www.dmu.ac.uk
DE MONTFORT UNIVERSITY
LEICESTER • BEDFORD • MILTON KEYNES

A DISTRIBUTED INFORMATION SHARING COLLABORATIVE SYSTEM

(DISCS)

NEIL ANDERSON

PH.D THESIS

December 2002

www.dmu.ac.uk
DE MONTFORT UNIVERSITY
LEICESTER • BEDFORD • MILTON KEYNES

A DISTRIBUTED INFORMATION SHARING COLLABORATIVE SYSTEM

(DISCS)

NEIL ANDERSON

SUPERVISOR: PROFESSOR H S ABDALLA

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

December 2002

ABSTRACT

Keywords: E-Commerce, Innovative Collaborative Product Development, Information Sharing, Concurrent Engineering, Virtual Teams, Object Orientation, and Object Database.

This research presents a distributed collaborative information system that facilitates virtual multi-disciplinary collaborative product development activities. Thus enabling organisations to compete in increasing global competition to deliver the right product at the right time and at the least cost. This necessitates the development and utilisation of effective collaborative computer systems that support the entire product development lifecycle.

The research in this thesis addresses the problem of distributed collaboration from the perspective of an internet based system that provides mechanisms to allow integrated information sharing from the initial concepts through to the realisation and the marketing of a product. The developed work effectively removes the barriers caused between, people with different expertise, incompatible software and hardware, and the non-standardisation of data formats.

The problem of distributed collaboration caused by ineffective knowledge sharing leads to the late detection of discrepancies and conflicts in the product development lifecycle when it is costly in terms of time and money to correct. Thus lowering development costs, improving the time to market, increasing innovation and product quality.

This research project addresses the problem of knowledge sharing in a more efficient way by the development of a novel conceptual knowledge sharing system that supports an innovative integrated collaboration framework for the entire product development lifecycle.

The major objectives of this research work are to: (1) develop an architecture that facilitates collaborative information sharing across multiple disciplines that is capable of supporting the entire product development lifecycle, (2) integrate the research into collaborative environments from the fields of Concurrent Engineering, Computing

Science and Cognitive Psychology to overcome the barriers to collaboration, (3) develop a collaborative software environment for product development, and (4) demonstrate the feasibility of distributed conceptual information sharing and structuring via a case study.

The developed system comprises of an information sharing protocol and a set of modules concept mapping, data sharing and structuring, data standardisation, object oriented database, conflict resolution, and Hypertext Transfer Protocol (HTTP) accessed document modules.

The work has been seen as a vital step forward in the attitudes and the approach taken to the distributed collaboration problem. The macro situation must first be considered and then supported through micro domain considered systems. The problems with the disparate nature of the systems of today are the result of the restricted approach of the previous research on micro sub systems that neglect concern for the holistic integration of people, processes and technology as is addressed by this research.

ACKNOWLEDGEMENTS

Those who have supported me during this research are many and fill all the facets of my life. They have provided invaluable support in the intellectual, emotional, economic and spiritual realms of my life during this time.

My indebted gratitude goes to Professor Hassan Abdalla, my supervisor for his unwavering faith, support, advice, guidance, kindness and friendship. He had a confidence in my abilities to complete this research work when mine had faltered. His wise supervision has made possible what I thought was not.

Thanks to the EPSRC's generosity for providing me the economic means to complete this research and to De Montfort University for the facilities and environment of learning where I completed my research.

Many thanks also to the members of the Concurrent Engineering Research Group who have shared my highs and lows especially, Dr M Shafik, Dr E Shehab, Mr W El-Ganzoury, Mr Kevin Tseung and Mr K Edalaw. Thanks also to Dr A. Ahmed, UMIST, and Dr A Gayretli, Nottingham University, for their valuable help. A special thanks goes to Mr R Bierton of the HCI Research Group, DMU, who provided many invaluable comments.

My personal thanks go to those who privately offered unequivocal support. My wonderfully kind and wise fiancé Samantha Fenton who words of confidence and advice guided me through the difficult times and whose laughter and congratulations brightened the highest moments, she deserves a major share of this degree and its potential benefits, helping me achieve to the best of my ability. To my parents and family who offered wisdom, support, belief and economic means and putting up with my struggles offering kind words. Thanks to my fellow PhD researchers at De Montfort University who shared the disappointments and the success, Dr A R Graves, and Mr R Bates. And finally my friends who helped me believe in myself Mr A Heathcote, Mr R Payne, Mr P Burns, and Mr N Whittle.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

SYMBOL	DEFINITION
AI	Artificial Intelligence
API	Application Programmers Interface
BAD	Brain Aided Design
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CDP	Concurrent Design Process
CE	Concurrent Engineering
CENTEX	Concurrent Needs & Technologies Experimentation
CERC	Concurrent Engineering Research Centre
COM	Common Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial off the Shelf (products).
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DBMS	Database Management System
DISCS	Distributed Information Sharing for Collaborative Systems
DMU	De Montfort University
DTD	Document Type Definitions
EDI	Electronic Data Interchange
EJB	Enterprise Java Bean
EPSRC	Engineering and Physical Sciences Research Council
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
ICWE	Integrated Collaborative Working Environment
ISAM	Index Sequential Access Methods
ISO	International Standard Organisation

IT	Information Technology
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
MONET	Multi-media System for Conferencing and Application Sharing in Distributed Systems
NURBS	Non Uniform Rational B-Spline
OMG	Object Management Group
OO	Object Oriented
OODB	Object Oriented Database
OT	Object Technology
PDES	Product Data Exchange Specification
RMI	Remote Method Invocation
SDAI	Standard Data Access Interface
SGML	Standard Generalised Markup Language
SMIL	Synchronised Multimedia Integration Language
SQL	Standard Query Language
STEP	Standard Exchange of Product
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator
VAT	Value Added Tax
VENTO	Virtual Enterprise Organiser
VM	Virtual Machine
VR	Virtual Reality
VRML	Virtual Reality Markup Language
WWW	World Wide Web
XLL	Extensible Link Language
XML	Extensible Markup Language
XSL	Extensible Style Language

Chapter 1

CHAPTER 1. Introduction

1.1. Overview

Novel product development systems facilitate high quality, flexible, cost effective, timely to market products, these are crucial elements to be competitive in a global market. The prime sphere requiring improvements to be made within current practices of product development is in the area of distributed product development team communications. Widely available technologies such as the telephone, email, and more recently the advent of instant messengers are too generic to provide an effective distributed environment for successful information sharing. Successful utilisation of distributed product development teams requires advanced communication tools and Concurrent Engineering (CE) techniques.

CE used successfully as an approach to product development sustains competitive advantage through the integration of lifecycle requirements early on in the design phase resulting in the faster and cheaper manufacture of superior quality products. The key to success is accurate, timely and effective communication from a range of distributed parties within the conceptual / design phase thus reducing errors and therefore costs.

Design information needs to be available to all concerned parties from the outset. Accurate and timely information is essential for superlative product development. Collation and communication of multifaceted requirements is fundamental to ensure that a proposed design is as accurate and correct as the available information resources allow.

It is vital that the information gathering process be undertaken at the earliest stages in the product development to ensure that the impact of accurate design decisions is maximised. Problems discovered early in the design process (particularly during the first 20% of the cycle time) are easier to solve than those discovered later (Prasad (1996)).

Important in accurate distributed communication is the group creation of collective comprehensible models. Conventional systems use complex data collation models that place a restricted domain on potential users of the technologies. The shared model needed to facilitate effective benefits requires a basis in natural human cognition avoiding unnecessary depiction complexities yet flexible enough to represent complex data.

1.2. Rationale for an Innovative Product Development Platform

Product development raises a number of design and function criteria, such as cost, quality, function, expected maintenance, and intended market. All of these areas require effective communication among a group to avoid potential pitfalls later in the development lifecycle. Focus on the accuracy and completeness of distributed information sharing at the earliest opportunity embraces the philosophy of Concurrent Engineering rather than trying to deal with what is an integrated problem in piecemeal.

The increase in collaborating organisations sharing expertise to stay competitive leaves a void where previous facilitating tools are minimal in function. Currently each tool only provides part of the solution; they lack the integrated completeness, which is in itself a prime enabling factor.

The need for distributed product development teams arises from today's highly competitive global markets. Product development teams often require expertise from around the globe; previously geographic co-location was the only mechanism available to achieve performance from such teams. However co-location can prove very expensive or even impossible. The more skilled a team member is the greater likelihood their skills are in demand; commonly they are working concurrently, on multiple projects at a time (CENTEX (1999)). Hence making co-location impractical.

The actual design costs are only a small proportion of the overall product development cost, however the design phase determines the majority of the overall production costs.

The conceptual phase represents an even smaller proportion of the overall product development cost, and has consequently a larger impact on the future product's costs, and therefore should be considered as of greater significance than is generally the case at present.

The benefits that consequently derive from an improvement in the communication of a distributed concurrent engineering team are a reduction in costs, a reduction in time to market, an improvement in business processes throughout the product lifecycle, and an improvement in the final applicability to market for the product. The entire product development lifecycle is considered in this research project providing the previously mentioned improvements.

Costs are reduced via a variety of benefits to the activating team; the enhanced communication brings about greater efficiency in terms of understanding, speed of realisation and of improvement in processes. These benefits assist in avoiding the problems, which often bring about modifications later on in the lifecycle. These changes are costly to make and would be reduced with effective communication of ideas and requirements. The time to market is effectively reduced in much the same manner, with a reduction in time-consuming changes later in the lifecycle the overall development period will be decreased.

Business processes are placed under increased scrutiny within such a communication medium, problems are highlighted with greater ease and there is sufficient means for discussion to resolve them.

Improving the applicability to market requires the involvement of the end users and customers of the product, if their input can be gathered at the stages where change can be made accordingly the final product will stand a much greater chance of satisfying the end market. Admittedly if the product has an inherently long development period when user ideals are subject to change then the emphasis of this input should be weighted correspondingly.

The target user domain is greatly expanded from the usual special technically skilled members of a project team. This is achieved via the presentation of an information structure which can be created and reviewed at varying levels of complexity from obtaining just the essence of an idea to a detailed model which could be produced and manufactured, and is hence vastly accessible. This then facilitates the inclusion of end product users, assembly teams, sales, and marketing, i.e. categories of people who have relevant input that makes a big difference if captured early on within the product lifecycle. With the more traditional and thus sequential design approach this input would arrive at a much later stage when redesign is costly, even prohibitive.

The anonymity of many within a workforce, a sense of a lack of importance and active contribution in larger corporations is often due to their lack of opportunities to contribute. This would also be improved as a side effect of involving more people early on in the design stage of a project. The sense of ownership of the project would greatly increase motivation aspects. The insight from views other than high-level management can be obtained; the value of this contribution should not be underestimated. The daily issues, which lack consideration at the initial stages then, represent barriers and problems later on in the lifecycle; these can then be rectified before development is effected.

As with the application of any new methodology or technology to a business there will be a resistance to the change. The research work of (Abdalla (1999)) reports that the companies surveyed 41% had a sufficient resistance to change. Concurrent Engineering as opposed to segregated and serial activity chains invites this period of upheaval and change. A similar situation occurs when Object Oriented technologies are employed. If a company is willing to persevere with one then the upheaval required for the other will be reduced, the attitudes and mindset for change will already have been established.

The research work of (Anderson and Abdalla (1999)) tells us that DISCS is an essential step towards realisation of e-business in an organisation or a group of organisations. The research work of (Anderson and Abdalla (2001)) and (Anderson and Abdalla (2002)) state that the system offers a wide range of benefits based upon the improvement of the communication between product development team members. This improvement in communication is derived through a linked repository of globally accessible information. The research of (Anderson and Abdalla (2002)) declares that related information, concerning conceptual analysis, design and workflow processes can all be retrieved from each other. This reinforces the understanding of the material retrieved and hence the communication of ideas between team members. If the understanding of all team members can be increased then an extra emphasis on involving a wide range of people at the earliest opportunity then a variety of common problems can be communicated to the rest of the project team and dealt with at the point of lowest cost. A lifecycle of problems can be identified at the beginning stages, improving upon the number of change requests, marketing, assembly, testing usability, through to the ecological disposal of the product.

The developed system offers the facilities to create information repositories that have real world value, throughout the entire product development lifecycle, these information repositories offer benefits in terms of reduced costs, decreased turn around times, increased team-building qualities, a reduced time to market and an increase in applicability and quality. Hence improving both production processes and business processes such as distribution.

This thesis concerns the integration of technologies and methodologies for a system, that would be most beneficial to large globally distributed corporations, with non co-located specialists or collaborating companies who cannot afford staff the time to co-locate for individual project meetings.

1.3. Aims and Objectives of the Thesis

This thesis expands on the previous work in this area, by introducing techniques to share information throughout the entire product lifecycle. Many previous works assume that the product lifecycle has progressed sufficiently to be ready for the input of design details before their systems become applicable to the problem, however design inconsistencies are likely to have already been introduced.

The core aim of this thesis is to develop and integrate both on a system and cognitive level technologies and techniques for distributed conceptual information sharing. Facilitating a diverse, comprehensive, comprehendible, shared information platform to be assembled by a distributed collective allowing the product development lifecycle to be encompassed from the earliest stages through to completion in a consistent manner.

The main objectives of the research are to:

- Design an architecture for a distributed information sharing platform;
- Develop a prototype detailing the information sharing platform concept;
- Integrate concepts from the fields of Engineering, Computing Science and Cognitive Psychology;
- Produce a flexible methodology for sharing information across multiple disciplines;
- Enable potential problems / benefits to be identified at the point of minimal change resource expenditure;
- Demonstrate the feasibility of distributed conceptual information sharing and structuring via case studies.

1.4. Organisation of the Project

The remainder of this thesis is covered in the following 7 chapters. **Chapter 2** consists of a comprehensive literature survey. The resources covered bring together the fields of Concurrent Engineering, Software Engineering, and Cognitive Psychology.

Chapter 3 describes the processes deployed in collaborative product development and e-commerce. It also details the proposed architecture for collaboration introducing the components of the research and explains a sample scenario.

Chapter 4 addresses the core area of the work, the concept mapping approach. This chapter details the rational for the approach and how it facilitates the requirements of distributed collaboration.

The collaborative system is described in **chapter 5** covering the facets of collaboration and how each component of the work addresses them. Subjects covered are the

Information Sharing Protocol, the Data Sharing and Structuring, Data Standardisation, the Object Oriented Database, Conflict Resolution and HTTP Accessed Documents.

Chapter 6 demonstrates the validation of the system and the significance of the research from the preceding chapters.

In **chapter 7** and **chapter 8** the conclusions are presented and the elements and domains that can be considered for future developments.

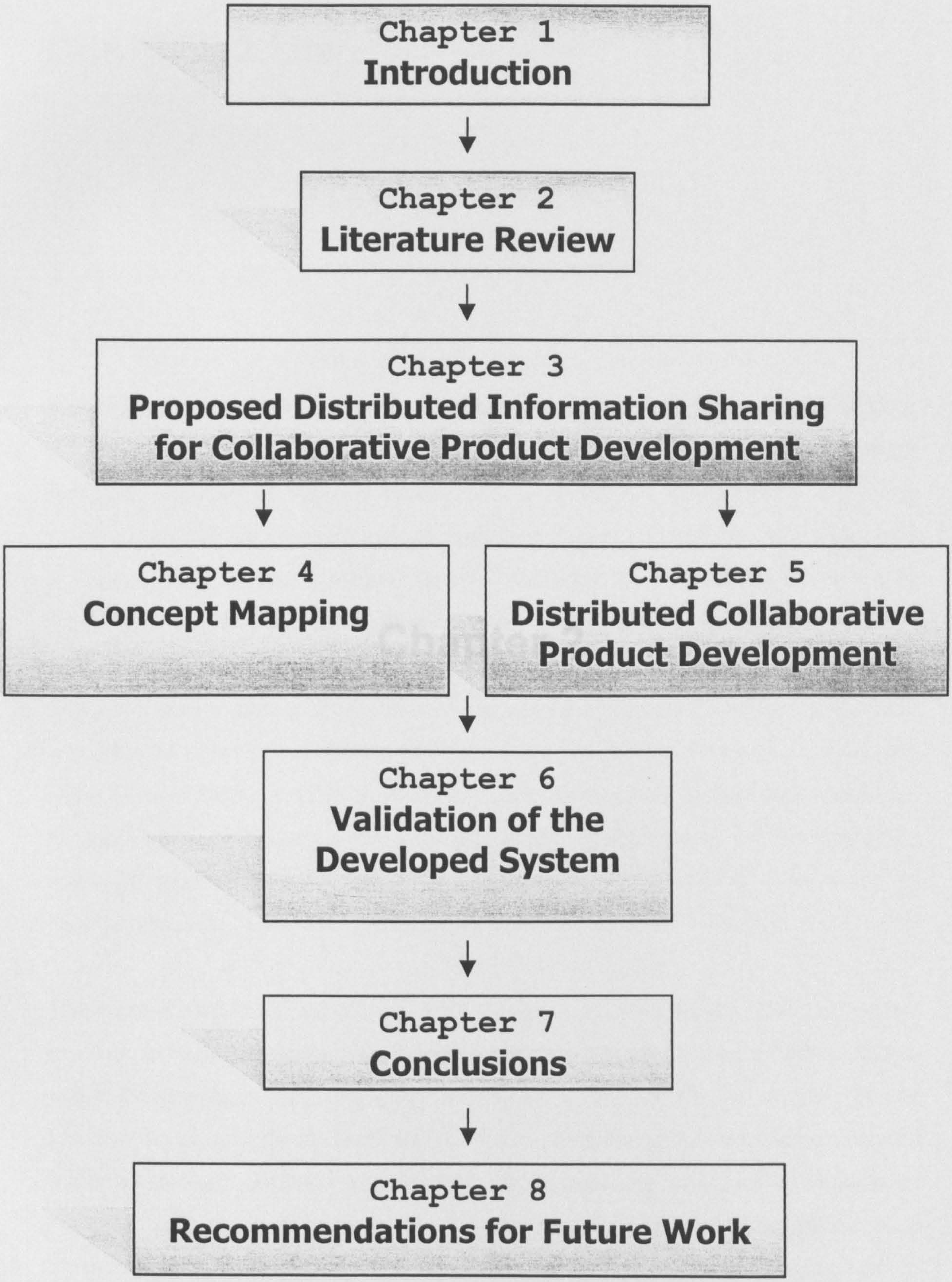


Figure (1-0-1) Thesis Structure

Chapter 2

CHAPTER 2. Literature Review

2.1. Introduction

Global competition is now prevalent in a much wider and ever growing range of markets. These markets by their very nature are extremely competitive with a wealth of factors to consider as opposed to localised competition. E-business is a globally reaching solution encompassing these increased factors of competition. This thesis provides a vital advance towards effective attainment of an e-business solution for either a single or multiple organisations. Products entering into such markets have to be appropriate, of a high quality, and have low cost to be able to compete. Products wishing to obtain such qualities to survive in these harsh market conditions require the formation of tiger teams. These tiger teams are comprised of experts from varying fields of expertise working together on high priority projects, pooling their knowledge of requirements, procedures, constraints and solutions. Tiger teams as a mechanism are highly effective when co-location is not a problem. However when co-location is not a practical option the use of the tiger team is a viable solution.

There are a variety of facilitating tools designed to enhance the productivity and performance of virtual teams. Such tools previously concentrated on a limited domain within the product development lifecycle to a high level of detail. For example the use of virtual reality to show the projected view of an envisioned product has used advanced technologies such as caves and headsets to increase the sensation of immersion. Primitive tools do exist to share data among team members and there are the early developments of tools that focus on integrating multi-functional components, such as the text mark-up for a virtual reality model.

In this chapter, the relevant previous work available in the literature concerning the area of Internet based communications and distributed product development is reviewed. The support and contribution to the research is detailed for each work concerned.

2.2. Concurrent Engineering

The following factors have been identified as requirements for success, effective communication, and a systematic involvement of customers, suppliers, distributors, powerful information infrastructure, and effective use of modern technology (Abdalla (1999)). These essential factors were revealed in the IMS-Test Case 3 "Global Concurrent Engineering". The results are from research into the best practices of an international consortium of companies from a wide range of business sectors including automotive, aerospace, telecommunications, shipbuilding and information technology.

Innovation is discussed within the research of (Ahmed and Abdalla (1999)). Innovation is most crucial at the conceptual stages of design. The administration of innovation is an aspect that is still lacking is most of today's businesses. Management are often afraid to stifle subjective and frequently intangible factors such as creativity, knowledge, learning and culture. There is a need to capture innovation and creativity in an enjoyable and non-threatening environment in suitable manner to provide management planning and structuring.

The idea of Supertitive Strategy, a practice in which companies don't just compete with others (considered a base level for survival) but raise above this onto a new level, is put forward in (Ahmed and Abdalla (2000)). Six pillars of a supporting framework are put forward, innovation, critical competencies, resources utilisation, and reward schemes, benchmarking techniques and stakeholders relationships. The first three are factors dealt with in the research presented in this thesis. Innovation is tackled through the support for the holistic environment of conceptual sharing while capturing it in an unobtrusive manner for structuring and analysis. The critical competencies required involve knowledge sharing, and effective use of information systems. Effective resource utilisation is facilitated by an accurate shared communication of the overall picture concerning requirements of various product development team members during product development.

Limited communication via application sharing can be achieved through previous works such as the Multi-media System for Conferencing and Application Sharing in Distributed Systems (MONET) project (Srinivas (1992)) undertaken by Concurrent Engineering Research Centre (CERC). MONET facilitates the creation and operation of virtual teams via an application sharing mechanism. Technologically MONET uses UNIX-BSD sockets and TCP/IP protocols to provide the communications infrastructure. Thus allowing X-Windows applications to be shared by the members of a virtual team such as CAD, art packages, and text windows. The scope of the system is restricted to a UNIX network; the portability is hence limited to within this operating domain. Control is achieved through floor control or chalk passing mechanisms. Application sharing works only to a basic level, there is no centralised repository or integrated structure to the information shared.

The need for a corporate repository concerning the design history and associated rational is identified in the research work of (Londono (1992)). The need for an integrated framework is put forward to enable information to be available to the right people at the right time.

Transparent data access through a model directed view, along with the inherent security problems, and integration with existing databases are the problems broached in the previous research work of (Jagannathan (1992)). Transparent access is achieved through the use of a model that is the mechanism the user interacts with, to obtain the retrieval of the data without the need for the knowledge of its actual physical location. This premise is essential in effective information sharing, the user should only be concerned with the data itself not with the technical requirements of where and how to retrieve or publish it.

The model driven approach is further investigated in the research by (Jagannathan (1993)). The work examines the idea that the human readable model is conceptual and does not have to be identical to the underlying schema of any of the information repositories in the enterprise. The theory of model browsing is introduced stressing the lack of a requirement for the user to know where and how the data is physically stored. The necessity for views of the relationships between components in remote repositories is concluded.

The relevance for the importance of portability to a system that intends to provide tools for a facilitating team is the focus of the previous work in (Jagannathan (1996)). The introduction of the use of the World Wide Web (WWW) as a front interface via web browsers is discussed, along with the concept of using backend processing and serving the results to a "dumb" browser is deliberated. The limitations of HTML as a mechanism are shown with conceptual wish-list recommendations given.

The difficulties of communication within a project team and the awareness of unconsidered classes of information that may not have a place within a traditional data storage mechanism are presented in the research work of (Gadient, et al. (1997)). The lack and organisation of documentation concerning the various idiosyncratic domains, for instance different manufacturing equipment is highlighted. Consideration for a receiving group's requirements is a commonly raised point, many parties simply signing off their work without concern for its interpretation.

The term distributed concurrent engineering is introduced in (Kim, et al. (1998)). This refers to not just a dispersed tiger team but also a variety of computer systems, a reference to the portability issues discussed in (Jagannathan (1996)). The research within (Kim, et al. (1998)) detailed a comprehensible set of five requirements for a distributed Concurrent Engineering system:

- The system should use open standards, to widen the participation of a variety of parties, including suppliers and customers.
- Natural communication between team members should be facilitated within such a system.
- A Database Management System (DBMS) that is capable of managing the storage and retrieval of the large amount of information typical for a reasonable sized company should be used.

- 3-D interactive graphical representations of parts and products should be available for users to manipulate.
- Effective conversions between the data stored in the DBMS, the standards used for data transfer, and the visualisation standards should be designed and developed.

The research work in (Kim, et al. (1998)) continues with a discussion of a basic mark-up technique consisting of the addition of a box of text to a part.

The evaluation of the impact of the concurrent design process on product quality costs and time to market are studied in the research work of the Concurrent Needs & Technologies Experimentation (CENTEX) project (CENTEX (1999)). A new generation radio system is the target project to be evaluated. A combination of virtual co-simulation and workflow / process flow technologies are used to create an Integrated Collaborative Working Environment. The expected benefits of CENTEX are:

- Enhanced competitiveness based on improvement of customer satisfaction, time-to-market and a decrease in R&D costs as well as recurrent costs through implementation of Concurrent Design Process (CDP).
- Leadership in scaleable CE process implementation through a pragmatic and realistic progressive implementation approach based on co-disciplines working environment (ICWE).
- A generic and modular approach of the CE process strategy and methodology implementation useable by smaller enterprises as well as large industrial companies.

Problem identification and resolution is discussed within (Krause and Doblies (1998)). The general delay in problem solving is identified as the time between problem detection and actual problem solving. This delay is attributed to the time taken to:

- Identify the people who have to be involved in the problem solving process.
- Find a common time slot for a meeting.
- Travel to the meeting place.

The travel time vanishes if it is possible to set-up a virtual meeting. Without the necessity to travel the common time slot necessary for the meeting is reduced and therefore easier to find. Identification of the important people can be facilitated by directory services and by the right composition of the virtual team. Presented also is a list of requirements for applications wishing to address the problem solving difficulty:

- Simple to use.
- Low cost.
- Directory services for all virtual team members, i.e. across multiple companies.
- Scheduling of conferences.
- Invitation of users or virtual teams to conferences.
- Ability to join and leave a conference without restarting or terminating the conference.
- Multi-point conferences, i.e. conferences between more than two persons.
- Whiteboard for general discussions, preferably with multiple pages and the ability to save these to disk as minutes.
- Sharing of CAD applications, spreadsheets, and word processors.

The benefits of Concurrent Engineering can only be leveraged depending on the successful transfer of engineering information among the participants in a design project (Vergeest and Horvath (1998)). It is pointed out that the little considered component in most modern research on exchange of CAD information concerns the exchange of the semantics and the hence the actual meaning of the data shared.

Cultural changes are required when adopting and accepting the different approach of Concurrent Engineering and especially when combined with distributed working practices. The work of (Al-Ashaab and Ruiz (1998)) confirms views upon the issue of cultural changes required when adopting a system such as proposed in this thesis. A change in work practices is required and this change should be taken into consideration. The problems arising from the use of mixed compatibility CAD/CAM systems is also discussed.

A number of considerations for developing a suite of facilitating tools for Distributed Concurrent Engineering are detailed in (Vergeest and Horvath (1998)) and (Al-Ashaab and Ruiz (1998)). There is not a fully integrated system in operation, which deals with the portability, communication, and sufficient integration aspects that are hence required for successful deployment. The most important yet least developed aspect concerns the enhancement of the semantics of the information shared, as presented in (Vergeest and Horvath (1998)).

There are significant long-standing problems disabling effective Concurrent Engineering as (Vergeest and Horvath (1999)) points out. Data exchange concerning geometry is fairly well understood, however there is frequently a mismatch between the required semantics of the shared objects involved and the semantics they possess. If the inherent semantics concerning a CAD model is not conveyed accurately its geometrical correctness partially loses its own value.

Approaches to Concurrent Engineering that focus on the slightly more traditional approach of creating iterative physical models, (Tangelder, et al. (1998)) discount their use in the problem of global information sharing. A physical model approach is valid for a co-located product development team; this technique is also of greater importance later in the product development lifecycle where the potential savings in design costs decrease.

Product Data Management (PDM) systems are usually arranged in hierarchical fashion associated with the bill of materials (BOM) type of component relationship (Baxter, et al. (1998)). This data relationship technique has proved successful with a wide range of implementers. However those organisations that are mature in their experience with PDM realise that the control, storage, and sharing of data based purely on its physical structure within the product it represents is insufficient. Organisations have discovered that they use different types of structures such as those based on function, requirements, and manufacturing processes. Often these structures are not compatible with each other.

There is a basis for argument both for and against highly formalised information sharing, the work in this thesis looks at a compromise putting forward an information sharing technique, providing structure but without restrictive rigidity. The work in (Olsen, et al. (1995)) puts forward the notion that it is difficult for standards to provide satisfaction for the information sharing needs of organisations, as these needs are dynamic. Also stated in (Olsen, et al. (1995)) are that many software tools aimed at supporting collaboration often result in impairing the ability of team members to communicate. Through requiring specific representations software tools create barriers to information sharing by excluding the specialisation and fragmentation that occurs in engineering organisations.

Currently in the aviation industry about 40% of the development costs for large aircraft are the result of collaboration overhead, harmonisation, and information exchange, (Delhaye, et al. (2000)). Best practice in the US-Aerospace Industry for example the Boeing 777 project demonstrated that development costs were reduced by 50% and development times were shortened by 35% through the integration of process management issues, human resource management issues and information and communication technologies.

Although the basis of this thesis is concerned with product development, it is obvious to consider other aspects of Concurrent Engineering such as training when deciding upon information sharing techniques. Web based training (WBT) as researched in (Benzekri, et al. (2000)) offers the same distributed information sharing scenario as that found within a product development team aiming to collaborate. This similarity is apparent from the eight concepts of a WBT platform put forward:

- Organisation: description of the general platform organisation, with its distinct function elements and how they interact.
- User Interface: the importance of information display.
- Communication: the communication between platform users as well as between a user and the platform itself.
- Network: transparency over the network implementation and configuration.
- Tracking: recording and tracking of users actions, providing trace ability.
- Evaluation: in the WBT this refers to grading and certification, in actual product development this equally applies to the iterative process of concept evaluation.

- Course Structure: in the WBT this applies to the reuse of course modules and documents, within product development reusability and accessibility of documents is paramount.
- Course development: in the WBT, development is referring to a defined methodology allowing communication at various development levels of courses. Allowing communication in product development at levels extending throughout the lifecycle is vital to ensuring consistency and utilisation of resources.

The research work in (Vankan, et al. (2000)) advocates a combination of Concurrent Engineering with that of System Engineering, utilising the theories of CE with the product boundary as expressed in SE. This approach draws attention to the top down methodology of SE and the parallelisation of CE, techniques that are drawn together within this thesis.

Early supplier involvement is essential in Concurrent Engineering focussed product development. The advantage of the Japanese in their concept to market time capability has been attributed to the early involvement of suppliers. A system named WeBid aimed at facilitating this is described in (Huang, et al. (2000)). A list of benefits from the suppliers that are integrated into the product development lifecycle at the beginning are given in (Huang, et al. (2000)):

- Reduced development costs.
- Standardisation of components.
- Visibility of the cost performance trade off.
- Consistency between design and suppliers process capabilities.
- Reduced engineering changes.
- Higher quality products with fewer defects.
- Refinement of the supplier's processes.
- Availability of detailed process data.
- Reduced time to market.
- Early identification of technical problems.
- Reduced supplier's process engineering time.
- Improved supplier innovation.

The research in (Chen and Li (2000)) states that although Concurrent Engineering can be instrumental in faster product delivery from conception, the act of concurrency introduces overlaps and design process complexity. The work puts forward the decomposition of tasks into more tractable design undertakings that can then be tackled by a design unit.

The following sections show this may well hold the key to a truly integrated system, not only technologically but also cognitively.

2.3. Collaborative Product Development

Concurrent Engineering can be implemented in a diverse variety of ways. The utilisation of the virtual enterprise allows organisations to gain the benefits of CE through a distributed workforce, a primary symptom of today's globalisation.

VENTO (Virtual Enterprise Organiser) (VENTO (1998)) points out a few issues dealing with specific Virtual Enterprise requirements that should be taken into consideration in addition to those of CE implementations. VENTO considers the users needs in terms of:

- Adaptation ability to the varying requirements of the participating companies.
- Flexibility in the integration of new partners.
- Simple modification of existing modules and exchange among the partners.
- Support of geographically distributed applications and databases.
- Protection of the information from unauthorised access.

VENTO appears to be one of a few projects to consider the issue as a whole and an integrated problem domain, as opposed to separate localised solutions.

The work in (EDItEUR (1995)) presents a general background to Electronic Data Interchange; its most useful contribution is that concerning high-level management procedures and attitudes, which have an equal applicability to any Virtual Enterprise implementation:

- Obtain commitment from key management.
- Establish a plan.
- Set up a project team.
- Appoint a project manager.
- Determine EDI business contact(s) within the company and at intended trading partners, and EDI technical contact(s) within the company, at EDI software and network service suppliers, and in intended trading partners technical departments.
- Ensure that staff involved in the project; receive appropriate briefing and / or training in EDI concepts.
- Identify legal issues for agreements with network service suppliers and with trading partners, and in respect of taxation authorities (e.g. VAT on invoices).
- Monitor progress at all stages.

The research in (Al-Ashaab, et al. (2000)) presents a virtual enterprise system named SPEED (Supporting Plastic engineering Development). The scenario of the virtual enterprise is represented as follows. Product engineering takes place in either the USA or Europe, tooling is undertaken in the Far East while manufacturing and final assembly is done in Mexico. The SPEED system supports the virtual enterprise through the capture and sharing of the manufacturing process and resource capabilities of the injection moulding over the Internet.

For a realistic virtual enterprise system (Rezayat (1999)) states that to accomplish this task effectively, electronic access to design and manufacturing information within the extended exercise must be web based because of the universal interface, open standards, ease of use, and ubiquity. This web-based system must be incorporated with enterprises information authoring and management systems such as CAD. Integrated Product development is the key to this research work. Within (Rezayat (1999)) it is stated that 50%-80% of all components in products from Original Equipment Manufacturer (OEMS) are fabricated by outside suppliers with this trend expected to continue into the next century. To facilitate this, (Rezayat (1999)) tells us that there is the need to access information throughout a product's lifecycle by everyone associated with its design, creation, sale, distribution and maintenance.

Detailed in the work of (Krause and Doblies (1998)) is a system called WebProM Collaborative Environment (WCE). The system focuses on facilitating a wide user audience to enable the incorporation of as many companies as possible. It achieves this through a shared product library, multimedia email, and conferencing system. The main focal point is the product library. Stated in (Krause and Doblies (1998)) is that the migration to a global economy is widespread. The technologies accompanying this process are changing rapidly and implementing the latest innovation is expensive in time and money. In particular small enterprises do not have the resources required to confront this problem. To be viable, a system must not be too expensive and be easy to implement in a business environment. The research work of (Krause and Doblies (1998)) presents an overview of the requirements for an environment to distribute and exchange information:

- Simple to use, especially no special knowledge like HTML necessary.
- Low cost.
- Availability, i.e. data must be available to all authorized users all the time.
- Fast access to data, i.e. minutes instead of days.
- Data formats that can be viewed and processed by all users.
- Documents organised in net like structures.
- Revision control for documents.
- Gateways to existing databases and engineering data management systems.
- High security, that includes access control secure storage, and secure transmission.
- Directory services for all virtual team members, i.e. directory services across multiple companies.
- Workflow support to ensure the right person performs the appropriate action.
- Support for multiple languages.

The delay in problem solving is quantified in (Krause and Doblies (1998)) into three main time-consuming activities:

- Identification of the people to be involved in the problem solving process.
- Finding a common time slot for a meeting.
- Travel time to the meeting place.

It is argued that through the use of a virtual meeting, travel time is eliminated, and finding a common time slot is much easier without the necessity to travel involved.

As for the requirements for information sharing, similarly (Krause and Doblies (1998)) details a list of requirements for an application to assist problem solving:

- Simple to use.
- Low cost.
- Directory services for all virtual team members, i.e. directory services across multiple companies.
- Scheduling of conferences.
- Invitation of users or virtual teams to conferences.
- Ability to join and leave a conference without restarting or terminating the conference.
- Multi-point conferences, i.e. conferences between more than two persons.
- Whiteboard for general discussions, preferably with multiple pages and the ability to save these disks as minutes.
- Sharing of CAD applications, spreadsheets, and word processors.

Further to these requirements for specific stages within the product development lifecycle, (Krause and Doblies (1998)) put forward a list of requirements for the heterogeneous environment necessary for successful supply chain integration:

- Support for all major operating systems and major types of computers, i.e. workstations and personal computers.
- Support for all-important data formats being used by various application software packages.

- Support for all-important types of networks, like Ethernet, Token Ring, ATM etc.

The research work of (Haas (2000)) highlights the ideal nature of the web browser for arbitrary client / server interaction. The web browser manages user requests, provides an adaptable and configurable environment for user interaction, copes with a multitude of media formats (audio, video, virtual reality), and provides mechanisms to handle unknown media types. Sharing of knowledge through the virtual environment is a complex task, requiring changes in culture in a large organisation. Two kinds of knowledge must be gathered, managed and preserved; explicit knowledge and tacit knowledge. Explicit knowledge is available in the form of technical reports, meeting protocols, product specifications, etc. Tacit knowledge resides in the brains of the employee in the form of experience, previous rationales, best practices and lessons learned. Knowledge management is regarded as the driving force for successful operations in the future due to specialisation and the "brain drain effect" of experts from intense competition.

The popularity and growing adoption rates of virtual enterprise systems can be seen in (Clarke (1999)). ProductVision is a system from Unigraphics Solutions concentrating on the visualisation of CAD data without the need for expensive training. General Motors purchased between 20,000 and 30,000 seats of the software.

Another marketed product is CoCreate's OneSpace, (MacKrell (2001)). OneSpace is based on a virtual conference room, shared CAD models and mark-up of data in a real time environment. This is aimed at in the same manner as ProductVision in providing facilities for viewing CAD data in an inexpensive environment by people without CAD training, helping include people who would otherwise be excluded from contribution by not having the appropriate IT skills. This mechanism expands on the scope of those that can contribute to the design hence helping facilitate a virtual enterprise and CE practices, however systems such as OneSpace are still not dealing with the most cost effective point of information sharing, instead providing solutions that have relevance later in the product development lifecycle.

It is the lack of IT training of possible design contributors that systems such as OneSpace and ProductVision are circumnavigating. This is reiterated in the research work of (Anderson and Abdalla (1999)) where the utilisation of information structures similar to that of our own internal cognitive representation aim to reduce the barriers to information access.

Introduced in (Huang, et al. (1999)) is the term Design for X (DFX), this relates to an umbrella of philosophies such as design for assembly, design for environment and others. DFX hold essentially many similarities with CE. The research work in (Huang, et al. (1999)) points out that many of the systems aimed at DFX and CE are standalone, a vital limitation inconsistent with the DFX tenet of fostering teamwork and collaboration. The work shows a DFX shell system extended to provide a web-based interface. Using a bill of materials (BOM) structure to manage the product components.

The research work of (Roy, et al. (1997)) is based on the sharing of VRML based CAD models that provide a basis for visual presentation and further analysis. This work asserts that collaborative product development systems do not assist in the conceptualisation, generation, specification and refinement of product designs in a collaborative fashion on the WWW. The work is aimed at providing a solution to this via the CAD model sharing. At the time of the research the frameworks in existence provided multimedia support for exchange of graphic images of components, but did not facilitate their modifications and evolutions. It is put forward that the corporate intranet initiatives have been in the direction of databases designed only to be information resources and not do assist in the activities of collaboration. Even though this research makes progress on this situation it still provides an information sharing and management system based later in the product development lifecycle than would derive the greatest benefits.

Further research based on the sharing of design information primarily as solid models can be found in (Chang, et al. (1999)). However the research does highlight the useful consideration that most users do not have access to the high bandwidth Internet or use a common operating system and so virtual collaboration has to provide mechanisms for coping with these technical limitations. For the purpose of the research in (Chang, et al.

(1999)) Java and VRML are considered suitable technologies for providing an effective level of platform independence.

The concept of the Virtual Project Office (VPO) is introduced in (Pallot (2000)). The integration of organisations poses problems in that they each have their own developed project management systems and practices. It is proposed that changes when adapting to a common system for collaboration by answering to the customers needs should be necessary in order to allocate the project objectives. Such a system should (Pallot (2000)): "*dramatically increase the creativity and innovation potential that will soon become two of the major factors of competition to be successful on the global market*".

The CyberEye system is explained in the research work of (Zhuang, et al. (2000)). One of the important considerations is that of both synchronous and asynchronous collaboration. An effective collaborative environment should take into account both of these interactions. With the domain of the virtual product development team is the global consideration of meeting time slots. Although a virtual environment makes the finding of a common time slot much more possible, to allow the design process to proceed uninhibited by the technology synchronous and asynchronous communications within the same platform must be available. This concept is clearly illustrated in the time / space matrix of (Zhuang, et al. (2000)):

	Same Time	Different Time
Same Place	**Face to face Interaction**	**Synchronous Interaction**
Different Place	**Synchronous Distributed Interaction**	**Asynchronous Distributed Interaction**

Table (2-1) Communication Space Time Matrix (Zhuang, et al. (2000))

A feature based modelling system is extended into a distributed architecture in the research work of (Lee, et al. (2001)). The work brings up the granularity of updates to client model views. This is intended to assist the problem of the current high bandwidth consumption of such CAD / CAM collaboration systems. The model is kept current on the client through a series of data sets consisting of, New, Deleted and Updated facets. In addition to this work the need for additional collaboration support is expressed through the investigation of integrating a Process Centric Engineering Design Workspace (PEDWorks). PEDWorks is based on the collaborative sharing of process design graphs that can be updated with the current state of each process.

