM.

- Conceptual design is an area where companies seldom use software; 60% of respondents use paper and verbal communication for this. 75-80% of respondents wished this activity to be computerized and improved.

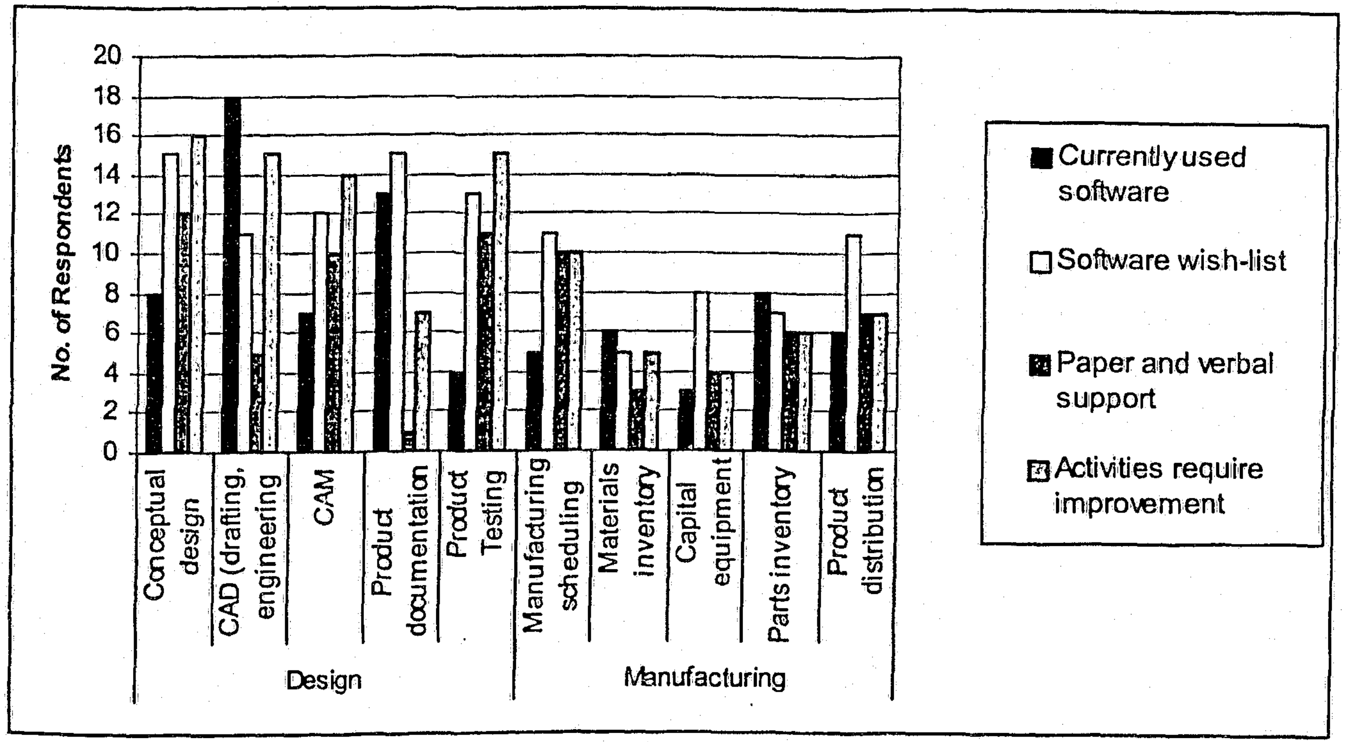


Figure 2:Number of responses for design and manufacturing operations

Furthermore, the survey found that companies are more concerned about design than about manufacturing.

To measure this, an **index of significance** was calculated as the relative measure of the issues in a given company using the formula:

Index of significance = (sum of the responses in a row) X 100/(number of respondents X number of columns)

where each *row* represents a particular engineering function, and each *column* represents a characteristic of note arising from the survey questions.

Table 1 shows the indices of significance calculated from the survey responses. Figure 3 shows the cumulative significance for design-oriented tasks versus manufacturing tasks, and Figure 4 shows the indices for the constituent elements of design versus manufacturing. It is clear that the respondents view the impact of PLM on design tasks as more important than its impact on manufacturing. This suggests that PLM systems should address design-related issues primarily, and that **conceptual design** is the most significant activity impacted by PLM.

Table 1: Index of Significance for different criteria of the survey

		Currently used software	Software wish-list	Paper and verbal support	Activities require improvement	Index of Significance
Design	Conceptual design	8	15	12	16	64
	CAD (drafting, engineering analysis, interference, ...)	18	11	5	15	61
	CAM	7	12	10	14	54
	Product documentation	13	15	1	7	45
	Product Testing	4	13	11	15	54
Manufacturing	Manufacturing scheduling	5	11	10	10	45
	Materials inventory	6	5	3	5	24
	Capital equipment inventory	3	8	4	4	24
	Parts inventory	8	7	6	6	34
	Product distribution control	6	11	7	7	39