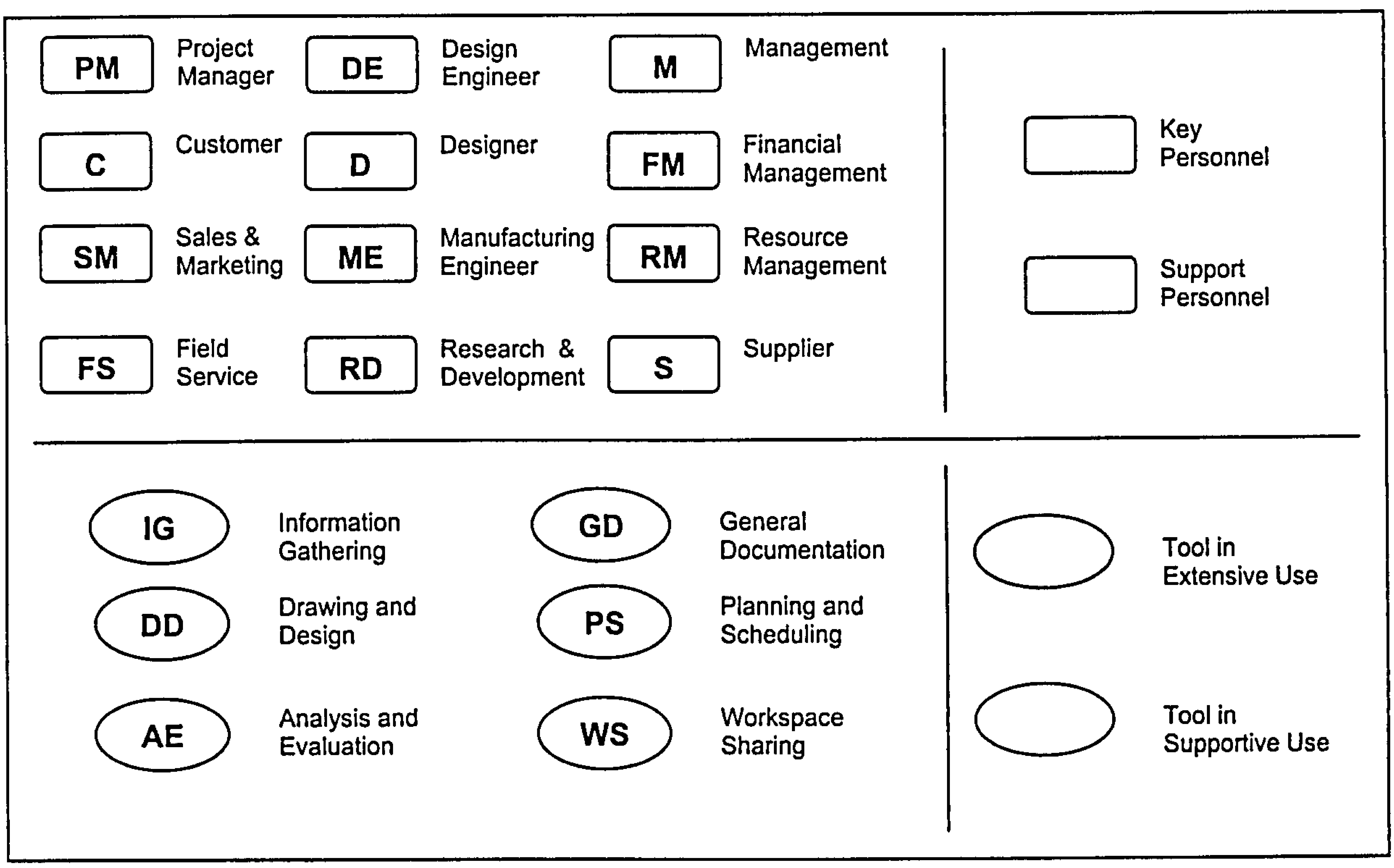
The results of an exploratory study conducted to gain insight into the practical issues of concurrent engineering within the context of a virtual collaborative environment is undertaken in the research work of (Sky and Buchal (1999)). Some of the inefficiencies discovered in a dispersed team setting include:

- Designers record background information, results of reasoning and calculations in private notebooks.

- Design information in the form of text and graphics can be captured and shared electronically, however much of the intent is lost.
- Meetings are the main method for resolving inconsistencies and design conflicts, when misunderstandings occur during meetings, they can lead to increases in development time and design costs.

The emphasis on the research in (Sky and Buchal (1999)) is to consider the importance and value of people relating this to the emerging Sociotechnical Model of Organisation. The integration of the considerations of the classes of people and tools can be seen in the adapted product life cycle model from (Sky and Buchal (1999)):

Legend:



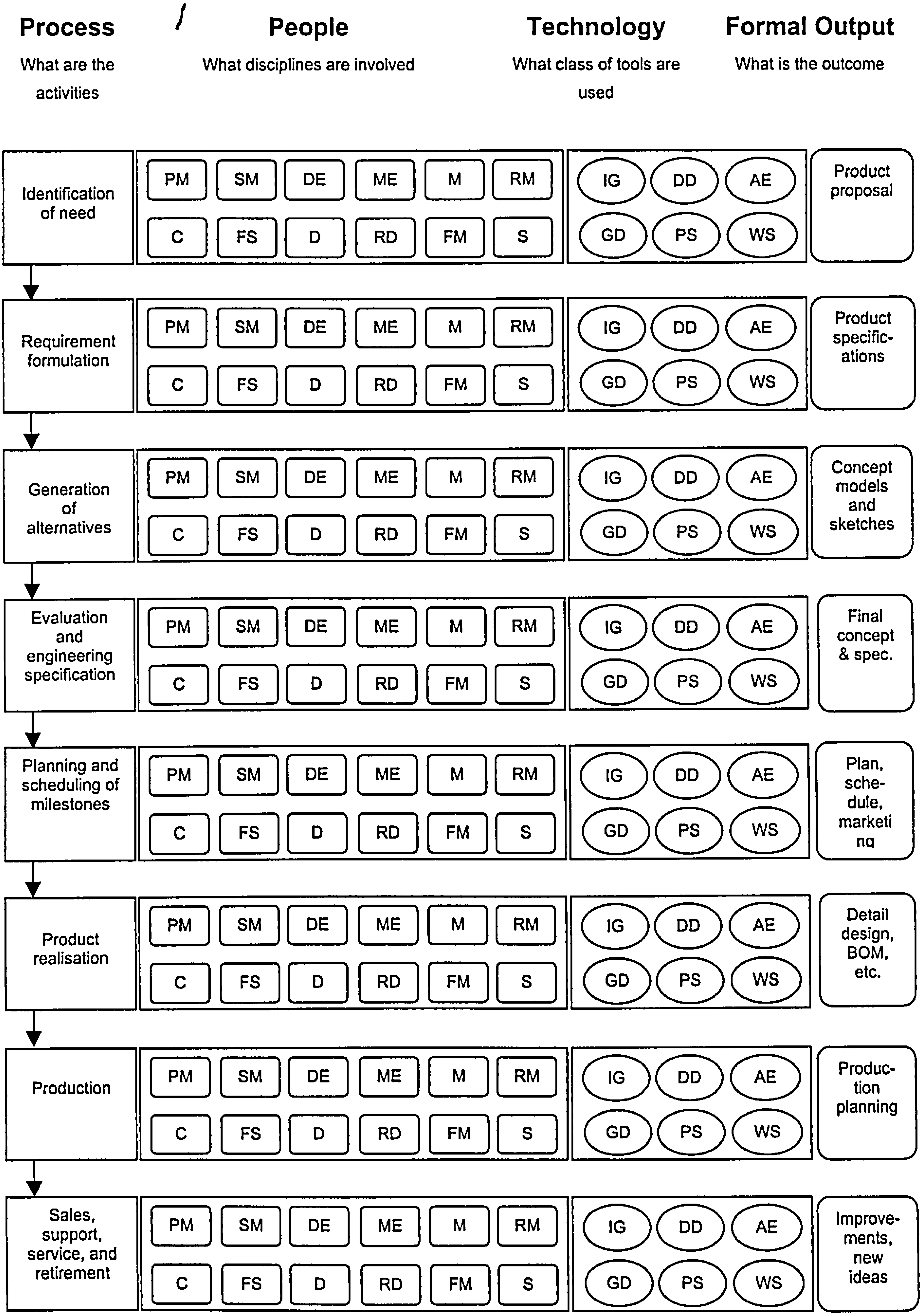


Figure (2-1) Sociotechnical Model of Organisation (Sky and Buchal (1999))

The work in (Lai, et al. (2001)) is based on a courseware system for education learning. It is important to combine efforts such as this within the domain of product development information sharing because much of the research aims have commonality. The information concerning the courses must be updated remotely, and the information must be presented in an interface in an effective manner such as to promote learning. Graphical hierarchies, show the task layout and component interaction giving an appropriate context and plan to the learning. If a learning task is done in context then it will be understood far better than an isolated undertaking.

The concept of a Virtual Design Studio is introduced in the research of (Tay and Ming (2001)). It is the integration of a variety of pre-existing tools on the desktop of 2D scratch and 3D model tools, virtual reality with CAD technologies, remote file transfers, video conferencing, and white-boards.

2.4. Concept Mapping

A concept map is similar in appearance to the more familiar brain storming diagrams. It is the representation of a group of related and interconnected concepts and their relationships to each other. These maps, for people familiar with data structure terminology, are similar to graph data structures however each node contains information pertaining to a concept. A concept consists of information describing an idea and the relationships between itself and other related concepts. It contains textual descriptions of its intent, and links activated in a similar manner to those you would find embedded within an HTML document. These links are part of the navigation system through the vast amounts of data stored within an engineering database. A concept map is productive as an information storage medium and communication mechanism because of its close relationship to how we cognitively store and process information ourselves.

Concept mapping has close links with the blackboard metaphor as used in the research work of (Lander and Corkhill (1996)). *"Imagine a group of human specialists seated in a room with a large blackboard. The specialists are working as a team to brainstorm a solution to a problem, using the blackboard as the workplace for cooperatively* *developing the solution. The session begins when the problem specifications are written onto the blackboard. The specialists all watch the blackboard, looking for an opportunity to apply their experience to the developing solution. When someone writes something on the blackboard that allows another specialist to apply his/ her expertise, they record their contribution on the blackboard, hopefully enabling other specialists to apply their expertise. The process of adding contributions to the blackboard continues until the problem has been solved."*

Stated in (Smirnov and Chandra (2000)) is that the base requirement for a collaborative system is the ability to capture knowledge from multiple domains and store it in a form that facilitates reuse and sharing.

According to (Frese (1987)) a mental model possesses two characteristics:

- It represents some part of reality.
- It is to some extent similar to that part of reality.

The model may resemble reality in three respects:

- Functions, in which case only the output of the process is modelled.
- Regards to process, in which the steps in the processes are mapped.
- Structure, in which case the structure generating the process is mapped into the model.

Stated in (Frese (1987)): *"A user of a computer system needs a mental model for planning, problem solving, communicating ideas and stimulating creative thinking. A model for communication purposes should be simple, yet expressive, comprehensive, yet understandable, use conventional means of expression, yet cover the relevant aspects of the problem."*

Knowledge representation, comprehension / understanding, memory, and creativeness are all inherently composite facets of a whole, hence the inclusion of, (Johnson (1992)) who identifies three representational families of knowledge:

- Propositional representations, these represent knowledge as a set of discrete symbols or propositions, concepts, objects and features, and relations.
- Analogical representations, i.e. mental images. These have a close correspondence between the representing and represented world. The variables, which construct this world, are assumed to be continuous e.g. pictures and maps.
- Procedural representations, this concerns the knowledge of actions and their representation, e.g. the set of actions required to ride a bicycle. This information is assumed to only be accessible through the partaking of the activity in question.

Procedural representations are considered generally out of the scope of the work presented within this thesis, except for their presentation in an extracted form from a subjective interaction with the real world by a participant.

Detailed within (McKnight (1993)) is an experiment carried out to test the effects of hypertext links in learning new material. It was discovered that some kind of mapping system was required to enable efficient and beneficial traversal of the information, as opposed to just a series of documents with links to other documents. Learning and comprehension go hand in hand, learning is a by-product of understanding. An expert in a field has previously formulated mental frameworks and schema within which to structure new information about their field. E.g. (de Groot (1965)) tells us that: "*chess masters are able to reconstruct a board position of more than 20 pieces after only five seconds study, but only if the position is legal; for randomly positioned pieces the masters perform no better than novices*". To improve a non-expert's understanding, a preview of the framework, which is required to comprehend the forthcoming information, would prove productive. Hence when sharing concepts and knowledge between distributed team members of differing domain specialities, a cognitively applicable framework for the knowledge represented is paramount to its accurate understanding.

The Bead prototype presented in (Bass (1993)) is a graphical based system for the retrieval of articles. It presents a bibliography showing the relevance between articles in terms of their graphical distance from each other, producing a map of the bibliography. The idea is to create an environment, which moves away from the query language style of interrogation. He also introduces the concept of a morphology, this is essentially the look of a knowledge representation structure, the visual attributes of which can be assigned meaning or inherently have meaning. A morphology can be assigned attributes to produce different visualisations of the knowledge structure for different users.

Various fundamental theories concerning knowledge association, recall and building are presented in (Buzan and Buzan (1995)). Buzan combines together many of the ideas expressed in the above relevant literature within the context of Mind Mapping™. In simple terms Mind Mapping™ is an expansion of brainstorming. It extends upon brainstorming by including images and key symbols in the representation; the theory for this inclusion is based upon research conducted by Ralph Haber and R. S. Nickerson who demonstrated the power of the brain to recognise images. Mind Mapping™ also strongly stresses the addition of other visual elements as is corroborated by (Bass (1993)). The Mind Map™ is in essence an image in its own right. "*The Mind Map harnesses the full range of cortical skills - word, image, number, logic, rhythm, colour and spatial awareness - in a single uniquely powerful technique.*"

Addressed in the research work of (Ahmed and Abdalla (1999)) is the process of ideas filtering for:

- Generated Ideas
- Collecting / Sorting to hold or reject.
- Checking / Ranking
- Review
- Selection of valid ideas

Described in (Ahmed and Abdalla (1999)) is the method for creation of a vision:

- Generate Ideas
- Cluster Ideas
- Refine Ideas
- Prioritise Ideas

These points are encompassed within a flexible concept representation and sharing system. Such a representation is especially adept at facilitating a concept generation and review environment.

The consideration of human factors within the domain of advanced manufacturing systems is detailed in the research work of (Udo and Ebiefung (1999)). Self-interest was discovered to be a prime consideration. The greater the interest someone has in his or her work the greater effort to succeed on that task. Increasing an employee's involvement and their perceived ability to contribute to a project increases their personal sense of ownership of a project and thus increases this self-interest. Facilitating input from a variety of people previously excluded from the early phases of the product development lifecycle makes possible an environment of interest and greater motivation.

Stated in (Heintz and Acar (1994)) is the idea that the frequency of information (such as in a group work discussion tool) speeds up convergence of ideas or interests, and that convergence affects potential for innovation positively. Also cited is that of the communication channel richness that can be used as a predictor of convergence. Hence with a variety of information depths and perspectives, convergence and thus innovation should occur quickly. Within (Heintz and Acar (1994)) it is also stated that there is an improvement in the quality of the decisions made by groups with computer support.

The importance of effective communication is also examined in the research of (Ahmed and Abdalla (1999)). They state that there are three barriers that prevent companies from having an effective communication.

- Barriers between people (effectively removed under Concurrent Engineering).
- Barriers between computer systems (due to incompatibility).
- Barriers between data exchange between companies (because of a lack of standardisation).

An essential prerequisite to successful cooperative design is communication among the participants in the design task states (Branki (1995)). Branki also puts forward the idea that computer-based tools that are aimed at supporting design, are often lacking when it comes to information sharing and coordination. These applications often create new communication hurdles for the product development team by isolating information at the boundaries of the tools themselves. This is prevalent where the tools support specific design tasks.

Introduced in (Wu, et al. (2001)) is the term Integrated Enterprise Concurrent Engineering (IECE). This is in response to the fact that much research and action has been taken in modelling and implementing CE within an enterprise but in the age of globalisation there is a great need to consider systems that extend beyond these boundaries to suppliers, connecting manufacturers and customers. A key aspect of the work in (Wu, et al. (2001)) is the development of a system that is capable of modelling the involvement of these parties in the design process, interchanging information and materials. The research of Wu is based on a variant of Petri Nets, called Trans-Nets with aim of providing an abstraction of information, removing complexities while retaining the ability to analyse the more important characteristics of the system.

Utilisation of specialised knowledge and the reduction of interpersonal friction are stated as the aim of collaboration for engineers in (Yoshimura and Takahashi (2001)). The research work here and in the research of (Danesh and Jin (2001)) is based on the sharing of mobile agents to assist designers by conveying constraints and conceptual opinions of other collaborators. This method could also be employed with concept maps having agents attached to each concept assisting the conveyance of other designer's ideas.

(Caillaud, et al. (1999)) proposes research into a knowledge-based system to minimise risks. The preposition of the research is that knowledge is a requirement for Concurrent Engineering. The work puts forward four information-sharing requirements:

- Reliability
- Accessibility

- Compatibility
- Respectful of legal aspects

The research work by (Caillaud, et al. (1999)) also states that only a part of the knowledge used in CE has to be shared to prevent domain specific knowledge clouding the issues that are involved. Concept mapping facilitates the integration of knowledge for future reuse and allows a close focus on a topic.

The research work of (Liu, et al. (2001)) states that brainstorming extends the boundaries of participant's problem spaces and elicits as many ideas as possible. It is a particularly suited method for extracting customer requirements. Liu bases requirements gathering on a series of structured interviews. A concept map could be set with a predefined framework that can be the basis of the areas of knowledge to be obtained from the customer, drawing conciseness on the issues involved.

Reported in the work of (Chan, et al. (2001)) is a system that allows multiple users to collaborate on the development of UML (Unified Modelling Language) model. The work brings in the importance of the granularity available for collaboration. With concept maps the level of user granularity can be very small. This means that users can collaborate much more effectively without having to lock large sections of data while they write their contributions. The importance of the facilitation of synchronous and asynchronous communications is stated in both (Chan, et al. (2001)) and in (Zhuang, et al. (2000)).

The method of problem reduction is discussed in (Eodice, et al. (2000)). This top down and tree structure approach is also utilised in the concept mapping. Each concept map itself being essentially a tree, however the idea that each concept map itself can become a node in an overall tree structure takes this idea a stage further.

Web based workflow is detailed within the research work of (Leong, et al. (2001)). Workflows are defined in term of graph data structures that as previously mentioned are similar to the concept map structure. With requirements and effectively milestones and goals being defined within concept maps, generating views to show workflow dependencies is a natural extension of the mechanism reducing the need for separate information workflow management systems.

A series of groupware technologies are discussed in the research work of (Gardoni, et al. (2000)) that state the need to keep track and record the problem solving process to enable the reuse of captured experience. The technologies used in the work are, e-mail, public files, forms, shared files, shared agendas, forums and workflow forms. Although these are relatively simple mechanisms for information sharing the work does discuss the advantages and disadvantages of structured and non-structured information, both of which are encompassed through the concept mapping. In fact concept mapping is a context-based mechanism for structuring what is often non-structured information.

The research work of (Lindeblad, et al. (2000)) puts forward seven tenets a computer system should satisfy for assisting in collaborative product design:

- Visualisation of the concept should be realistic.
- Designers should be able to share perspectives to the concept.
- Transferring data should be easy.
- The system should work with various kinds of equipment.
- Integration of the computer system to product design work practices.
- The designer's awareness of design process should be supported.
- Design rationale should be tracked.

Stated in (Struck, et al. (2000)), "*Usually a customer codes knowledge of his/her needs, e. g. into text, drawings, or verbal messages*". This knowledge is then subjectively coded by a supplier into useable requirements. Any subjective stages are inherently generative of inconsistencies and errors. If customer can easily use an interface to provide contextual information about their knowledge this can provide platform to reduce these information discrepancies.

The research work in (Sadeh, et al. (1998)) uses a Blackboard architecture to provide a queue of tasks or messages which require resolution that are then categorised by either the system or users into contexts. Messages can be ignored that are deemed of lower priority. This adheres more closely to a bottom up approach that allows faster problem identification but also leads to a greater chance of incomplete domain exploration and hence increases in risk.

The conceptualisation process is described in the previous work of (Hague and Taleb-Bendiab (1998)):

- The generation of solutions to meet the stated needs.
- The evaluation of these solutions to select the one that is the most suited to match the design requirements.

Also put forward in (Hague and Taleb-Bendiab (1998)) are the following acts in the conceptual design stage:

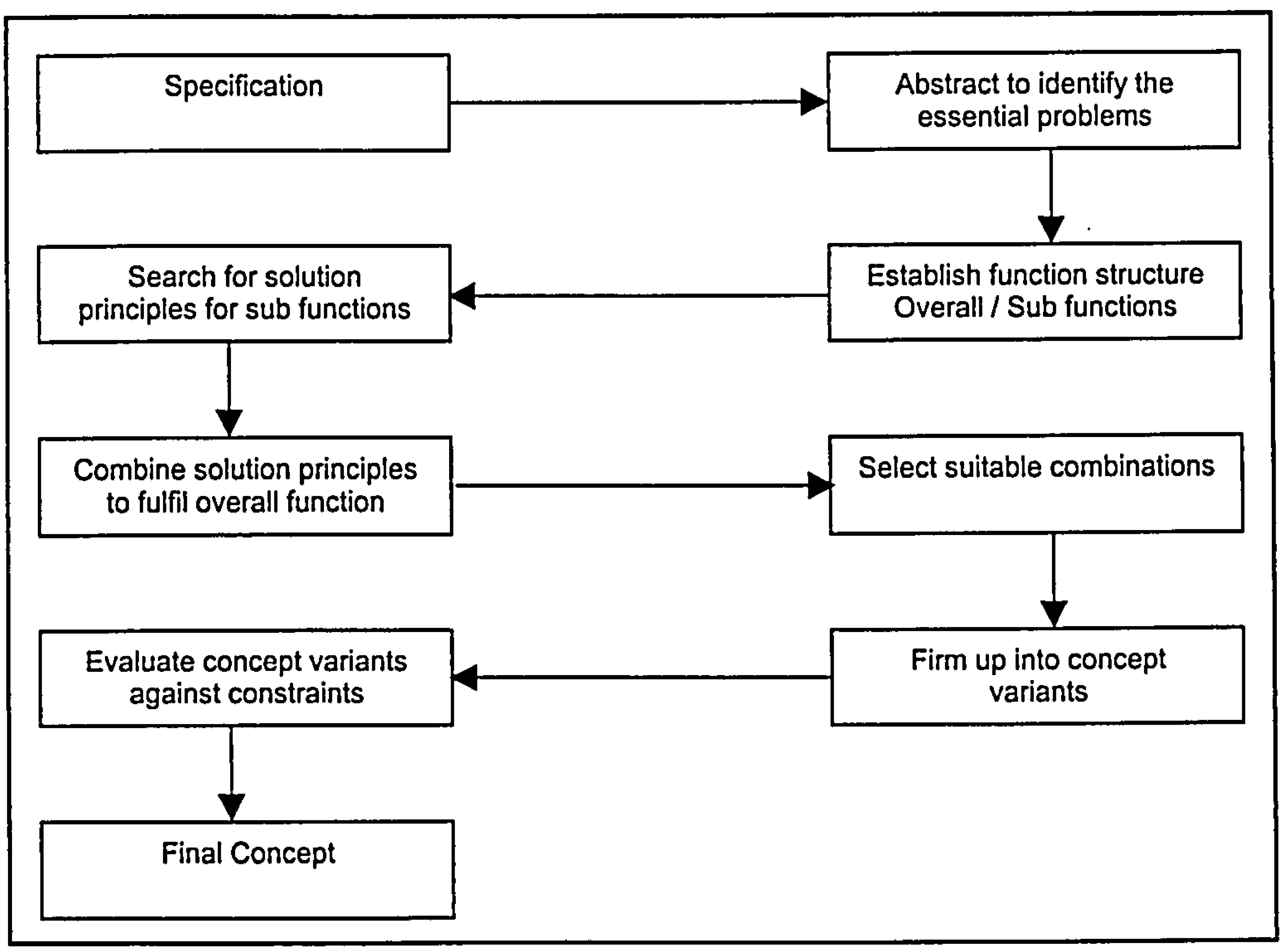


Figure (2-2) Acts in Conceptual Design (Hague and Taleb-Bendiab (1998))

Put forward in (Hague and Taleb-Bendiab (1998)): "*Fundamental to any process adopted, is the realisation that it must be as flexible as possible, so as not to negate* *what is ultimately the key to the overall design process – the designer and their expertise, knowledge and problem solving ability.*"

Innovation as the controller of value creation in business is addressed (Ahmed and Abdalla (2000)). Also stated in the research is the two techniques used in the development of new ideas; brainstorming or systematic analysis of the current system, market requirements, competitors capabilities, best practice capability and corporate manufacturing strategy.

Introduced in (Ahmed and Abdalla (2000)) is the following model of innovation:

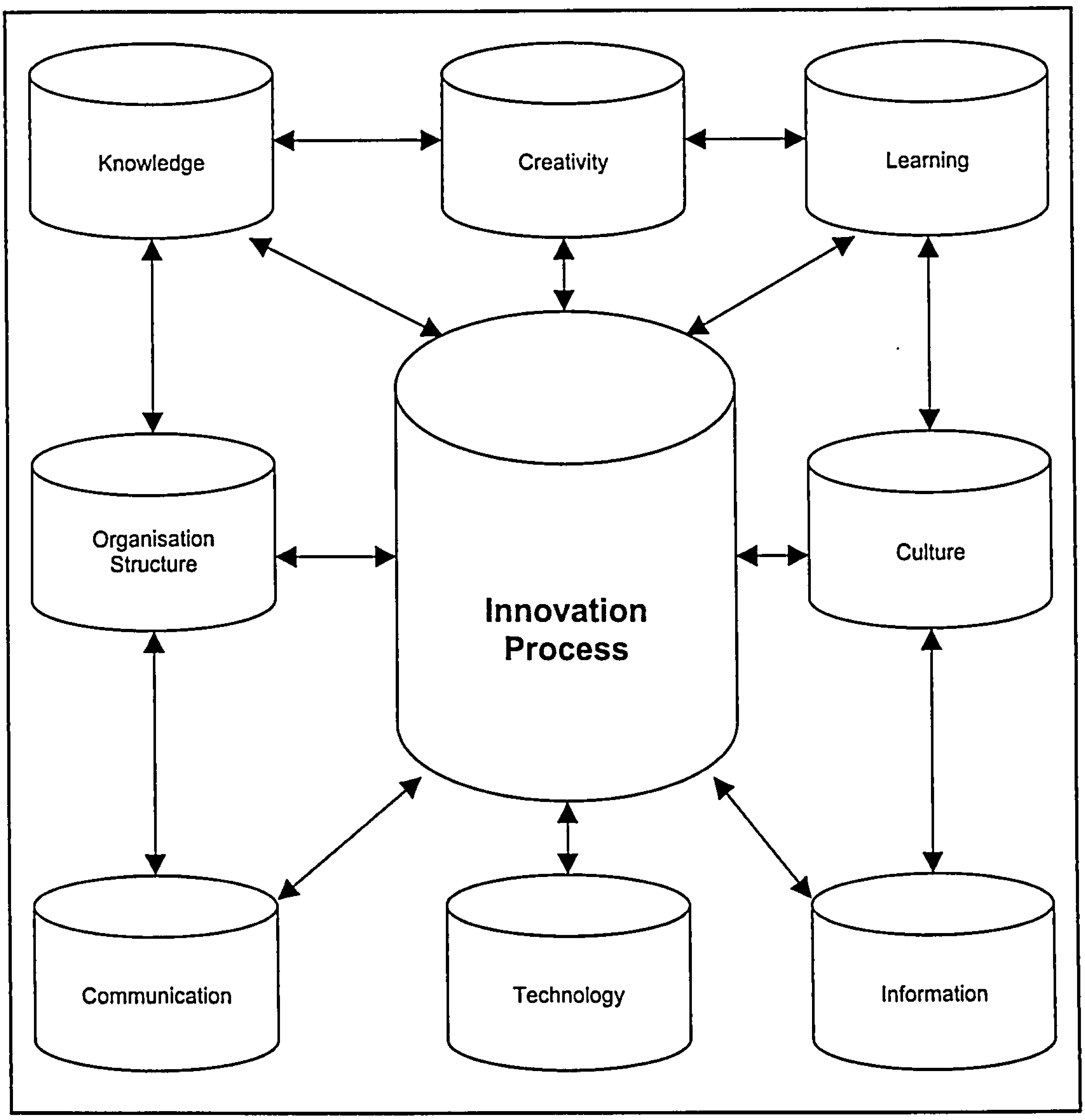


Figure (2-3) Model of Innovation (Ahmed and Abdalla (2000))

2.4.1. Concept Map Mechanism

XML is a suitable enabling language for the persistence and manipulation of the concept map data. A good introduction to the purpose of XML and the domain needs of the web is provided in (Bosak (1998)), which the simple HyperText Mark up Language (HTML) cannot provide anymore. XML differs from HTML in three major respects:

- Information providers can define new tag and attribute names when required.
- Document structures can be nested to any level of complexity.
- Any XML document can contain an optional description of its grammar for use by applications that need to perform structural validation.

Part of XML's philosophy is to provide a data creator with "*a data format that does not bind them to particular script languages, authoring tools, and delivery engines but provides a standardised, vendor-independent, level playing field upon which different authoring tools and delivery tools may freely compete.*"

An example list of the types of applications where XML would play a useful role is given in (Bosak (1998)).

- Legal publishing.
- The government drug approval process.
- Collaborative CAD / CAM efforts.
- Collaborative calendar management across different systems.
- Any corporate network application that works across databases, especially where policies must be enforced: purchase orders, expense requests, etc.
- Exchange of information between players in any broker-organised business: insurance, securities, banking etc.

Stated in (Bosak (1998)): "*a variety of XML applications are those in which users may wish to switch between different views of the data without requiring that the data be downloaded again in a different form from the Web server.*"

It is this wide scope of possibilities that will ensure the real use, support and enhancement of the technology.

DTD's are addressed in (Connolly (1998)); "*XML makes it cost-effective to capture community ontologies as Document Type Definitions to decentralise the control of specialised markup languages. The emergence of richly annotated data structures catalyses new applications for storing, sharing, and processing ideas.*" As opposed to the limited tag set you would find within HTML, through the use of DTD's, XML allows new tag definitions to be created and understood. This technology has many possible uses, one being to codify the concepts of a community so that there is a common level of understanding.

Shown in (Matthews (1998)) is a useful overview of various initiatives being undertaken with XML to demonstrate its wide-ranging applicability and value. Project ranging from MathML, an XML based method for representation of mathematics to Schematic Graphics Markup Language, a basis for representing graphical objects as opposed to binary.

In a technical document (Walsh (1999)), some examples of XML syntax are given for those familiar with HTML or Standard Generalised Markup Language (SGML). For the assistance of the readership of this document one such example is quoted below:

```
<?xml version="1.0"?>

<oldjoke>

<burns>Say <quote>goodnight</quote>,
Gracie.</burns>

<allen><quote>Goodnight,
Gracie.</allen>

<applause/>

</oldjoke>
```

The previous example is enough to understand for those who have experience of HTML, the only notable difference being the "<applause/>" tag, which due to the use of DTDs being optional requires a forward slash to denote, there is no required end tag. Much like the <HR> tag in HTML. Described in (Walsh (1999)) are the six kinds of markup that can occur within an XML document; elements, entity references, comments, processing instructions, marked sections, and document type declarations.

Ref: CR: 19981006 (1998) is a brief document listing the industry initiatives under way using XML as their basis. They number near 100 at the time of the release of this document, again underlining the potential of this technology, for the structuring of information.

The work of (Bray (1998)) discussed the important ability of XML to decouple the presentation from the content and the power of embedding meaning within the document. The example of XML markup of email texts is given, showing how a table of contents could be derived from such marked up content. There is also mention of the controversial decision to ignore poorly formed XML documents. That is to say that there is to be no tolerance given, no XML processor is allowed to recover, to try and ascertain what was the intent of the author as is the case with many of the web browsers concerning HTML today.

Posed in (Rees (1998)) is the most positive and simple reason for the use of XML, its combination of power and portability. SGML has the power, HTML the portability, however XML works as a combination of both. The following is the vision of the people involved in developing XML:

- XML shall be straightforwardly usable over the Internet.
- XML shall support a wide variety of applications.
- XML shall be compatible with SGML.
- It should be easy to write programs that process XML documents.
- The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
- XML documents should be human-legible and reasonable clear.

- The XML design should be prepared quickly.
- The design of XML shall be formal and concise.
- XML documents shall be easy to create.
- Terseness in XML markup is of minimal importance.

Presented in Connolly (1997) is a section on the Synchronised Multimedia Integration Language (SMIL). This is an application of XML and schedules multimedia presentations on the WEB. Screen locale and time are attributable parameters.

Introduced in (Lie and Lilley (1998)) is the eXtensible Style Sheets Language (XSL). XSL is an enhanced language built upon foundations of the Cascading Style Sheets (CSS) language. XSL allows a more dynamic approach; CSS would give the capability to render paragraphs etc, whereas XSL could generate a table of contents. XSL works by pattern matching an element in the document parse tree then mapping to objects, which can be rendered in the document.

The capabilities of XSL are addressed in (XML (1998)). For example the reordering capability, XSL allows a document to be reordered without requiring a subsequent download. There is greater context sensitivity, XSL takes into account all the ancestors, descendants and siblings of an element allowing a higher degree of flexibility based on the context of an element within a document.

(XML (1998)) also introduces eXtensible Link Language (XLL), this is the enhancement of the common "<A HREF=" statement found in HTML. XLL allows compatibility with URL linking, bi-directional links, addressing within an XML hierarchy and indirect links; providing an intermediate document resulting in greater maintainability when updating links.

The use and integration of parsers for XML can be found in (St. Laurent and Cerami (1999)). A parser is required to extract the data from an XML document in a form suitable for processing. Parsers come in a variety of flavours, validating and non-validating, tree-based and event driven. A non-validating parser in essence has the purpose of checking the tag structure is well formed. A validating parser will check the

XML against a corresponding DTD for correctness. The data from an XML document is accessed through the interface to a parser. This can be in an event driven structure similar to that of event handling found in various programming languages such as Java and in MFC (Microsoft Foundation Classes) of Windows. Each item of data triggers an event that can be collected in a queue for subsequent processing. A tree-based interface builds a traversable tree of the XML document allowing a greater degree of flexibility of processing at the expense of a more complex parser.

2.4.2. Cognitive Psychology

The utilisation of cognitive mapping theory applied to information structuring and sharing is one of the keys behind this research.

Stated in (Downs and Stea (1997)) is that cognitive maps are organised representations of some part of the world. These are mental maps, which can represent structures such as your route to work, and other physical relationships in the real world. However as a natural mechanism they can be capitalised upon as a useful storage mechanism with the aid of visualisation to assist in the cognition of information.

Addressed in (Laszlo (1996)) is that cognitive maps are used to understand your spatial environment. Thus you are using cognitive maps when in a virtual reality world. You are mapping where objects are within this world so that you know where to find them again with a relative set of movements in this world.

Put forward in (Downs and Stea (1997)) is that everyone uses them, they are a map of your world in your head however due to the variety of experiences everyone has of the world, no two people lead exactly the same lives they are personal to everyone. Hence a personal cognitive map becomes a snapshot of the world, as someone perceives it to be.

Stated in (Laszlo (1996)) is that cognitive mapping when used in the real world to solve spatial problems requires the map holder to know three things, whereness, whatness, and whenness. Whereness means knowing the location, distance and direction of the next waypoint or destination. Whatness refers to the knowledge of characteristics of the objects / people that are within the destination or waypoints. Whenness means that the destination or waypoint has a predictable pattern, or when it is available or present.

These three components of map creation and usage have different characteristics for different people. Taking a real world example two people on a simple walk home from a bus stop, one blind the other sighted have very different methods of way finding. The sighted person's cognitive map consists of street names and house designs, the blind man's more on the undulations of the pavement.

2.5. E-Commerce in Product Development

2.5.1. Product Modelling

The work of (Feiner (1993)) concerns the integration of 2D windows into an augmented reality system. The research details how the use of a mixed 2D and 3D environment can be best used to convey information. Although augmented reality is outside the scope of this research, an insight into the advances upon virtual reality for the 3D aspects is advantageous considering the benefits of enhanced cognition of information.

The research work of (MacIntyre and Feiner (1996)) is based upon the creation of a system called COTERIE. This work is to facilitate the creation of distributed virtual environments. The work is centred on virtual worlds. However alternative information update mechanisms are described such as the update scheme where an update is applied to all copies as opposed to the invalidation scheme where one update causes all other copies to then be invalid requiring a new copy to be obtained. Techniques such as these that are designed for high-speed virtual reality (VR) systems are transferable within the collaborative information-sharing domain.

A comprehensive understanding of the various areas within 3D multimedia interfaces is given in (MacIntyre and Feiner (1996)) detailing the terminology of considered domains such as virtual worlds, artificial reality and ubiquitous computing. Ubiquitous computing would be the ideal environment to use a collaborative system maximising the potential for access. This classification refers to "*an environment in which a large number of computers, ranging from hundreds of palm-sized to a few wall-sized units will be seamlessly integrated into our immediate surroundings, connected by wireless network. Ubiquitous computing also naturally integrates all the traditional digital media; text, audio, synthesised graphics and video.*" The benefits of such an environment in the distributed collaboration realm are such that the differing information mediums available can utilise the large number of displays per person such that entire displays can be given over to a single environment for long periods of time. Audio would have great semantics to call attention to various components of such an environment.

Detailed in (MacIntyre and Hollerer (1997)) is an Augmented Reality system that is combined with mobile computing. The system gives supplementary information during a walk around a university campus. Mobile computing is not within the scope of this research but for an integrated system potential expansions and a perspective of where the field may be headed should be taken into account to allow for smoother transitions to future technologies. Considering choices such as XML as previously mentioned can make for relatively easy adaptability to these other domains.

The research work of (Ottosson (1998)) discusses the use of VR in assisting product development. He attempts to define VR as the following: "VR means that an individual encounters 'synthesised' experiences created by computers and performed in such a way that the experiences to some degree are experienced as real experiences by the user. The individual will through uncontrolled and controlled reactions in real time influence the computer generated VR." A good point raised is that of how little in the way of the business benefits have been measured after the introduction of VR. The benefits are currently assessed through impressions and feelings. However well known manufacturers such as BMW and Volvo are currently employing the technology in both development and publicity. Shown in (Ottosson (1998)) is that ergonomic configurations can be enhanced through the use of VR and are indeed one of its most suitable domains. The term brainstorming is also expressed as Brain Aided Design (BAD) and is stated as making a good companion to virtual reality.

Identified in (Gomes (1998)) is that a virtual artefact has a variety of attributes, geometric, design intent, manufacturing, cost, and part number references and documentation references. These require a close integration with each other but each require their own separate considerations in terms of a suitable interface, the successful integration of this is vital.

The integration of a CAD and an expert system is explored in the research work (Abdalla and Knight (1994)). The work builds an interface between an expert system that contains extensive information concerning manufacturing facilities and product features and a CAD solid modelling system. An expert system with such data provides assistance with design for manufacturability. The system provides the facility to recognise geometric forms directly from the database of a CAD solid modelling system.

The research work of (Roucoules and Tichkiewitch (2000)) is based on developing a design system on a functional approach as opposed to the products geometrical structures and sub assemblies. It is an extension of the CAD – Expert system approach incorporating a sharing mechanism through shared server.

3D work in the simulation of object imported from CAD solid modellers is examined in the research work of (Horváth, et al. (1998)). Computations such as friction and collision of trajectories are evaluated to support the preliminary design of mechanical equipment.

Mobile Augmented Reality is introduced and developed in the research work of (Feiner, et al. (1997)). Augmented Reality is an extension of Virtual Reality in overlaying corresponding views of computer-generated 3D on top of real time views of the real world. The work of (Feiner, et al. (1997)) is based on providing overlaid information about a university campus through a head worn system allowing the user to walk about the campus and receive information relevant to their location. This kind of technology is not directly explored in this research but its consideration when defining portability requirements for software and avoidance of potential legacy data through non-system dependent data schema allow these exciting developments to become a possible combinable reality with a collaboration system.

Talk of the future of combined multimedia interfaces to provide the most accurate in data communication and data sharing is presented in the work of (MacIntyre and Feiner (1996)). This research encompasses elements of user interaction such as spatial audio and the use of haptic interfaces. Haptics is the term applied to the use of tactile feedback to the user. It discusses the use of both virtual and ubiquitous environments. It is the integration of all these media that will provide the future of distributed collaboration as bandwidth and computing power become more accessible and affordable thus must be considered when proposing future collaboration tool developments.

It is the accessibility to virtual reality that is fast becoming accessible in the form of plugins or extensions such as that from (Autodesk (2000)). This concerns the WHIP plugin this allows the display and manipulation of CAD solid models within a browser plugin. Feature such as this allow the development of more integrated works on a faster more portable scale. Providing maximum future accessibility to the information to be shared is vital to ensure the realistic utilisation of a distributed information sharing system.

Virtual Reality Modelling Language (VRML)

A comprehensive description of VRML 2.0 and how to use it is given in (Lea (1996)). This is combined in the work with the integration of Java.

Historically in 1995 VRML 1.0 was launched but the limitations of the new specification were soon obvious. A world could be generated within VRML 1.0 and navigated through. However nothing in the scene could be interacted with, a frustration that prompted the development of VRML 2.0.

(Web-3D-Consortium (1999)) is the home of VRML and provides many resources on the subject. See Appendix 1 for the capabilities of VRML 1.0 and the extended capabilities of VRML 2.0.

A technical paper is presented in (Kim, et al. (1997)) describing the method of converting NURBS (Non Uniform Rational B-Spline) surfaces in to VRML. Currently as the [WEB99] specification shows, VRML only supports polygons. The system described is that of Cyberview which is also mentioned in (Kim, et al. (1998)). The system is designed to enable collaborative teams to share STEP data using the open standard of VRML as a visualisation mechanism.

2.5.2. Object Oriented Database

Stated in (Kim (1990)) is that there are two basic strategies, which have a practical application to this research, relational and object-oriented. Although this research aimed at utilising current applications of object technology to this domain, there is little point in forcing an improper solution for the sake of it. Persistent storage is possible in an object oriented programming language, however this would be missing some of the vital properties required for concurrent access.

A comprehensive overview of Object Databases is given in (Barry (1996)) and presents a path through the considerations of their use against the factors concerning relational implementations, via a consideration of the complexity of the data to a set of implementation issues.

Data Complexity

The choice of an object-oriented database was made for a set of reasons, the first and generally the most relevant and important being the relative complexity of the data. Data can be determined to be complex based on a set of considerations (Barry (1996)).

- The data lacks a unique natural identification.
- There are a large number of many to many relationships.
- There is a heavy use of traversals to access the data.
- A frequent use of type codes exists.

The information to be stored is complex engineering data. Common sense suggests that these would be complicated in nature however it should be established that this is the case.

Nodes within a drawing have no unique identification without using the information stored itself. A section within a drawing could feasibly be connected to many other parts and these also to many other parts. Exploding the parts induces a heavy traversal of the nodes; this traversal could take place in a huge variety of ways. Type codes would have to be used with the data because much of it would similar; e.g. various types of screws would need a form of identification.

Both comp.databases.object, and (Chaudri (1998)) indicate that the incentive for creating object oriented databases originated from the need to store this kind of complex interrelated data and thus in theory should prove well suited.

Relational or Object Based?