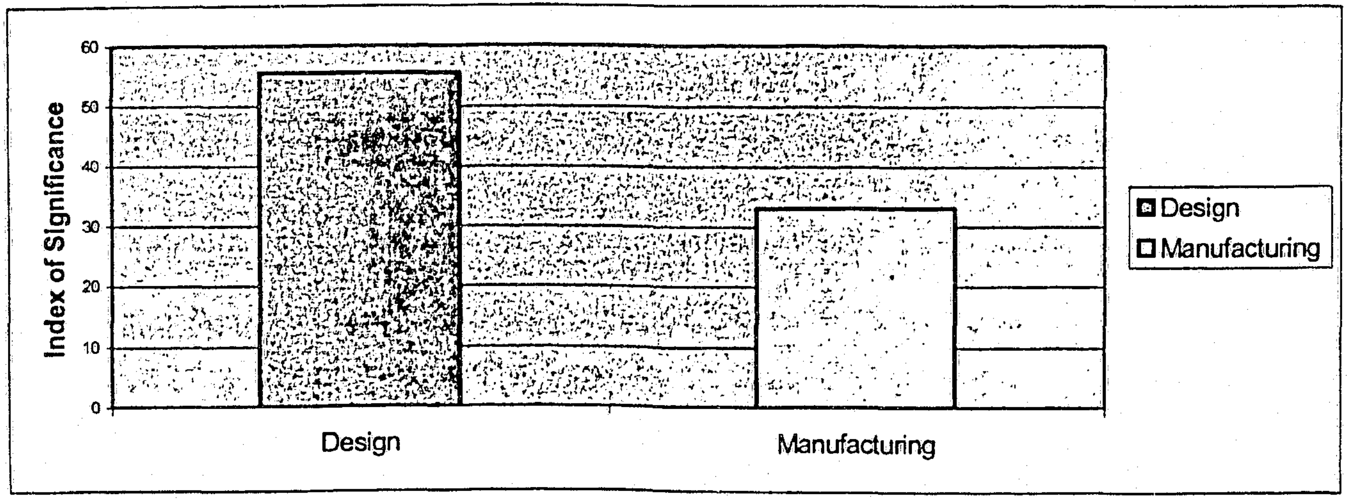


Figure 3: Design is more significant than manufacturing operations in companies

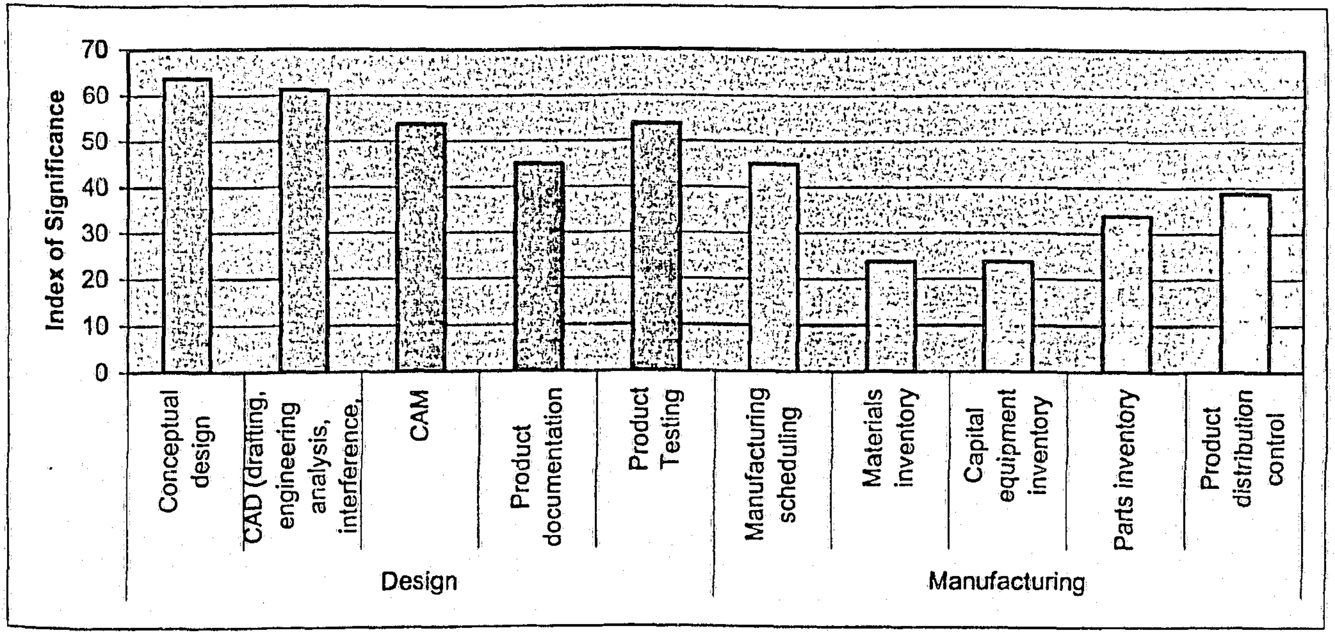


Figure 4: Index of Significance for different units of the operations

Question 6

How do you control Engineering Changes and related activities?

The majority of respondents use a combination of human and computer resources for change management. Change management is one of the most challenging tasks of a company. None of the companies use a fully automated change management system. 41% companies rely on paper and verbal change management procedures. The rest use some sort of software tool for change management.

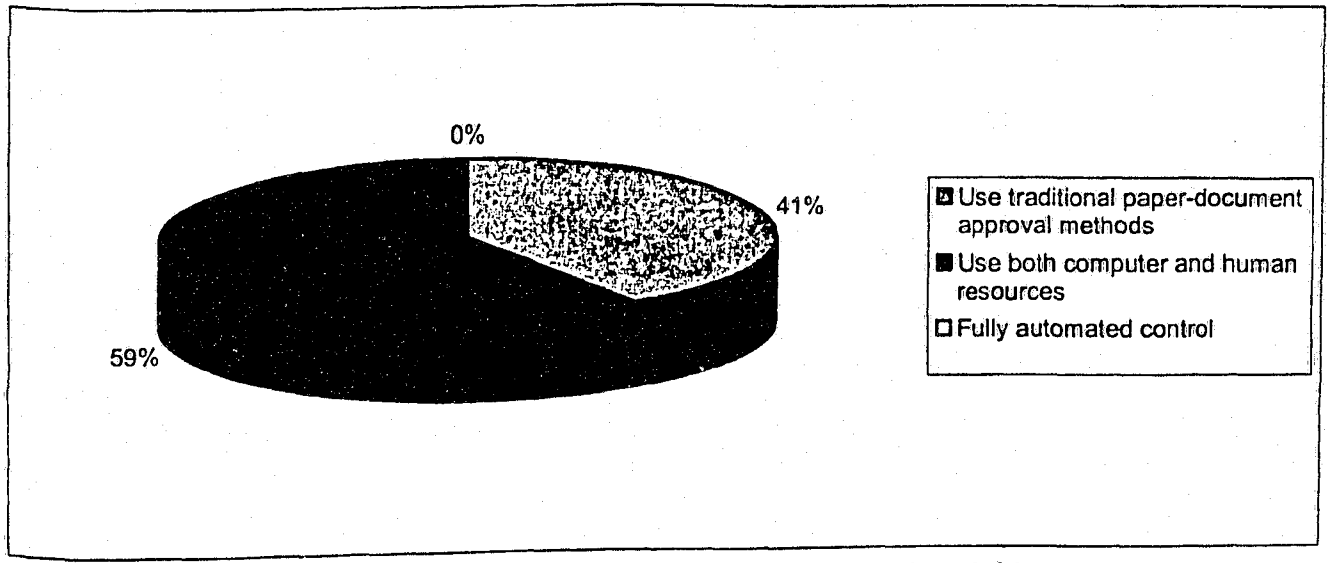


Figure 5: Engineering changes and related activities

Change management is the most vulnerable area for the respondents that require improvement. Research has shown that companies can save up to 10% of the annual turnover by building a robust change management system [1].

Question 7

How do you track revisions of documents?

Revision control is a very important aspect of document management in an engineering company. Manual tracking of revisions is most common (65%), even though it is time-consuming and error-prone compared to computer-based methods. While all companies require revision control, some feel that revision control is not required beyond CAD files.

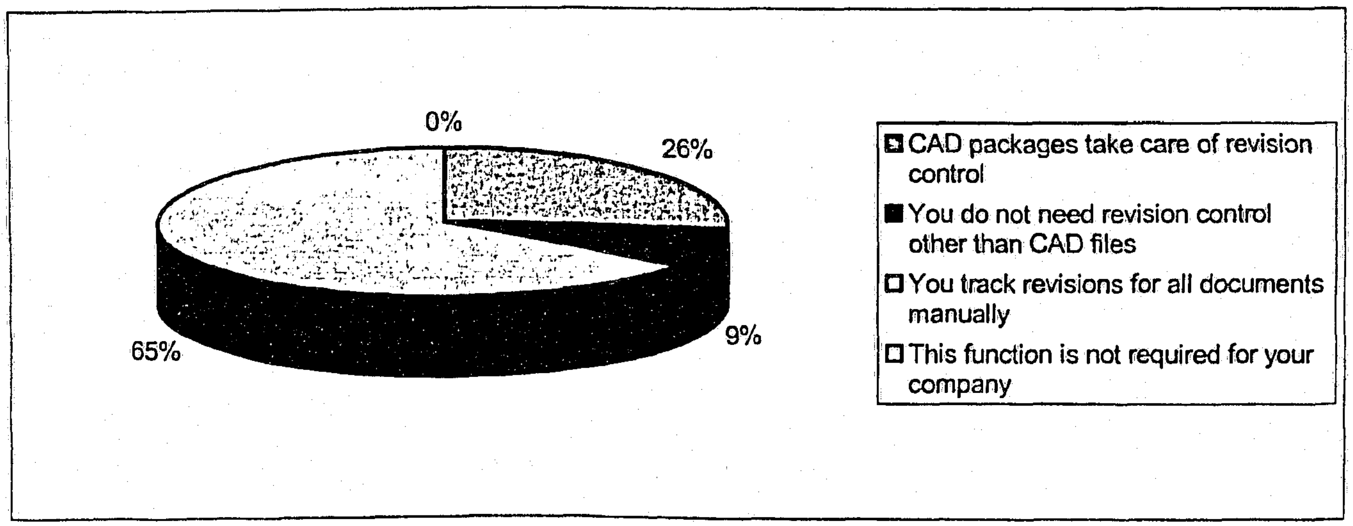


Figure 6: Tracking revision control

Question 8

How do you control configuration management (CM)?