In (Kim (1990)) and (Barry (1996)) there are a variety of hybrid database systems, which combine aspects of the relational and the object model, however these were dismissed because they only tackle the problem at a superficial level. They can allow the programmer to access a similar API to an object-oriented database, however the underlying database is a relational one. Time is wasted converting the data, and the greater data complexity the greater the importance of subsequently generated lag. These hybrid systems, do offer benefits of both types of database management system but these benefits seem more tailored to people who have training and data input already invested in relational databases and thus are more reluctant to commit further IT resources to making a full crossover to Object Technology (OT). Hybrid database systems add further software layers increasing the workload of the system for the ease of migration. Migration is an important issue for companies with existing large data repositories however there is only so many derived performance benefits you can achieve from old technology.

The work of (Barry (1996)) tells us that if data is complex which database is the more suitable. An object database deals with highly complex data very competently.

Complex data can be modelled in a relational database however this is difficult to do. Data often needs to be deconstructed upon entry into the database and reassembled on retrieval. When there are lots of relationships between the data, the multiple joins required in reconstructing it become severely degrading to the systems performance.

An argument around this for a relational database supporter is that the data could be de-normalised reducing the number of joins needed. However this results in repeated data that in turn is an inherently inefficient use of resources, and further more allows the possibility for the data to become inconsistent during updates, a strong possibility in a concurrent access system.

Within an object oriented database many to many relationships can be expressed directly instead of having to join the data as is the case with a relational system, the data can be traversed as a graph data structure. The successive index searches and joins which the database would have to undertake to traverse the graphs generally means that it would be slower than the object oriented database for this type of data.

Stated in (Barry (1996)) relational databases are generally based on B-Tree or Index Sequential Access Methods (ISAM) these are the most common access methods. The data must be portrayed as a 2D table. The relational model defines the structures, operations and design principles to be used with these tables. SQL provides ways of combining tables to express the relationships among data which are almost completely absent in most B-Tree / ISAM programming libraries.

Why even consider relational databases at all considering the negative points made previously. Well relational databases have extremely fast querying facilities; the technology has been around long enough now to be refined time and time again to produce some highly optimised systems. This has also led to some highly sophisticated error handling for database operations. There is also far better support in terms of software tools for relational databases.

An object database allows the user to store objects directly without conversion to a tabular format. The application code requires no SQL or database calls, as is the case with the alternatives, the object database allows transparent program integration.

An object database provides many features, which must be considered. The work of (Barry (1996)) puts forward the following questions that should be answered or at least considered before implementation.

- What is the computing environment?

The computing environment for the database could be defined within the system; most of the system is intended to be portable however consideration was given to stipulating that the database server should run from a specific operation system. The chosen database system runs via a Java virtual machine as described elsewhere in this thesis. This gives a variety of computing environments, which support a Java virtual machine, making this component of the system portable.

- Will the data be distributed?

The data at least for the purposes of this research, is based around a central server the issues of concurrent access increase the complexity of maintaining a distributed database.

- Will be the distribution be local or over a large geographic area?

If the data in an expanded implementation were to be distributed, then this would be structured over a large geographic area.

- How many users will work concurrently?

The actual numbers are dependent entirely on the number of people concurrently working on product development. Ideally a representative from each domain of the product development lifecycle would be present in the final discussions concerning a project. This would cut down on unnecessary system load and tangent inducing sub topics. Discussions could also be resolved within a targeted domain group, and their conclusions and suggestions linked into the wider picture.

- How much data is likely to result from your most common queries?

The searches upon the database form the traversal of nodes with the linked structure of the system, hence each search result would turn up a single structure be that a 3D model a concept map or a URL. However should the system be used to check for example a list of parts within a model then this could result in a large data set.

- Will you be using methods as part of your choice?

Methods used for introspection of the stored objects provide a powerful query mechanism especially when activated via an RMI invocation.

- Will you be accessing existing data from non-object sources such as a relational database?

A migration strategy is not within the scope of this research however, with the Standard Data Access Interface (SDAI) implementation, the system should provide a level of access to data stores, which provide this interface

Visual Interaction

There is becoming a pronounced need for information retrieval systems to remove some of the mystery and complications surrounding database access. Improvements in this domain will widen the scope of possible accessors to the data, as (Catarci (1999)) says:
"*Until now, the most widely used database query languages have actually been programming languages which require knowledge about language syntax, technical background, and information of both the system application domain and its interaction mechanisms. Such languages do not help to understand the meaning of data, nor do they provide any guidance in satisfying the users needs. In general they do not fulfil the requirements of user friendliness and ease of use...The information consumers urge both effective visualisations (in 2D and 3D) and retrieval tools able to overcome the existing dichotomy between browsing and querying.*"

ObjectStore vs. Jeevan

After it had been decided that an Object Database was the correct approach to the problem domain, the question of which one would be suitable arose. There is an up and coming plethora of ODBM systems to choose from each with a variety of capabilities. The real world choice came down to Jeevan and ObjectStore.

(Design (1997)) gives us the ability to:

- Create and modify persistent Java objects.
- Store and access data in the same query format, as it exists in the application.
- Describe, store, and query complex data used in sophisticated software applications, as well as data traditionally managed by relational database applications such as MIS programs.
- Persistently store data independent of the data type.

"The Java interface to Object Development Client (ObjectStore) is for Java and C++ applications that require multi-user high-performance persistent storage for large databases with full database features such as failover, online backup, fine grained concurrency, and security...ObjectStore is not constrained by the number of objects being managed. It can work well for distributed databases of virtually unlimited size."

ObjectStore supports:

- Applications that interface with databases and servers on local or remote machines.
- Java applications and C++ applications that can access the same data.
- Multiple concurrent users.
- Collections with indexed look-ups and queries.
- Databases consisting of multiple segments, which are variable-sized regions of disk space that ObjectStore uses to cluster objects stored in a database.
- Operating on multiple databases in a transaction.
- Cross-database and cross-segment references.
- Online backup, failover, archive logging.

The ObjectStore API facilitates an application to:

- Create, open, close and destroy databases.
- Start, commit, and abort transactions.
- Read and write database roots, which provide starting points for navigation among persistent objects.
- Store objects in a database, retrieve and update those objects.
- Store collections of objects with indexed look-up.

The other consideration (Jeevan, 1999) allows you to:

- Create an object database to store Java objects.
- Directly map the application classes to database schema.
- Directly communicate with the database without any middle layer driver (such as JDBC driver).
- Index the database on primitive object attributes of reference objects.
- Dynamically relate any two objects in the database.
- Retrieve Java objects using simple but powerful queries.
- Update the retrieved object by calling its methods and store it back in the database.

Jeevan essentially consists of three supplementary classes to Java; the database is a relatively simple offering compared to ObjectStore. The Jeevan (W3Apps (1999)) database is missing the transaction, security, failover and concurrency operations which are critical to the management of an information store in the DISCS domain. ObjectStore has become the database of choice and several STEP implementations have also been proven on this foundation to be successful.

2.5.3. Design Consistency

The maintenance of consistency when concurrent engineering is distributed across multiple functional perspectives becomes a significant challenge with major potential impacts on product cost, quality and requirements (Klein (1994)). Detecting understanding and resolving conflicts requires a representation of the interdependencies between design decisions involved in the conflicts. The distribution of concurrent engineering across time implies the need to remember the reasoning underlying design decisions made throughout the product development lifecycle. It is imperative to store this design rationale to be able to make intelligent modifications to previous design decisions. When an artefact is designed there is a long series of deliberations and tradeoffs, however the underlying history, intent and logical support for the decisions captured is usually lost or at best represented as a scattered selection of personal notebooks.

The research work of (Abdalla (1998)) states that the introduction of constraints from a variety of functional sources, such as design specifications, manufacturing facilities, and targeted costs as early as possible during the design stage leads to a significant reduction in the product development lifecycle, and the amount of rework needed.

On the nature of constraints (Hashemian and Gu (1995)), tells us that constraints vary in type and nature; therefore any constraint-based approach to design should provide for a variety of constraint types from different aspects of a products lifecycle. "In a constraint-based approach, a design problem is considered as determining the values of design variables while satisfying a set of constraints which represent the design requirements imposed by several aspects of a product's lifecycle. This concept can be considered as an emulation of the team approach by representing a members expertise with a set of constraints." The work of (Hashemian and Gu (1995)) describes a system that is based on the constraint network approach. This is essentially a nodal graph data structure with each node in the graph either an entity or a design variable with entities and variables always linked via a constraint and variables linked by constraints. Much of the work in solving the constraints then becomes an algorithmic problem of traversing this network of constraints. A solution to a constraint network is a set of quantified variables that do not violate any constraints. A view of the structure of a constraint network from (Hashemian and Gu (1995)) can be seen as follows:

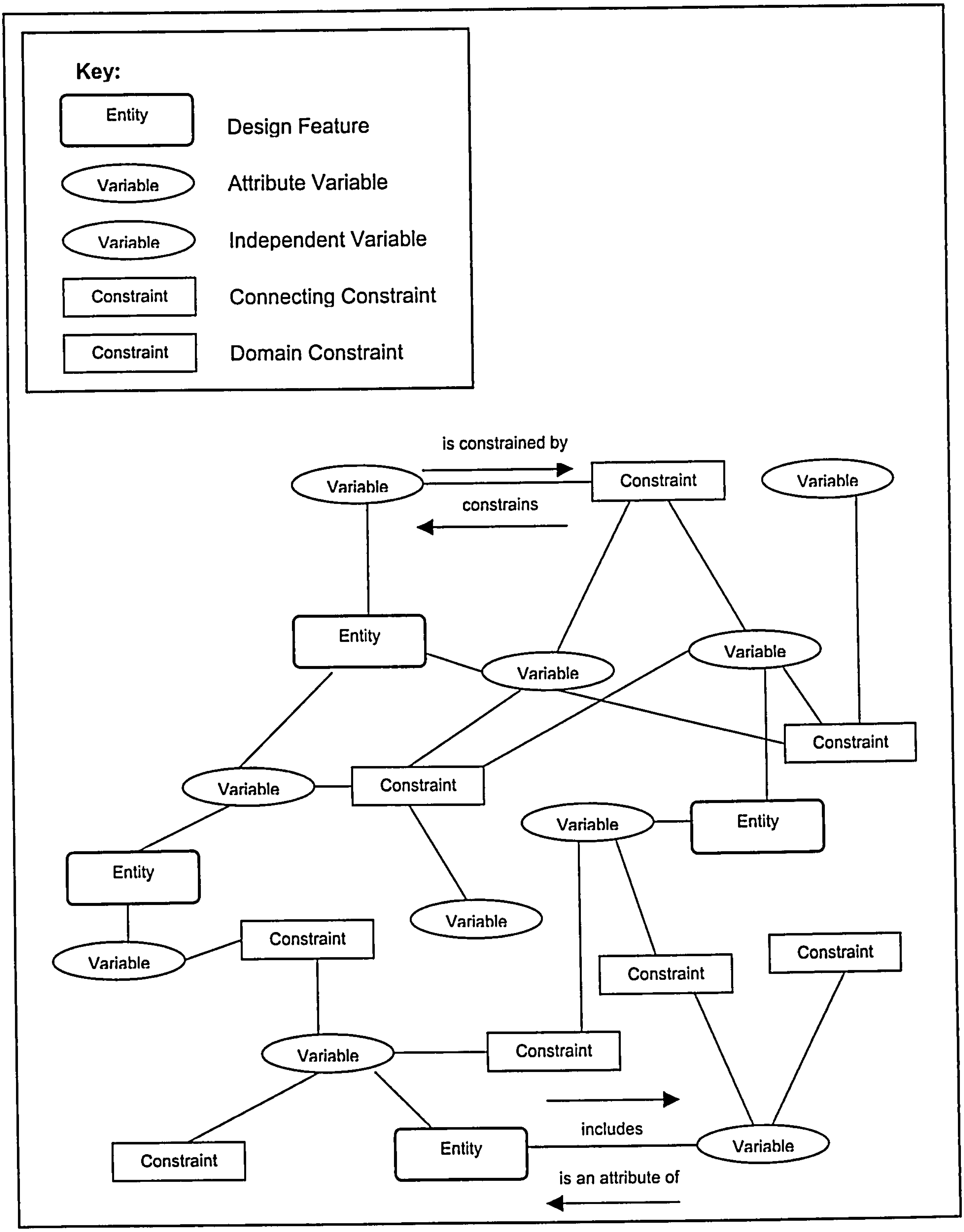


Figure (2-4) Example Constraint Network

The issue of constraints validation is tackled in the research work of (Yoo and Suh (1999)). The constraints are defined in the EXPRESS information modelling language from the STEP standard. These are then distributed into local and global databases. Subsequently entered data stored in the databases are validated with the allocated constraints. The local distribution of constraints in schemata increases the efficiency in processing what is usually a large data set. EXPRESS allows the definition of four types of integrity constraints, uniqueness constraints, local constraints, existence constraints, and global rules. The constraints are effectively organised within a hierarchy, the local schemata validate the local constraints, and these are then passed upwards through the chain of systems until they are validated by the global rules.

Concurrent team behaviour in terms of interactions, conflicts and conflict resolutions is investigated through the use of single function software agents in the research work of (Berker and Brown (1996)). The use of single functions agents is in contrast to a great deal of the work in the area that utilise large powerful knowledge intensive agents. The purpose behind this is to discover the core interactions to acquire a better understanding of design agent conflict management. As a mechanism for investigation the single function agents are too fine grained to be efficient enough in reality. The researched work in (Berker and Brown (1996)) attempts to identify how functionality should be grouped into large more effective agents.

The work of (Barták (1998)), demonstrates a series of constraint network solving mechanisms. The simplification of the network based upon its design variables are presented and also the search algorithms for detection of a solution to a network.

The work of (Matta and Corby (1998)) has the aim of providing a library of generic knowledge components. These knowledge components aim to model:

- How products were initially decomposed
- How teams were designated – which tasks were designated to which team
- How teams were composed – which fields they specialise in
- How design propositions were considered
- When decisions were made and who made them

- How the teams arrive at those decisions, etc.

If knowledge can be captured in such a way, these knowledge models can then retain the contribution of domain experts in a reusable format for the purposes of professional training and decision-making guides. It was discovered how difficult it is to extract potential conflicts from designers for fear of offending other members of their hierarchy. The enabling of the past through a knowledge library allows conflicts to be resolved with the cumulative experience of designers available to deciders.

2.5.4. High Performance and Platform Neutral

There is a great need to develop systems that address real world considerations in the area of collaboration, many systems developed are built upon old and inflexible technologies. Platform independence is key to true e-business solutions and the Java language provides a useful tool with which to accomplish this, along with other necessary considerations discussed below.

Java named after an Indonesian island is a relatively new language surfacing within the last four to five years it has had a far greater take up in that time than C++ achieved in its first ten, five times that in fact. One of the main reasons for this is the industries support for the language, many of the big industry giants are behind Java, providing new tools, API's and documentation. This ever-increasing support for Java is encouraging many developers to adopt it.

The main choice between object-oriented languages these days is between C++ and Java, Smalltalk also has a strong following, but the main industry support is placed in the direction of the first two.

Covered in (Flanagan (1997)) is the Java 1.1 language and Application Programming Interface 9 API). A quote from an early Sun paper on a description of Java is:
"Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language."

Object Oriented Programming Language

From a programming point of view this means that the emphasis is on the data and related methods rather than the procedural train of thought. OO is considered to be an extremely powerful paradigm, explanation of which is outside the scope of this document.

Interpreted Byte-Codes

The Java compiler generates byte-codes for the Java Virtual Machine (JVM), rather than native machine code. This byte code is in layman's terms a halfway point between machine code and a human readable format. It is these byte codes which give Java its platform independence, the byte code can then be transferred to a wide range of platforms, and then executed by a Virtual Machine (VM), this interprets the code at run time to produce executable instructions.

Architecture Neutral and Portable

The byte-code approach is useful to not just those who wish to sell a single product and wish it to be marketable across multiple platforms but those who are developing distributed systems. It is not just the byte-code approach however, which determines its portability, but the removal of any implementation dependant aspects of the language, for instance Java explicitly specifies the size of each of the primitive data types, as opposed to C which depends on the platform.

Dynamic and Distributed

Java is dynamic by virtue of its ability to load a class into a virtual at run time. These can then be dynamically instantiated. There is also a high-level support for networking in the URL class.

Small Domain of Constructs

There exist a relatively small number of constructs in Java, and the syntax was designed to be familiar to a large family of existing programmers, many of the language constructs are the same as in C and C++. Java has also omitted the use of pointers; the referencing and dereferencing of objects is done autonomously. Garbage collection is also in place to avoid the need for memory management.

Robust Type Checking

There is a variety of built in functionality within the Java language. These allow for a reduction in the number of common problems found in other programming languages. Java is a strongly typed language; there is extensive compile-time checking for potential type mismatch problems. There is a requirement for explicit method declarations, allowing the catching of method invocation errors. The lack of pointers as mentioned above and the exception handling make for more robust programs.

Secure Distributed Code

Security is important in any language, however due to the much touted distributed aspects of Java; the importance of security is greatly increased. At a low level much of Java's security protections go together with the robustness, without allowing direct access to memory a large number of possible security hazards are ruled out. Checking is also performed upon the byte-codes to ensure that the code is well formed and will not over or underflow the stack, this is to stop any malicious code being given the opportunity to take advantage of any weakness in the Java interpreter. A more commonly known and more obvious aspect of the security is the "sandbox" model. This is where untrusted code can be allowed to "play" without damaging the "real world". For example within this environment such code cannot access the filesystem. In Java 1.1, there is also the concept of being to add a digital signature to approve a section of code from a trusted source to allow the running without so many restrictions.

High-Performance Just In Time Compliers

Due to its interpreted nature Java will never be as fast a compiled language such as C or C++, however the benefits Java offers are comparable to those of high level scripting languages but with much higher performance. With the advent of Just In Time (JITs) compilers that produce machine code from the byte-codes for a certain CPU at run time, Java's performance may well not be that far behind the lower level languages.

Multithreaded Support

Within a GUI application it's relatively easy to visualise multiple events or tasks happening at once. Playing music while reading a document etc. Java is a multithreaded language; there is support for multiple threads of execution for handling different tasks.

Described in (Chauvet and Lerman (1997)) is a set of various distributed object request broker mechanisms. Consisting of a Java implementation following the Java object model, a C++ one in the COM object model, a Java one following COM, C++ following CORBA, and one in Java following the distributed Java object model. The flexibility of CORBA has a great many advantages however the components comprising the work in this thesis are all Java based, reducing the need for CORBA's anything to anything solution. A Java-to-Java solution is what is required. This can be facilitated via Remote Method Invocation (RMI), or HORB that consists of a lightweight Java object broker.

The steps required in the RMI development process is detailed within the work of (Callaghan (1998)).

Another approach would be to use the Enterprise Java Bean (EJB) service as described in (Johnson (1998)) and (Thomas (1997)). EJB provides a framework for the inclusion of Java Bean server components into a multi-tiered client server architecture. Stated in (Thomas (1997)), "*Enterprise JavaBeans improves the productivity of application developers. The Enterprise JavaBeans environment automates the use of complex infrastructure services such as transactions; thread management, and security checking. Component developers and application builders do not need to implement complex service functions within the application programming logic.*" The critical issue (Cattel (1994)), is the lack of addressing how an ODBMS fits into Object Management Group (OMG's) environments.

2.6. Design Sharing in Product Development

Stated in (Fowler (1995)) the STEP standard is used to facilitate the unambiguous representation and exchanging of product data, through the full life cycle of the product.

The essence of the aims of STEP is to provide a consistent data exchange format, and provide an interface to various application systems such as CAD, CAM or CAE.

XML is an alternative for this data representation however apart from its specificity for the problem domain, i.e. exchange of product data, STEP offers strong support for the extended lifecycles of product data required in the modern information age.

Life times of the applications software, system software and the hardware which were all used to create the data, will have much shorter lifecycles than the data itself is stated in (Fowler (1995)). Application software has a typical lifecycle of 3 to 5 years if that, systems software, 5 to 10 years and the hardware even less with cycles of less than 3 years. Thus the integration of such a standard for storage and interpretation can be a vital mechanism in overcoming the future migration problems (Hares (1994)) that are rapidly occurring with legacy systems today. There is a strong drive behind the philosophy to relate data to the products and processes that the data describes, as opposed to the computer systems, which create the data.

The research work of (An, et al. (1995)) is based on an early implementation of a data exchange mechanism following the STEP file format. (An, et al. (1995)) does state; "*As an international standard of product data exchange, Product Data Exchange Specification (PDES) / STEP is intended to provide information for all engineering applications without considering hardware, software, or processes.*" Alternatively put in (Urban (1993)), "*The major concepts of PDES / STEP aim to develop a neutral product information model and to support the complete representation of a product throughout its lifecycle.*"

The research work pf (Peng and Trappey (1998)) constructs six STEP-compatible models to demonstrate the following:

- product definition
- product structure
- shape representation
- engineering change
- approval
- product scheduling

Also highlighted in (Peng and Trappey (1998)) the adverse effects of incompatible computer based systems in terms of information delays, lack of configuration control, poor productivity, and usually poor quality and slow release to market.

STEP is effectively constructed within a three-layer model, i.e. the physical layer, the logical layer, and the application layer. The physical layer provides the STEP physical file format, file exchange, an application-programming interface and database sharing. Principal product representation entities for all phases of the product lifecycle are defined in the logical layer. These entities are then classified by their properties into specific parts called integrated resources. The application layer provides an application protocol to support various types of applications for example shipbuilding. EXPRESS is the language used by STEP to define product data. EXPRESS classifies and constructs integrated resources by their "*data entities, attributes, rules relationships, functions and constraints.*"

Figure (2-5) below shows the STEP document structure (Fowler (1995)):

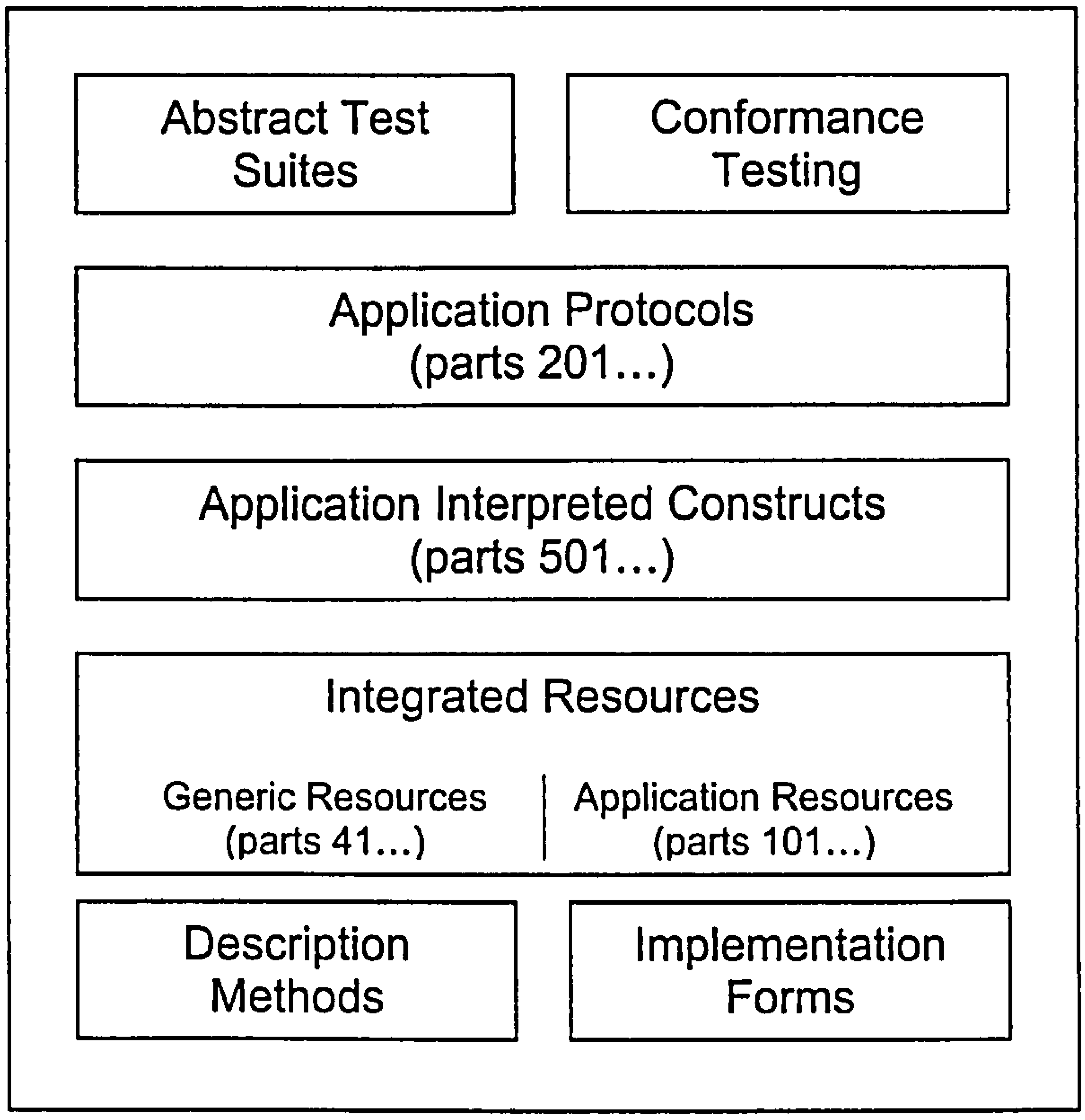


Figure (2-5) STEP Document Structure

2.6.1. Standard Data Access Interface (SDAI)

The SDAI is a component within the Implementation Forms of the STEP standard; this concerns the connection of applications to the database of STEP data via an Application Programming Interface (API). (Goh, et al. (1994)) tells us that the problem with the built-in data storage systems of traditional CAD / CAM systems which were there to hide data structures from the user is that now there are a wealth of incompatible data representations. Thus leading to a difficulty in sharing information between applications. The application programming interfaces of databases allows data independence from applications. The follow up problem to this is that these database

API's are then different themselves. The SDAI provides a constant interface for applications to adhere to. Stated in the research work of (Goh, et al. (1994)) is that the routines within the SDAI that they developed were awkward since the SDAI was developed upon an existing database interface, a closer integration with the internals of the database from the start would provide a much-improved implementation. ObjectStore's (ObjectDesign (1997)) Java interface allows this integration. The requirements for such a data access interface is as follows (Goh, et al. (1996)), query, data modification, meta-data query, and validation. Concluded in (Goh, et al. (1996)) is that STEP requires available tools integrated into a common environment.

There are four defined levels of implementation for data sharing in the standard. These four levels are as follows (Fowler (1995)):

- Level 1: passive file transfer.

A pre-processor translates data from the format of the sender into a STEP physical file; this file is then transferred to the destination system. A post-processor reads the data and translates it to the file format of the second system.

- Level 2: active file transfer.

This is an extension to the first level; the data is translated into a "working form" which then allows manipulation as part of the translation process.

- Level 3: shared database access.

This again is a further extension of level 2, instead of the file being translated for the purposes of exchange it is now for the purpose of storage. A standard interface is provided to the underlying database to store and manipulate the data as a usual part of a database's services.

- Level 4: integrated knowledge base.

This is a research issue implementation and concerns the combination of Artificial Intelligence (AI) with STEP.

The conversion between STEP and CAD data to the extent of maintaining a CAD database is dealt with in the research of (Nakamura (1993)). Ideally a STEP model can be regarded as a superset of all existing CAD models. The difference in semantic scope between various CAD models when they are translated via a STEP data model means that only the intersection between the two CAD models can be translated exactly. Concerning practicality, situations exist in which the exact meaning of the data cannot be kept for the above reason.

The work of (LKSOFT (1999)) defines the requirements of a complete Java binding to the SDAI:

- A robust programmers interface. The users should not be able to bring the SDAI out of control.
- A well-defined and useful object-oriented API.
- Support for early binding support without dictionary information.
- Support for late binding support with dictionary information.
- Support for all 7 conformance classes of SDAI.
- Support for all language features of SDAI.
- Support to access p21 files.
- To work on one and also on several computers through a network.
- To have a specification which allows several implementations.

The thesis work of (Loffredo (1998)) defines a variety of STEP database implementations building benchmarks for use and testing against various SDAI implementations constructed upon both relational and object oriented databases. This purpose of the research effectively discovers the performance characteristics of the various implementations proving once more that an object oriented database with its powerful relationship modelling characteristics is the most sensible method to explore further.

Level three implementations concerning the SDAI implemented on both relational and object oriented databases building on the work of (Loffredo (1998)) can be found in the research of (Yeh and You (2000)) and (Liu, et al. (2000)).

The utilisation of off the shelf tools is a realistic entry point into the implementation of systems based upon the STEP standard as in the research of (Lindeblad, et al. (2000)) who choose the ST-Developer v7 from Steptools Inc. This tool suite is also based upon the research from (Loffredo (1998)) and is rapidly becoming the de-facto entry method into the world of STEP.

2.7. Summary of previous research work

Literature relevant to the various aspects related to this research works has been detailed. The literature shows a need for research into the areas of the integration of people, technology and the product development lifecycle phases. Much of the previous work that had been carried out has limited consideration for the integration of product development lifecycle phases or producing vague results pertaining to more integrated solutions. The present research is directed at the integration of previously unconnected sub systems both in the semantic and technological contexts.

It is envisioned that this work will be of assistance to people involved in the integration of such technologies and / or the construction of collaborative systems for use in a distributed nature.

Work examined within this literature survey shows trends to avoid tackling the overriding issue, that of fully integrated information sharing throughout the entire product development lifecycle. Work undertaken focuses on single facets of information sharing such as the MONET project (Srinivas (1992)) that allows application sharing without further consideration of the interconnectivity of the data, the context viewed or its relationships to other related information.

The research work of (Jagannathan (1992)) takes into account that the information should be related but also considers this from a technical perspective without assisting the user with a conceptual level of viewing. The system described places a greater emphasis on access to the data rather than an integrated viewing structure.

The work of (Kim, et al. (1998)) is one of the more complete solutions so far with use of STEP and use of the portable 3D viewing mechanism VRML. However again the considerations are narrow within the product development lifecycle, models can be marked up with comments and stored in a central facility for retrieval, however there is no support to take ideas from the initial conceptual level to a realistic stage where these models can be generated. There is neither this early lifecycle support nor the facility to relate later lifecycle stages such as maintenance information or ecological disposal considerations.

The work reviewed showed that a number of systems to assist distributed product development teams had been developed. However there is little work performed in the earliest stages of product development where the potential impact of correct communication is high. The key limitations of the reviewed works were that:

- The earliest stages of product development were not considered for instance concept generation;
- Reuse of design decisions for future training or learning mostly unconsidered;
- Essential aspects in information communication and comprehension generally ignored in preference of high cost modelling systems that have value in further more detailed phases of the product development lifecycle;
- Proposed methodologies suitable only for a technical team at the exclusion of input for non technical members of a product development team;
- Much of the collaboration effort concentrated on the technical efficiency of sharing the information rather than human efficiency and understanding of the information shared;
- There is a lack of an information model or framework capable of providing the required flexibility, comprehension, accessibility, media, portability, reusability, and support for entire product development lifecycle;

- Integration of both the necessary technologies and the cognitive human aspects of collaboration have not been adequately considered.

In short much of the work reviewed is either aimed at tackling a single type of product or a limited subset of the product lifecycle, there is yet to be a system that is capable of sharing product data throughout the lifecycle and is flexible across product classifications.

2.8. Scope of the present work

It has been ascertained through a comprehensive literature survey that supportive collaborative design at the earliest possible stage is a paramount requisite for distributed collaboration software tools. Therefore if communication in the conceptual stage can be achieved accurately, efficiently and in a timely manner, this impacts on further stages reducing the need for reworking and effects quality and time to market positively.

The development of a set of best practices and techniques for integrated distributed collaborative information sharing is proposed. Firstly, a methodology driven by advances in cognitive psychology to represent information in a portable, communicable, flexible yet structured manner has been developed. Secondly, the proposed system provides an environment that fulfils many of the information sharing recommendations encountered in the literature. Thirdly, technologically a solution is proposed that provides equivalent properties of information sharing partnered by a human focused approach.

To achieve the aims and objective of this research work the following key stages are proposed:

- Development of a methodology for information representation, sharing and modification at the earliest stages of design such as concept generation;
- Integration of principles of cognitive psychology into the information structuring;

- Propose an integrated technical schematic for a system able to support the specified methodology;
- Construct a prototype demonstrating key factors in the research.

Chapter 3

CHAPTER 3. Proposed Distributed Information Sharing for Collaborative Product Development

3.1. Overview

The principal objective of this research was to develop a set of best practice techniques for collaborative information sharing and implement these in a prototype computer system for the facilitation of distributed virtual teams operating within the context of Concurrent Engineering. The work requires the integration of a set of both technical and cognitive protocols to enable effective information sharing throughout a distributed collaborative team. Providing such integration empowers the communication between collaborating team members to a novel and semantically enhanced level. The developed techniques provide a means for incremental capture of the data, a means to share this among the team members and provide extra cognition and semantic assistance concerning this information.

The work draws together information sharing techniques from the sphere of Concurrent Engineering however the work is also based on the theories of cognitive psychology together with Software Engineering best practice for implementation considerations. The work provides a platform for future investigations based on a thorough research of the real problems and solutions in the domain of information sharing.

This research proposes techniques and technologies explored in a software prototype to aid distributed product development Concurrent Engineering teams through accurate, timely and distributed information sharing to produce products through a generic approach, which are:

- Reduced in cost;
- Improved in quality;

- Brought to market in a reduced time period;
- Have through development awareness improved the business processes involved throughout the product lifecycle;
- Increased in their applicability to the marketplace.

The proposed improvements to products and processes under development from a distributed team are gained through the corresponding enhancements in the collaborative communication and information management environment. Product and process advancements are in turn derived from:

- Improved communication between team members.
- Increased common conceptual & semantic understanding.
- Context linked support for multi-media information components, concept frameworks, CAD models, image, documents, video, and web pages.
- Enhanced technical access characteristics through the consideration of portability in tackling multi-platform incompatibilities.
- An encompassing consideration for the entire product lifecycle.

This chapter describes the developed approach for information sharing in a manner with which to obtain a greater coherence and understanding for distributed product development team members. The innovations and solutions for current information sharing problems are detailed and the context of the work is explained in an example scenario.

3.2. Introduction

Modern product design inherently necessitates the involvement of multiple specialists each providing a contribution to the development. Traditionally this approach has led to the specialists providing contributions in a sequential design model. This approach essentially builds upon the work of the previous stage with each specialised phase handing over the current state of the product development to the next expert party. Following a series of design stages in this serial manner introduces design barriers. A design barrier breaks the flow of information between the involved parties; product development specialists do not receive adequate information in a comprehendible format or functionally in a timely manner. Maximising the skills and efforts of a collaborative team requires that all parties have access to understandable, accurate information at the right time. This problem is complex in a co-located team however for the distributed team additional factors are relevant for these issues. Access to information is now not only broken through design processes but also through technical, geographical and temporal barriers. The following figure shows a typical traditional design sequence:

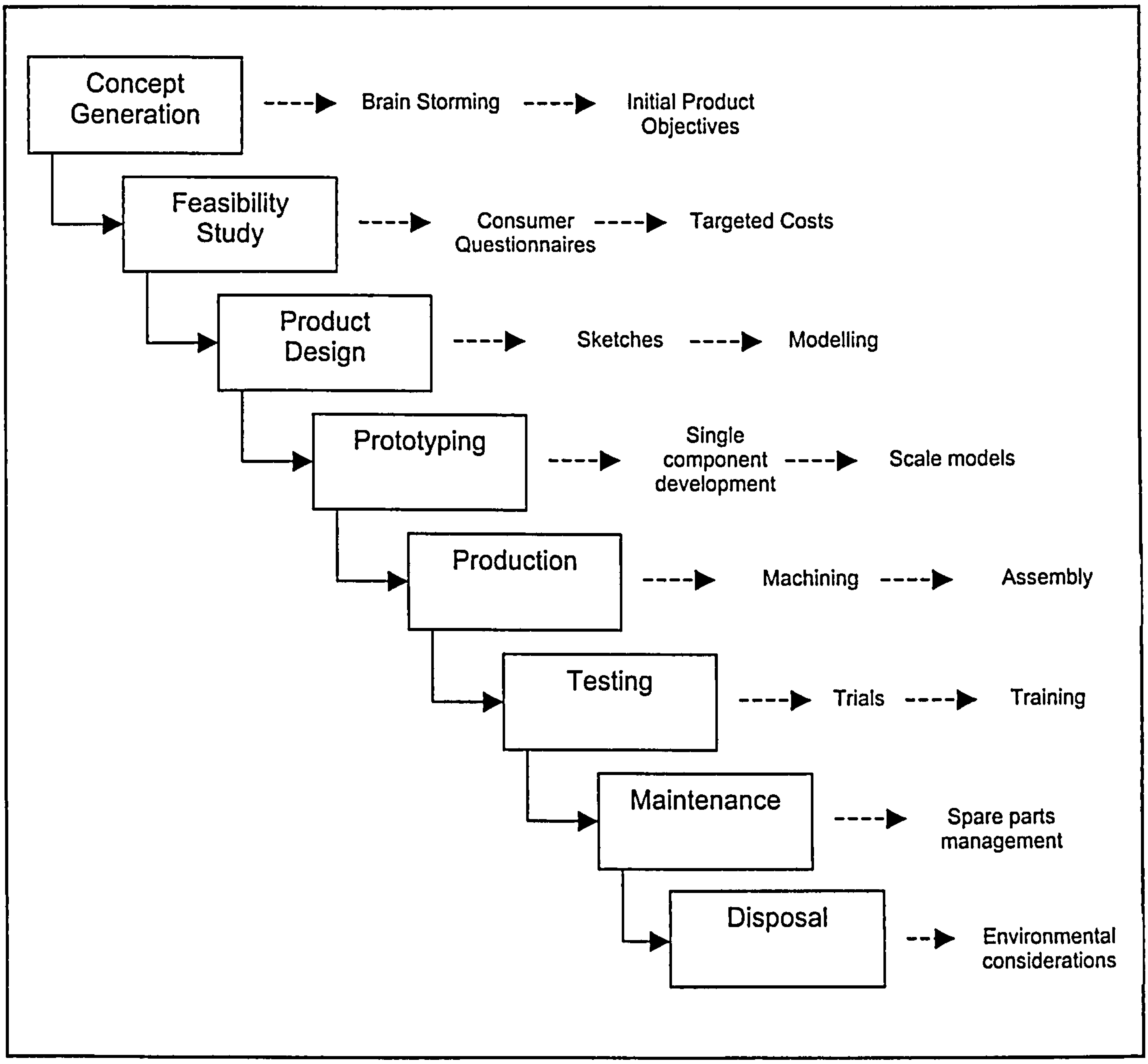


Figure (3-1) Traditional Design Sequence

The Concurrent Engineering approach provides key concepts to facilitate advanced product development. CE theory attempts to remove the design process barriers between product development phases. The essential element of CE is the increasing of the competitiveness of a product through the incorporation of product development lifecycle considerations into the earliest stages of design. These product development lifecycle considerations range from the initial conceptual design through to disposal including manufacturability, marketing, cost, assembly, serviceability, and even ecological considerations such as recycling. This approach requires the integration of a product development team from the start of the product development lifecycle. Without this integration requirements and considerations from each group of specialists or design phase are not communicated to other members of the team. These requirements if not accurately communicated lead to unforeseen problems in later development stages in turn involving much iterative rework until the problem is solved. This not only increases the product's time to market but also raises costs and inherently reduces quality as other sacrifices are made in order to complete development. The resulting product is at best competitive, lacking innovation and unable to achieve a supertitive level of competition (Ahmed and Abdalla (2000)). The following figure based on the Product and Process Wheels of (Prasad (1996)) shows an integrated lifecycle based on the ideas of CE:

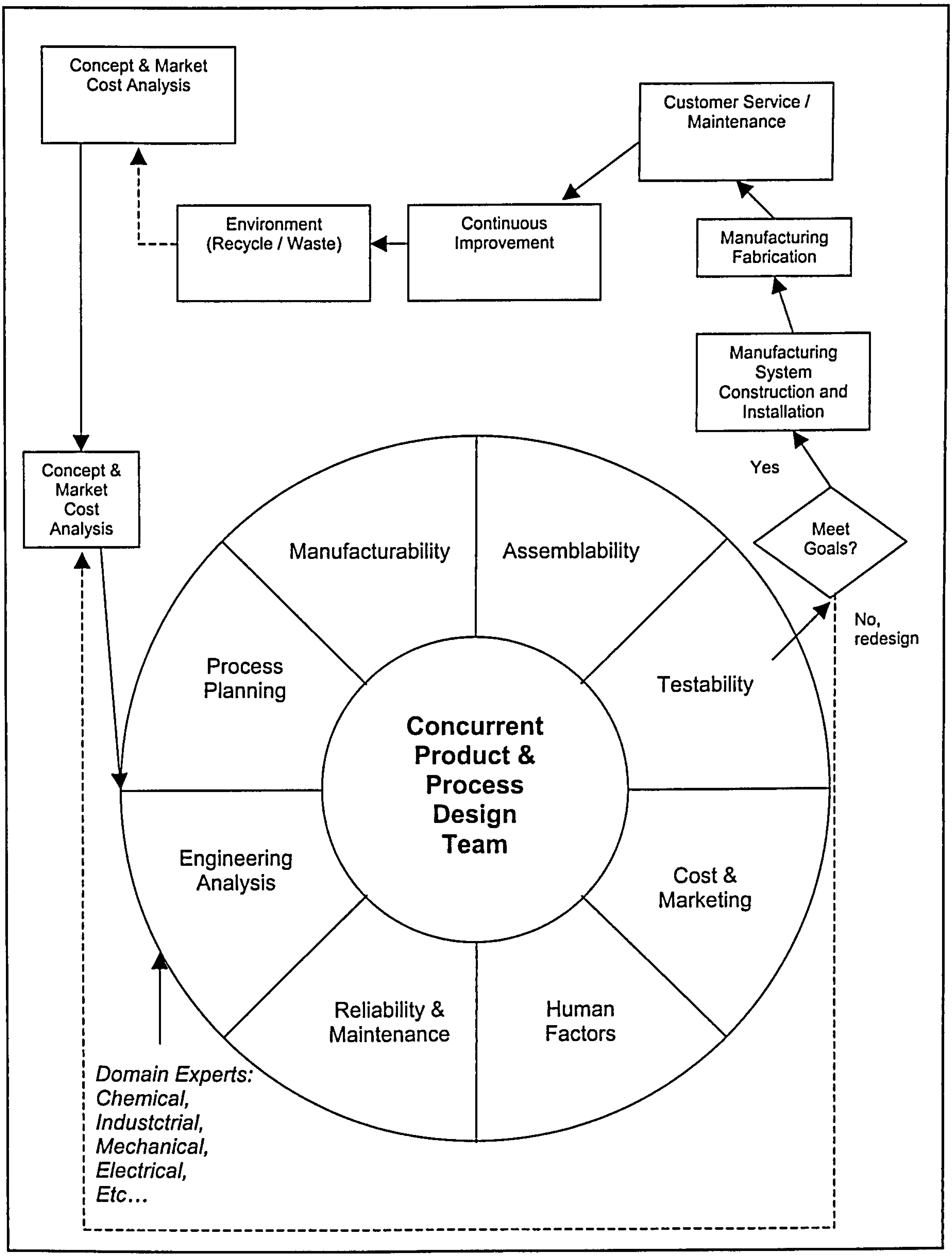


Figure (3-2) Integrated CE Lifecycle based on Product and Process Wheels of (Prasad (1996))

For CE to be successful in the modern information age effective leverage of domain experts is essential. These experts often have the keys to innovation and competitive advancement. The effective implementation of CE hinges on the integration of all of these experts from a diverse spectrum of domains at the earliest opportunity in the product development lifecycle. Co-located CE is still a challenge to many organisations however competition requires product development to take place in an ever-increasing global market. Globalisation necessitates organisations to utilise global resources and the greatest resources in the current market are people and information. In order to provide world-class products organisations need to utilise worldwide resources, in the context of this thesis the utilisation of a globally distributed product development team. Co-location is usually prohibitively expensive in terms of both time and cost for a globally distributed team. Notably skilled team members are certain to be in high demand and thus working on concurrent activities.

Utilisation of the CE approach in a distributed environment relies heavily on supporting software tools. These tools play an essential role in managing communications, team coordination, design and simulation. An innovative tool dealing with the issues pertaining to the lifecycle phases of maximum potential savings is required to reduce problem occurrence in later stages.

3.3. Collaborative Product Development Processes

Product development throughout the lifecycle, information sharing through multiple complexity levels for a diverse user audience, technology and data integration, information cognition, and a truly global approach has been considered in this research. Current collaborative product development approaches utilising the Internet attempt only to tackle sub components of the product development lifecycle these provide good communication in those arenas but do not leverage the full benefits of the Concurrent Engineering philosophy.

Concurrent Engineering aims to reduce the product development lead-time, increase the quality of the product, and reduce costs resulting in a significant increase the industrial competitiveness of the final product. Essential to this development philosophy is the need to utilise and integrate all available design information and expertise throughout the lifecycle however the earlier this integration takes place the greater the competitive advantages introduced through a consequential reduction in expensive reiterations. Essentially a "get right first time" approach by making available in an accessible manner the requirements and design input from a multifaceted design team. There are a variety of approaches each with a more specific focus. DFX (Design For X) encapsulates these, for example design for assembly, machine-ability, cost and others. To ensure maximum utilisation of the product development team an approach is necessary that allows for design input to bring a focus to development that facilitates a DFX approach. Designing for a single emphasised goal encourages an unbalanced development process the opposite of the Concurrent Engineering philosophy. A supportive design environment needs to allow input for the product to be designed to a range of goals and requirements. Chapter 2 highlights a diverse support for this integrated approach however there is a highly apparent need for research to be undertaken for integrated advancement of technologies, methodology, and the most important factor the design team themselves.

Existing tools and methodologies are too narrow in scope in terms of both the product development lifecycle and the design team themselves. A narrow development scope often serves only the design stage that system is focused upon adequately, however this then in turn contributes to the problem of design barriers through the generation of more application specific data that is isolated from subsequent / previous design phases. The communication is limited essentially to the current design phase. Product development systems often implement complex methodologies that have a detrimental effect through the building of new barriers to accessing the design information for the majority of those that can contribute. These are two of the main areas that require consideration to enable the product development team to maximise their gains from collaboration.

To facilitate the development of a product that does have a reduced lead-time with improved quality requires a suitable software support system. Such a system necessitates that the product development lifecycle needs to be accommodated from the initial stages of conception to disposal in a methodology that can represent high level data in a flexible enough format to also be suitable for use through an iterative process to represent detailed complexity.

3.3.1. Product Development

The development of a new product to arrive in the market place requires the following steps, product evaluation, development planning, product design, prototyping, and marketing. The initial definition of a product is the vital starting point in the development of a new product.

Integration of product development concepts aims to improve the effectiveness of the development and the evaluation of product development ideas and processes. These objectives can be defined as such:

- The development of products with great consumer need, accuracy and anticipated quality;
- The design of new improved processes and the consideration of process capabilities to facilitate efficient product development at improved competitive prices;
- The reduction of the product development design iteration time and consequentially the time to market;
- Consideration of manufacturability aspects and minimised disruptive design changes that in turn facilitate high productivity.

The 11 proposed principles of integrated product development by (Associates (2001)) are as follows:

- Understand the customer and manage requirements;
  Better customer relationships, frequent communication, and feedback systems lead to better understanding the customer's/user's needs. Methodologies such as Quality Function Deployment (QFD) aid in defining customer needs and translating those needs into specific product, process and quality requirements.
- Plan and manage product development;
- Use product development teams;

Early involvement of marketing / program management, manufacturing, material, test, quality, and product support personnel in product development provides a multi-functional perspective and facilitate the parallel design of product and process, reducing design iterations and production problems.

- Integrate process design;

  The design of manufacturing and product support processes must be integrated with the design of products in order to optimise the performance, availability and life cycle cost of the product.

- Manage costs from the start;

  This can be achieved by: (1) developing a greater awareness of affordability and life cycle costs, (2) establish target costs and manage to those targets, and (3) managing non-recurring development costs by effective planning; incremental, low-risk development; and managing project scope.

- Involve suppliers and subcontractors early;

  Suppliers know their product technology, product application, and process constraints best. Utilise this expertise during product development and optimise product designs to the capabilities of the "virtual factory" which includes these suppliers.

- Develop robust designs;

  Quality engineering and reliability techniques such as Design of Experiments provide an efficient way to understand the role and interaction of product and process parameters with a performance or quality characteristic leading to robust designs and enhanced reliability.

- Integrate CAE, CAD & CAM tools;

  These tools, when intelligently and cost effectively applied, can lead to a streamlined development process and project organisation.

- Simulate product performance and manufacturing processes electronically;

  Analysis and simulation tools such as FEA, circuit simulation, thermal analysis, NC verification and software simulation can be used to develop and refine both product and process design inexpensively. These tools should be used early in the development process to develop a more mature design and to reduce the number of time-consuming design/build/test iterations for mock-ups and developmental prototypes.

- Create an efficient development approach;
  Align policies, performance appraisal, and reward systems to support these development objectives and team-based approaches.
- Improve the design process continuously;
  Continued integration of technical tools, design activities and formal methodologies will improve the design process. Use benchmarking as an objective basis for comparing the organisation and its products to other companies and their products and identifying opportunities for improvement.

3.3.2. E-Commerce / Business To Business

The domain of E-commerce currently encompasses a wide range of innovative applications some are essentially metaphors of real world commerce these can effectively be grouped into:

- The E-store;
- Advertising;
- E-Procurement;
- Marketing / Branding;
- Collaboration Tools.

It is this last category of collaboration that this research work is primarily concerned with however valuable lessons can be learned from the implementation of the other systems with roots in the same environment.

3.3.3. The E-Store

These systems based their interface on a metaphor of the high street shop, browsing a selected range of products online arranged into categories or located through searching. These are then selected and placed in to a basket that serves as a temporary placeholder until all the other selections have been made. The user then "proceeds to the checkout" such a system allows the items that are selected, to be totalled and the users financial information for the purchase to be taken.

3.3.4. Advertising

Revenue raised through the means of advertising on the Internet operates in a similar manner to that found in the field of television. A suitably interesting programme is produced and the more viewers it obtains the higher the price can be charged for the commercial break period. Concerning the internet it is relatively easy to gain accurate figures for the number of visitors to a web site and thus a suitable financial compensation can be charged to organisations wishing to include the now common "banner" advert. The primary aim for sites should be content that is desirable to the majority to increase the number of different visitors, it is less useful to obtain a high throughput of visitors but find that it is the same person checking back several times a day for an update. In addition such sites should be of small page size, as they will have to bear the burden of the user having to download an advert as well. The longer a user has to wait for a page to download the greater the chance they will avoid looking at the site again.

3.3.5. E-Procurement

Tendering for services is often an untimely and laborious process, submitting a sealed bid for a service that you can provide usually results in a long turn around time and often only allows a small number of suppliers to submit a tender for what they can provide. E-Procurement allows an organisation to advertise for a much wider range of competitive tenders that can be processed efficiently, quickly and in some cases if certain fixed criteria are not met, and then these can be filtered out automatically. This leads to the organisation obtaining a higher quality bid that they would otherwise have been able to achieve, so the system doesn't directly bring in more revenue but saves on potential expenditure and the addition benefits of obtaining the right supplier.

3.3.6. Marketing / Branding

Quite simply this is an online advert. It serves to provide the view with either company information for the purpose of contacting them, information about upcoming products, advertising their full product range that a local store many not be able to carry. Some sites use this medium to further compound their image; rather than provide websites that are small and fast and provide you with information quickly, they serve to give a multimedia presentation increasing your opinion of their capabilities and products.

3.3.7. Collaboration Tools

The web is an environment for much opportunity and creative solutions to existing problems; one area of high contention is that of using the web for distributed team collaboration. These teams may be a product development team, a construction development team, any team where there is a need to coordinate actions and designs. These can provide a wide range of web-based tools, file sharing, email, message boards, discussion forums and so on. To raise revenue from this environment the most effective methods are to charge a company either for the full use of product, on a per project basis, or license each user seat. This approach relies on providing global access to a service that would usually require co-location to achieve.

3.3.8. Web Based Collaboration

The advent of a global market fuelled dramatically through the explosion of the popularity of the Internet often now requires in turn a global design and product development team. There have been various attempts at distributed application sharing (Srinivas (1992)), but the genuine efforts at e-commerce solutions to support supply chain integration and design team support need to be based in the utilisation of the web browser as the client. The web browser provides extensive and varied support for the display and often a degree of manipulation of media. Such a flexible client operates initially via an information pull scenario minimising operating loads on a server based on pushing information. The capability of the web browser to retrieve extra software components or plugins to achieve enhanced functionality on demand provides a level of extensibility often unavailable in a more bespoke platform. With a client based in the browser environment there is no need to install additional client software to participate. Essentially any collaborative software system with the serious intent to integrate not only technologies but also the product development team must provide suitable access to the system in a manner that appreciates the disparate nature of globally distributed operating environments.

Figure (3-3) shows a typical scenario of a modern collaborative network, a mix of local area networks (LAN's) dial up connections and multiple operating systems. The World Wide Web has already bridged these differences; it is a natural extension of these technologies that a collaboration tool should be based to provide the widest possible access to potential contributors.

The web provides a suitable platform for the basis of software collaboration tools however there is a constant competitive rivalry between the developers of technologies for use on the Internet. Organisations such as Microsoft, Sun Micro Systems, Netscape, Opera provide varying frameworks for development. This volatile development environment provides a range of innovative technologies however it also introduces extra barriers to the issue of platform independence, as there is much to be gained through tying users to a specific platform. The antitrust case of Microsoft and Netscape in late 1990's and early Millennium demonstrates this aspect of the arena. There are two main areas to be considered in the utilisation of web-deployed system, the client and the server. Platform independence in terms of the server although useful is less important in the context of the type of system proposed in this thesis. A server / network of servers can provide services for a large number of clients. An organisation can configure the server side of the equation in preparation for collaboration however it is the client side that holds the key to global information sharing through an accessible interface. Client interfaces are required to maximise portability within the limits of the industry competition previously mentioned. HTML 4.0 is effectively the most portable of the web technologies however this is only a simple text mark-up language ill suited to an application interface. Married with JavaScript it can provide a graphic designer with scope for an inventive human computer interface.

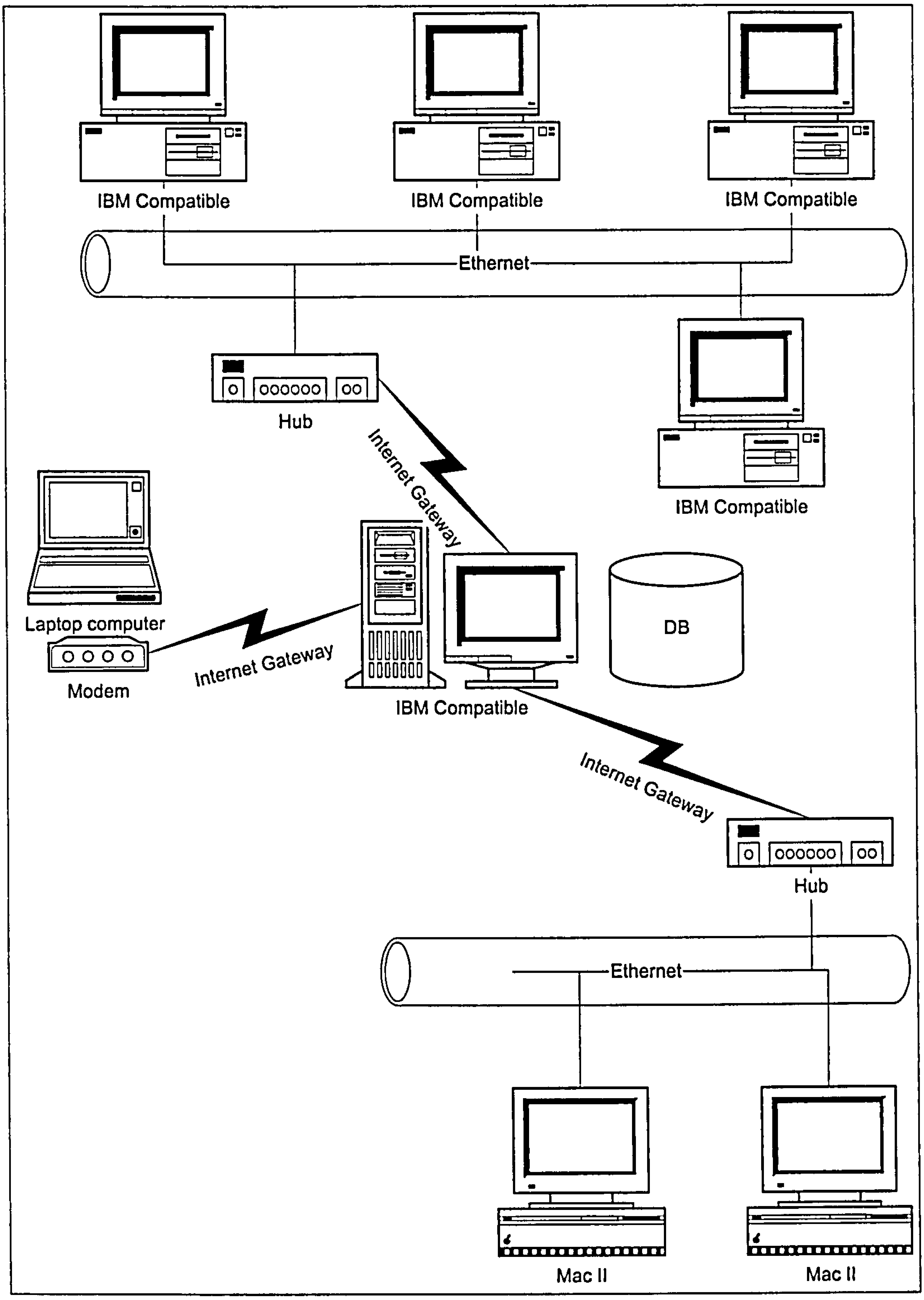


Figure (3-3) Multiple User Connectivity To The Internet

JavaScript however is better suited to providing basic interaction in web pages rather than powering an application interface. In the 'browser wars' the Document Object Model, a hierarchy of components in a web page, is often implemented differently in various browsers providing inconsistent results and differing programming interfaces.

This essentially directs collaboration based on a Web centric E-Commerce environment to necessitate the use of a Java-to-Java solution. This allows for a traditionally powerful interface running within a JVM (Java Virtual Machine). The JVM provides the environment for the application or as is the case with Java, an applet, to run seamlessly independent of the incompatibility problems of the client browser. The JVM provides the interpretation for the web browser. Running java server applications on the web server allows a direct connection between the interface and the server breaking away from the traditional pull only mechanism to a push / pull scenario.

Collaborative product development requires a strong technical solution complementing human factor considerations along with Concurrent Engineering philosophises.

3.3.9. Distributed Information Sharing System

The development of e-commerce and collaboration support tools especially in the arena of flexible multi-media web browser clients is increasing rapidly in significance. The potential for collaboration is vast with emerging technologies swiftly adopted such as Java and XML. Collaboration however should provide reasonable effective access for all members of a collaborating team alongside an intuitive information format that provides:

- Flexibility of representation;
- Simplicity in depth; information should be represented in a methodology capable of swift information capture yet able to represent complex data;
- Intuitive information format;
- Short learning curve;
- Support for contextual representations;
- Multi-media support;

- Support for both synchronous and asynchronous communications within the information representation.

A software based information representation has been developed to facilitate solutions to these problems.

3.4. Overall Architecture of the Proposed System

A conceptual level of information sharing and storage is vital to the next generation of collaborative systems to be able to utilise the knowledge the virtual team possess in a timely manner suitable for tackling product development issues at the phase where the future impact of decisions is at its greatest. Collaborative efforts thus far in the researched literature do not tackle the product development lifecycle as early as this research deems is necessary to further maximise the increase in the organisations competitiveness.

The proposed system consists of a web-based concept-mapping module, data sharing and structuring module, data standardisation module, object oriented databases, CAD solid modelling, a conflict resolution module, and HTTP (hypertext transfer protocol) accessed documents including 3D model viewing software. The proposed system architecture is shown in Figure (3-4).

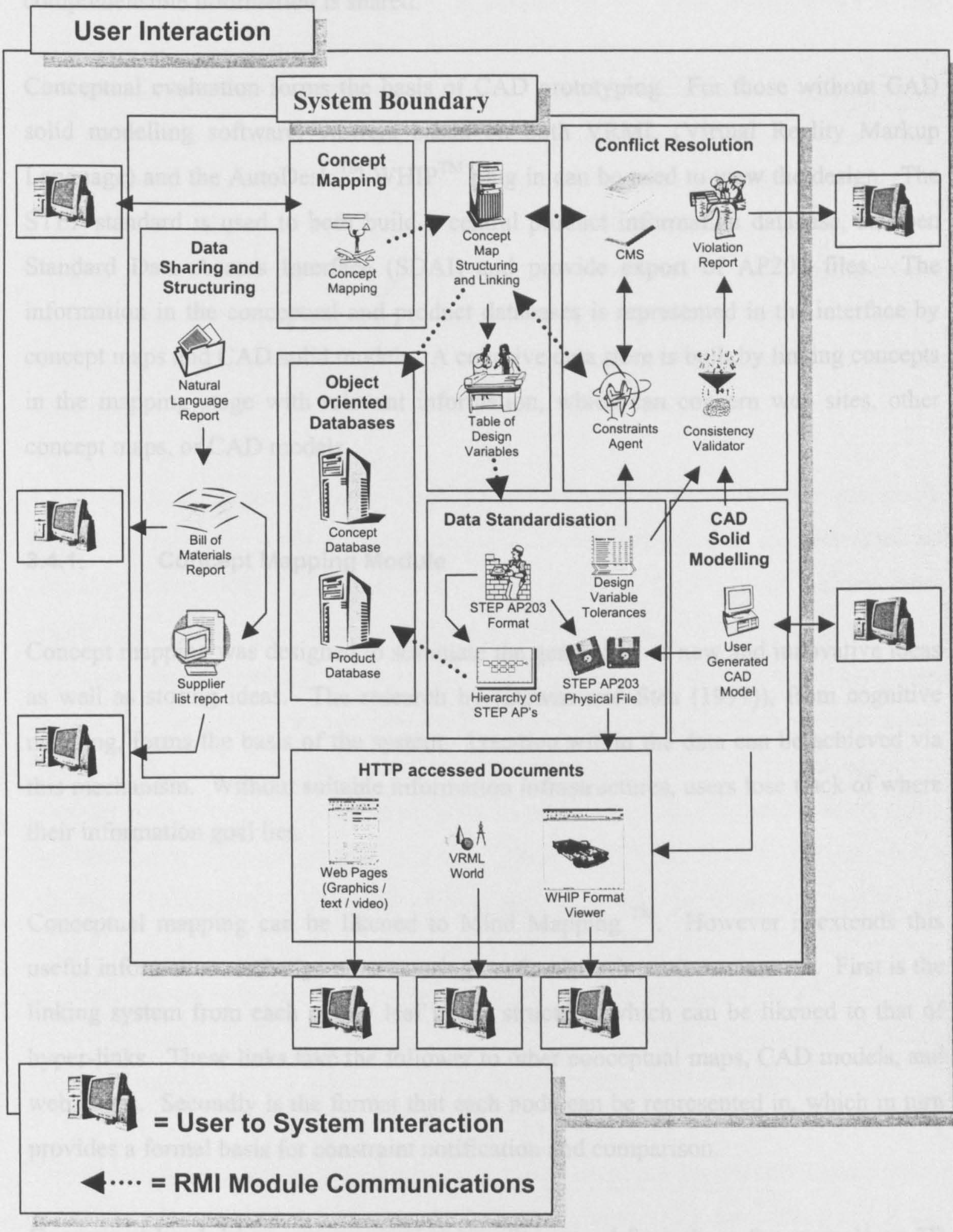


Figure (3-4) Proposed System Architecture

Fundamentally significant conceptual information is gathered at the start of the product development lifecycle. These valuable data are stored in a central concept database, which is structured using XML (extensible Markup Language) ensuring formal and comprehensible information is shared.

Conceptual evaluation forms the basis of CAD prototyping. For those without CAD solid modelling software, viewers based on both VRML (Virtual Reality Markup Language) and the AutoDesk ™ WHIP™ plug in can be used to view the design. The STEP standard is used to both build a central product information database, an open Standard Data Access Interface (SDAI) and provide export of AP203 files. The information in the conceptual and product databases is represented in the interface by concept maps and CAD solid models. A cohesive data store is built by linking concepts in the mapping stage with relevant information, which can concern web sites, other concept maps, or CAD models.

3.4.1. Concept Mapping Module

Concept mapping was designed to stimulate the generation of new and innovative ideas as well as storing ideas. The research by (Downs and Stea (1997)), from cognitive mapping, forms the basis of the system. Location within the data can be achieved via this mechanism. Without suitable information infrastructures, users lose track of where their information goal lies.

Conceptual mapping can be likened to Mind Mapping ™. However it extends this useful information exchange or structuring mechanisms in various manners. First is the linking system from each node / leaf in the structure, which can be likened to that of hyper-links. These links take the follower to other conceptual maps, CAD models, and web pages. Secondly is the format that each node can be represented in, which in turn provides a formal basis for constraint notification and comparison.

Conceptual level driven communication benefits extend from those for example on 3D models of pipe arrangements for chemical plants. Utilisation of a 3D model allows maintenance teams to be taken through the expected maintenance routine, in a virtual environment so that their input could be obtained. However 3D modelling is slow in comparison to an overview driven conceptual analysis. Obviously deep intricate detail cannot be considered at the conceptual level. However higher level thought trains, which otherwise would not be visible to the rest of the team, could be taken into account in the least amount of time. More detailed viewpoints such as the 3D models can then be linked into the system for further analysis. The research project offers various levels of detail for information to be presented; this facilitates new team members with a system, which not only allows them to gain the essence of project requirements quickly, but also allows them to delve into regions of greater detail if required. Much in the same way that a contents page in a book provides a framework for subsequent information, a concept map can offer an intuitive assistive cognitive medium for the comprehension of the various data types encountered in product development.

This research demonstrates the development of concept mapping to bridge geographical boundaries and time zones. Hence addressing factors identified in research work by (Zhuang, et al. (2000)). This previous research pinpoints the importance of a system, which can deal with synchronous and asynchronous distributed interaction. Asynchronous communication is a vital factor, considering that globally distributed product development teams are very much affected by working in different time zones. Concept mapping, developed in the system and described in this thesis, seamlessly provides both synchronous and asynchronous communication. Team members using the browser-based concept map tool are provided with the facility to construct a conceptual representation of a product. The greatest benefits are gained through the use of the system in real time, sharing access control during conceptual collaboration. However the concepts can also be worked on offline when the global distribution of the team members prohibits real time interaction. Thus the system is in accordance with Zhuang's requirement for synchronous and asynchronous distributed interaction.

The developed concept map tool was built using advanced Java technologies, Swing Java RMI (Remote Method Invocation), SAX (Simple API for XML) and DOM (Document Object Model) providing a platform independent creative framework tool. The tool allows the viewing of the XML based concept maps retrieved from the previously described concept database. Each user has access to a browser-based client, which provides access to the maps in the manner of a shared drawing tool, with control from menus, toolbar, dialogues and direct manipulation of the map area. These maps are viewable by the team as a whole and there is the facility provided to create links from added concepts to other relevant documents such as other concept maps, text, video, CAD models, etc. Storage of the concept map structure and the links embedded within the concept is implemented in XML.

The concept mapping technique has the capability to provide a precise organisational level, imparting a framework for subsequent media such as CAD solid models to be incorporated.

3.4.2. Data Sharing and Structuring Module

Data have to be structured both logically and technologically to contribute to the success of a product's development. The Data Sharing and Structuring module interacts with the concept mapping providing the infrastructure for sharing and structuring of data within the system.

Concept map structures and data related to material properties are integrated in the developed data sharing and structuring module. These concept maps can be parsed to produce a natural language report. Concept map data can be processed to allow configuration of the information for a specific report or user group resulting in, for example, a bill of materials or a supplier list. Table (3-1) illustrates some of the wide range of the possible media, which can be context linked by the data structuring module.

Media	Linkable to:
Concept Maps	Design Variable Table, WHIP model, VRML, CAD Solid model STEP AP203 file, Video, Graphics, Web pages, Plain text
Design Variable Table	N/A
WHIP .dwf (drawing web format)	WHIP model, Video, Graphics, Web pages
VRML world	VRML, Video, Graphics , Web Pages
HTTP embedded graphics	Web Pages, Graphics, Video, VRML
Web pages	Web Pages, Graphic, Video, VRML
Plain text documents	N/A

Table (3-1) Linkable media types

Processing of a concept map is capable of generating a table of design variables, which in turn forms the basis for building a CAD solid model or alternatively a mapping into STEP Application Protocol (AP203).

The diagram in Figure (3-5) illustrates a possible data structure that DISCS could accommodate, based upon an example set of documents concerning a new aircraft concept. It highlights the structural connectivity and hence the navigation routes the information can be traversed in; concept map structures can be connected to other purposeful documents such as maintenance documents. The diagram has been centred on a more traditional lifecycle to show the data, which would be created in a time line fashion. However the proposed system is more dynamic and connective to data retrieval.

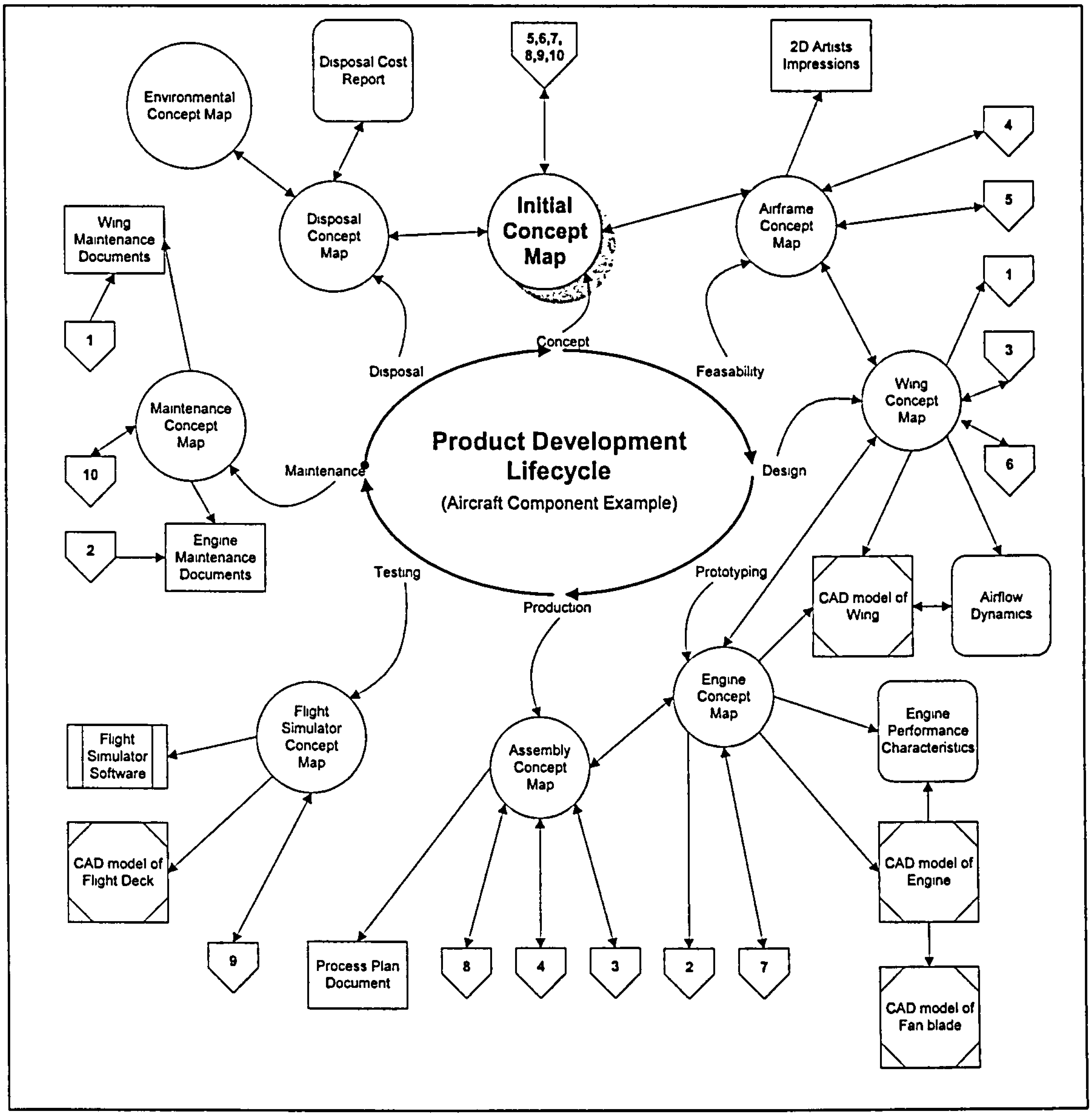


Figure (3-5) Data structure connectivity within the product development lifecycle

The structural representation of concepts was carried out using the advanced information structuring language XML. The eXtensible Markup Language (XML) is an ideal language to structure the concepts and their relationships. These relationships can exist between other concepts, other concept maps, VRML worlds, WHIP drawing for web format models, and web pages. The eXtensible Style Language (XSL) in conjunction with XML provides a greater flexibility for the display of the data. A DTD (Document Type Definition) has been constructed with which XML can be checked for correctness and consistency. This mechanism is also implemented to exploit a level of control over the structure of the concept maps, should an industry for example, find the concept maps spiralling out of control in terms of size and complexity. In this case a wide variety of limiting mechanisms can be put in place. This would force conciseness of the issue under discussion.

The data-structuring module, along with its corresponding DTD, parses an XML file for structural integrity. XSL allows a different appearance to be placed upon the concept maps. This can be utilised in the future to tailor the display to suit the user or media type. Interfacing to the system is achieved through use of the DTD. Other systems can interface to represent their information within a concept map structure defined by the DTD, which the system can then interpret and incorporate.

The relationship between the open DTD allows correctly formatted and well-formed XML data to be checked and validated as illustrated in Figure (3-6). This shows examples of some of the supported MIME (Multipurpose Internet Mail Extensions) types, the team members, and the linking support.

MIME types are defined and are integral to the linking mechanism, facilitating the storage of related data together with the relevant concept. Concept tracing is facilitated through the structural ownership of a concept, a contact point can be found for each individual concept.

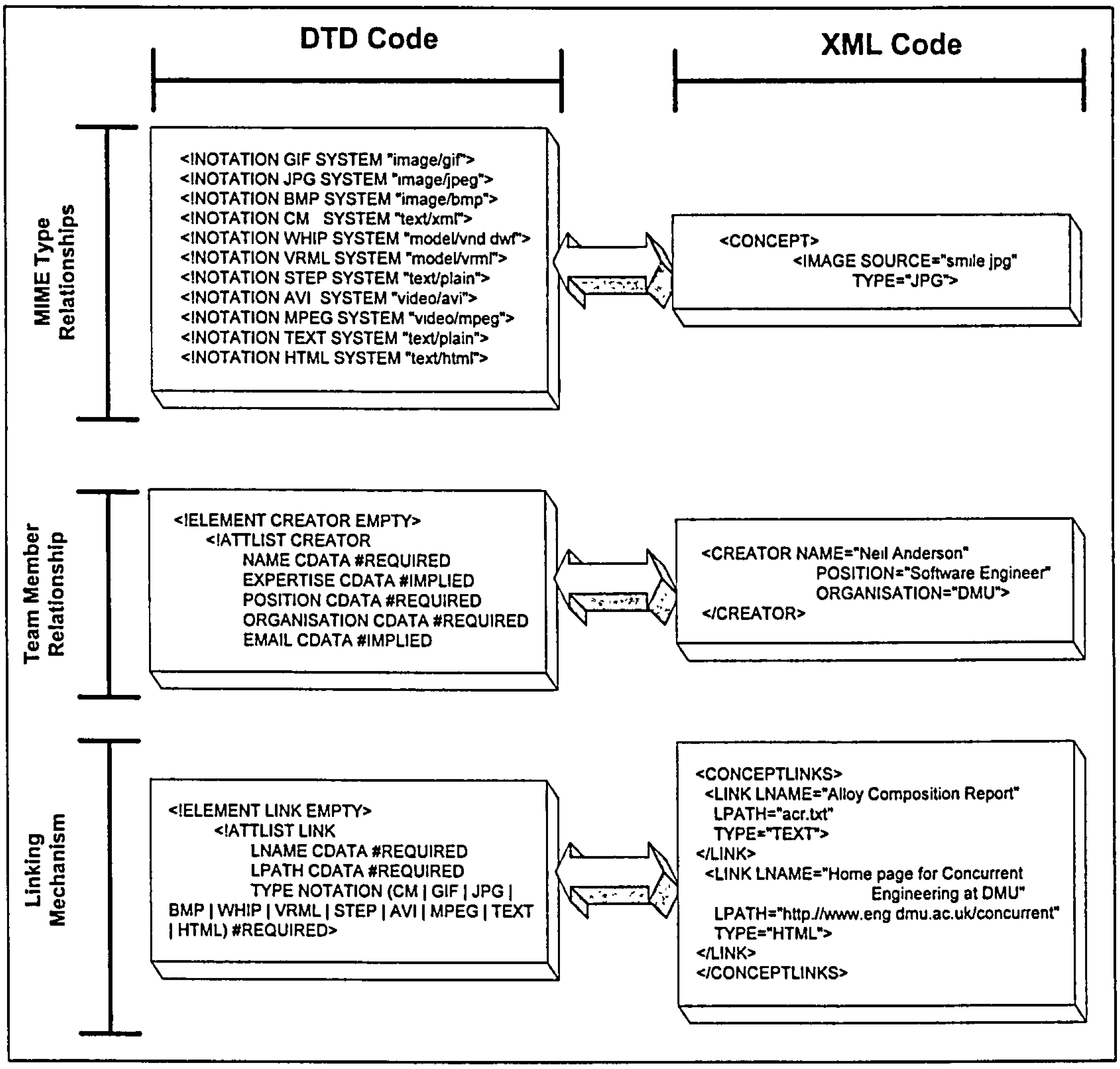


Figure (3-6) Document Type Definition and XML relationships

3.4.3. Data Standardisation Module

The lifespan of product data is a vital issue to be tackled. The provision for product data to effectively be application independent is essential in today's climate of rapid software change. The utilisation of standard technologies to develop the system greatly increases the lifespan of stored data. Structural meaning of the data is preserved outside the application's scope, preventing over time as technologies change, the loss of access to legacy data.

Collaborative product development necessitates standardisation of data using STEP. This research work has dealt with product data exchange to facilitate sharing of data at various complexity levels without conflict. There is a strong drive behind the philosophy to relate data to the products and processes that the data describe, as opposed to the computer systems, which create the data. Life times of the applications software, system software and the hardware, which were used to create the data, will have much shorter lifecycles than the data itself. Application software has a typical lifecycle of 3 to 5 years, systems software 5 to 10 years and the hardware even less with cycles of less than 3 years (Fowler (1995)). Thus the integration of a standard for storage and interpretation within DISCS is a vital mechanism in overcoming the migration problems (Hares (1994)), which are rapidly occurring with legacy systems today. The works of (An, et al. (1995)), (Urban (1993)), (Peng and Trappey (1998)), and (Loffredo (1998)) provide further detail on STEP and its application.

Standardisation applies not only to that of the data structures but also to the portability of the underlying modules. This is a highly important factor to consider where the deployment of such a system is not only likely to cross-geographic boundaries but also operating environments. The technological foundations of the architecture have a solid basis in portable technologies. No matter how sophisticated a system is, it will provide little impact in the organisations concerned if it cannot work within the bounds of the lowest common technology denominator between organisations. Information Technology investment in platforms cannot be disregarded, but should be exploited, (Anderson and Abdalla (1999)).

3.4.4. Object Oriented Database Module

Object Oriented Databases are ideally suited to the storing of complex product development data. A concept database and a product database are used within this research. The concept database stores the concept map XML data structures and the product database stores the STEP product data. Each database is connected through links embedded within the concepts.

The embodiment of linking media types is further demonstrated, in Figure (3-7), which shows both the design iterations and the relationships to the object oriented databases. It starts from the initial top-level meeting through to a set of documents taking into account the nested management infrastructures in today's organisations. The diagram is divided, showing the information flow on the left and the stored linked data model on the right.

A sharable model of integrated knowledge through context-linked media is stored in the object-oriented databases. The information that is captured by various experts, is information model driven. This allows navigation by users with a wide range of computer literacy levels and knowledge bases. The system allows all the information concerning the product during development and afterwards to be collated in useful manner in a central repository for access by those concerned.

3.4.5. CAD Solid Modelling Module

CAD solid models form an essential element in the data structure of a DISCS data framework. They provide the consolidation of data from iterations of concept mapping. To assist with the interface to STEP AP203, Unigraphics CAD Solid modelling version 16 has been used in this research.

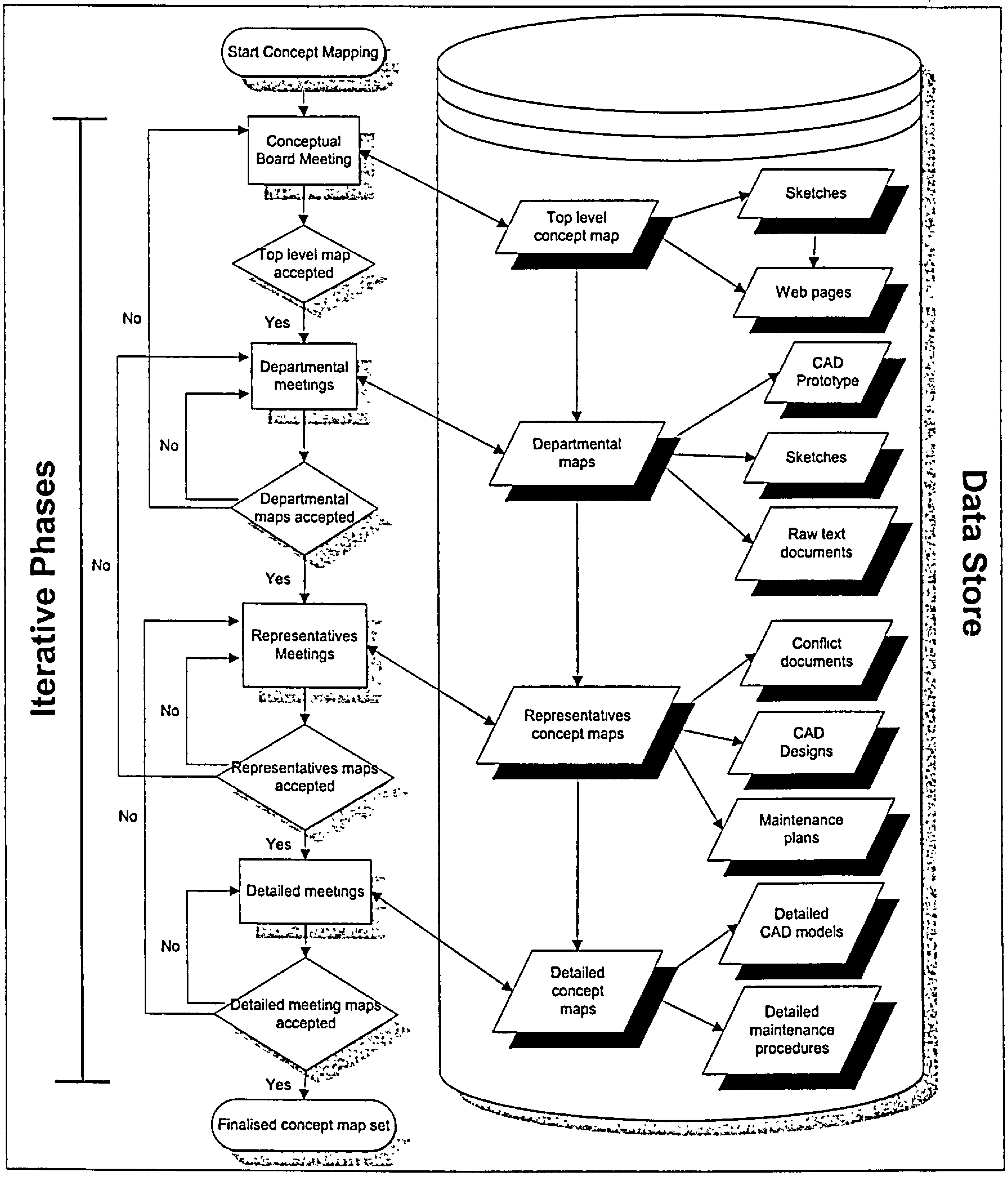


Figure (3-7) Object Oriented Database & Information iteration flow

3.4.6. Design Consistency Module

Conflict resolution throughout product development is one of the purposes of shared communication. Conceptual analysis of different team member viewpoints allows conflicts to be identified at the earliest opportunity. It provides a basis with which to reach a consensus.

Concept map structures checked by a constraint agent would alert the creator of a concept to any conflicts within the constraints defined by it and other existing concepts. A link to CMS (Cambridge Material Selector) incorporates the inclusion of material properties as constraints. Concept maps form a hierarchy; for those that define constraint parameters these are analysed for confliction with parent concepts. Notification of a conflict occurs both at the time of a conflicting concept to the user creating the concept, and via email to the user who created the opposing concept.