It is generally acknowledged that highly complex products require robust CM. Since half the respondents do not do CM rigorously, this suggests that the respondents are largely companies that do not develop highly complex products. We note that this conclusion may be the result of variables not accounted for in the survey.

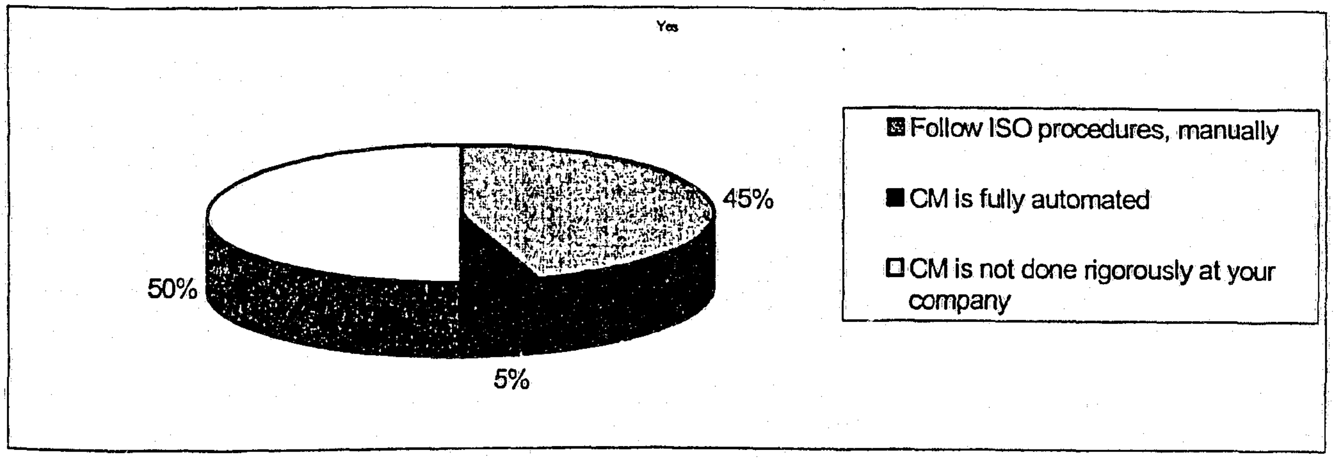


Figure 7: Configuration management control

Question 9

What kind of software do you use at your organization?

Many respondents (32%) use CAD packages. PLM software is used the least (4%). PDM tools are not widely used (14%) either. This may be because of the relative maturity of CAD software compared to PDM and PLM software. If one assumes that PDM and PLM software will mature over time as CAD software has, then it is clear that adoption of PDM and PLM software will increase with time, and that productivity and product quality will increase as a result.

We are not suggesting that current PLM approaches will become industry standards. CAD software itself has evolved in relatively unpredictable ways over the past 20 years, in response to new developments in software, in computer hardware (such as the wide scale adoption of PC-based computing), and in methodological improvements in product design engineering. We expect a similar evolution to occur for PDM and PLM, although at an accelerated rate compared to CAD software.

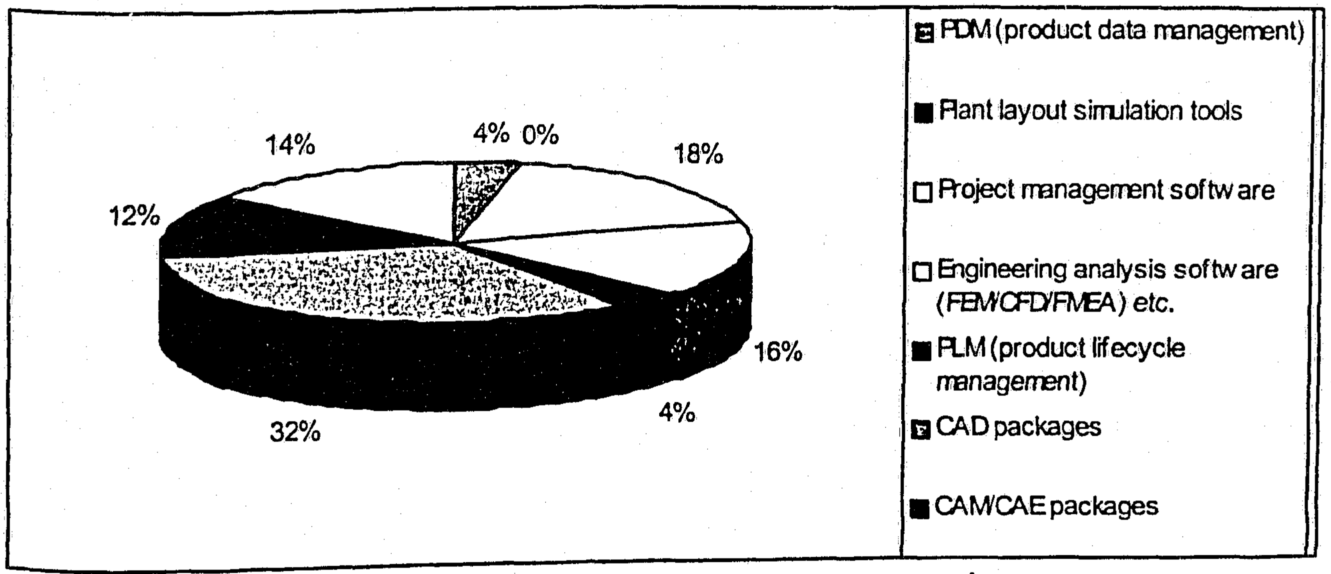


Figure 8: Type of software used in companies

Question 10

Do you have to manage multiple CAD formats?

Overwhelming majority of the respondents require the management of multiple CAD formats. This is another important aspect of the companies, especially since most current PLM solutions are tied to particular CAD packages.

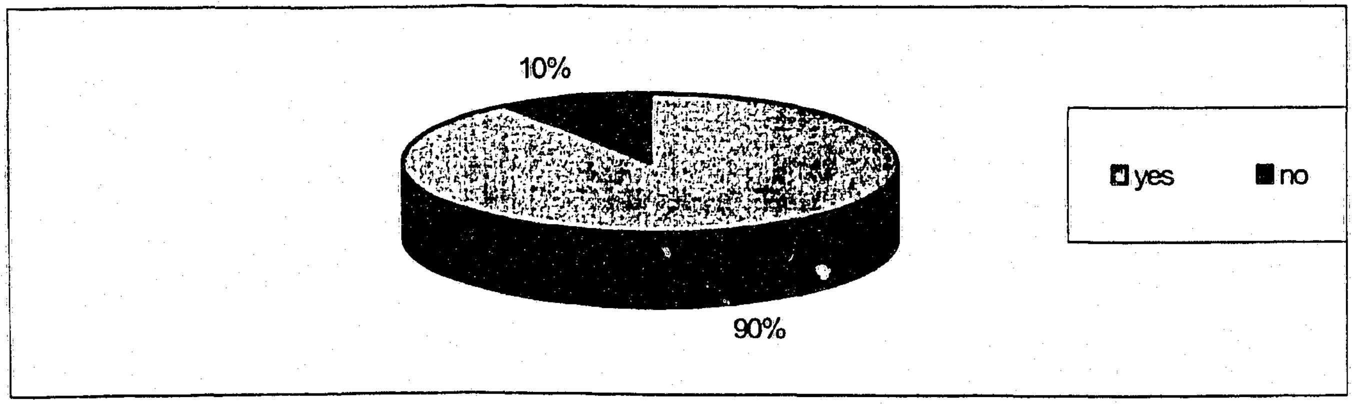


Figure 9: Management of multiple CAD-formats

Question 11

Is data exchange between software packages an issue for your company?

The majority of respondents (85%) identified data exchange as a significant issue. This is expected as a result of having to work with multiple CAD formats (Question 10).

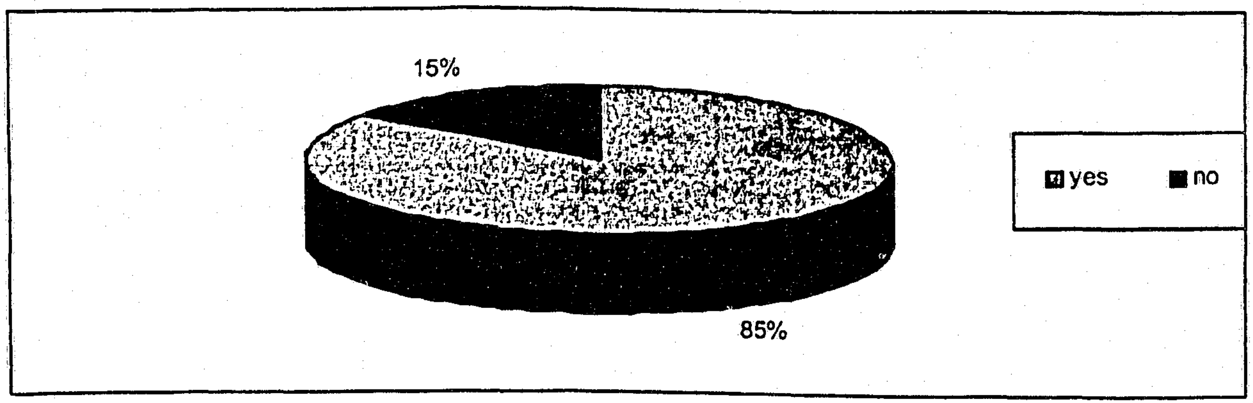


Figure 10: Data exchange issue in the companies

Question 12

Are you aware of the benefits that PLM software offers?

Awareness of PLM software is strong among the respondents. Most of the respondents know about the benefits of PLM. This suggests a mismatch between what existent PLM software provides and what users expect of it.

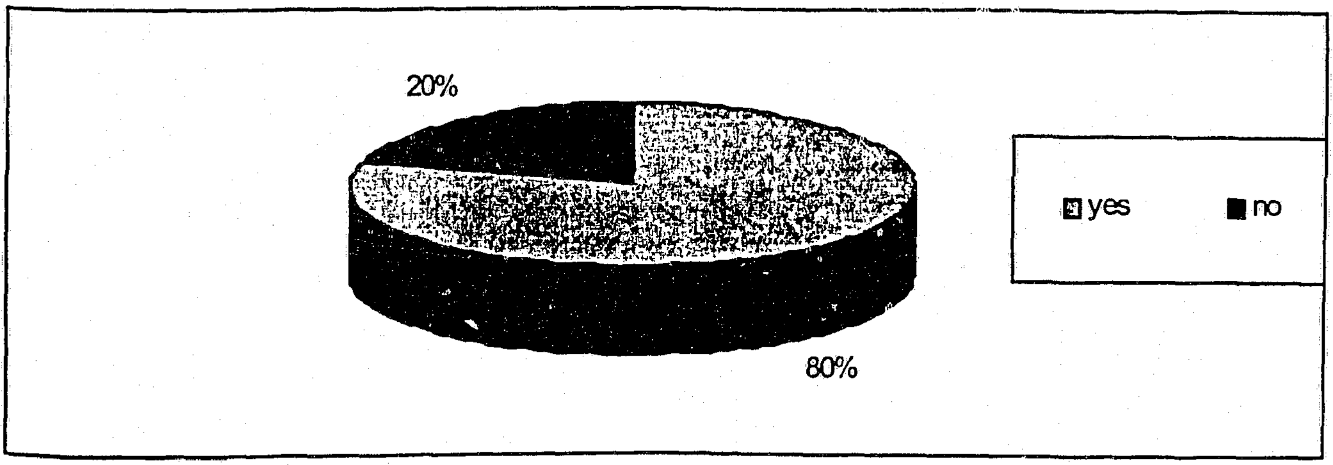


Figure 11: Awareness of benefits that PLM offers

Question 13

Have you considered purchasing PLM software?

Most of the respondents considered buying a PLM software. This indicates that the respondents are interested in achieving the benefits that PLM should provide and are proactively pursuing their attainment.

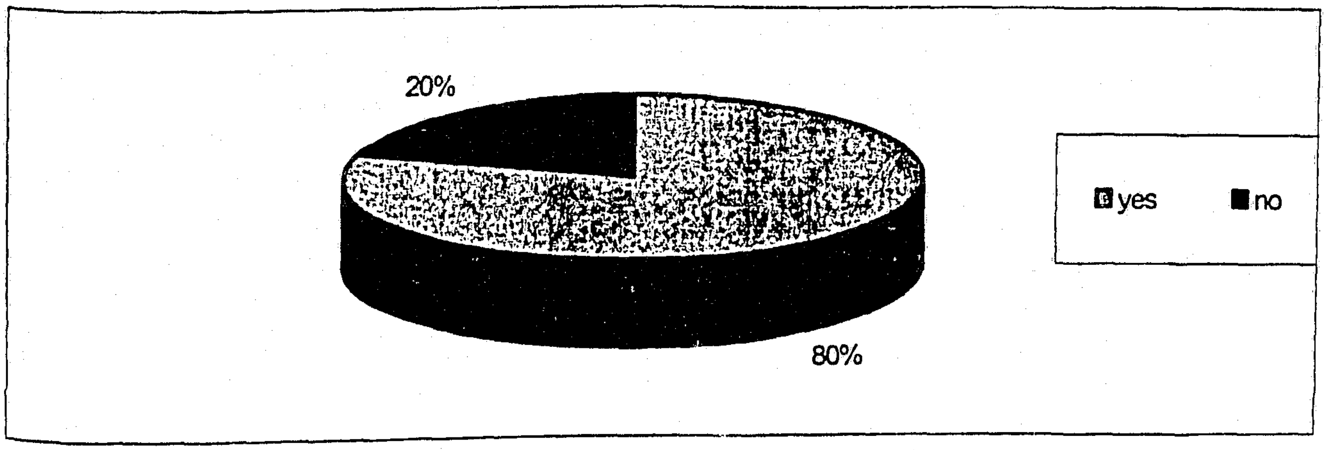


Figure 12: Procurement considerations for PLM

Question 14

If yes (to question 13), have you purchased it?

Very few companies actually purchased PLM software although the benefits are known. This reinforces the proposition that there is a mismatch between what existent PLM can provide and what users expect of it.