Consistency validation compares a design variable tolerance consensus set with a user generated CAD model. This provides conflict alert support for those without STEP capability for whom the design variables are propagated directly to an AP203 format.

3.4.7. HTTP Accessed Documents

The system has been developed to utilise the growing base of technologies for the web. Within the previously mentioned linking mechanism is the facility to input Uniform Resource Locations (URL) to web based documents. The connection from concepts to other web based data utilises the status of the web today. Suppliers are likely to have a corporate web site allowing their details to be incorporated in concepts that relate to their products. There is the facility to incorporate other web-based systems. Within the developed system is the capability to display CAD solid models via a web-enabled format (Drawing Web Format [.dwf]). This enables discussion for those without expensive CAD solid modelling software.

Utilisation of URL facilitates the wide range of developed data formats to be accessible through DISCS increasing the scope for information structuring, which are usually limited by web browser technology.

3.5. Example Scenario of the Proposed System

The first stage of operation is the capture of the conceptual requirements. Followed by the creation of supplementary material to the concepts, which could be carried out in a variety of comprehensive Commercial off the Shelf (COTS) products. These could then be attached via a relationship connection to a concept to provide a semantic association for retrieval. An iterative loop during the conceptual phase would ensue until a firm set of conceptual foundations had been established and agreed upon by the virtual team. It is these foundations that further details can be constructed upon. 3D model prototypes would be constructed from a solid basis and again could have a relationship between themselves and the conceptual level from which they were derived. Supplementary documentation, which supports explanation of the models, can be linked to the model from various nodes existing in the model. This documentation can be selected as supporting design documentation or as maintenance guidance manuals.

3.5.1. Example

This example scenario will give an improved understanding of the possible impact that an improvement in the communication of a project team can facilitate the use of concurrent engineering over a distributed project team. The automotive industry is the chosen example with a new car as the subject.

Using the company structure with the design department in the USA and construction in Britain, the following shows how input that would be unlikely to be introduced into a project can be obtained and has substantial value.

The concept tool could in theory accept a large number of participants, however discussions generally evolve to a useful conclusion with smaller groups. With this in mind the initial discussion would take place within the high-level design team to collect the initial thoughts concerning the new car. Instead of all the members of this group travelling to a design centre the concept tool can be used to collect ideas from Customers, Sales, Marketing, Design, Finance etc. Viability at this level is purely to discover a market for the car. Concept maps would be created as the first point of call for the next stages in the project. Any textual / sketched ideas can be linked to the concept they represent, clarifying the idea.

Using the market and ideas defined in the concept maps a CAD team could construct a basic outline for the car this could be taken from the linked preliminary sketches in the concept maps. The purpose of this model is not only to enable the visualisation of a goal but also to create a point of contact where more concept maps, CAD models and text documents can be linked.

In Britain where construction is undertaken, a virtual meeting takes place using the concept tool; members of this team review the proposed ideas to add their views at this early stage. Members of this team include assembly line representative, purchasing, testing etc. Assembly notes that the chassis for the new car would be of similar dimensions to that of an existing model.

Assembly points out that the time lost in refurbishing / setting up a new plant to deal with new models often causes a lot of teething problems, these could be avoiding by reusing the existing chassis design, saving on retraining the workforce. Purchasing also adds that new supply chain structures have to be set up and often yield problems with initial quality and adequate supplier relationships, using the same chassis could also reduce these. A quality engineer present concludes that the quality control mechanisms in place could also be reused, allowing them to be built on and improved. Thus the reliability of the product chassis components would be increased, hence increasing its quality at little cost along with the process improvement. All of this is stored within a concept map linked to the chassis in the model and in the initial concept map.

Joint Meeting

A representative from the British meeting is present along with members of the American design team, the previous meeting is clear from the concept map, and the members can all view the proposed ideas. Technical documents on the existing chassis have been linked to the initial CAD model from the first meeting; these can be viewed in the context of the discussion that is clear to all present. The input from the quality control engineer can be viewed along with a set of statistics; he has linked to his concept within the map. All the information required for the decision can, either be retrieved directly from the concepts of interest or extracted in real time from the present virtual team members.

In all of the meetings supplementary communication tools are available such as video conferencing, email, bulletin boards, a full suite is available to enable as rich a communication experience as possible.

Scenario Conclusion

In summation within this example a high level decision was made which effected quality, purchasing, assembly, training, and supply chains. A small idea given by an assembly worker sparked off a debate in a forum allowing consideration by the relevant staff. This then facilitated other benefits to be identified and worked out until it becomes an effective solution. The information sharing mechanism developed enables the virtual meetings to offer a closer experience of the communication benefits that face-to-face meetings can provide, possibly even being more productive with the information context structuring and linking mechanism considered. The information is available when it is required, where it is required, and retrieved in the context it is required.

3.5.2. Iterative Flow

The system is suitable for both synchronous and asynchronous communications. For both the generation and retrieval of product data there is an intended iterative flow for its use. This flow of course is highly flexible. However just as the information sharing mechanism allows a practical structuring of product data to obtain information from a product development team in an efficient manner, there is a similar information flow into the system as follows the most beneficial data structuring.

The expected system flow is outlined in the example scenario above, however a more independent and generic explanation follows:

Conceptual Board Meeting

It is at this stage that the initial people involved in the product idea along with those who have the power to decide whether to proceed or not meet. The output from this meeting should be a high level concept map showing the pros and cons of the product enabling some judgement to be formed as whether to proceed with more undertakings on the matter. To assist any explanations within the concept map, sketches and any possible related web pages could be linked in for reference.

Departmental Meetings

The departmental stage is where each considered department using the concept maps and the assisting information from the initial stage to act as an instigator, produces a concept map, concerning that department's issues and concerns in the matter. From this concept map it could be expected to link basic CAD models - simple prototype ideas, 2D sketches, and text documents concerning memos etc.

Department Representatives Meetings

The purpose of this gathering is to bring together the concept maps from the previous separate meetings; a representative familiar with their department's concept maps and assisting information contributes on behalf of the department. The individual departmental maps can be linked into initial map to allow all participants a view of the concerns and problems foreseen by the other domain specialists. At this level various other documents are linked into the map to give a more comprehensive view of the problem domain. This would include conflict documents, detailed CAD models, environmental considerations, maintenance plan, etc.

Detailed Meetings

These concern defined problems within a problem domain and scope. They require the virtual attendance of a few specialists. At this level the concept maps are only used as anchor points to relate new documents to, the real work added in fine detailed CAD models, manufacturing process documents, and intricate maintenance procedures.

Retrieval of Information

This is a stage when the product is under review, a problem has slipped through the net, training is being given, maintenance documents retrieved, etc. For this stage a new concept map can be generated to give an improved view of earlier constructed documents. This needn't replace existing work merely supplement the ease of information retrieval.

Iteration Points

This sequence is by no means intended to be a single run through, each stage can return to the previous, should it be required to obtain the input from the people at that level.

3.6. Summary

The proposed information sharing approach has been presented in this chapter, showing the importance and relevance of tackling the problem of collaborative working from the conceptual level of product development. The work comprises a multifaceted system providing support for the product development process from initial stages through to completion that is demonstrated in an example scenario. This chapter has detailed some of the objectives of the work by integrating the following aspects in the proposed approach:

- The adoption and utilisation of the Concurrent Engineering philosophy, taking into account the consideration of subsequent downstream product development activities. Harnessing information resident within the collaborative team to achieve a group awareness of other domains requirements and suggestions.
- Collaborative product development detailing the objectives, principles and rational behind integrated product development. The drive behind different facets of E-Commerce covering the different mechanisms of realistic revenue generation. The practical considerations of web-based collaboration based upon the realistic state of the available development environment have been considered.
- The provision of a collaborative framework that facilitates information to be shared with increased conceptual and semantic understanding in turn provides high quality, suitable products with a reduced lead-time. A collaborative system that provides an information sharing and structuring framework based on the human cognition system is considered in the proposed system.
- Utilisation of flexible and portable information and Internet technologies, to facilitate extensibility and cater for the real world collaboration requirements of mismatched operating environments.
- A combination of both macro and micro information structuring context linked to supporting resources.
- The natural adoption of an information structure that satisfies both asynchronous and synchronous communications within the single information model is

integral to the proposed system. Integrating knowledge from real time communication and serial contacts is essential in avoiding disparate islands of unrelated information.

- A portable and flexible user-friendly interface designed for use within a web browser providing a high level of client access.
- Utilisation of current technologies, OODB, Java, XML.
- State of the art information modelling utilising XML and its incorporated DTD's to structure and evaluate consistency, providing a data format that is independent from the system that created it.
- Linking to other information objects through HTTP provides an extensible information model utilising the web browsers inherent ability to present a large range of information formats or alternatively direct the user to a plugin that can provide the capability the user requires.

Chapter 4

CHAPTER 4. Concept Mapping

4.1. Overview

The development and detailing of concept maps in both the semantic and technical contexts are presented in this chapter. The considerations for the many facets of information sharing are demonstrated in this work and a method for bridging both cognitive and technical barriers is shown. A communication mechanism that allows distributed and multiple domain specialised teams to share and structure information from the initial phases through to disposal and maintenance is encompassed in this research.

4.2. Introduction

The utilisation of all available information at the earliest possible stage in the product development lifecycle with the intent to increase competitiveness in industry is key within the Concurrent Engineering philosophy. The exploitation of accurate and timely information at these vital and most influential stages of development requires a greater involvement of costs and effort at this early point in the lifecycle.

A facilitating information-sharing platform is essential for global teams at this early stage of the product development process. Investment in information sharing at the core stages allows designers to design more effectively for X. With improvements in the collection and collation of essential ideas, restrictions and requirements, the designer's productivity is increased through a reduction in unnecessary design iterations.

Maximisation of a global team's expertise in contributing to the design rationale is paramount, much of the previous research effort as detailed in Chapter 2 has been focused on this harnessing of a distributed team but in a limited context and within a limited scope focusing on less influential stages in the product development lifecycle.

The necessary elements for global distributed information sharing can be classed as follows:

- Supportive of synchronous and asynchronous communications (Zhuang, et al. (2000));
- Facilitate as wide user domain base as possible;
- Allows the combination of external context linked media;
- Structures information in a fluid and flexible manner;
- Encourages effective design innovation;
- Enables knowledge to be divested in a manner with strong coherence to the manner of retrieval;
- Both complexity and simplicity should be effectively represented within the same shared manner;
- Continuity of data structuring to allow for a high level of reuse by a wide range of interested parties, for example specific product training or design rational learning.

For an information-sharing platform to provide effective assistance and not conversely a hindrance as a new barrier to information access, these requirements should be encompassed in a proposed solution. Access to the system to divulge and contribute information should not be impeded (except for deliberate security purposes) and there should not be new barriers to information comprehension.

The collation of team contributions into a coherent structure on a shared platform while facilitating global accessibility at the conceptual stage through the development and utilisation of conceptual mapping provides a solid foundation for subsequent design phases.

4.3. Concept Maps

In modern industrial times there has been an ever-increasing pressure for production units to be able to produce products faster, of an improved quality, with greater applicability to market and for a lower cost. Instead of the traditional product development lifecycle as has been detailed earlier, there is a drive to greater levels of concurrent advancements breaking down the barriers of handing over the development at various phases.

A design brief will be out of date from the moment it is produced if not before. Traditionally this will be because the product development team at the conception of a project are largely uninvolved. New requirements will come to the fore as missing contributions are identified when other domain specialists are later consulted.

The design team effectively needs to be expanded to include the whole product development team. This extended design team may not all actually work on producing solutions to the problems but they all have a role to play in contributing necessary, accurate and timely information at the earliest moment of opportunity.

Expanding a development team early in the lifecycle has a multitude of implications:

- How to accommodate a range of view points each with their own requirements;
- How to provide an environment where input can be gathered from team members;
- How to maintain both structure and flexibility of the information gathered;
- How to encourage innovation while maintaining a design focus;
- How to allow those of different technical vocabularies to contribute;
- How to facilitate unique information formats to be incorporated;
- How to provide a facility for real time and linear communications;
- How to facilitate both concurrency and yet coherence.

Solving the problem of expanding the development team at the conceptual phase to allow for a greater diversity of contributions, in turn provides a greater stability for the rest of the design and development to be based upon than is currently possible.

Design teams are heavily involved in the early product development lifecycle, however they can only produce a design within the accuracy and relevance of the information they have at their disposal. This availability of information to the team at an essential early phase in the product development lifecycle for it to make a substantial difference to the future development of the product is a core purpose for the introduction of concept maps.

Concept mapping provides a unique medium with which to convey ideas and logical trains of thought. It aims to tackle many of the previously mentioned problems with information sharing in the product development lifecycle.

Its raw fundamental theories lay in that of Tony Buzan's successful mind mapping techniques (Buzan and Buzan (1995)). These have been designed to utilise the basic cognitive functions of the brain for thought processing, recall, and structuring.

Companies such as Boeing have already invested in such techniques to simplify their maintenance manuals. Such methods however can be utilised for much more than this, not only is the format for information structuring highly conducive to new thought generation, being closely related to the inherent information processing mechanisms in the brain within a group context it can also be applied to the conveyance of human thought.

These methods generally have in the past been applied to narrow fields of investigation at any one time contained within one mapping diagram (see Figure (4-1)). This example is actually based on the subject of mind maps itself. You will notice a similarity to that of brain storming diagrams and the purpose is similar yet utilising other facets of human cognition.

A passage from (Buzan and Buzan (1995)) explains the idea of combining knowledge from multiple sources into a single collaborative knowledge effort:

"In our daily lives we learn a myriad of information that is unique to each of us. Because of this uniqueness, each of us has knowledge and a perspective that is strictly

ours. Therefore it is beneficial to work with others during problem-solving tasks. By combining our Mind Map knowledge with others we further the associations that we as well as others make."

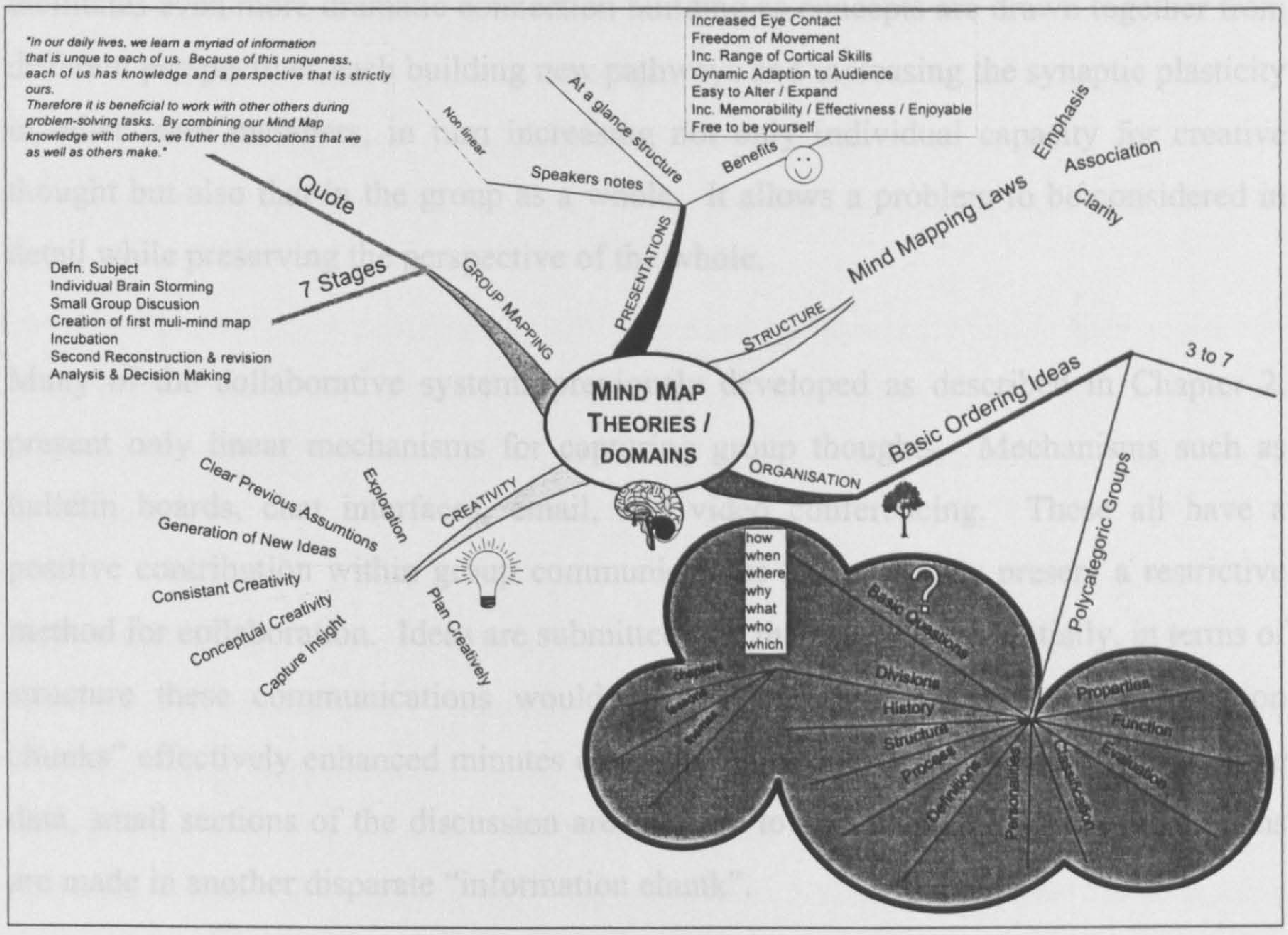


Figure (4-1) Mind Map

In the simple case the brain exists as a series of neurons connected through their respective axons and treelike branches called dendrites (Calvin (1997)). The strength of the connections is termed synaptic plasticity. The greater the synaptic plasticity between neurons the greater the chance of one thought firing the neurons related to other thoughts. Thus increasing the relationship between two concepts promoting recall and understanding. The more often a thought is held the greater the synaptic plasticity between the involved neurons. This is the brains mechanism for increasing its efficiency, the greater the synaptic plasticity the easier it can resolve thought.

Conceptual level mapping is very effective at triggering related thoughts and increasing the synaptic pathways between previously little connected concepts. This in turn increases the probability of novel thoughts and previously undiscovered insights, hence promoting creativity. In a collaborative group model as is presented in this work, it facilitates even more dramatic connection building as concepts are drawn together from different perspectives each building new pathways and increasing the synaptic plasticity in other team members, in turn increasing not only individual capacity for creative thought but also that in the group as a whole. It allows a problem to be considered in detail while preserving the perspective of the whole.

Many of the collaborative systems previously developed as described in Chapter 2, present only linear mechanisms for capturing group thoughts. Mechanisms such as bulletin boards, chat interfaces, email, and video conferencing. These all have a positive contribution within group communications however they present a restrictive method for collaboration. Ideas are submitted into the medium sequentially, in terms of structure these communications would have to be later gathered in "information chunks" effectively enhanced minutes of the meetings. There is little granularity to the data, small sections of the discussion are difficult to identify and further contributions are made in another disparate "information chunk".

Relationships between concepts are not made and would only be held in the individual perceptions of the group participants leading to probable differing subjective interpretations of matters. Essentially there is no contextual framework for new input to be placed, rendering it useful generally as a separate unit of knowledge without contributing to the whole perspective. There is little in the way of building positive new associations between ideas, they are merely a transaction of comments without structure within the knowledge of the product development in its entirety or relationship to the natural capabilities of human cognition.

The natural structure of a concept map embodies flexibility. Such a structure is highly effective for use when giving a presentation as opposed to a rigid linear series of notes. A linear format can be made from them for the sake of structured talk while allowing tangents to be explored in discussion.

The following are some of the identified benefits of using such a medium for a presentation (Buzan and Buzan (1995)):

- Increased eye contact with the audience;
- Freedom of movement, a single map can cover a subject area avoiding the need for many separate pages that must be maintained throughout the presentation;
- Dynamic adaptation to the audience. With a linear series of notes it is difficult to move about the subject area upon questions or adapt to time constraints.

In the domain of group collaborations it is an asset to have information gathered in the same manner as it can be used to redistribute. This saves a great deal of time in collating the resources for a topic then filtering out the essential elements for a presentation.

The flexible nature of the information representation overcomes the usage problems of some other methodologies. There have been many attempts at forming methodologies for representing design many of them complex with large symbol libraries to understand before being able to provide a contribution. These methodologies although aimed at providing a consistent interface to design, they also present an extra barrier to gaining the maximum contributions from a design team. Not all parties involved will know or perhaps even have the background to learn a complex design methodology, consequently only those who understand it thus excluding other potential valuable contributions will use the methodology, or the use of a design representation in such a manner is discarded altogether.

The concept map format is also highly adaptable not just representing the generation of high-level ideas. It lends itself to being capable of representing a variety of hierarchical information such as product assemblies, and part feature descriptions.

Figure (4-2) shows a concept map example concerning the features of a connecting rod, more specifically focused upon those of a holes type and how they correspond to a feature based tree structure.

Utilising the concept mapping structure in this way allows an effective hierarchy of maps to be generated during the lifecycle of a product each following an increasing level of represented detail. Taking development from its initial concepts to a representation of the actual features concerning a part and their requirements, for example, dimensions, tolerance, and materials.

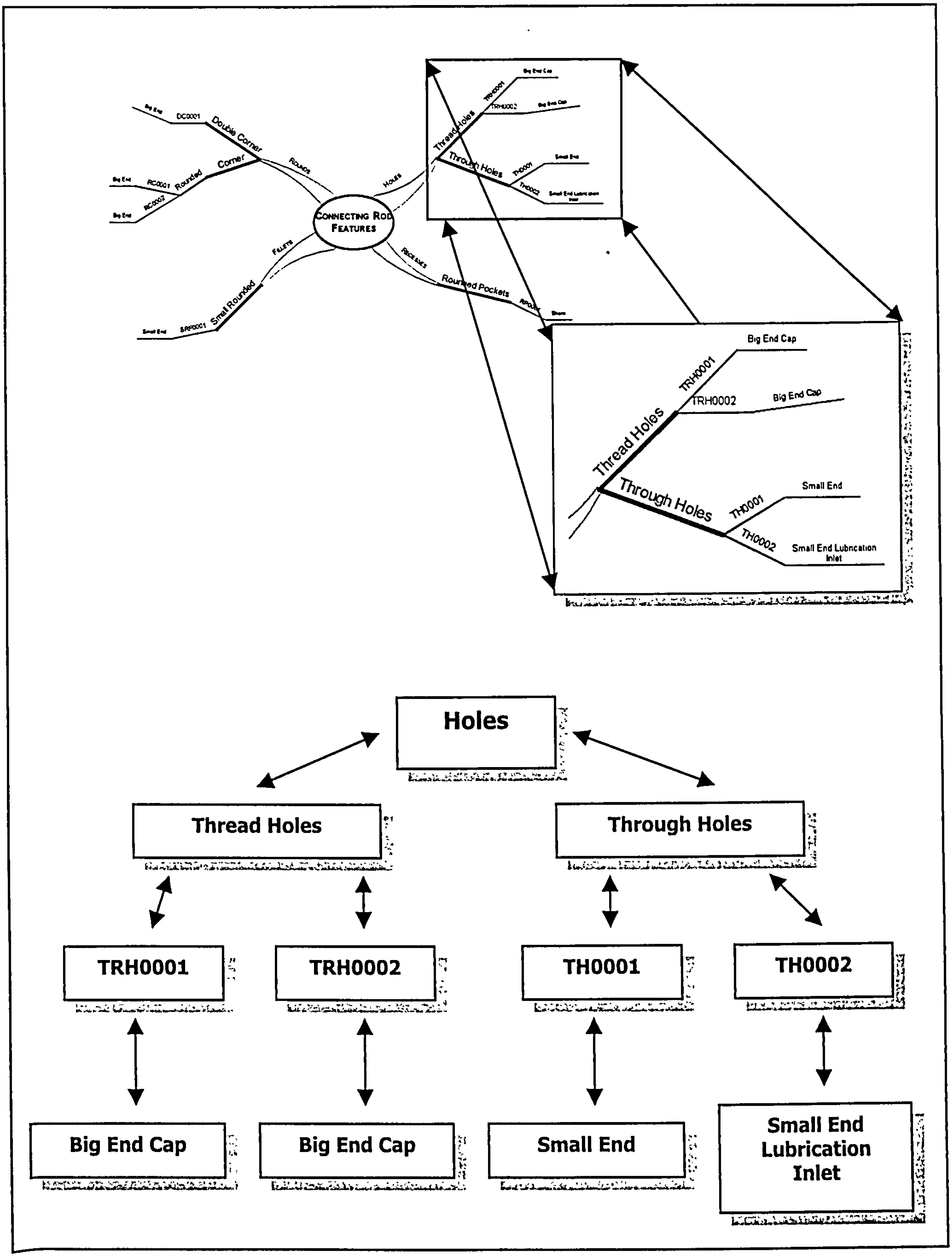


Figure (4-2) Concept Map and Corresponding Feature Tree

4.4. Data Structuring Within a Conceptual Framework

The novel approach taken with this research has been to develop methods for improving collaboration throughout a distributed product development team. The focus of this has been the facilitation of gathering important information from diverse sources and structuring these in a natural manner to aid cognition. Structuring of information to facilitate understanding and communication must revolve around a simple yet powerful framework.

The provision of the following factors is essential to the structure of a collaborative framework:

- Simplicity;
- Extensibility;
- Availability;
- Adaptability.

4.4.1. Simplicity

The structure must be simple to understand for the majority to be able to contribute which is the purpose of utilising a multi-domain team. Simplicity also encompasses consistency; structured information must be presented in a consistent manner to aid accurate comprehension from one data set to another.

4.4.2. Extensibility

The brain tends to look for pattern and completeness, termed gestalt (Buzan and Buzan (1995)). Where there are open-ended options the brain attempts to fill them in, its what leads people to collect every piece in a set, and makes jigsaw puzzles and crosswords addictive. This is an essential trait for every day perception however it can be used in a more powerful manner. Linear data representations are highly restrictive, it is difficult to insert new threads of thought / information and the brain has little impetus to provide completion. However in extensible information environments the brain is constantly searching to expand and fill the available area. Using a pen and paper approach the brain generally attempts to fill the bulk of the page, in the computer implementation the space is virtually limitless.

4.4.3. Availability

Information must be available in a timely manner to those who have the responsibility to contribute their own thoughts on a problem. If communications are primarily linear based such as email or text based chat transcripts it requires a greater investment in time to locate individual issues. Information within the mapping structures is presented in effectively a tree structure that allows rapid traversal to the issues of concern. Information must also be available through technical and temporal barriers.

4.4.4. Adaptability

Product development information can in reality come in a vast variety of forms, it is important for the core information-structuring format to be adaptable for the situations it is expected to encompass. Facility must also be provided for the unknown, inclusion of formats and legacy / inflexible data that cannot easily be manipulated into new structures.

4.4.5. Concept Map Navigation

Concept maps as has been previously described are more than just an individual manner for representing information and collaborating on a topic. They are also a navigation means for data mining. Mapping information provides a means with which to see the overall picture while being able to focus on a sub section of the task.

The idea of having a central node and expanding it through a series of branches to smaller further refined sub-topics can be greatly extended through the provision of a facility to treat a map as a central node in itself with other maps providing the sub branches. When this is taken a step further allowing not only maps to link to other similar more refined maps but also to other types of data organisation and allow them in turn to link to other maps then you have a highly fluid and flexible system for information representation and conveyance (see Figure (4-3)).

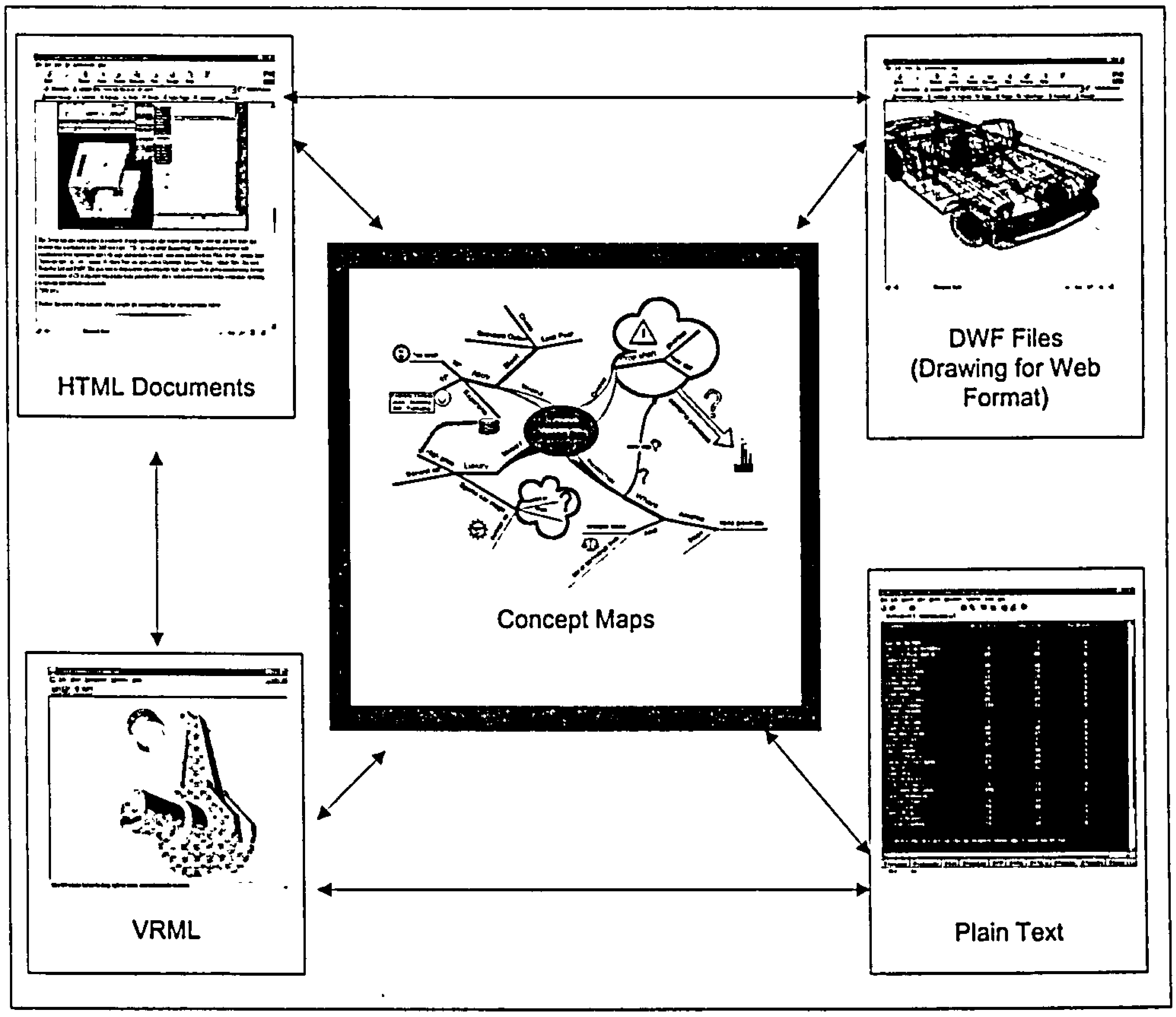


Figure (4-3) Data Organisation

Building an information structure such as this is performed during the communications of the product development lifecycle. It is relevant to another cognitive mechanism, cognitive mapping. Cognitive mapping is the process of building an internal representation of structures and relationships in the real world (Downs and Stea (1997)). An internal representation of a map allows us to perform tasks such as navigating to work and back. To navigate a cognitive map requires according to (Laszlo (1996)) three things to be known to the map holder:

- Whereness;

- Whatness;
- Whenness;

Whereness relates to the knowledge of the location, distance and direction of the next waypoint or destination. Whatness means the knowledge of characteristics of the objects or people that are within the destination or waypoints. Whenness refers to the identification of the waypoint either by means of pattern or its presence.

Concept maps utilise this natural cognitive mapping mechanism that we all use instinctively. Whereness is provided in the manner of waypoints through branches and concept points in the micro scale, in the macro consideration whole concept maps can represent a waypoint of navigation. A consistent representation for each concept's format or in the macro case a familiar structure to the map as a whole provides support for whatness. Whenness is derived as the data mining results become more relevant to the search in question and when the answer is reached or the end of the branches is reached then the final waypoint has been identified.

This relationship to our own internal mechanisms for cognitive mapping has various benefits:

- A perceived location within information allows newly acquired information to be learnt within a context enhancing its value;
- Visualising information to be arranged within a space facilitates the memorising and hence further understanding of the information away from the computer, this makes possible the recognition of relationships between new information encountered in the real world and that in the data store;
- Time spent searching for relevant information can be reduced.

4.5. Concept Mapping Procedure

The nature of the work is to provide a flexible product development environment that allows the users a flexible approach however there is still a course of action to be followed for the construction of a concept map.

The procedure for building a concept map is illustrated in Figure (4-4). A concept map is built in the following order:

1. Central node (defines the intended scope of map);
2. Main branches (define relevant sub topics);
3. Sub branches (either textual or graphical);
4. Additional details.

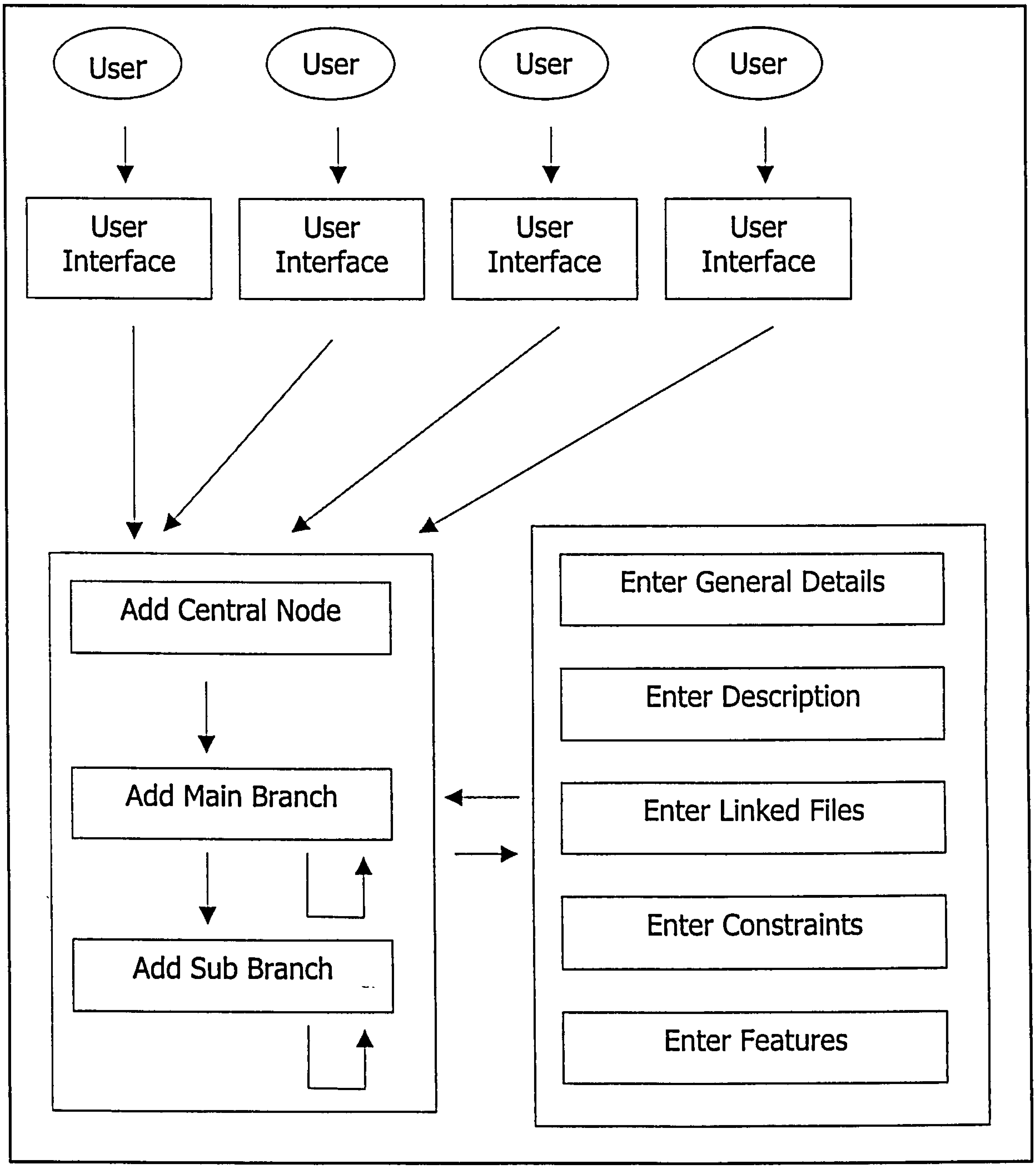


Figure (4-4) Concept Mapping Procedure

The central node of the map defines the topic and the scope that the concept map should encompass or relate to. A large central point such as this is a constant reminder to those working on the structure what the focus is. Each new area of discussion within this topic is built upon a main branch that constantly brings the users back to the main topic each time one is added rather than the end of the previous topic as in a linear representation. Sub branches are added to convey further details.

Each sub branch is also a placeholder for a variety of further information. A sub branch, via a dialogue provides; general details about the concept, an extra plain text description of its purpose, links to other locations (such as files, web pages), constraints in relation to other concepts, and if the concept is representing the actual features for an object relationships to other features can be added.

An example concept map is shown in Figure (4-5). The central node, main branches and numerous sub branches can easily be seen.

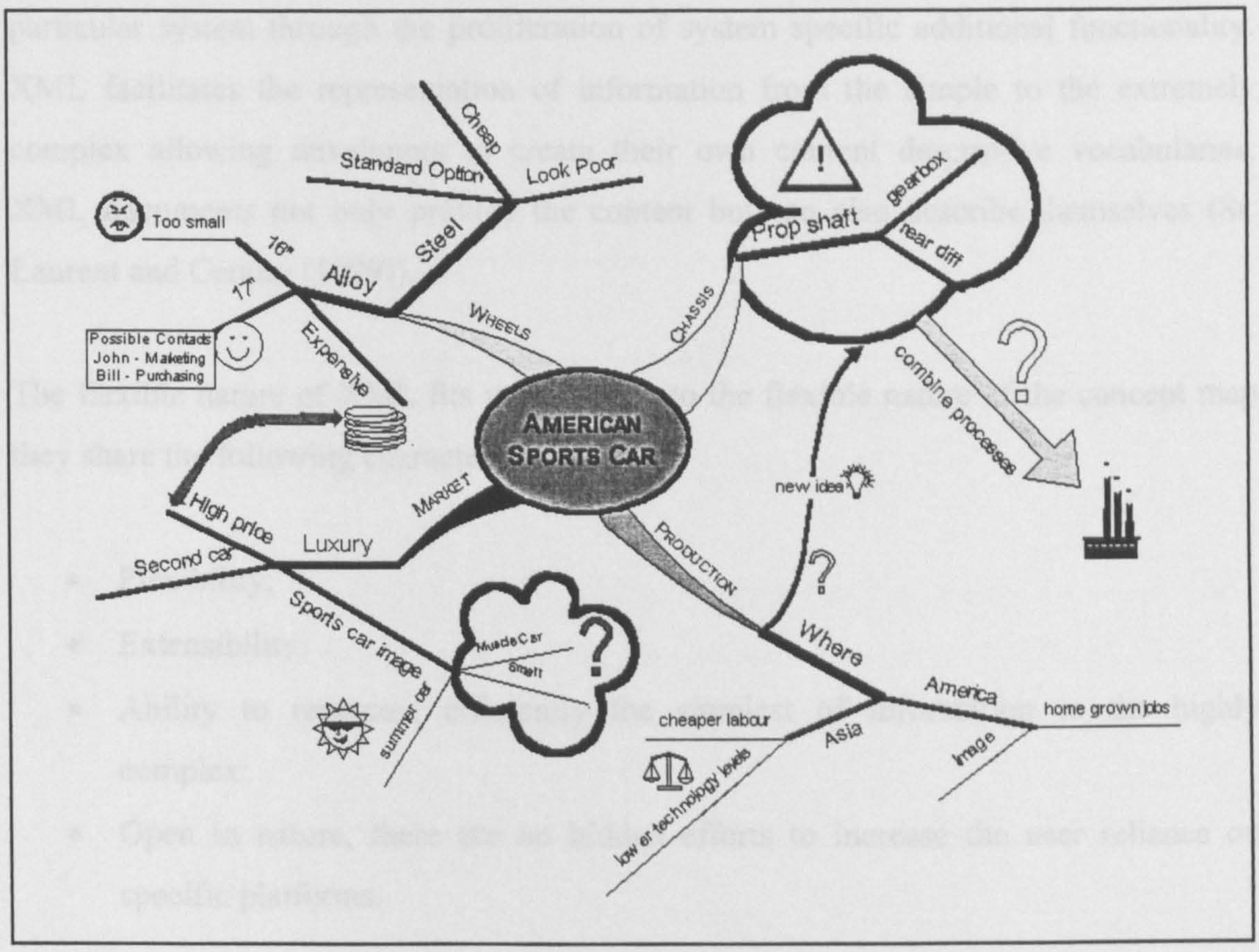


Figure (4-5) Example Concept Map

4.6. Flexible Technologies for Advanced Communication

A flexible collaboration tool necessitates the use of advanced technologies. In the domain of information representation there is a strong industrial influence behind the use of the XML (extensible markup language) and its application is suitable in this case.

XML can be thought of as another file format, however the current plethora of proprietary files formats lock people into specific applications and system, XML aims to liberate users from this situation. XML facilitates the storage of information in a hierarchical form, exactly the type of data concerned with concept maps. As opposed to other interchange formats such as Rich Text Format (RTF) and HTML, XML provides a much more consistent interface to access and modification.

Especially in the arena of web development there is much proprietary competition as different organisations try to gain the competitive edge by tying the users into a particular system through the proliferation of system specific additional functionality. XML facilitates the representation of information from the simple to the extremely complex allowing developers to create their own content descriptive vocabularies. XML documents not only provide the content but can also describe themselves (St. Laurent and Cerami (1999)).

The flexible nature of XML fits very closely to the flexible nature of the concept map they share the following characteristics:

- Flexibility;
- Extensibility;
- Ability to represent efficiently the simplest of information to the highly complex;
- Open in nature, there are no hidden efforts to increase the user reliance on specific platforms.

The characteristics of XML mirror those required for the representation of a concept map. XML is an ideal choice for the representation of concept maps and the main elements involved are shown in Figure (4-6).

The Java Virtual machine runs within the web browser running an applet that is the concept-mapping tool, this in turn contains an embedded Java based XML parser.