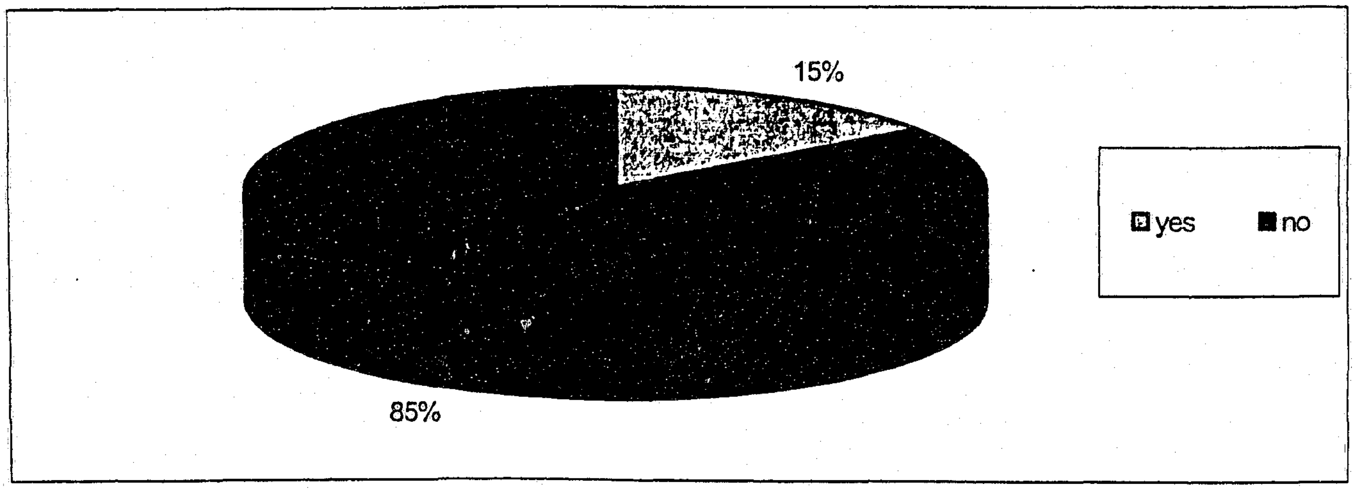


Figure 13: Number of companies who purchased PLM

Question 15

If not, why?

Of particular interest here is the even distribution of responses. This demonstrates that there are multiple obstacles – no one factor seemed to override the others. Of particular interest is the *lack of control* over the software. It appears that flexibility, extensibility, and configurability of PLM software is as significant a hindrance to its adoption as are more obvious reasons such as cost.

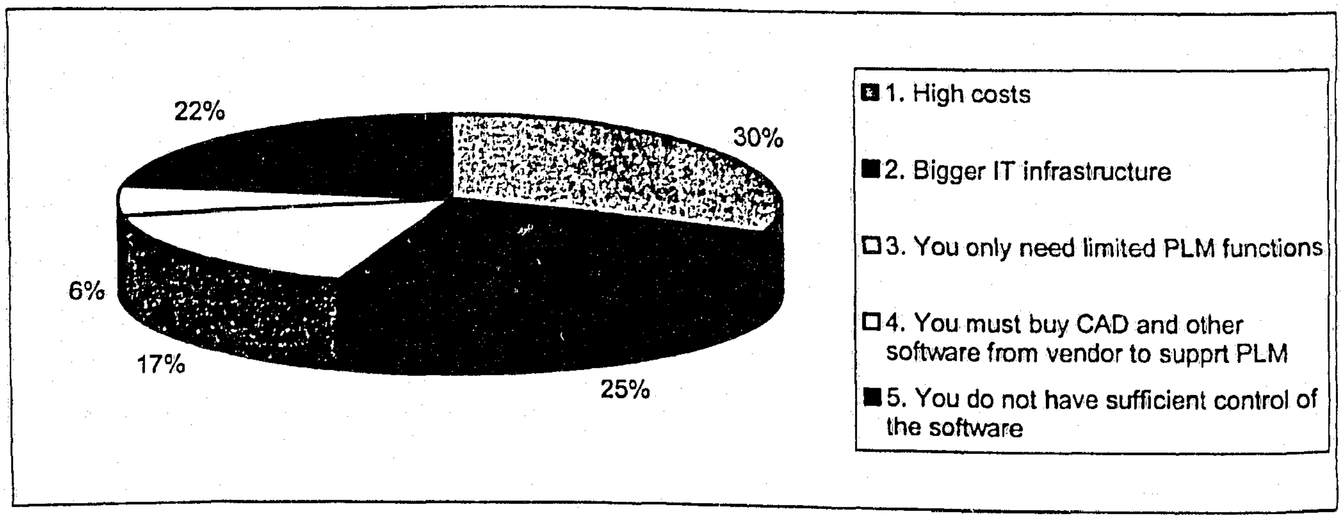


Figure 14: Reasons for inability to purchase PLM software

Question 16

We, at Ryerson University, are building Open Source PLM for SMEs. This means that SMEs would build the PLM collaboratively. In this way, this Open Source PLM would address all the needs of PLM and would be free for all SMEs. As SMEs continue using it, it is possible thta this Open Source PLM becomes the industry standard.

Would you like to be a part of this initiative?

Our initiative is welcomed with enthusiasm. 95% companies show the willingness to join OPLM development. This is extremely significant as it shows that SME's want a PLM and they must afford it.

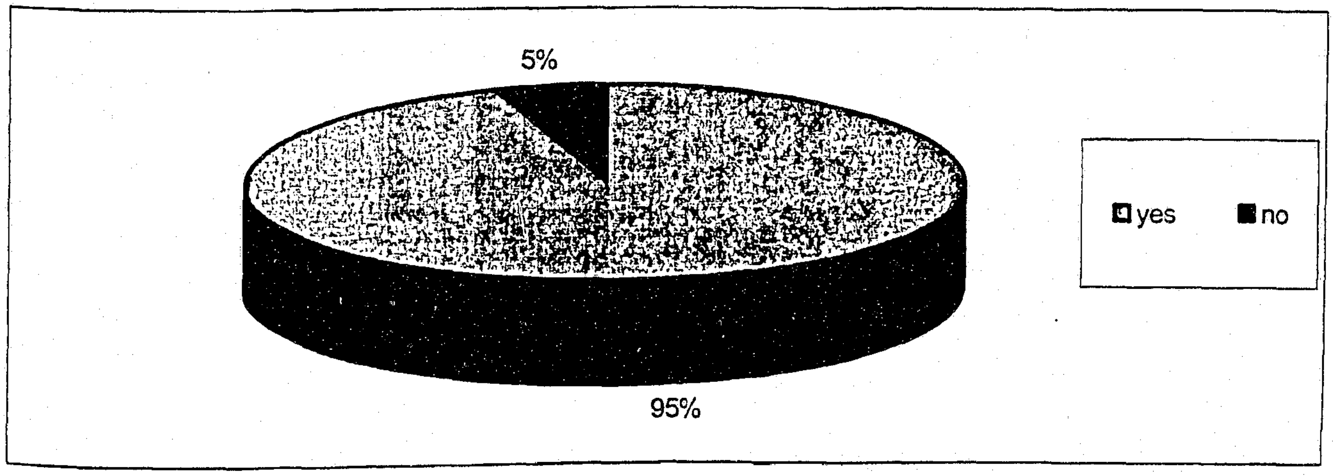


Figure 15: Number of companies who want to participate in building an Open source PLM

Individuals and companies interested in participating in the development of an opensource PLM solution may:

1. Contact Filippo A. Salustri at Ryerson University (416/979-5000 x7749, salustri@ryerson.ca)
2. Refer to the web site http://www.massdriver.com/plm.html for more information.

References

1. K. Rouibah and K.R. Caskey. 2003. **Change management in concurrent engineering from a parameter perspective.** *Computers in Industry*, 50(1):15-34.

Acknowledgments

The authors graciously acknowledge the financial support of Materials and Manufacturing Ontario, and Axiomatic Product Development, Ltd. For the work reported herein. We also acknowledge Perennial Survey Inc. who hosted our web based survey.

Appendix B

	Functional Requirements	UI subsystem	Description	
1	System must access information	YES	In order to access information, one has to use the UI to specify the nature of information.	X
2	System must display product information that is graphical, textual, organized, structured and dynamic	YES	UI must be able to display product information.	X
3	The system must interface with PDM, CRM, ERP etc. to access product information	NO	This has no direct link with with UI. This defines the interface between the OPLM system and systems outside OPLM.	
4	The system must maintain relationship in product information	NO	Relationship in product information would be maintained and contained outside the UI.	
5	~~The system must be able to compare product information.~~ THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.	~~NO~~	~~Same as above:~~	
6	The system must be able to access Internet	NO/YES?	when the system access internet, it has two types of functions: maintain the connectivity, display contents in a browser. We don not design a browser, so, the display part is taken care of. For connectivity part, UI is not involved.	?
7	The system must be secure in terms of access control, usage, external interference	YES/NO?	Security is maintained at a different level of OPLM. All the UI does is get input from user and layout the output information. Security check is done when a piece of ifnromation is fed by UI. Is there a relationshsip????	?
8	The system must track versions resulting from changes	NO	versions of information are tracked by other subsystems, not UI	
9	The system must be able to interface with email/wiki/video/teleconference etc.	YES/NO?	for email/wiki/wiki/telecon we will have other software (browser/mail) available for display purposes. To retract information from emails/wiki, UI plays no part.	?
10	The system must require little training	YES	UI must be designed such that a user needs little training. The display part of the system has to be designed like that.	X
11	The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			X
12	The system must notify changes to the users	NO	when a change occurs, the automatic message that needs to sent to the required party would be generated by version/change notification subsystem. UI is neutral here.	
13	The system must be universally searchable	YES/NO?	UI should be designed such that it allows the user to have as many options as it can be. There would be a strong search engine, is this requirement a part of that engine or UI needs customization to adopt to that engine? Unsure.	?
14	The system must be persistent	YES/NO?	by persistent we mean that the system continues to work reliably. This is true for the whole OPLM system. Is this requireent necessary?	?
15	The system must be robust	YES/NO?	Same as above. The whole system must be robust. We cannot design a subsystem less/more robust.	?
16	The system must be configurable at runtime	YES	this is true for UI and any other subsystem. If a change in source is done, it must be impleneted at runtime.	X
17	The system must support API's for new data types	NO/YES?	when a new data-type is available, the UI must be able to read it and display it correctly. UI must have this capability to display it.	X

Functional Requirements	API subsystem	Description	
System must access information	YES	API subsystem must be designed such that it can access information from external systems.	X
System must display product information that is graphical, textual, organized, structured and dynamic	YES	API subsystem will define how information from external system can be accessed/displayed by OPLM.	X
The system must interface with PDM, CRM, ERP etc. to access product information	YES	This is the purpose of the API subsystem	X
The system must maintain relationship in product information	YES	If some information of a particular product is outside OPLM, it must be accessed and updated to maintain the relationship information	X
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			X
The system must be able to access Internet	NO	No relationship with API subsystem itself.	
The system must be secure in terms of access control, usage, external interference	YES/NO?	same as box 7U	
The system must track versions resulting from changes	NO	same as box 8U	
The system must be able to interface with email/wiki/video/teleconference etc.	NO	Not related with API subsystem.	
The system must require little training	NO	Operation of API subsystem would not engage a user	
The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			
The system must notify changes to the users	NO	Not related with API subsystem.	
The system must be universally searchable	NO	Not related with API subsystem.	
The system must be persistent	NO		
The system must be robust			
The system must be configurable at runtime	YES	When adding or updating new API's, the system must incorporate them at runtime	X
The system must support API's for new data types	YES	Obviously	X

Functional Requirements	Security Subsystem	Description	
System must access information	YES	Security subsystem ensures security and access to information. Access to information must not be unlimited for all users.	X
System must display product information that is graphical, textual, organized, structured and dynamic	NO	Displaying information is not related with security (displayable info is already security-checked)	
The system must interface with PDM, CRM, ERP etc. to access product information	YES	While exchanging information with an external system, the security subsytem must allow such access	X
The system must maintain relationship in product information	NO	Not related with security subsystem. Retaining [roduct information is something outside security.	
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			
The system must be able to access Internet	YES	Obvious. All contents must be checked for harmfulness. This may be done in conjunction with firewalls, anti-virus tools etc.	X
The system must be secure in terms of access control, usage, external interference	YES	Obvious. All contents must be checked for harmfulness. This may be done in conjunction with firewalls, anti-virus tools etc.	X
The system must track versions resulting from changes	NO	not related with security subsystem.	
The system must be able to interface with email/wiki/video/teleconference etc.	YES	all such materials must pass security before they are allowed for further processing	X
The system must require little training	NO	Security subsystem does not interact with the user	
The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			
The system must notify changes to the users	NO	change notification tool processes the information that is already security checked	
The system must be universally searchable	YES	The search engine may or may not show the 'search results' for items that the user may otherwise be allowed/disallowed by the system	X
The system must be persistent			
The system must be robust			X
The system must be configurable at runtime	YES	All new configuration must be done at runtime	X
The system must support API's for new data types	NO	not related	

Functional Requirements	Revision control	Description	
System must access information	NO	Revision is not required when information is accessed	
System must display product information that is graphical, textual, organized, structured and dynamic	NO	Displaying information need not be versioned	
The system must interface with PDM, CRM, ERP etc. to access product information	NO	not related	
The system must maintain relationship in product information	YES/NO?	in case of assemblies or manufacturing information, revision plays its part separately as the relationship part. Not connected.	?
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			X
The system must be able to access Internet	NO	Contents on internet would not be versioned	
The system must be secure in terms of access control, usage, external interference	NO	not related	
The system must track versions resulting from changes	YES	Obvious	X
The system must be able to interface with email/wiki/video/teleconference etc.	NO	email/wiki/teleconference etc. are mostly time-specific operation	
The system must require little training	NO	not related	
The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			
The system must notify changes to the users	NO	This could be an afterwork of version control	
The system must be universally searchable	NO	not related	
The system must be persistent			X
The system must be robust			X
The system must be configurable at runtime	YES	any change/update to the subsystem must be done at runtime	
The system must support API's for new data types	NO	not related	

Functional Requirements	Internet access	Description	
System must access information	YES	internet can be used to access information	X
System must display product information that is graphical, textual, organized, structured and dynamic	NO	Internet access subsystem extablishes connections, keeps the communication etc.	
The system must interface with PDM, CRM, ERP etc. to access product information	NO	Not related	
The system must maintain relationship in product information	NO	Not related	
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			
The system must be able to access Internet	YES	obvious	X
The system must be secure in terms of access control, usage, external interference	YES	obvious	X
The system must track versions resulting from changes	NO	not related	
The system must be able to interface with email/wiki/video/teleconference etc.	YES?NO?	email/wiki etc. require internet connectivity. But they are external systems who already have the ability to find the internet.	?
The system must require little training	NO	not related	
The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			
The system must notify changes to the users	NO	not required, because contents that are available on the internet cannot be changed unless it is stored on the disk	
The system must be universally searchable	YES	An extended function of the search engine as well	X
The system must be persistent			
The system must be robust			X
The system must be configurable at runtime	YES	Same as others	
The system must support API's for new data types	NO	not related	

Functional Requirements	Information storage	Description	
System must access information	YES	If the user wants access to stored information, this is required	X
System must display product information that is graphical, textual, organized, structured and dynamic	NO	Not related	
The system must interface with PDM, CRM, ERP etc. to access product information	NO	Not related	
The system must maintain relationship in product information	YES	Product relationship is a piece of informtion that must be stored	X
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			X
The system must be able to access Internet	NO	Not related	
The system must be secure in terms of access control, usage, external interference	NO	Not related	
The system must track versions resulting from changes	NO	Function of version control	
The system must be able to interface with email/wiki/video/teleconference etc.	NO	interfaces	X
The system must require little training	NO	Not related	
The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			
The system must notify changes to the users	NO	Not related	
The system must be universally searchable	NO	Not related	
The system must be persistent			X
The system must be robust			X
The system must be configurable at runtime	YES	same as other subsystems	X
The system must support API's for new data types	NO	information storage is not concmed with API's with new data types	