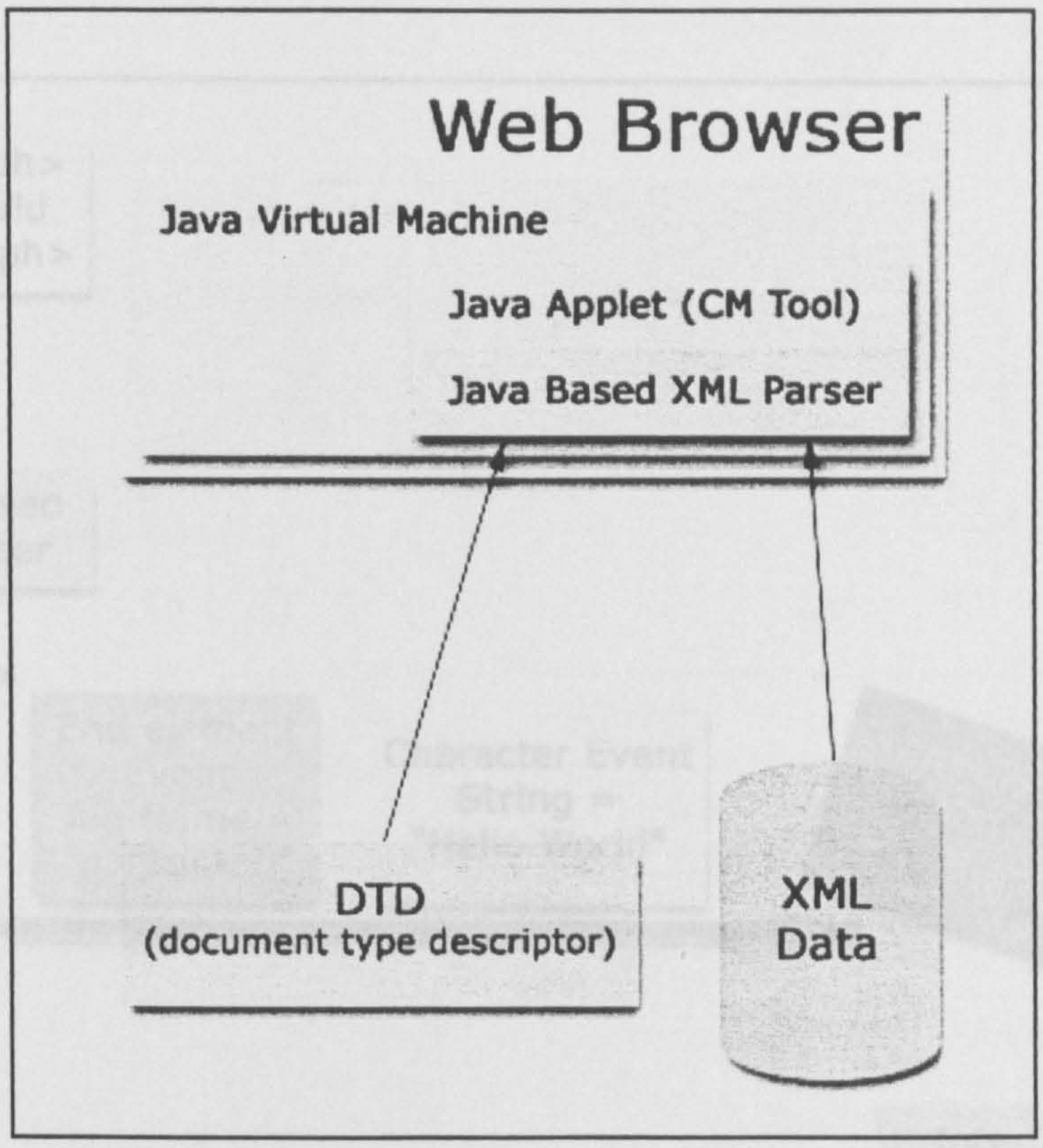


Figure (4-6) XML Parser Integration

The parser accepts the XML input and has the capability to treat this in two ways; event based or tree based. Event based processing generates three events for each XML tag the parser encounters, the start tag, the encapsulated data and the end tag.

Figure (4-7) shows how the XML is broken down into these events.

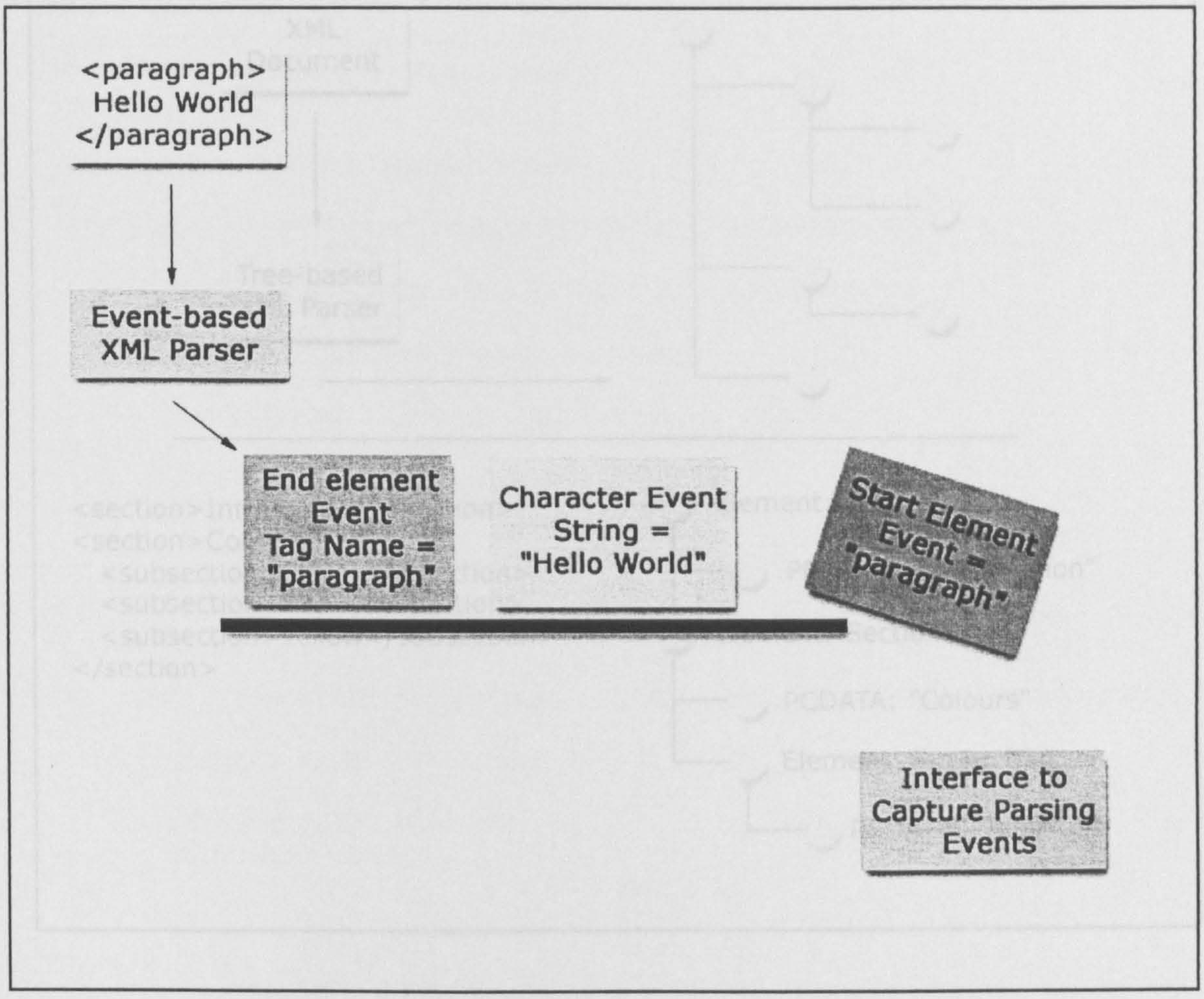


Figure (4-7) Event Based Parsing

Tree based parsing builds a hierarchical tree data structure from the XML tags as is shown in Figure (4-8). Each tag in tree form comprises of the tag type along with its data and any tags that it contains as leaf nodes of itself. Effectively allowing a large number of nested tags to form a tree structure.

The two different parser implementations have their own respective advantages and disadvantages. The event parsing is far better suited to linear processing of the XML tags, as would be used in batch processing systems where there is no need for a fine granularity of access required. For example processing a list of XML formatted emails to build a table of email addresses. Tree based parsing provides the developer with a more fine-grained environment where data at nodes can be read and modified without having to process the whole document each time.

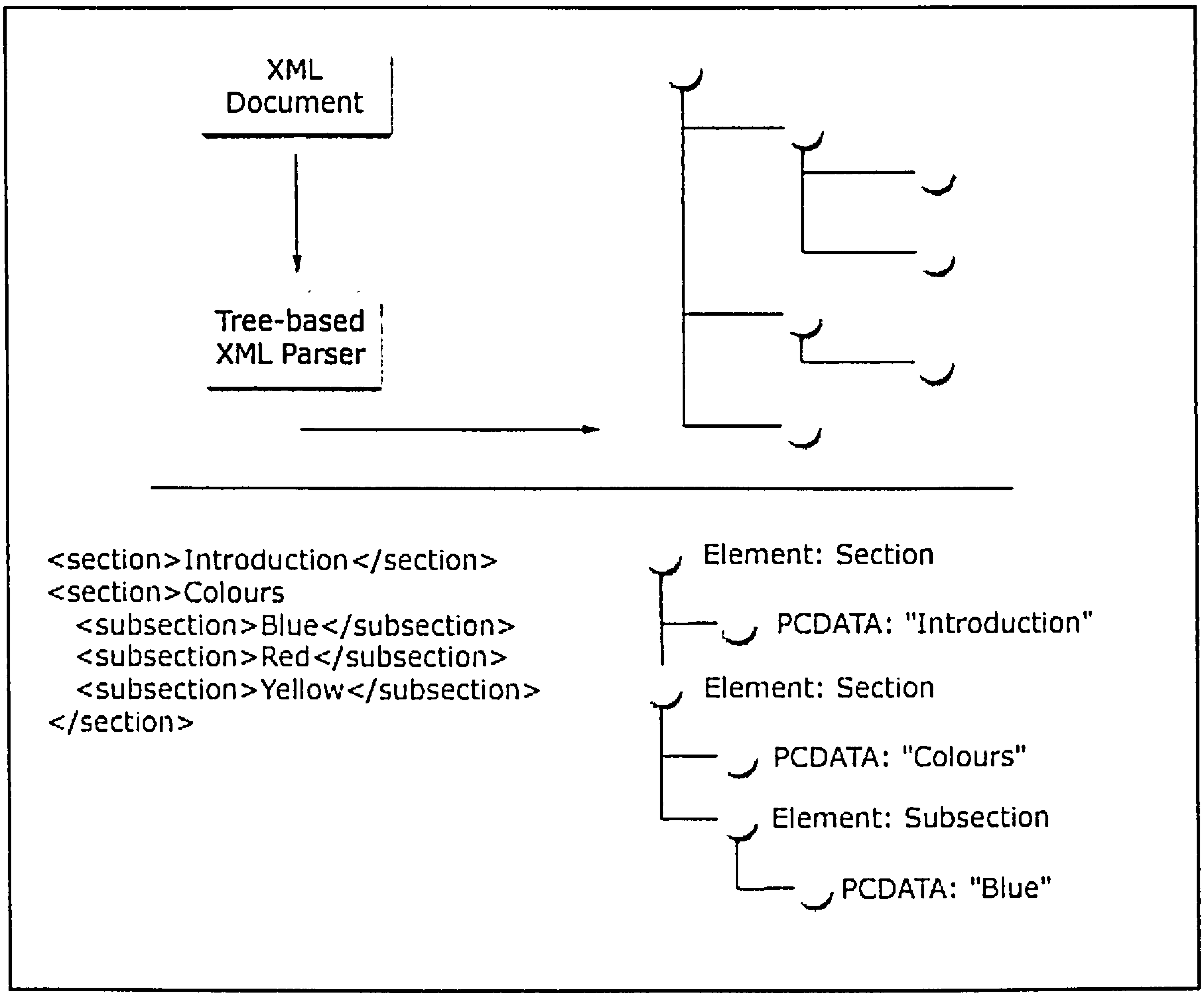


Figure (4-8) Tree Based Parsing

4.6.1. DTD

There is a need to define the rules for the XML that can be allowed to represent the concept maps. These rules are defined in the DTD (Document Type Definitions). These definitions state what the structure of the concept map is and what concepts are allowed to be formed in what order and where in the tree structure.

The following is the concept map DTD:

```
<!NOTATION GIF SYSTEM "image/gif">
<!NOTATION JPG SYSTEM "image/jpeg">
<!NOTATION BMP SYSTEM "image/bmp">
<!NOTATION CM   SYSTEM "text/xml">
<!NOTATION WHIP SYSTEM "model/vnd.dwf">
<!NOTATION VRML SYSTEM "model/vrml">
```

```
<!NOTATION STEP SYSTEM "text/plain">
<!NOTATION AVI  SYSTEM "video/avi">
<!NOTATION MPEG SYSTEM "video/mpeg">
<!NOTATION TEXT SYSTEM "text/plain">
<!NOTATION HTML SYSTEM "text/html">


<!ELEMENT CONCEPTMAP (MAINCONCEPT)>

  <!ELEMENT MAINCONCEPT (CONCEPTDETAILS, MAINBRANCH*)>

    <!ELEMENT MAINBRANCH (CONCEPTDETAILS, CONCEPT*)>

      <!ELEMENT CONCEPT (IMAGE? | BRANCH?)>

          <!ELEMENT IMAGE (CONCEPTDETAILS, CONCEPT*)>
            <!ATTLIST IMAGE
                    SOURCE CDATA #REQUIRED
                    TYPE NOTATION (GIF | JPG | BMP) #REQUIRED>

          <!ELEMENT BRANCH (CONCEPTDETAILS, CONCEPT*)>


  <!ELEMENT CONCEPTDETAILS (NAME, CREATOR, CONCEPTDATE, CONCEPTLINKS)>

      <!ELEMENT NAME (#PCDATA)>

      <!ELEMENT CREATOR EMPTY>
        <!ATTLIST CREATOR
                NAME         CDATA #REQUIRED
                EXPERTISE    CDATA #IMPLIED
                POSITION     CDATA #REQUIRED
                ORGANISATION CDATA #REQUIRED
                EMAIL        CDATA #IMPLIED
                PHONENO      CDATA #IMPLIED>

      <!ELEMENT CONCEPTDATE (#PCDATA)>

      <!ELEMENT CONCEPTLINKS (LINK*)>

        <!ELEMENT LINK EMPTY>
          <!ATTLIST LINK
                  LNAME CDATA #REQUIRED
                  LPATH CDATA #REQUIRED
                  TYPE NOTATION (CM | GIF | JPG | BMP | WHIP | VRML |
STEP | AVI | MPEG | TEXT | HTML) #REQUIRED>
```

The notation definitions are defining a series of MIME type for files that could be linked to individual concepts as is apparent later in the DTD. This includes images in the form of JPEGs, GIFs, and BMPs, plain text, html, and video formats.

The root tag is that of the CONCEPTMAP, this tag contains the rest of the tags that make up a concept map. It defines that itself containing a single tag in turn defined as MAINCONCEPT.

The main and central concept is then in turn defined itself; this contains the details about the concept itself and allows zero or more MAINBRANCH concepts to be represented within it.

The main branch concepts are then defined also containing the details about it represented by CONCEPTDETAILS and permits zero or more concepts to be bound within it.

The final node in the chain of concepts is the CONCEPT definition itself, this is allowed to be either an IMAGE concept or a BRANCH concept but not both and it must be at least one or the other.

The two types of CONCEPT are then defined, IMAGE and BRANCH. The IMAGE concept contains the concepts details and zero or more other concepts. This definition includes the attributes that must be provided; in this case for an image the source of the image and its type as define in the initial notation statements. The BRANCH concept is defined as providing its own details and also allows zero or more other concepts.

The core tag to this DTD is the CONCEPT DETAILS. This in turn consists of four other tags that must present, NAME, CREATOR, CONCEPTDATE, and CONCEPTLINKS.

The NAME tag purely consists of parsed character data.

The CREATOR element is empty and therefore has no sub elements and no content however it is defined by its attributes, name, expertise, position, organisation, email, and phone number. Each of these either must be present or defined as optional via the #REQUIRED and #IMPLIED statement respectively.

The CONCEPT DATE tag also just defines itself as parsed character data.

The CONCEPTLINKS tag is the hub of the linking mechanism in the concept-based system. It defines a set of zero or more links. A LINK tag is defined itself as having no sub element and no content however it provides a series of attributes that define a name for the link, a path that the link follows in the form of a URL (Uniform Resource Locator) and the type of data that the link will lead to. These attributes are all required should a link be specified.

4.6.2. XML

The XML that conforms to the previous DTD can be found in the following example. It should be self explanatory a brief background in the style of HTML and the previous DTD should provide the information necessary for its comprehension. XML is intended to be both human and machine-readable.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE CONCEPTMAP SYSTEM "conceptmap.dtd">
<CONCEPTMAP>

  <MAINCONCEPT>

    <CONCEPTDETAILS>
      <NAME>American Sports Car</NAME>
      <CREATOR NAME="Neil Anderson"
               POSITION="Software Engineer"
               ORGANISATION="DMU"></CREATOR>
      <CONCEPTDATE>12/1/2001</CONCEPTDATE>
      <CONCEPTLINKS></CONCEPTLINKS>
    </CONCEPTDETAILS>


    <MAINBRANCH>

      <CONCEPTDETAILS>
        <NAME>Wheels</NAME>
        <CREATOR NAME="Neil Anderson"
                 POSITION="Software Engineer"
                 ORGANISATION="DMU"></CREATOR>
        <CONCEPTDATE>12/1/2001</CONCEPTDATE>
        <CONCEPTLINKS>
        </CONCEPTLINKS>
      </CONCEPTDETAILS>

      <CONCEPT>
        <BRANCH>
          <CONCEPTDETAILS>
```

```
            <NAME>Alloy</NAME>
            <CREATOR NAME="Neil Anderson"
                     POSITION="Software Engineer"
                     ORGANISATION="DMU"></CREATOR>
            <CONCEPTDATE>12/1/2001</CONCEPTDATE>
            <CONCEPTLINKS></CONCEPTLINKS>
          </CONCEPTDETAILS>

          <CONCEPT>
            <BRANCH>
              <CONCEPTDETAILS>
                <NAME>17"</NAME>
                <CREATOR NAME="Neil Anderson"
                         POSITION="Software Engineer"
                         ORGANISATION="DMU"></CREATOR>
                <CONCEPTDATE>12/1/2001</CONCEPTDATE>
                <CONCEPTLINKS></CONCEPTLINKS>
              </CONCEPTDETAILS>

              <CONCEPT>
                <IMAGE SOURCE="smile.jpg"
                       TYPE="JPG">
                  <CONCEPTDETAILS>
                    <NAME>Alloy</NAME>
                    <CREATOR NAME="Neil Anderson"
                             POSITION="Software Engineer"
                             ORGANISATION="DMU"></CREATOR>
                    <CONCEPTDATE>12/1/2001</CONCEPTDATE>
                    <CONCEPTLINKS>
                      <LINK LNAME="Alloy Composition Report"
                            LPATH="acr.txt"
                            TYPE="TEXT"></LINK>
                      <LINK LNAME="Home page for Concurrent
Engineering at DMU"

LPATH="http://www.eng.dmu.ac.uk/concurrent"
                            TYPE="HTML"></LINK>
                    </CONCEPTLINKS>
                  </CONCEPTDETAILS>
                </IMAGE>
              </CONCEPT>
            </BRANCH>
          </CONCEPT>
        </BRANCH>
      </CONCEPT>