Functional Requirements	Search subsystem	Description	
System must access information	NO	Not related	
System must display product information that is graphical, textual, organized, structured and dynamic	NO	Not related	
The system must interface with PDM, CRM, ERP etc. to access product information	NO	Not related	
The system must maintain relationship in product information	NO	Not related	
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			
The system must be able to access Internet	NO	Not related	
The system must be secure in terms of access control, usage, external interference	NO	Not related	
The system must track versions resulting from changes	NO	Not related	
The system must be able to interface with email/wiki/video/teleconference etc.	NO	Not related	
The system must require little training	NO	Not related	
The system must be intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			X
The system must notify changes to the users	NO	Not related	
The system must be universally searchable	YES	core function	X
The system must be persistent			
The system must be robust			
The system must be configurable at runtime	NO	Not related	
The system must support API's for new data types	NO	Not related	

Functional Requirements	Information associativity	Description	
System must access information	NO	Accessing is not related with association	
System must display product information that is graphical, textual, organized, structured and dynamic	NO	display is not related	
The system must interface with PDM, CRM, ERP etc. to access product information	YES	it might be possible that a product has some information to associate with from external systems	X
The system must maintain relationship in product information	YES	Core function	X
~~The system must be able to compare product information. THIS REQUIREMENT IS SIMILAR TO REQ#4 BECAUSE WHEN THE RELATIONSHIP IS MAINTAINED FOR A PRODUCT, IT WOULD AUTOMATICALLY HAVE THE ATTRIBUTES THAT WOULD MAKE THE COMPARISON OBVIOUS.~~			X
The system must be able to access Internet	NO	Not related	
The system must be secure in terms of access control, usage, external interference	NO	Not related	
The system must track versions resulting from changes	NO	Not related	
The system must be able to interface with email/wiki/video/teleconference etc.	NO	Not related	
The system must require little training	NO	Not related	
The system must be Intuitive. IS THIS SAME AS REQ#10? 'INTUITIVENESS' HAS TO DO WITH THE DISPLAY, THAT IS THE UI. WE CAN CONSIDER INTUITIVE AND LITTLE TRAINING ALMOST THE SAME.			
The system must notify changes to the users	NO	Not related	
The system must be universally searchable	NO	Not related	
The system must be persistent			X
The system must be robust			X
The system must be configurable at runtime	NO	Not related	
The system must support API's for new data types	NO	Not related	

Bibliography

[1] **i-MB CAD Chronology**; website: http://mbinfo.mbdesign.net/CAD1960.htm. Last visited November 2003

[2] **Wikipedia, the Free Encyclopedia**; website: http://en.wikipedia.org/wiki/Main_Page, Last visited July 2005

[3] **Four Problems with ERP, The Enterprise System Spectator**; Website: http://fscavo.blogspot.com/2004/05/four-problems-with-erp.html, Last visited January 2005

[4] **Definition of PLM by T. Affuso**; Website http://www.ugs.com/about_us/newsletter/perspective/2004_june/exec_outlook.shtml , Last visited July 2005

[5] **Definition of PLM by Willium Berutti**; Website: http://www.cambashi.co.uk/research/plm_debate/plm_scm.htm , Last visited January 2005

[6] **Definition of PLM by Alan Christman**; Website: http://www.cimdata.com/PLM/plm.html , Last visited July 2005

[7] **Definition of PLM by IBM**; Website: http://www.3ds.com/index.php?id=95 , Last visited February 2004

[8] **Definition of PLM by Datamation Ltd.**; Website: http://www.datamation.co.uk/datamation/dmabout_press_press_releases_eol04.htm , Last visited August 2004

[9] **Definition of PLM by EDS; Website:** http://www.i-deas.hu/brossura/TC_visualization_brochure.pdf , Last visited March 2004

[10] **Definition of PLM by John Stark**; Website: http://www.johnstark.com/2PLM181.html , last visited June 2005

[11] **Definition of PLM by University of Michigan PLM Development Consortium**; Website: http://plm.engin.umich.edu/PLMDC-TR3-2004.pdf , Last visited April 2005

[12] **Definition of PLM by Bruce Jenkins**; Website: http://www.daratech.com/summit2002/main.html , Last visited July 2004

[13] **Product Lifecycle Management**; John Stark website http://www.johnstark.com/ , Last visited July 2005

[14] **Daratech Inc. Official Website**; http://www.daratech.com/ , Last visited July 2005

[15] **Product Lifecycle Management: What's Real Now**; *Michael Burkett, Jennifer Kemmeter, Kevin O'Marah; AMR Research, 2002*

[16] **Open Source Initiative**; Offical Website: http://www.opensource.org/ , Last visited July 2005

[17] **Open-Source Software, An Economic Assessment**; Stefan Kooths, Ph. D., Markus Langenfurth, Ph. D., Nadine Kalwey, *Diplom-Volkswirtin, December 2003*

[18] **Characteristics of open source projects**; *Capiluppi, A.; Lago, P.; Morisio, M.;* Software Maintenance and Reengineering, 2003. Proceedings. Seventh European Conference on , 26-28 March 2003 Page(s): 317 –327

[19] **Statistics Canada**; Official Website: http://www.statcan.ca/, Last visited June 2005

[20] **An Artifact-Centered Framework for Modeling Engineering Design**, F. A. Salustri, *Proc. 1995 International Conference on Engineering Design* (ICED '95), Prague, Czech Republic, pages 74 - 79.

[21] **Life Cycle Modeling: IEEE Documents**; Website: http://searchcio.techtarget.com/searchCIO/downloads/LifeCycleModelling.pdf , Last visited July 2005

[22] **Incremental model**; Website: http://www4.desales.edu/~dlm1/it532/class02/devincre.html

[23] **Information technology and product lifecycle management**; Thomas, V.; Neckel, W.; Wagner, S.; Electronics and the Environment, 1999. ISEE -1999. Proceedings of the 1999 IEEE International Symposium on , 11-13 May 1999 Page(s): 54 –57

[24] **Products lifecycle management system using radio frequency identification technology;** Soga, S.; Hiroshige, Y.; Dobashi, A.; Okumura, M.; Kusuzaki, T.; Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on Volume 2, 18-21 Oct. 1999 Page(s):1459 - 1467 vol.2

[25] **Product lifecycle based on J2EE in PDM**; Wan-Jun Yu; Da-You Liu; Ni-Ya Li; Quan Liu; Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on Volume 2, 2004 Page(s):872 - 877 vol.2

[26] **Workflow-driven collaborative session management in product lifecycle management via Internet;** Sun, D.W.; Xiong, X.H.; Ruan, L.W.; Liu, Z.J.; Zhao, J.M.; Wong, Y.S.; Engineering Management Conference, 2004. Proceedings. 2004 IEEE International Volume 3, 18-21 Oct. 2004 Page(s):1146 - 1150 Vol.3

[27] **An agent-based life cycle management for dynamic virtual enterprises;** Ouzounis, V.K.; Tschammer, V.; Computer Supported Cooperative Work in Design, The Sixth International Conference on, 2001 12-14 July 2001 Page(s):451 - 459

[28] **Study on improvement of workflow management system of PDM;** *Fan Xiao-dong; Xiong Guang-leng; Fan Wen-hui;* Systems, Man and Cybernetics, 2002 IEEE International Conference on , Volume: 3 , 6-9 Oct. 2002; Pages:5 pp. vol.3

[29] **An object model for product and workflow data management;** *Baker, N.; Bazan, A.; Chevenier, G.; Estrella, F.; Kovacs, Z.; Le Flour, T.; Le Goff, J.-M.; Lieanard, S.; McClatchey, R.; Murray, S.; Vialle, J.-P.;*
Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on , 26-28 Aug. 1998 ; Pages:731 – 738

[30] **Development of PDM framework and customization environment;** *Jeong Ah Kim; Jin Hong Kim; Namkyu Park;* Technology of Object-Oriented Languages, 1998. TOOLS 28. Proceedings , 23-26 Nov. 1998 ; Pages:40 – 49

[31] **PDM-based system for manufacturing process analysis and optimization;** *El-Mounayri, H.; Jiyu Zhang; Jie Chen;* Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Conference, 1999. Proceedings , 26-28 Oct. 1999 ; Pages:15 – 21

[32] **Managing design data: the five dimensions of CAD frameworks, configuration management, and product data management;** van den Hamer, P.; Lepoeter, K.; *Proceedings of the IEEE* Volume: 84, Issue: 1, January 1996, pp. 42 - 56

[33] **A step toward STEP-compatible engineering data management: the data models of product structure and engineering changes;** Ting-Kuo Peng, Amy J.C. Trappey; Robotics and Computer-Integrated Manufacturing Volume: 14, Issue: 2, April, 1998, pp. 89-109

[34] **STEPtools Inc.** ; Official website http://www.steptools.com/, Last visited March 2005

[35] **Usage Guide for the STEP PDM Schema**, Release 4.3, release 2002, url http://www.pdm-if.org/pdm_schema/

[36] **Web-centric product data management system within a team design problem** *Kelley, D.S.; Lucas, D.;* Information Visualisation, 2002. Proceedings. Sixth International Conference on , 10-12 July 2002 Page(s): 511 –515

[37] **Business Process Modelling and Design — A Formal Model and Methodology;** Koubarakis, M; Plexousakis, D; *BT Technology Journal* Volume: 17, Issue: 4, October 1999, pp. 23-35

[38] **Business process modelling: Review and framework;** Aguilar-Savén, Ruth Sara, *International Journal of Production Economics* Volume: 90, Issue: 2, July 28, 2004, pp. 129-149

[39] **A generic structure for business process modeling;** Fu-Ren, Lin; Meng-Chyn, Yang; Yu-Hua, Pai; *Business Process Management Journal* Volume: 8, Issue: 1, 2002, pp. 19 - 41

[40] **An agent-based architecture for analyzing business processes of real-time enterprises;** *Jun-Jang Jeng; Schiefer, J.; Chang, H.;* Enterprise Distributed Object Computing Conference, 2003. Proceedings. Seventh IEEE International , 16-19 Sept. 2003 Pages:86 – 97

[41] **Business Process Modelling and Design — A Formal Model and Methodology;** Koubarakis, M; Plexousakis, D; *BT Technology Journal* Volume: 17, Issue: 4, October 1999, pp. 23-35

[42] **IDEF website:** http://www.idef.com/, Last visited July 2005

[43] **Enterprise Modeling and Decision-Support for Automating the Business Rules Lifecycle; Rosca**, Daniela; Greenspan, Sol; Wild, Chris; *Automated Software Engineering* Volume: 9, Issue: 4, October 2002, pp. 361-404

[44] **Technical White Paper,** Syspro; Website http://www.syspro.com/PDF/bill_of_materials.pdf, last visited July 2005

[45] **Dr. Filippo Salustri'**s website: http://deed.megan.ryerson.ca/~fil/, last visited August 2005

[46] **Version control system comparison**; website: http://better-scm.berlios.de/comparison/comparison.html , Last visited June 2005

[47] **Configuration management handbook,** W.V. Eggerman, W. V. Blue Ridge Summit, PA: Tab Books, 1990.

[48] **Change management in concurrent engineering from a parameter perspective,** Rouibah, Kamel; Caskey, Kevin R., *Computers in Industry* Volume: 50, Issue: 1, January, 2003, pp. 15-34

[49] **Managing change: some important aspects;** *Small, A.W.; Downey, E.A.;* Change Management and the New Industrial Revolution, 2001. IEMC '01 Proceedings. , 7-9 Oct. 2001 Pages:50 – 57

[50] **Engineering Documentation Control Handbook, Configuration Management**; Frank B. Watts; *Second Edition,* Noyes Publications, Park Ridge, New Jersey, U.S.A.

[51] **Radically Accelerating Change Management; Optimizing the Change Process is Critical to Successful Product Development**; PTC publication, 2004

[52] **Defining the Engine Design Process,** J.C. Lockledge and F.A. Salustri, Journal of Engineering Design, 10(2) 109-124, 1999

[53] **Product Architecture Template: website: www.cc.gatech.edu/.../archDesign/sld002.htm, last visited September 2004**

[54] **A Relational Model of Data for Large Shared Data Banks**; *E. F. Codd* ; *Communications of the ACM*, Vol. 13, No. 6, June 1970, pp. 377-387. Copyright © 1970, Association for Computing Machinery, Inc. Website: http://www.acm.org/classics/nov95/toc.html

[55] **Open Source Software Engineering Tools**; Website www.tigris.org, Last visited June 2005

[56] **Computing appropriate representations for multidimensional data;** Wei Choong, Yeow; Laurent, Dominique; Marcel, Patrick; *Data and Knowledge Engineering* Volume: 45, Issue: 2, May, 2003, pp. 181-203

[57] **Business intelligence systems: design and implementation strategies;** Gangadharan, G.R.; Swami, S.N.; , 2004. 26th International Conference on Information Technology Interfaces 2004 Page(s):139 - 144 Vol.1

[58] **A software framework for developing distributed cooperative decision support systems**; website http://dicodess.sourceforge.net/, last visited July 2005

Glossary

AI	Artificial Intelligence
AP	Application Protocol
API	Application Programming Interface
APD	Axiomatic Product Development
BI	Business Intelligence
BoM	Bill of Materials
BPE	Business Process Engineering
BPM	Business Process Modeling
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CCA	Change Coordinator Agents
CMA	Change Monitor Agents
CPM	Change Propagation Matrix
CRM	Customer Relationship Management
CVS	Concurrent Version System
DAC	Design Automated by Computer
DFD	Data Flow Diagram
DICODESS	Distributed Cooperative Decision Support System
DSS	Decision Support System
ECM	Engineering Change Management
ECN	Engineering Change Notice
ECO	Engineering Change Order
ECR	Engineering Change Request

EDS	Electronic Data Systems
ERP	Enterprise Resource Planning
ETL	Extraction, Transformation and Loading
FEA	Finite Elements Analysis
FR	Functional Requirement
FTP	File Transfer Protocol
GeNIe	Graphical Network Interface
GUI	Graphical User Interface
HTTP	Hyper Text Transport Protocol
IDEF	Integrated Definition of Function Modeling
ISO	International Standardization Organization
IT	Information Technology
J2EE	Java 2 Enterprise Edition
MCO	Manufacturing Change Order
MIT	Massachusetts Institute of Technology
MMO	Materials and Manufacturers of Ontario
NPD	New Product Development
OEM	Original Equipment Manufacturer
OLAP	Online Analytical Processing
OMG	Object Management Group
OPLM	Open source Product Lifecycle Management
OSCON	Open Source CONference
OSI	Open Source Initiative

PDM	Product Data Management
PDMIG	PDM Interest Group
PIF	Process Interchange Format
PLCS	Product Life Cycle Support
PLM	Product Lifecycle Management
PTC	Parametric Technology Corporation
RAD	Role-Activity Diagram
RCS	Revision Control System
RDBMS	Relational Database Management System
RFID	Radio Frequency Identification
RID	Role Interaction Diagram
SADT	Structured Analysis and Design Technique
SAGE	Semi Automated Ground Environment
SCCS	Source Code Control System
SCM	Supply Chain Management
SDLC	Systems Development Life Cycle
SIM	System Identification Matrix
SME	Small and Medium-sized Enterprise
SMILE	Structural Modeling, Interface and Learning Engine
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
STEP	Standard for the Exchange of Product model data
TCP/IP	Transmission Control Protocol/Internet Protocol
UDDI	Universal Description, Discovery and Integration

UGS	Unigraphics Solutions
UML	Unified Modeling Language
VC	Version Control
VE	Virtual Enterprise
XML	Extensible Markup Language