    </MAINBRANCH>


  </MAINCONCEPT>

</CONCEPTMAP>
```

It defines a root node, the central concept called "American Sports Car" with an associated creator "Neil Anderson". This central node for the sake of conciseness contains a single main branch called "Wheels" this in turn defines a concept of type branch called "Alloy", finally an image concept called "17" that provides the source for its representing image and two links, a link to a report and a web page. The document is then closed with the corresponding end tags.

4.7. Summary

The concept mapping approach has been presented in this chapter; its origins are in cognitive psychology and its application. It has shown how the concept mapping process is inherently parallel to our own internal cognition systems and how this can be utilised to provide an enhanced collaborative framework. This chapter has explained:

- The necessary elements for global distributed information sharing.
- Concept maps and their role throughout the entire product development lifecycle with the emphasis on the involvement of the whole development team while shifting domain focus at each phase.
- Considerations for the expansion of the development team at an early stage in the product development lifecycle.
- Psychological and neurological explanations for the benefits of using the information mediums developed.
- The flexible nature of the information environment created to easily allow other activities without extra processing such as presentations.
- How the different levels of information complexity can be represented in one cohesive data model, facilitating future data mining with ease.
- The four essential tenets for a collaborative framework.
- Navigation within the framework.
- The procedure involved in creating such information repositories.
- The use of flexible technologies to complement and facilitate the flexible nature of the information framework.
- Explanations of both XML and its use in this context alongside the concept mapping DTD.

Chapter 5

CHAPTER 5. Distributed Collaborative Product Development

5.1. Overview

The chapter details a model for collaborative product development. The proposed system comprises of, data management and standardisation, conflict resolution, an object oriented database, and a scalable HTTP interface. The research work focuses on the integration of people, processes and product development phases. Collaborative product development necessitates team input from the earliest possible phases as addressed by the concept mapping in Chapter 4. However it is the extension of the concept mapping into a fully holistic system that this chapter discusses. Various components of the system are described and demonstrated within the collaborative development context of a new automobile. This situation demonstrates the geographic and temporal barriers facing global collaboration and shows how the proposed system addresses them.

5.2. Introduction

The philosophy of Concurrent Engineering does not dictate a specific methodology or technology but puts forward the perspective that as many potential factors of product development should be available for analysis in the design phase. This has been addressed partly through the concept mapping however to lever the benefits the collaboration has to extend beyond concepts and basic information sharing to detailed design and management.

Throughout the product development lifecycle the depth and diversity of the information in a collaborative effort continually expands. It is necessary to propagate the results of design decisions from early design stages effectively throughout the product development lifecycle. There is a necessity for the integration of the data generated by collaboration not only within each phase but also between development phases facilitating their transition. An example model demonstrating this centred on the concept maps is shown in Figure (5-1).

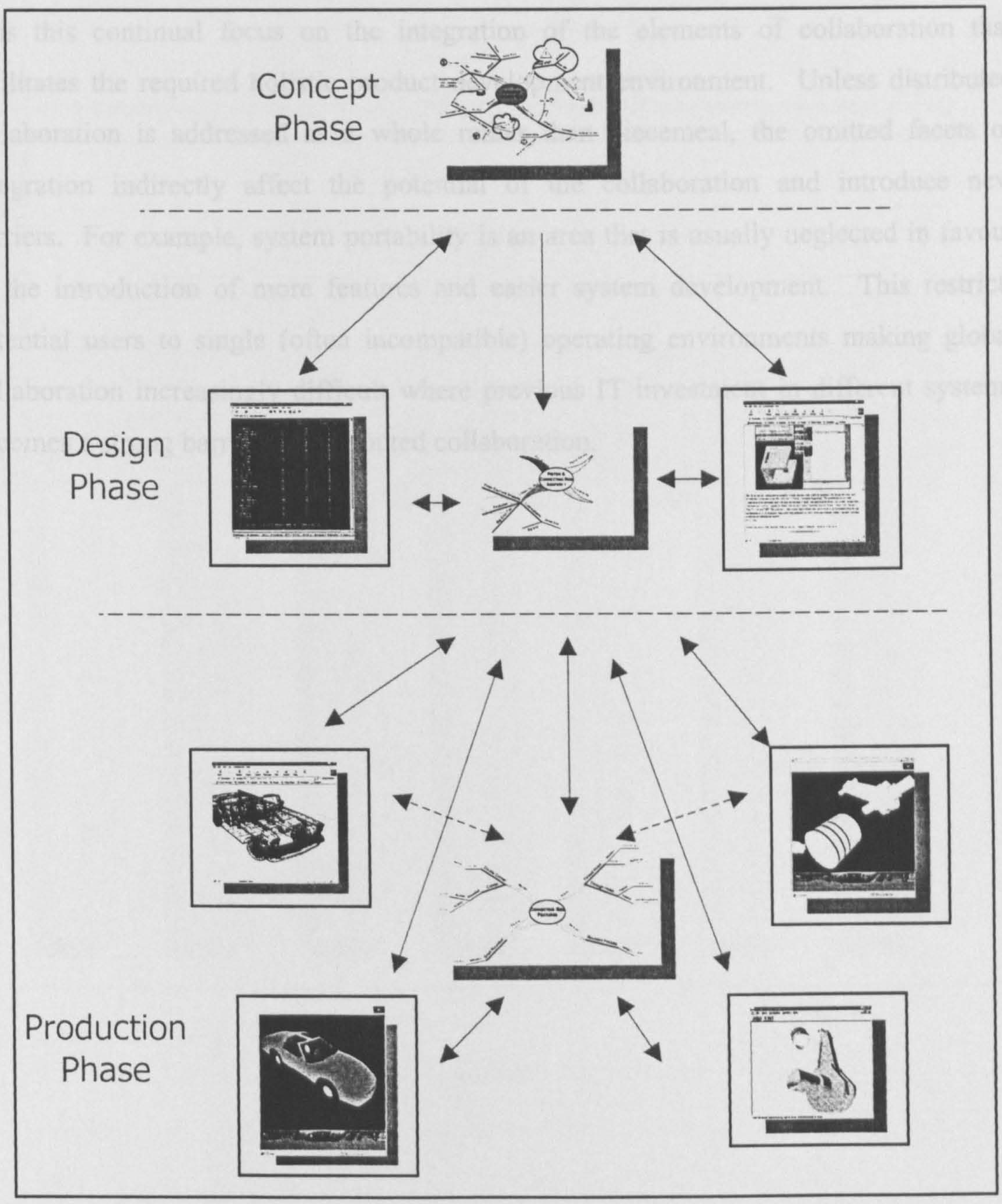


Figure (5-1) Collaboration Relationships

A conceptual representation addresses the needs at the beginning of product development however this representation needs to maintain continuity through subsequent development phases. In the collaboration pyramid the concept maps are the hubs for other information representations. They provide a framework not only for a single development phase but also facilitate the building of continuity between phases.

It is this continual focus on the integration of the elements of collaboration that facilitates the required holistic product development environment. Unless distributed collaboration is addressed as a whole rather than piecemeal, the omitted facets of integration indirectly affect the potential of the collaboration and introduce new barriers. For example, system portability is an area that is usually neglected in favour of the introduction of more features and easier system development. This restricts potential users to single (often incompatible) operating environments making global collaboration increasingly difficult where previous IT investment in different systems becomes a strong barrier to distributed collaboration.

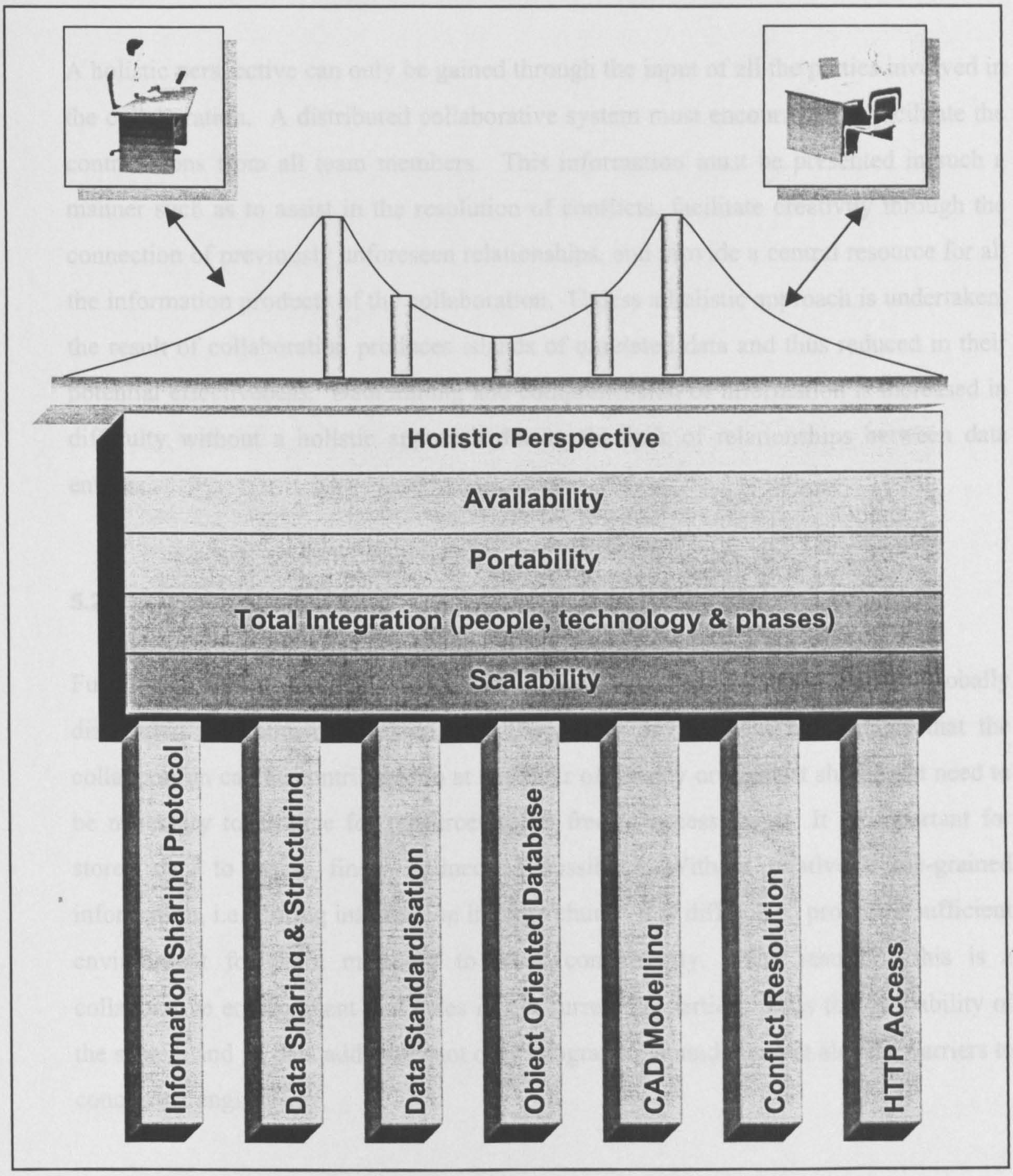


Figure (5-2) Collaboration Bridge

The collaboration bridge in Figure (5-2) represents the horizontal bridging tenets for a system necessary to maximise the collaboration between organisations. The supporting pillars of the bridge are effectively the sub sections of the collaborative system developed to address these issues.

5.2.1. Holistic Perspective

A holistic perspective can only be gained through the input of all the parties involved in the collaboration. A distributed collaborative system must encourage and facilitate the contributions from all team members. This information must be presented in such a manner such as to assist in the resolution of conflicts, facilitate creativity through the connection of previously unforeseen relationships, and provide a central resource for all the information products of the collaboration. Unless a holistic approach is undertaken, the result of collaboration produces islands of unrelated data and thus reduced in their potential effectiveness. Data mining and comprehension of information is increased in difficulty without a holistic approach due to the lack of relationships between data entities.

5.2.2. Availability

Full time access to retrieve or contribute to a collaboration database within a globally distributed environment is essential. The global environment necessitates that the collaboration can be contributed to at any hour of the day or night, it should not need to be necessary to arrange for resources to be free to access them. It is important for stored data to be as finely grained as possible. Without relatively fine-grained information, i.e. storing information in large chunks it is difficult to provide a sufficient environment for team members to work concurrently. The result of this is a collaborative environment that loses its concurrent properties. Thus the availability of the system and its data addresses not only geographic boundaries but also the barriers to concurrent engineering.

5.2.3. Portability

Portability in the context of the general computing environment is seen as a beneficial but not essential requirement, however in the domain of collaborative product development, the portability of a supporting system is of a far greater importance. Supply chain integration necessitates the ease of compatibility between computer systems. Increasingly supplier contracts are put out to tender, if the requesting organisation or the supplier necessitates a requirement for a single operating environment to be able to allow the collaboration to continue, this will introduce barriers between potential candidate suppliers.

Portability in the collaboration domain is essential in the modern computing era; there is an ever-increasing diversity among platforms despite efforts for standardisation and consistency. A platform in the context of this research corresponds to the operating system and the web browser of the client. Many systems are developed on a proprietary platform; clients are specially designed to run under Windows for example. This often provides faster and easier software development, greater possible features and reduces the complexity of the system using well-defined core libraries with many years of testing behind them. Single platform systems present new barriers to collaboration; each team member must be supplied with a suitable computing environment to be able to provide their contribution. This is usually a prohibitive task, purchasing and configuring new environments is costly both in time and money.

The end results of such problems are that the contributions from non-technologically capable partners are submitted in other non-integrated manners or worse still are omitted altogether from the collaborative process.

5.2.4. Integration

The core purpose of a collaborative system is to reduce or remove the barriers between the people processes and the technology deployed; effectively the purpose of the integration of these factors is to provide emergent benefits greater than the sum of the parts. Full integration is the key to collaboration efforts for the reduction of costly product development barriers. Strong integration brings with it benefits in terms of relationships between previously disparate data and more powerful data management. However a balance needs to be struck between this support for breaking down the barriers between sets of data and the prevention of the development of a system so tightly integrated that it in turn through lack of scalability then disregards the future accommodation of technological changes.

5.2.5. Scalability

With the constant advance of product development and information technologies a collaborative environment designed and developed for today will provide short lived benefits unless it is based upon scalable technologies that can further the existing integration with as yet unknown data representations. The developed research in this thesis utilises an open system approach whose data format in this case the DTD is not hidden or buried in inaccessible complexity, it is represented in the format of an internationally recognised data standard. The system also facilitates scalability through the provision of direct access to external data representations through utilising HTTP linking. A scalable collaborative environment allows the collaborating organisations the flexibility to produce structures that provide greater relevance to their industrial domain without constricting them unnecessarily. Scalability also involves the created datasets independence from the system that created it. The framework as described in Chapter 4 is based on XML that allows the data to be processed independently of the system. It is only through the advent of such information standards like XML that tight system integration alongside data independence is achievable.

These proposed collaboration tenets are supported through the developments of this research; data sharing and structuring, data standardisation, conflict resolution, an object-oriented database, integrated CAD solid modelling and HTTP accessed documents.

These supporting systems are related in context in the following sections.

5.3. Information Sharing Protocol

Central to a collaborative system is the underlying information protocol that provides the data transfers and management of the information flow. This usually is a factor that restricts the potential of many of the facets of a collaborative system. Providing a collaborative environment based around the multiple clients and server architecture generally restricts system development to a single language or operating environment.

However with the advent of web-based technologies and the increased portability of the clients there is now the potential for only the server side software to be restricted. This restriction can also potentially be reduced through the use of the portable Java run time environment.

5.3.1. CORBA

A java-to-java solution has been the focus of this research with no specific legacy system to consider the development effort for CORBA is unwarranted. However in the interests of supporting long term scalability the potential for CORBA was investigated.

CORBA (Common Object Request Broker Architecture) provides a means with which to connect services together through providing each component with an interface defined by CORBA's IDL (Interface Definition Language). This would be the suitable approach for systems with substantial investment in legacy systems. Each existing component is wrapped in an interface to allow other objects to call the functionality within it. Within a single process the IDL is compiled into client stubs and object skeletons. The client invokes an operation on the IDL stub that acts as a proxy. Passing through the stub on the client side, the invocation continues through the ORB (Object Request Broker) and the skeleton on the implementation side to get to the object where it is executed.

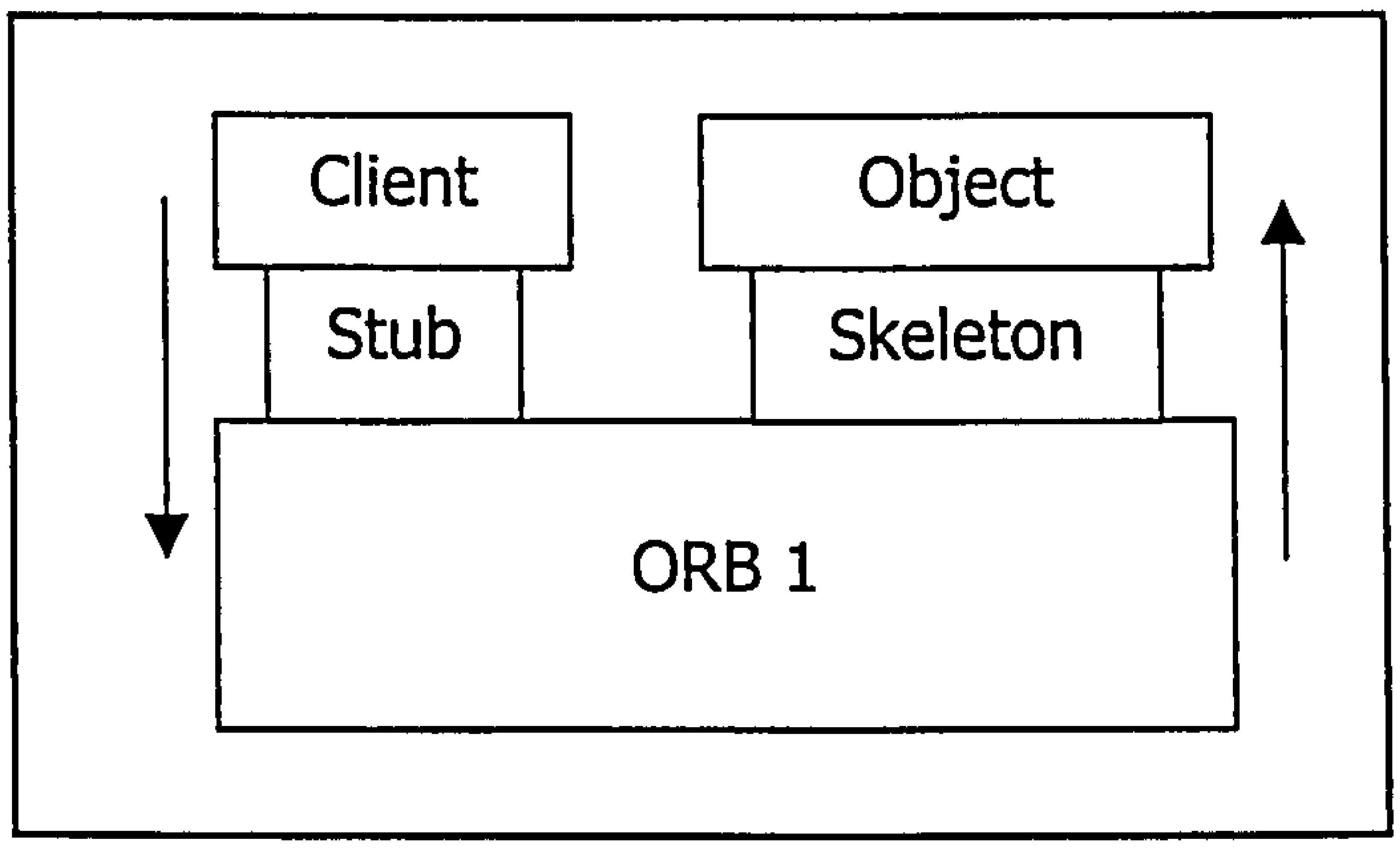


Figure (5-3) Request from Client to Object Implementation (Group (2002))

To access remote objects the client first obtains its object reference, there are a variety of mechanisms for this. When the local ORB receives the request for the object and discovers that the target is remote the invocation is routed over the network to the remote object ORB that in turn handles the request.

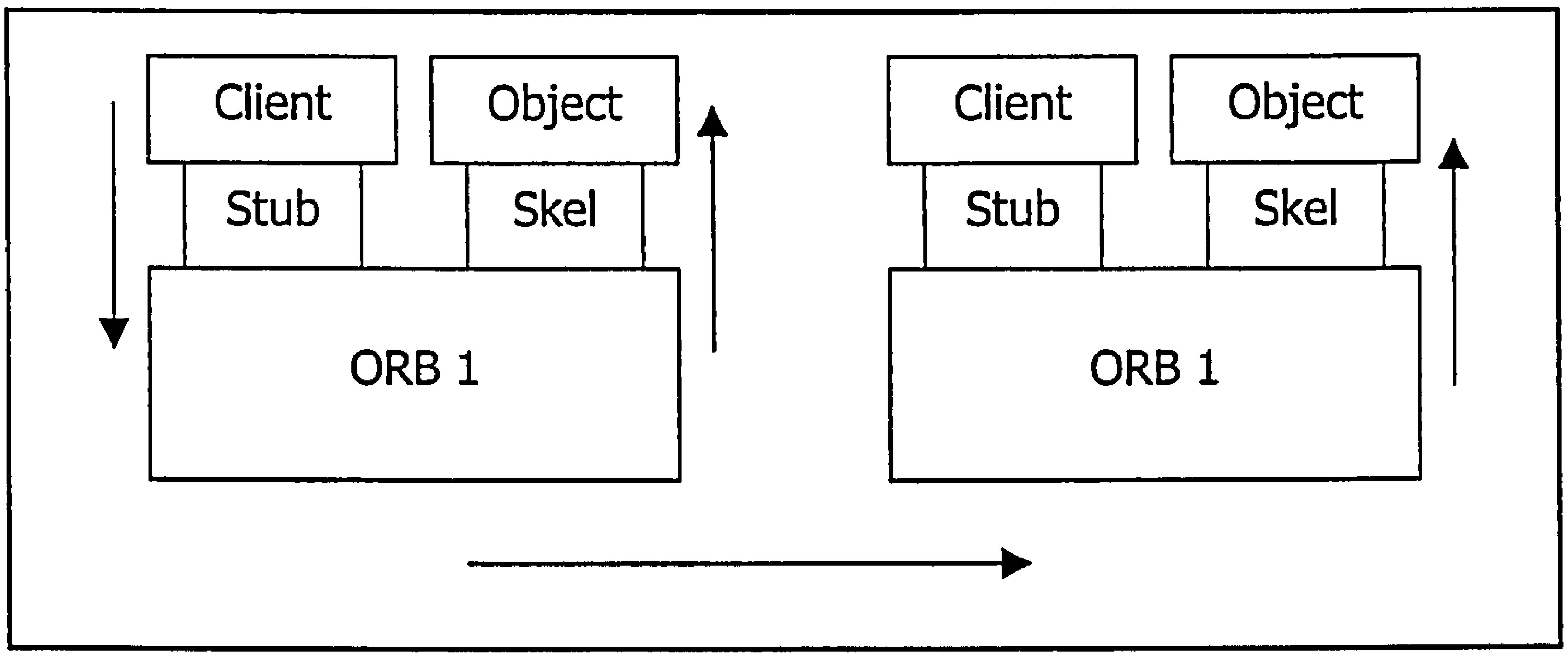


Figure (5-4) ORB to ORB Communication (Group (2002))

The development effort for the implementation of CORBA is substantial but much less than the redevelopment and complete porting of existing software to a common language and environment. However for a Greenfield software development the provision for remote object method invocations can be achieved at a smaller development cost with Java's RMI networking interface.

5.3.2. RMI (Remote Method Invocation)

An RMI application is usually comprised of at least two programs, a client and a server. A server application creates the remote objects and generates references for them. These references are available from potential clients to locate and invoke methods upon the objects that they denote. Java's RMI provides the mechanism that enables the client and the server to pass data to each other.

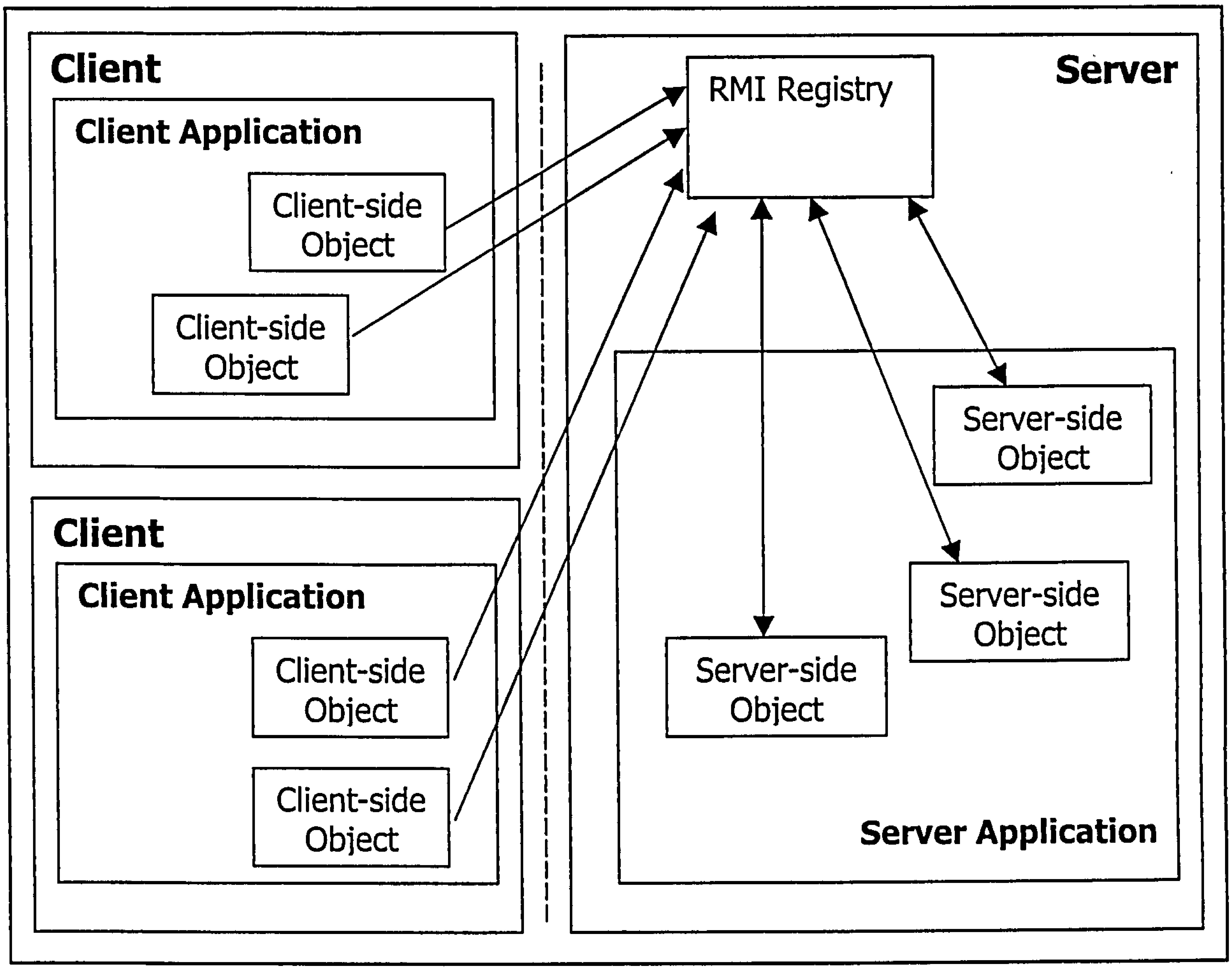


Figure (5-5) Client Server RMI Based Object Invocations

There are two mechanisms available within RMI too obtain references to remote objects. Applications can pass and return remote object references as part of its operation or the RMI registry can be utilised. A combination of methods has been used in this research. The RMI registry provides a centralised location for applications to register remote objects. In this case the server creates objects and registers them with the RMI registry that then binds a name to the object. The client then connects to the server and looks up the name in the server's registry and invokes methods upon it.

This method for client server communications allows many clients to connect to a single server application. However in this research it is necessary for bi-directional communications to take place. A server also needs to be able to communicate to a client. To accomplish this a reference to the client object is passed when the client invokes a method on the server, this is then stored providing the server with a means to in turn invoke methods on the client.

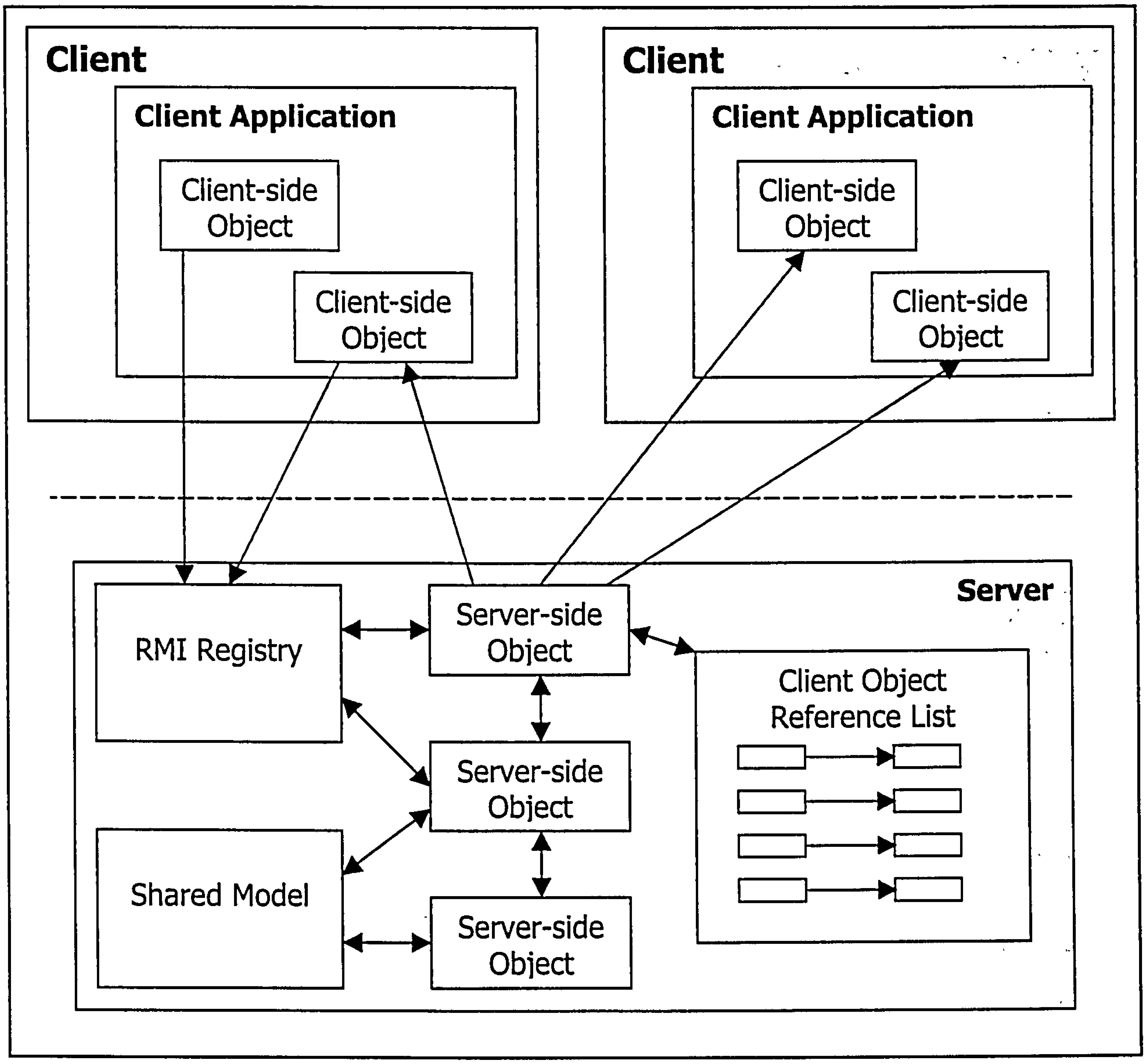


Figure (5-6) Server Maintained Client Object Reference List and RMI Model

Figure (5-6) combines the two RMI approaches, the client with the facility to access the server objects through the RMI Registry and conversely a server maintained object reference list to the client objects. This configuration allows an effective implementation of the MVC (Model View Controller) design pattern. The server maintains a shared model that is updated by server objects with methods called via RMI from a client. When a change to the model is made the server iterates through its maintained client object reference list and in turn invokes methods on the client to reflect the modifications on the multiple views.

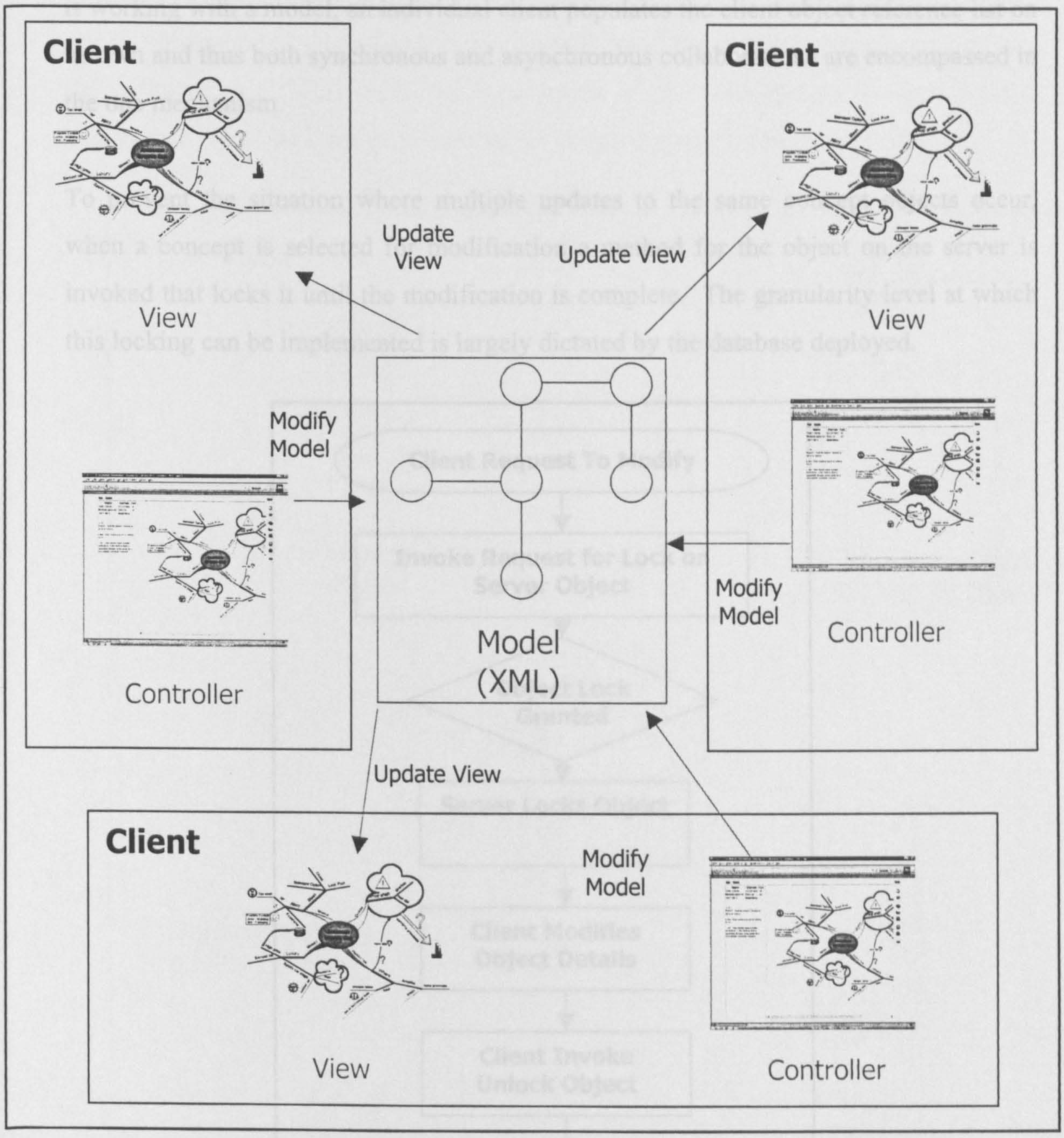


Figure (5-7) MVC (Model View Control Architecture)

This bi-directional RMI mechanism effectively implements a broadcast system for the synchronous update of clients. Concerning the event that multiple users are viewing the same concept map (considered as the model in Figure (5-7)) the server invokes methods on those clients to update the changes to them. Considering the case when a single user is working with a model, an individual client populates the client object reference list on its own and thus both synchronous and asynchronous collaborations are encompassed in the one mechanism.

To prevent the situation where multiple updates to the same concept objects occur, when a concept is selected for modification a method for the object on the server is invoked that locks it until the modification is complete. The granularity level at which this locking can be implemented is largely dictated by the database deployed.

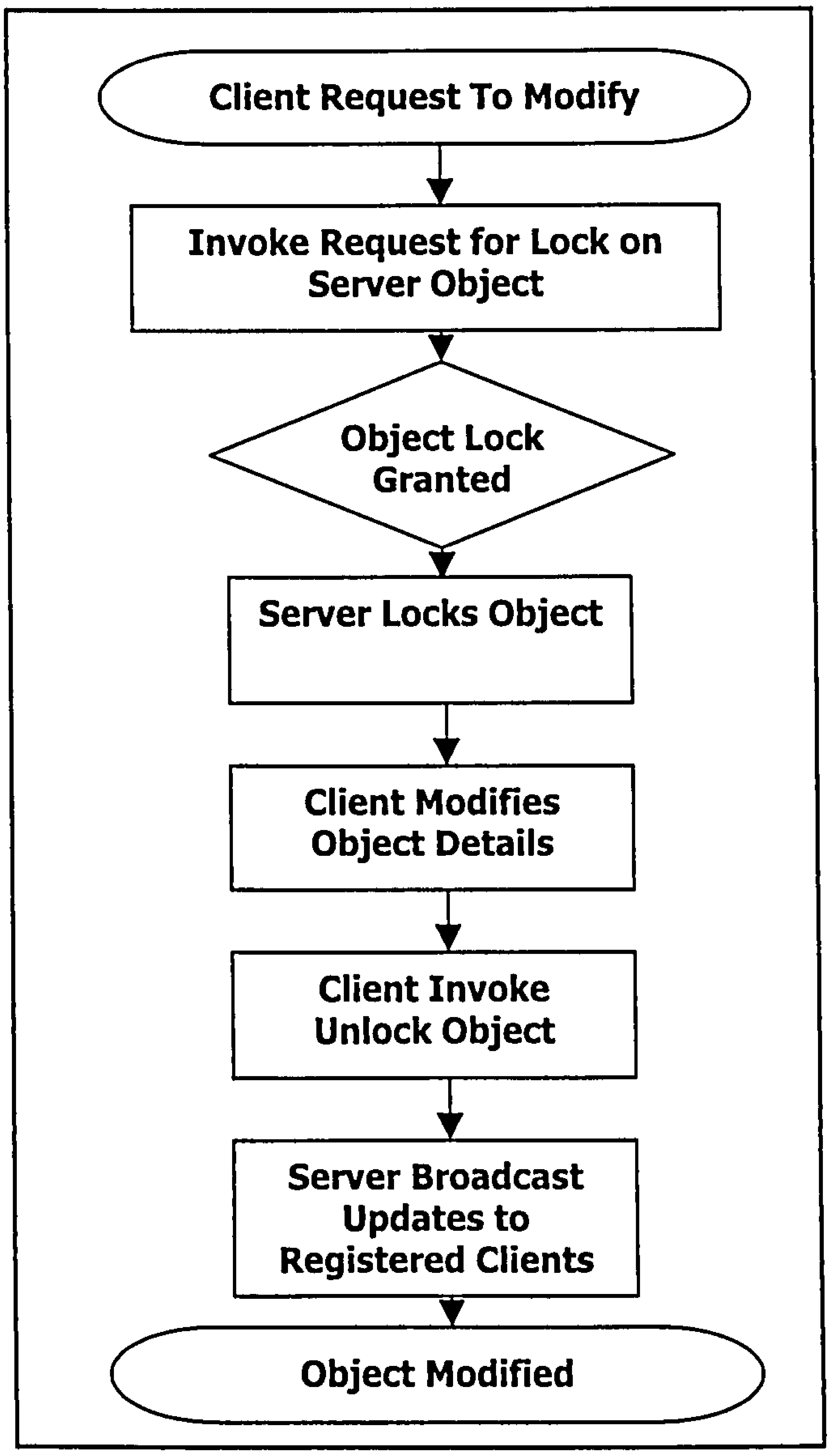


Figure (5-8) Client to Server Object Locking for Modification

5.4. Data Sharing & Structuring

Supply chain integration requires the effective structuring and sharing of product development data; this is essential to the facilitation of successful collaborations. In the context of the multinational automobile collaboration the information is structured in a top down manner. This facilitates departments of one development location (UK) to collaborate yet at the same time involve the design department from the (USA).

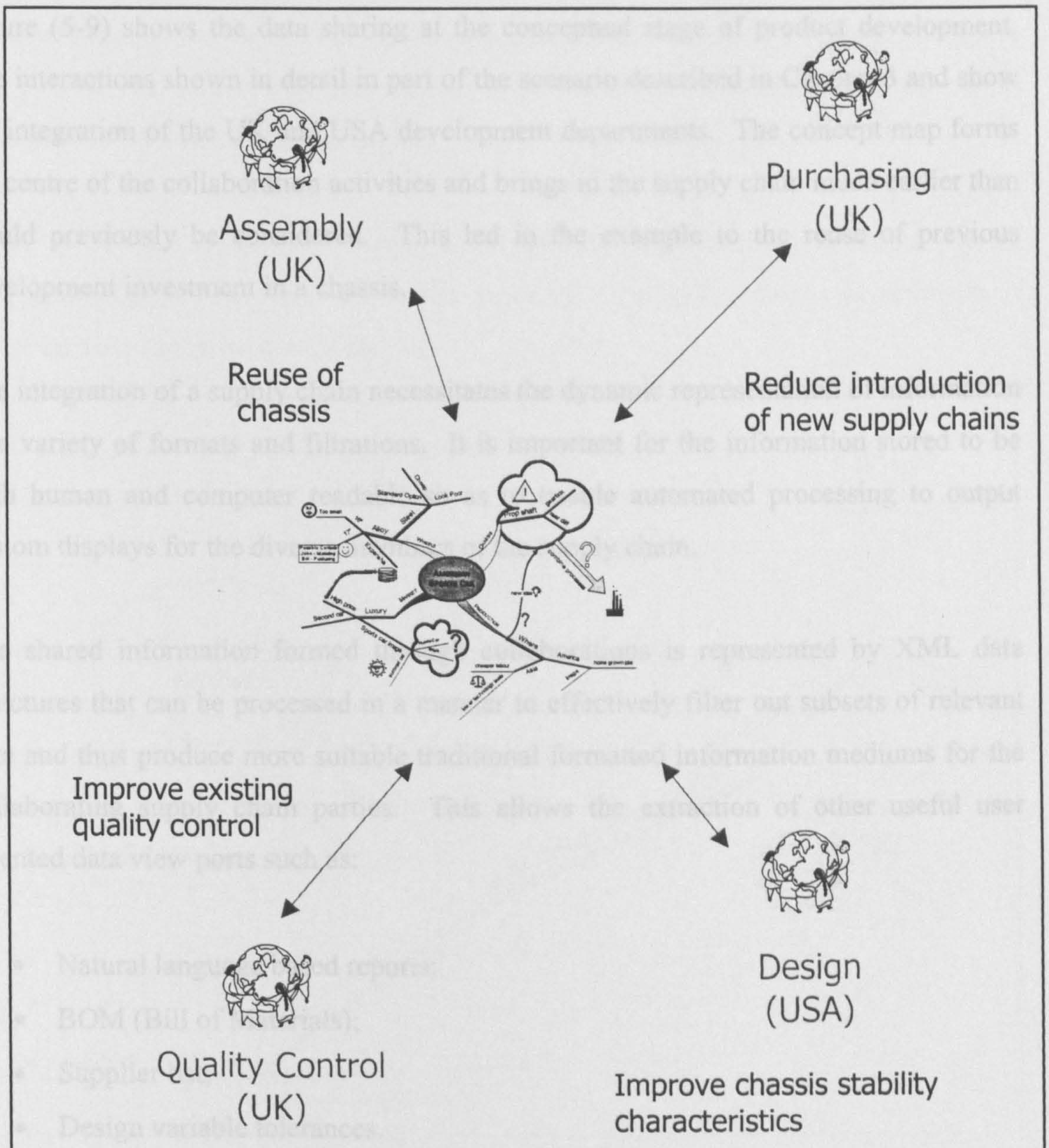


Figure (5-9) Conceptual Stage Context

The process of concept mapping requires suitable structuring of the data in order to facilitate the manipulation and granularity of access required. As has been covered in Chapter 4 the concept maps themselves are represented in XML with their corresponding DTD. The concept maps themselves can be used, as has been detailed for alternative uses such as presentations however the underlying format can extend much further than that.

Figure (5-9) shows the data sharing at the conceptual stage of product development. The interactions shown in detail in part of the scenario described in Chapter 3 and show the integration of the UK and USA development departments. The concept map forms the centre of the collaboration activities and brings in the supply chain much earlier than would previously be considered. This led in the example to the reuse of previous development investment in a chassis.

The integration of a supply chain necessitates the dynamic representation of information in a variety of formats and filtrations. It is important for the information stored to be both human and computer readable so as to enable automated processing to output custom displays for the diverse members of the supply chain.

The shared information formed through collaborations is represented by XML data structures that can be processed in a manner to effectively filter out subsets of relevant data and thus produce more suitable traditional formatted information mediums for the collaborating supply chain parties. This allows the extraction of other useful user oriented data view ports such as:

- Natural language based reports;
- BOM (Bill of Materials);
- Supplier list;
- Design variable tolerances.

5.4.1. Natural Language Reports

A natural language report can be generated like a table of contents based on the branches of the concept maps that have text files linked to the concepts. The result of this processing would be a document framework for milestone / progress reports or even user guides or maintenance handbooks to be created. A complex language processor is required with a real knowledge for the semantics of a language to produce a more comprehensive document. However in the same manner that a software case tool can process UML (Unified Modelling Language) to provide the programmer with function headers and the outline of the code, exporting to natural language document can provide an author with the outline of a document. Reducing the time from collaboration to written documentation.

Figure (5-10) this report for example could take the form of an early industry press release, the outline of the situation can be exported, reducing the time an expert needs to spend on assisting more menial tasks.

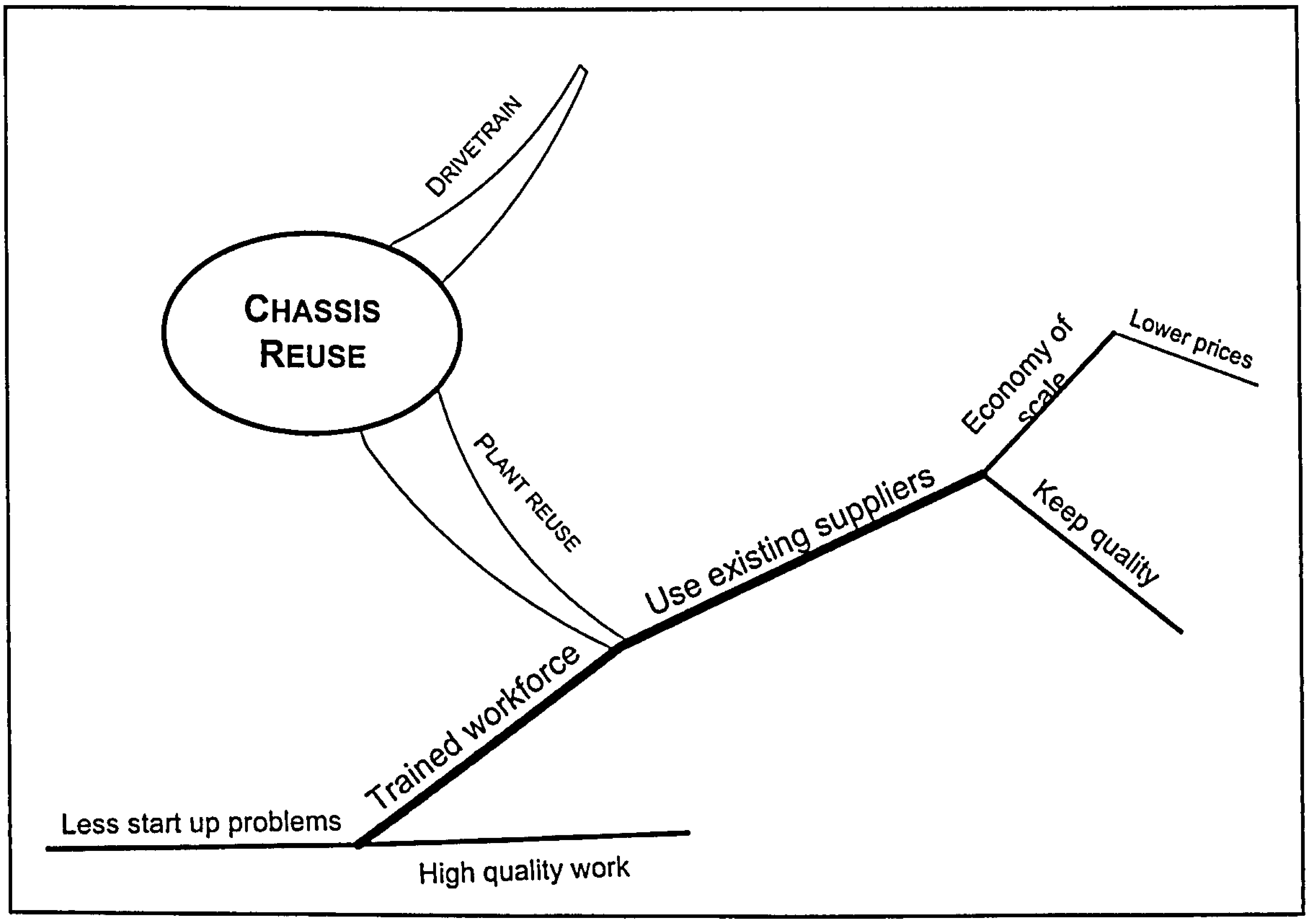


Figure (5-10) Partial Concept Map – Chassis Reuse

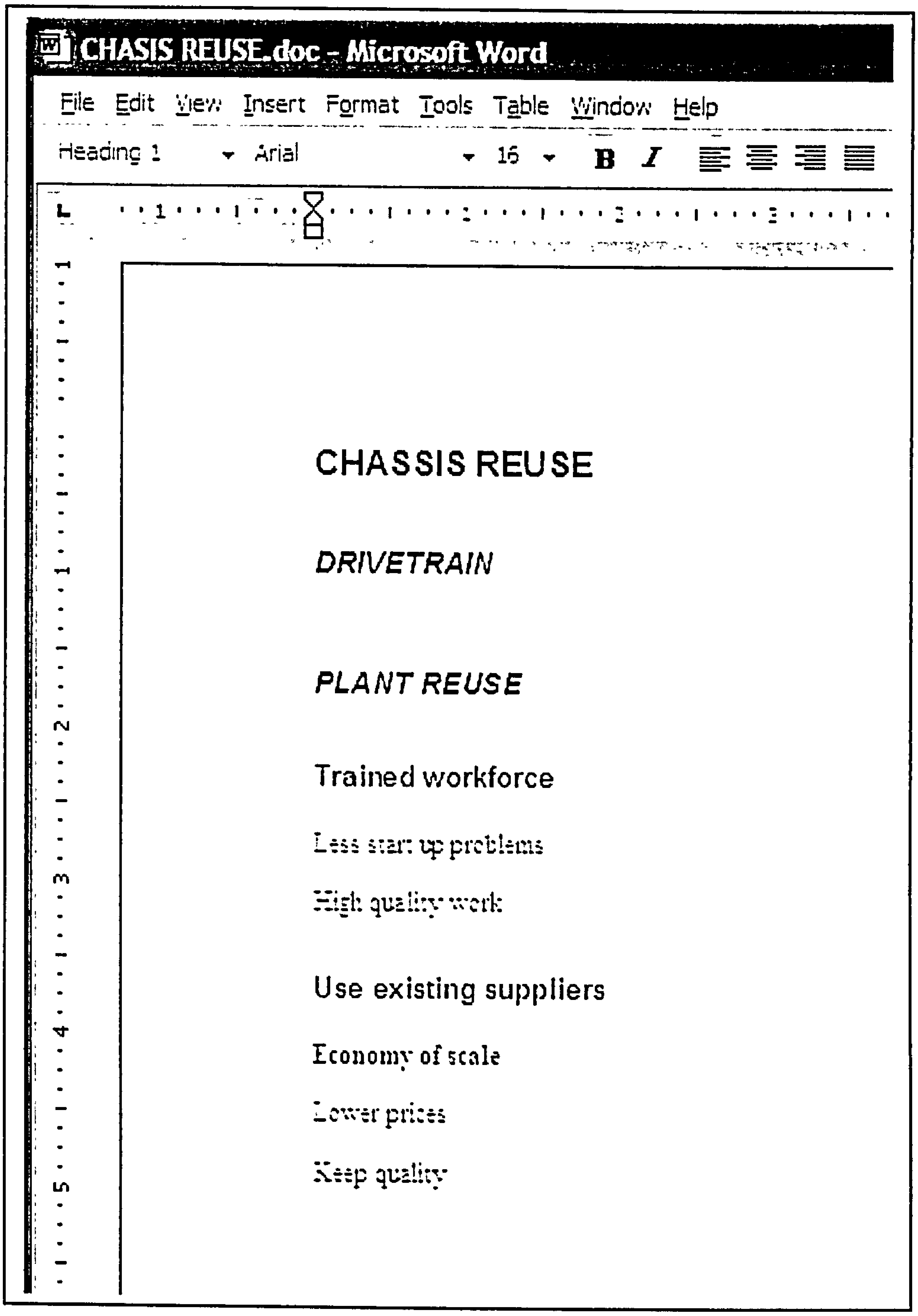


Figure (5-11) Natural Language Document Output

5.4.2. BOM (Bill of Materials) & Supplier List

The information stored throughout collaboration represents high-level early concepts that are iteratively broken down into increasingly complex components. These different levels of complexity are capable of representation by the concept maps. Thus facilitating formats to be represented such as the parts and sub assemblies along with their associated materials.

Supply chain integration can be further improved by greater completeness at the time of concept definition. At more complex levels of representation where concept maps are representing the actual product parts and features as has been described in Chapter 4, each feature provides the facility to associate the materials required to produce it.

Collaboration based on the Concurrent Engineering philosophy requires the earliest possible involvement of the collaborating parties; further to this these parties must contribute as early as possible. It is this early contribution of necessary information that provides greater completeness of concepts at creation. When information is of greater completeness at the time of creation it makes the retrieval and processing of information for later lifecycle tasks much more simple. The association of a supplier to a concept representing a part, either for the part itself or the required materials allows a supplier list to be generated for general contact purposes such as a company address book. Collaborating part and material suppliers linked to concepts thus have the facility to be made aware at an early stage of potential material requirements to assist in turn their own projected material and process planning. This information once approved by the collaboration can then be automatically processed into a more formally structured BOM for official parts ordering.

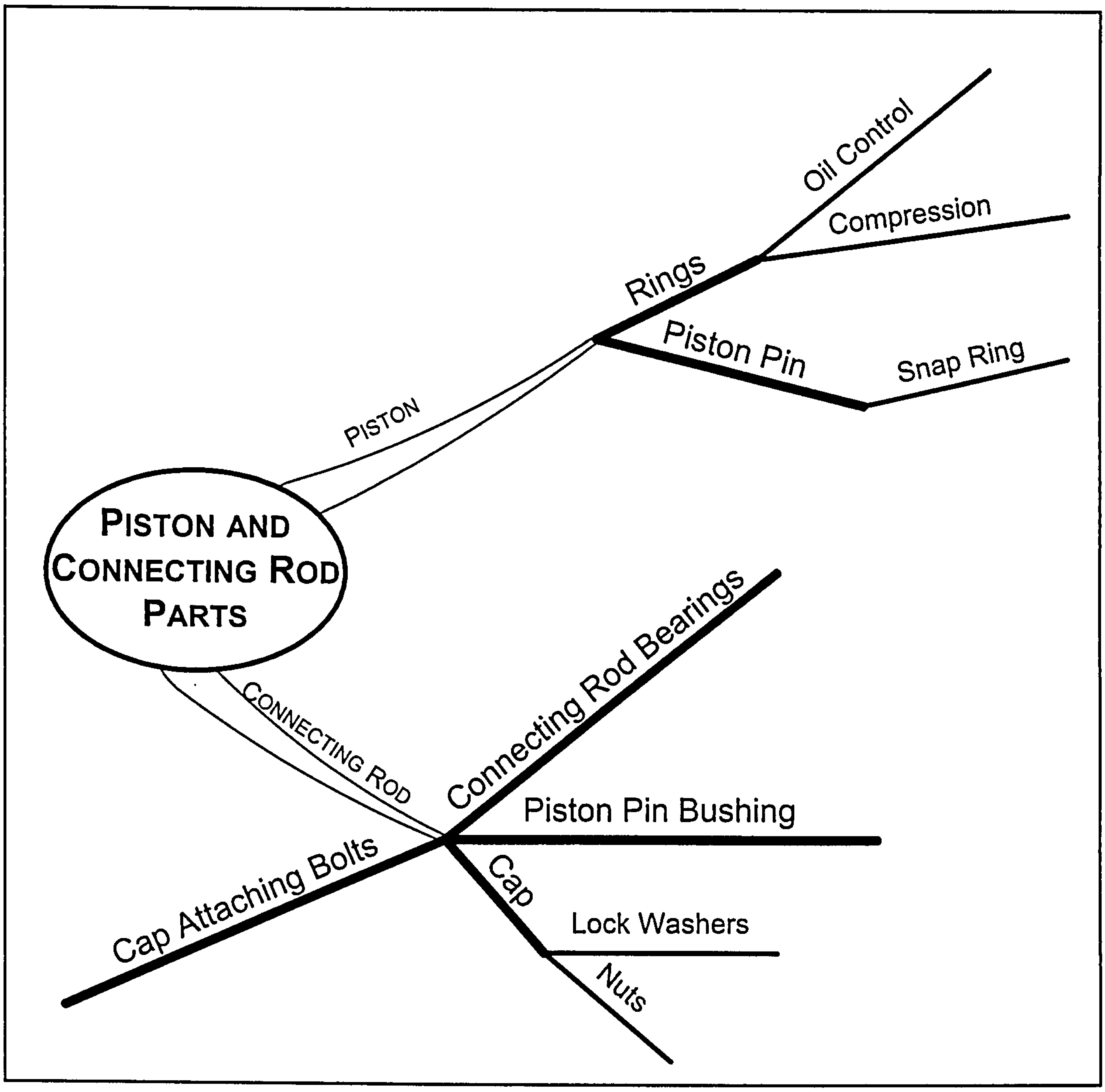
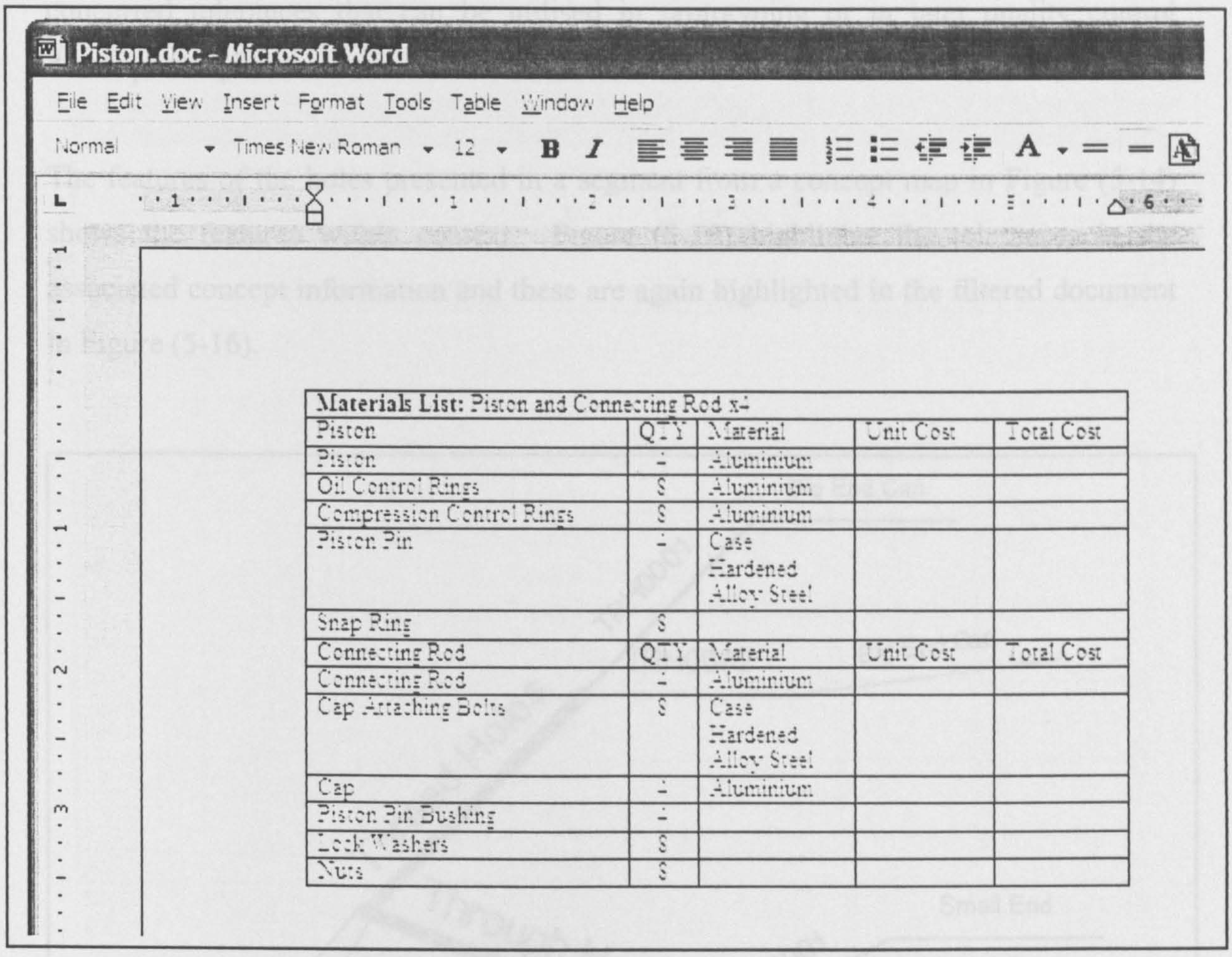


Figure (5-12) Partial Concept Map – Piston and Connecting Rod Parts



Materials List: Piston and Connecting Rod x4				
Piston	QTY	Material	Unit Cost	Total Cost
Piston	4	Aluminium		
Oil Control Rings	8	Aluminium		
Compression Control Rings	8	Aluminium		
Piston Pin	4	Case Hardened Alloy Steel		
Snap Ring	8			
Connecting Rod	QTY	Material	Unit Cost	Total Cost
Connecting Rod	4	Aluminium		
Cap Attaching Bolts	8	Case Hardened Alloy Steel		
Cap	4	Aluminium		
Piston Pin Bushing	4			
Lock Washers	8			
Nuts	8			

Figure (5-13) BOM (Bill of Materials) Document Output

5.4.3. Design Variable Tolerances

Integration of other activities required in the early design stages facilitates the lowering of costs and the reduction of unforeseen problems occurring later in the product development lifecycle. A main purpose of collaboration is to identify requirements and raise conflicts based upon those requirements.

Complex requirements can be specified as design variables and associated with a concept. These design variables represent the tolerances of agreed constraints upon aspects of the product under development, a consensus reached by the collaboration.

The set of design variables that represent a product can be extracted in the same manner as the materials and supplier information. This effectively results in a checklist of concurred tolerances that can be utilised in prototyping or in later quality control development phases.

The features of the holes presented in a segment from a concept map in Figure (5-14) shows the features within context. Figure (5-15) highlights the tolerances in the associated concept information and these are again highlighted in the filtered document in Figure (5-16).

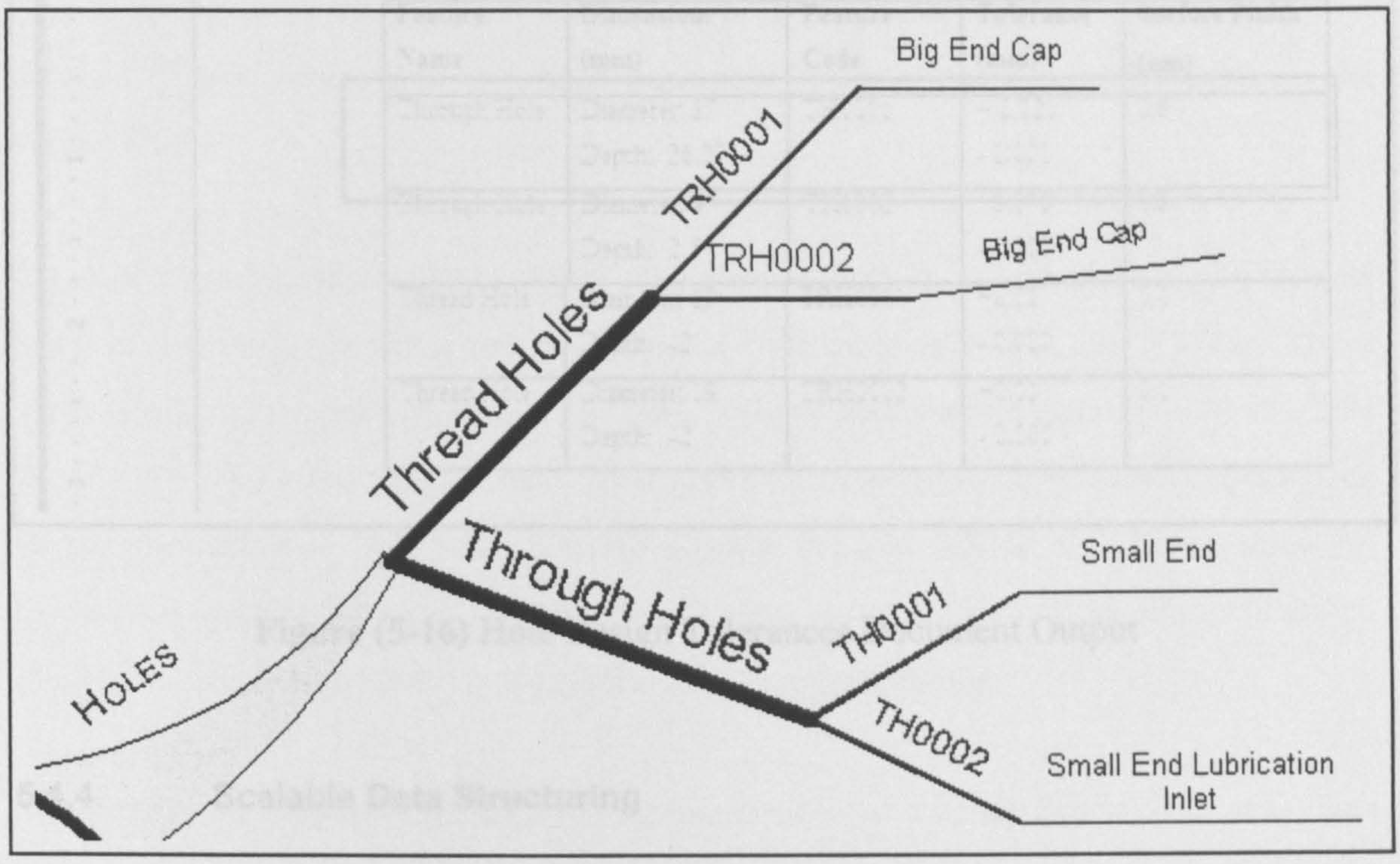


Figure (5-14) Partial Concept Map – Hole Features

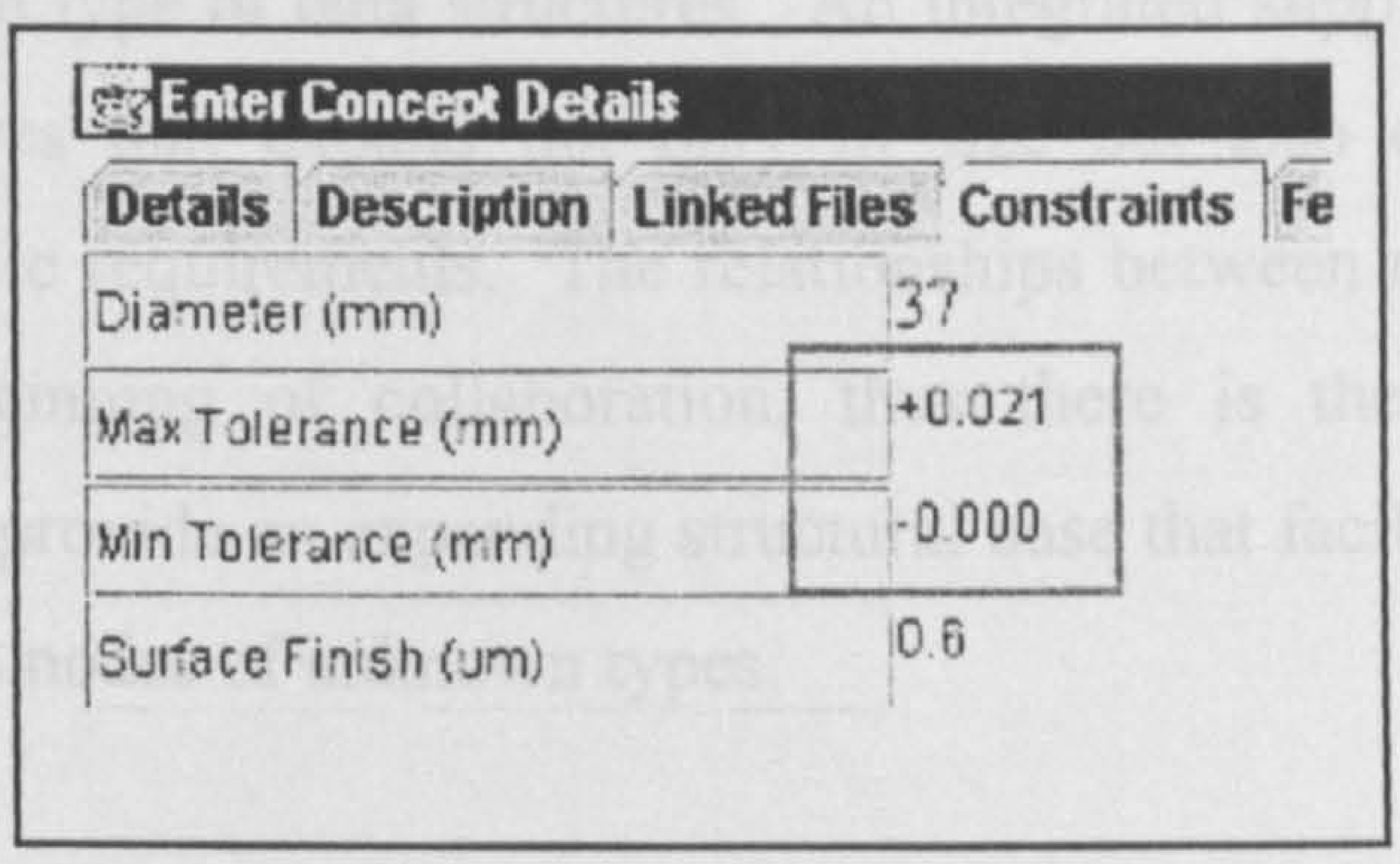
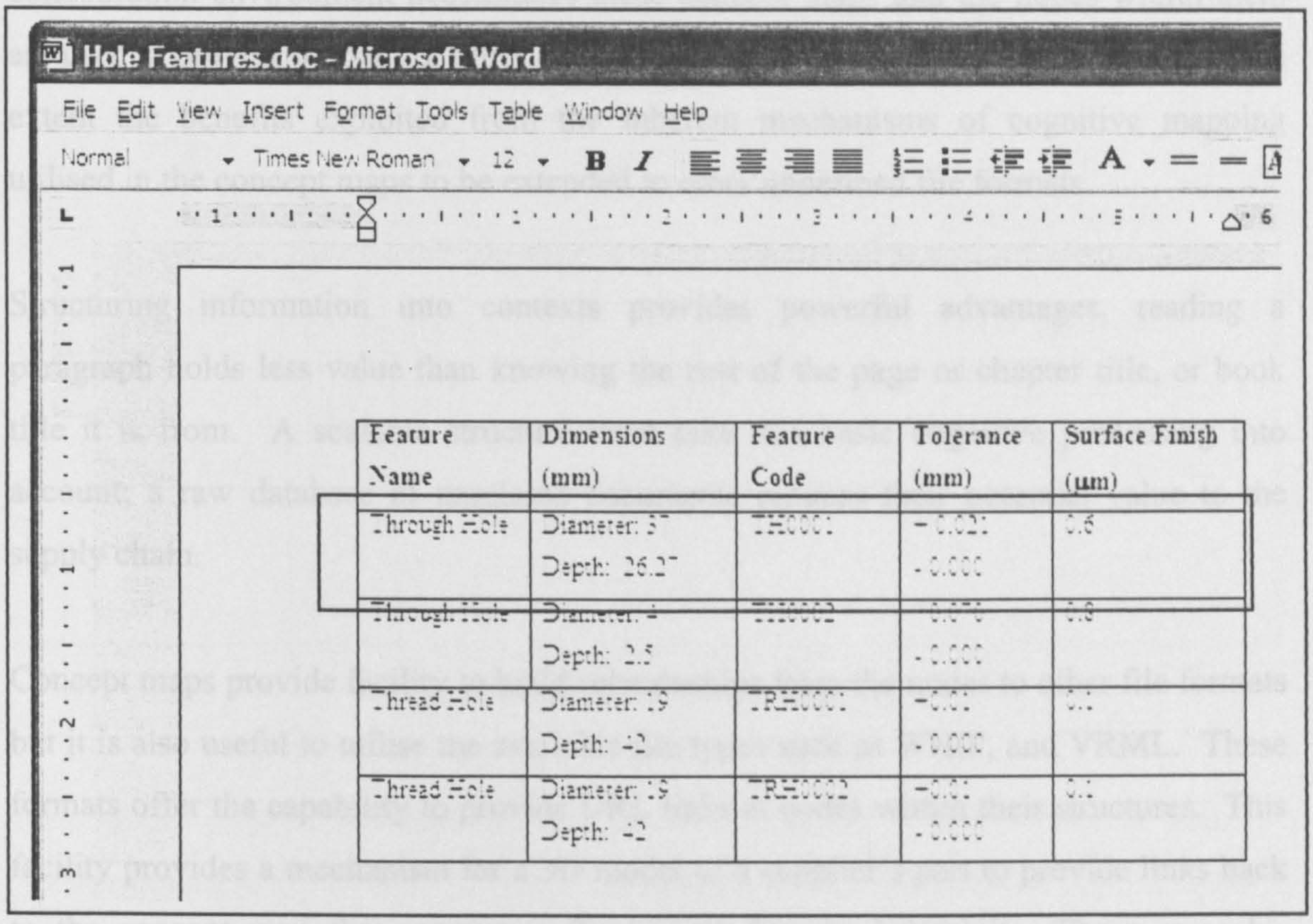


Figure (5-15) Feature TH0001 Highlighted Tolerances


Hole Features.doc - Microsoft Word

Feature Name	Dimensions (mm)	Feature Code	Tolerance (mm)	Surface Finish (µm)
Through Hole	Diameter: 37 Depth: 26.27	TH0001	+0.021 -0.000	0.6
Through Hole	Diameter: 4 Depth: 2.5	TH0002	+0.070 -0.000	0.8
Thread Hole	Diameter: 19 Depth: 42	TRH0001	+0.01 -0.000	0.1
Thread Hole	Diameter: 19 Depth: 42	TRH0002	+0.01 -0.000	0.1

Figure (5-16) Hole Design Tolerances Document Output

5.4.4. Scalable Data Structuring

Scalable data structures are essential to enable the effective integration of the supply chain. Collaborative product development results in the creation of a relatively unknown quantity and type of data structures. An integrated supply chain requires the provision of structures that expand not only in size but also capability to fit the collaborations dynamic requirements. The relationships between nodes of data is also unknown at the beginning of collaboration, thus there is the requirement for a supportive system to provide an expanding structural base that facilitates the creation of relationships between nodes of unknown types.

The relationships between nodes of data are core to the utilisation of our internal cognitive maps and the benefits associated as such described in Chapter 2. These mechanisms have within the micro sense been exploited in the use of the concept maps.

The consideration of the macro data structures required in the product development collaboration environment necessitates these concept maps and the nodes within them are used to provide a hook for other data types to be linked to. This enables, to a certain extent the benefits exploited from the inherent mechanisms of cognitive mapping utilised in the concept maps to be extended to other undefined file formats.

Structuring information into contexts provides powerful advantages, reading a paragraph holds less value than knowing the rest of the page or chapter title, or book title it is from. A scalable structure must take this basic cognitive processing into account; a raw database of unrelated documents reduces their potential value to the supply chain.

Concept maps provide facility to build relationships from the nodes to other file formats but it is also useful to utilise the available file types such as WHIP, and VRML. These formats offer the capability to provide URL links at nodes within their structures. This facility provides a mechanism for a 3D model of a supplier's part to provide links back to the concept map that represents the logical thought behind it. These same bi-directional relationships can be formed with most common files types when combined and embedded in the relatively simple HTML document format.

The facility for these relationships to be created provides a means for scalable structures to be formed by the collaboration. There is little restriction placed on what information can be shared and what relationships can be formed amongst the nodes of data.

Supply chain integration and collaboration utilising these relationship building mechanism results in the creation of a network of linked information. This allows cyclic networks to be constructed. Concept maps form the backbone of this network effectively resulting the imposition of a tree structure over a graph data structure. A graph information structure allows nodes to be connected to each other as follows:

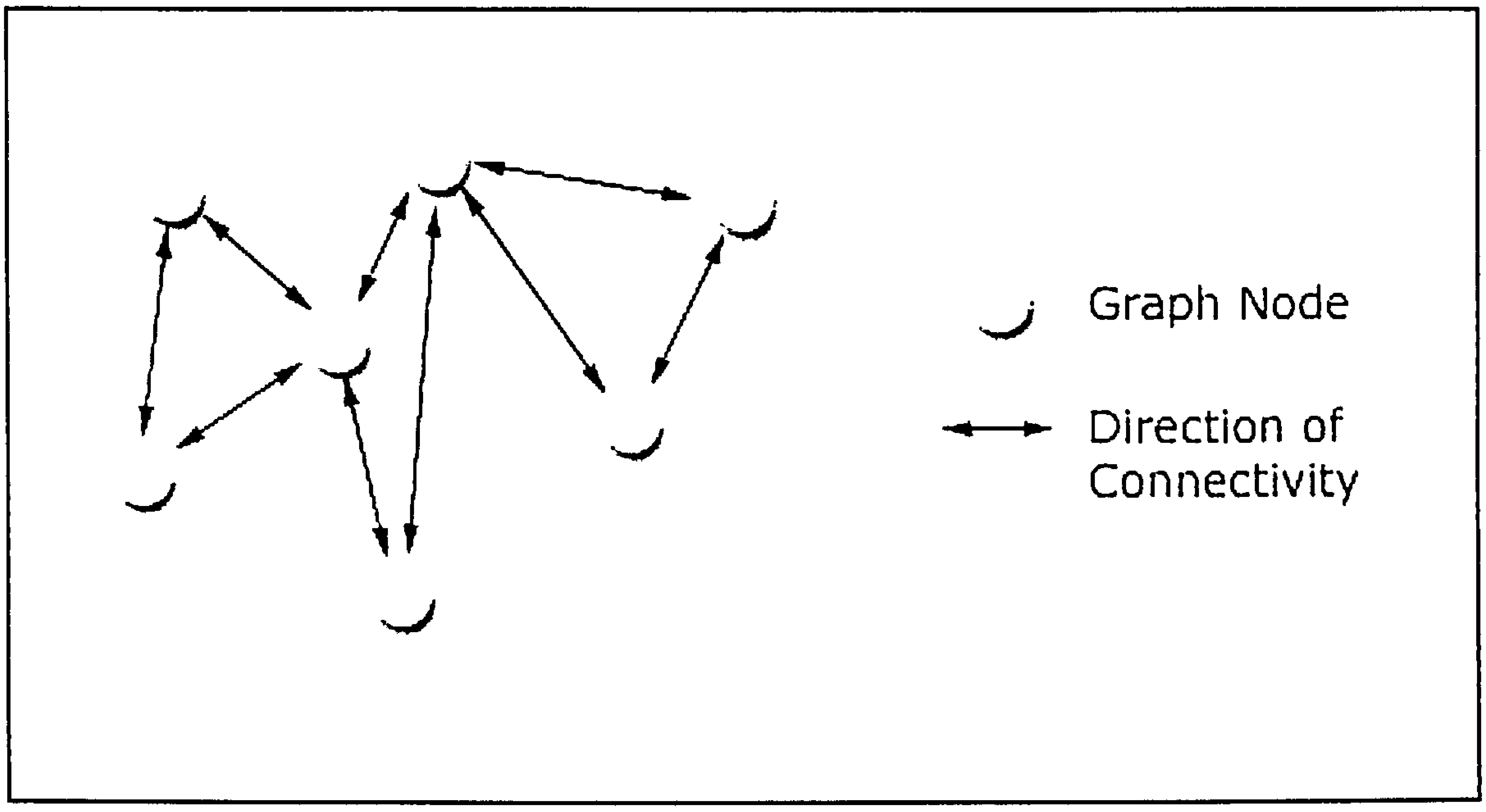


Figure (5-17) Graph Data Structure

This graph information structure can represent either a single concept map of interconnected nodes at the micro level or a group of connected files at the macro level of representation. Graph data structures are essentially subsets of one another and very well suited to an environment of scalability facilitating the representation of a concept map that can in turn form a node of a larger graph network.

The product development lifecycle dictates a starting point and to a certain degree completion. This necessitates a form of a linear flow to the creation of product development data. In its purest form a scalable graph data structure is just a set of nodes that can be connected in any manner, however this results in a confusing web of information. It is the purpose of the concept maps to form the backbone of the framework with other file types being related to expand upon the detail level available from this medium. The concept maps represent the main entry point for each stage in the product development lifecycle linked to the main concept map of the previous phase. This consequentially enforces a tree structure on the graph structure throughout each development phase. Figure (5-18) highlights this preferred framework.

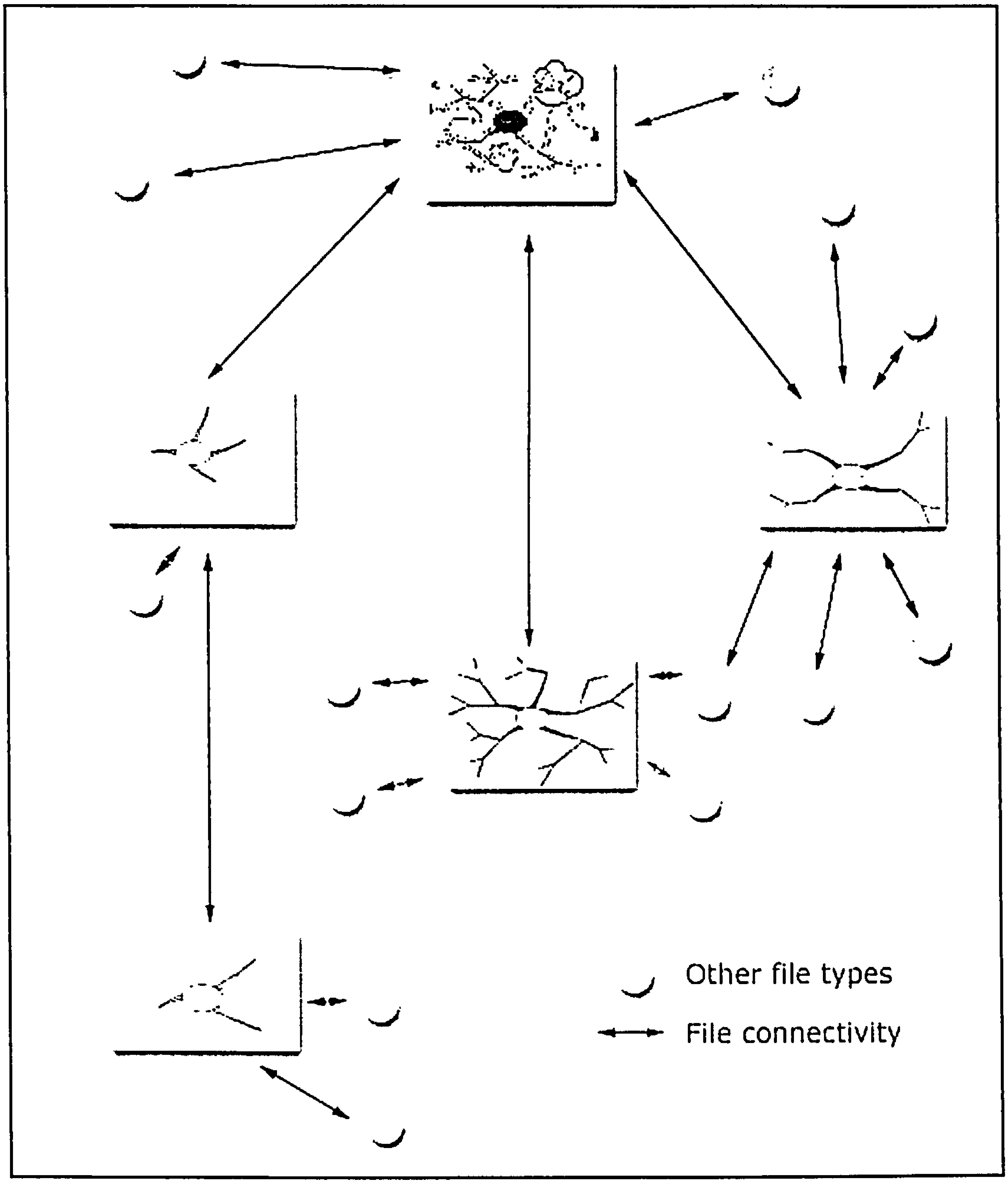


Figure (5-18) Tree and sub Graph Structures of Concept Maps and External Files

Scalability as has been shown is essential in the way the data produced through use of the system is structured, however scalability is also essential in terms of the actual system itself. The lifecycle of a system in place today is generally short lived, it is usually safe to assume its functionality or performance and quality will be superseded within a couple of years. When collaborative efforts take place they represent a significant investment from the organisations involved. This is not just the IT investment involved but also the product data stored. It is essential for product data to be accessible and comprehensible for many years during and after the product development lifecycle. Therefore a system that provides a shared collaborative environment must provide avenues and effective means with which to scale it, in order to be able to apply it to both current and unknown requirements.

The purpose for providing a collaborative environment is to break down the barriers to collaboration in doing so it is often easy to put new barriers in their place; this must be addressed. A system that is highly limited in its information representation formats will serve a purpose for only a short period of time as new technologies push forward. Consideration must be made for a system that allows not only improvements to it but also provides flexible mechanisms for integrating and connecting to other systems. This scalability consideration concerns both the problem of platform compatibility caused by the lack of consistency between mixed organisations with differing IT policies and also the expansion of the capabilities of the environment.

Scalability can be achieved through the integration of the now ubiquitous URL. An embedded link in a concept map node is an address for a URL. This link provides a mechanism for increasingly detailed information in various file formats to be related. Utilisation of this linking technique makes available the connectivity to other systems providing they can accept submitted data through an HTTP GET or POST request. This functions in the same manner that information is submitted through a web page such as online shopping orders. Information could be submitted to a server-based program for additional processing. The submitted data to an external system could represent an entire concept map structure, providing other systems with the means to process this data. This system of linking facilitates potential connectivity to online workflow, project management, and calendar tools. This would enable the expansion of the system from an information sharing conceptual collaborative environment to a system that can relate actual work progress to milestones, and the arrangement of online collaborative meetings through external diary applications.

Using a URL as a means of providing a scalable interface has its limitations and thus other means of scaling the system should also be provided. One of the benefits of a URL is its standardised form; this is considered when the information representation of the concept maps is in XML. The concept maps must conform to the concept map DTD, this provides a means of knowing the rules and the form that can be used. The utilisation of the open structure of the DTD presents an ideal mechanism for providing application developers with a set of rules for the data that they will expect. The collaboration's information can be exported in XML making the data processing much more viable for other applications. As put forward earlier it is possible to parse the XML to produce greater process specific output such as a BOM. This capability to interface to external applications provides the opportunity to expand the capability of the system to facilitate for example presentation software such as Microsoft's PowerPoint. Concept maps could be turned into the framework of a presentation allowing the user to directly take a presentation straight from the current status of the work thus ensuring consistency. The provision of capability for external applications to import / export easily to the environment supports the breaking down of information islands that currently prohibit much in the way of collaboration.

5.5. Data Standardisation

Integration of the supply chain necessitates standardisation at varying levels. Standardisation is essential as part of the protocol with which organisations collaborate. In an environment that supports multiple organisations there must be strong standard elements to provide a basis for collaboration.

Standardisation must consider the common platform, data formats, and human computer interface. See Figure (5-19).

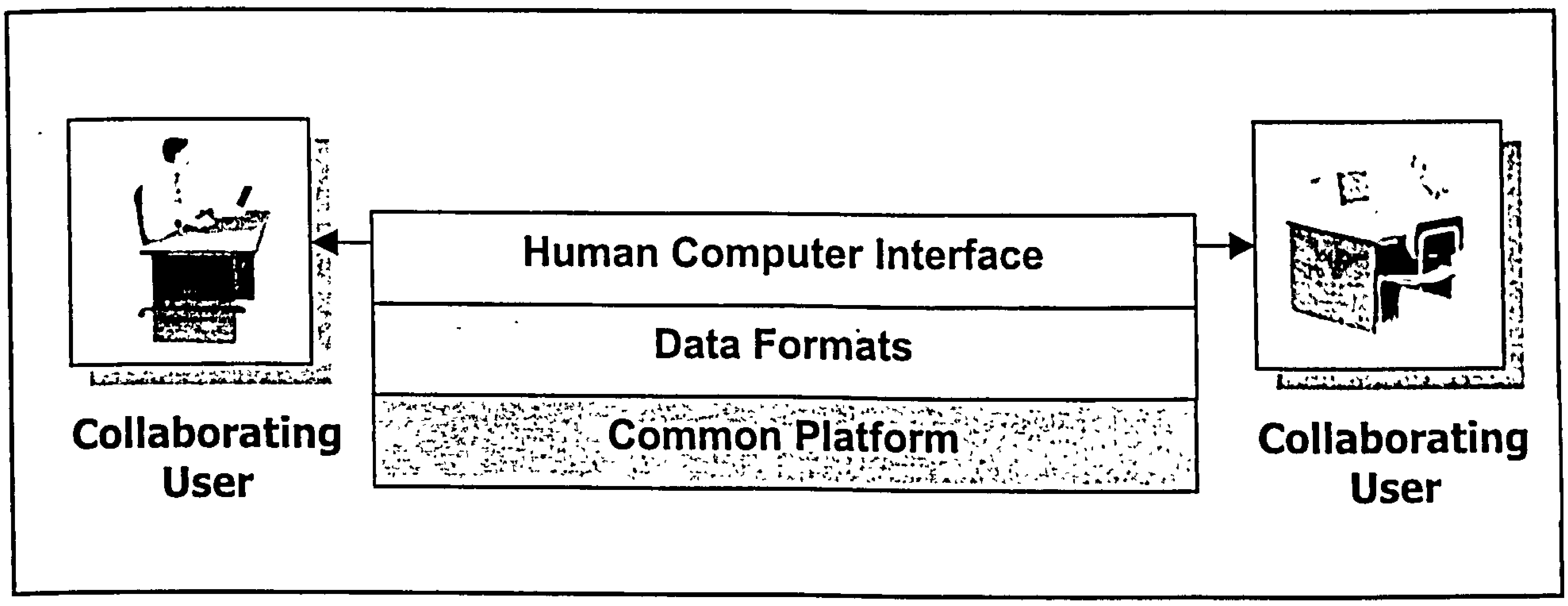


Figure (5-19) Data Standardisation Elements in Collaboration

5.5.1. Standardisation in the Human Computer Interface

Distributed product development collaboration is essentially founded on the accurate sharing and conveyance of the collaborating parties viewpoints. All facets of standardisation are important but it is the human computer interface that the users will be confronted with as their guide or barrier to the other parties.

There are generally accepted user interface de-facto standards, such as the placement of screen furniture, the menu bar, toolbar, and document body. Each of these can be further decomposed into standard items, for example the file menu on the left, and the help on the right. Style guides for interface design are common, however it is usually the knowledge that is represented that is far from common.

The human computer interface has the responsibility to deliver the collaborative efforts of others; the concept mapping described provides a consistent manner for this. Standardising the way collaboration takes place is multi-faceted. The onus for representation of knowledge in this work is placed upon the concept maps. However as has been mentioned earlier there is a requirement to standardise the access methods to the collaborative formats.

Globally distributed collaborative product development crosses not only geographic but also temporal boundaries. Software interface development utilises de-facto interface standards to facilitate the ease of use of the interface. In the same way that interface elements should remain consistent from application to application, the same interface should be used for access to the collaborative environment regardless of temporal and geographic factors.

Geographic factors should be overcome through the use of web-based technologies, providing access to the same material regardless of locale. This material consists of the user interface embedded within a web page thus providing exactly the same experience. This also provides a suitable method for updates to the system. These can be undertaken at a single location and reflected upon all the users systems without the need to apply a large number of patches to installed software.

The integration of people in the product development process necessitates that the issue of common meeting slots be addressed. Concurrent Engineering philosophy requires that the parties involved in the product development lifecycle are involved from the earliest possible stage. Domain specialists are generally the people involved, experts in their own fields, these are generally people whose skills are in high demand and thus finding a common meeting slot for the number of people required is difficult. Integration of these people in the global setting is even more difficult when combined with the world time zone differentials.

The majority of systems today provide mechanisms for these different communication types as previously termed, synchronous and asynchronous collaborations. For example chat room style facilities are provided for real time communications or video conferencing. These facilitate synchronous collaboration and require a common meeting slot to take place. Asynchronous or sequential collaboration is easily facilitated by email for example. The problem with using such mediums to address synchronous and asynchronous collaboration as previously discussed is their lack of integration, the information itself is dispersed, but also the lack of uniformity in the interface. Information needs to be represented in the same manner and not separated as a means of solving the synchronous / asynchronous problem.

Figure (5-20) shows this problem in the context of the automobile product development example. The concept map in the design stage has various elements linked to it as previous detailed, in this case the time difference between London and Frankfurt is minimal and the parties involved can utilise a common time slot with which to collaborate. However the party in Chicago find their available time slot at 3pm renders them unavailable to join in the collaboration that happened earlier in day. Thus the parties in London and Frankfurt contribute to the concept map and link items in real time together and the department in Chicago accesses the same data representations to add draft CAD models at a later time.

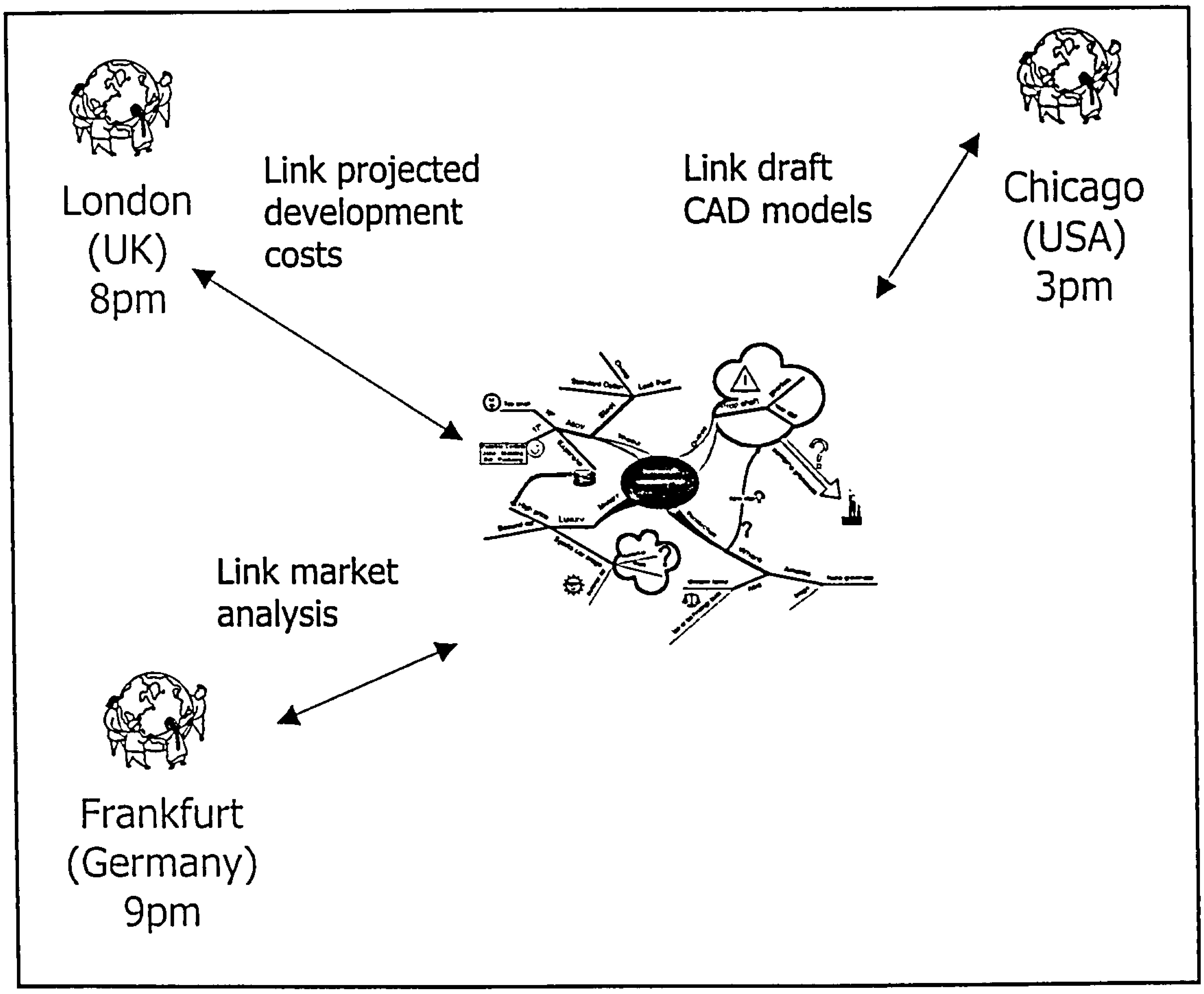


Figure (5-20) Global Time Different Collaboration

5.5.2. Standard Data Formats

Integration of the supply chain necessitates that the collaboration utilises a common repository for the sharing of information. It has been stressed the importance of the comprehension of information, however to underpin this requires the employment of independent data standards.

Just as human communication requires common factors to enable comprehension, the systems involved in the collaboration also need certain standards to facilitate a common base to exchange data.

The worldwide access and scalability of a web-based system have made it the obvious environment for a global collaborative system from the view of functional capability. However the benefits that can be leveraged from global connectivity also are the cause of problems in the development of common standards for information exchange. There are a number of organisations pushing forward their versions of browsers, client and server programming languages and platforms, each looking to gain competitive advantage from the others. There is no de-facto method for web based systems development at present; each technology has its own advantages and disadvantages.

Standards for data formats envelop the Internet community at present, the W3C (World Wide Web Commission) generally spur the advance of these standards and recommendations. However many of these standards are corrupted through extensions of browsers by vendors and organisations to attract a greater number of users, this in turn provides a testing ground for ideas and whether they are taken up by the development community as a whole or not. This situation leads to an unstable development environment for web based applications, ever evolving standards and technologies with constant modifications make it difficult to adhere a set format for information representation and hence collaboration.

There are however in this diverse range of technologies several that have widespread industry backing, XML and Java are those used in this research. XML is utilised in this research as a mean of providing a stable and portable collaboration data format.

XML is a relatively stable standard within this dynamic development environment and thus was chosen for the representation of the concept maps ensuring their compatibility and comprehension for at least the medium term future. Chapter 4 details how XML has been used to facilitate information sharing across the Internet far beyond the now ubiquitous web page.

Collaboration efforts require serious investment in technology and the data that is stored in supportive repositories. The parties involved in the supply chain are committing product development data to a specific collaboration environment and thus this data must be independent from the application that created it. XML provides the means to create a data format that can be processed by a wide range of software tools. A simple web browser for example with a style sheet could format the data. This separation is usually not necessary in the general computing environment however product development data has a long life that usually far exceeds that of a software system regardless of how scalable it is.

Concept mapping provides a very powerful mechanism for collaboration, however there is a need to provide greater facility for the complex product detail involved in the collaboration of a supply chain. This requires the representation of the product development data itself. It is possible for XML to be used in this realm, however the developments in this area have been limited and relatively primitive.

Collaborative product data exchange formats are as diverse as the efforts in the web - based community. There is however a standard in the form of STEP (Standard for the Exchange of Product) aimed at encompassing all the facets of product data exchange and thus has been considered in this research work.

STEP forms a group of standards that represent the sphere of product data exchange on a formal basis. This formal basis primarily provides a consistent data exchange format independent of single CAD systems to avoid the need for the many translations between collaborating organisations thus avoiding the errors involved in translation between incompatible formats. STEP shares various characteristics with its less specific alternative XML in that it aims to be entirely platform independent and be understandable when separated from the creating application.

Implementation of STEP is a vast undertaking; there are now expensive development environments to assist with this area such as from (STEPTools (1999)), this area has also been slow to accommodate integration with web based applications with developments focusing rightly on the foundations of product data exchange.

STEP can be implemented on a variety of complexity levels from the file format itself to shared databases of fine granularity to a full knowledge base. Large organisations are still in the relatively primitive stages of implementation concerning STEP and this research effectively is restricted to realistic consideration of the file format level. This however is fully extendable from this basis itself into a fully fine-grained database implementation.

The ISO 10303-21 part of STEP describes a standard representation for a file that details data instances conforming to data models specified in the EXPRESS language.

There is an ever increasing development of the Application Protocols (AP) for STEP, these are a series of parts of the standard that use the Integrated Resources (see Figure (2-5)). These Application Protocols form the basis for industry and domain specific representations such as printed circuit assemblies, shipbuilding, and automotive mechanical design.

A file can represent a model based on the EXPRESS definitions in an Application Protocol such as AP203. A simple sample EXPRESS definition is given below:

```
ENTITY through_hole;
  Diameter           : FLOAT;
  Depth              : FLOAT;
  Code               : STRING;
  Tolerance          : FLOAT;
  surface finish     : FLOAT;
  contact            : person;
END_ENTITY;

ENTITY person;
  First_name    : STRING;
  Surname       : STRING;
END_ENTITY;
```

An instance that corresponds to this EXPRESS data model and thus would be stored in the physical file format would be as follows:

```
#1 = THROUGH_HOLE('37','26','TH001','0.021','0.6', #2);
#2 = PERSON('Neil', 'Anderson');
```

The Application Protocol AP203 represents 'configuration controlled design' and consists of the data models suitable for supporting output from design variables extracted from the concept maps. This allows a facet of the system to export useful accurate design information directly into a strong standard for the exchange of this product information. Figure (5-21) shows this process.

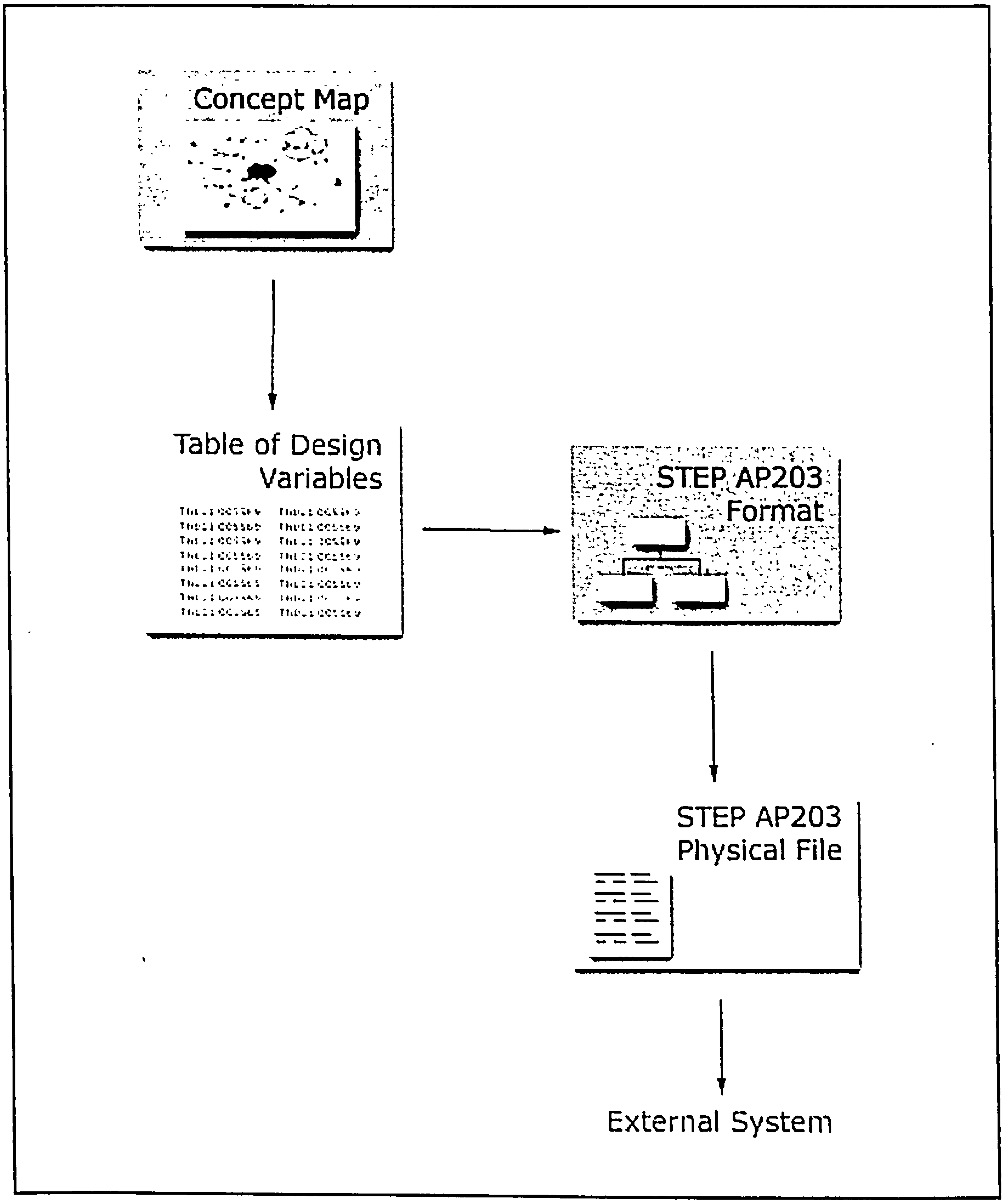


Figure (5-21) Concept Maps to AP203 Physical File Format

5.5.3. Common Platform Standardisation

Product development collaboration is facilitated through standardisation of the human computer interface and in the data formats used and it is essential that these are supported by effective standardisation of the common platform. The common platform in this research is facilitated by Java and the HTTP protocol.

Adequate human computer interfaces and data formats can be developed on platforms using different technologies. Utilisation of HTTP and the web facilitates a common platform based on Java to be available to all parties without dependence on specific operating systems or environments.

A common platform in this research is centred on an applet that provides the application interface for the collaborating parties. This communicates with a central server via the RMI methods in section 5.3.2.

5.6. Object Oriented Database

Integration of the supply chain during collaborative product development necessitates the maintaining of a large number of complex relationships between data entities.

Section 2.5.2, details the choices between relational and object oriented databases. An object-oriented database has been chosen for this research work due to the inherent capability to represent complex relationships present in the product development data. There are effectively two databases one for the storage of the concept map structures, and their relationships and one for the actual product data.

The concept map class structure is represented in UML and the class structure developed in Java and is stored as such in the Object Database.

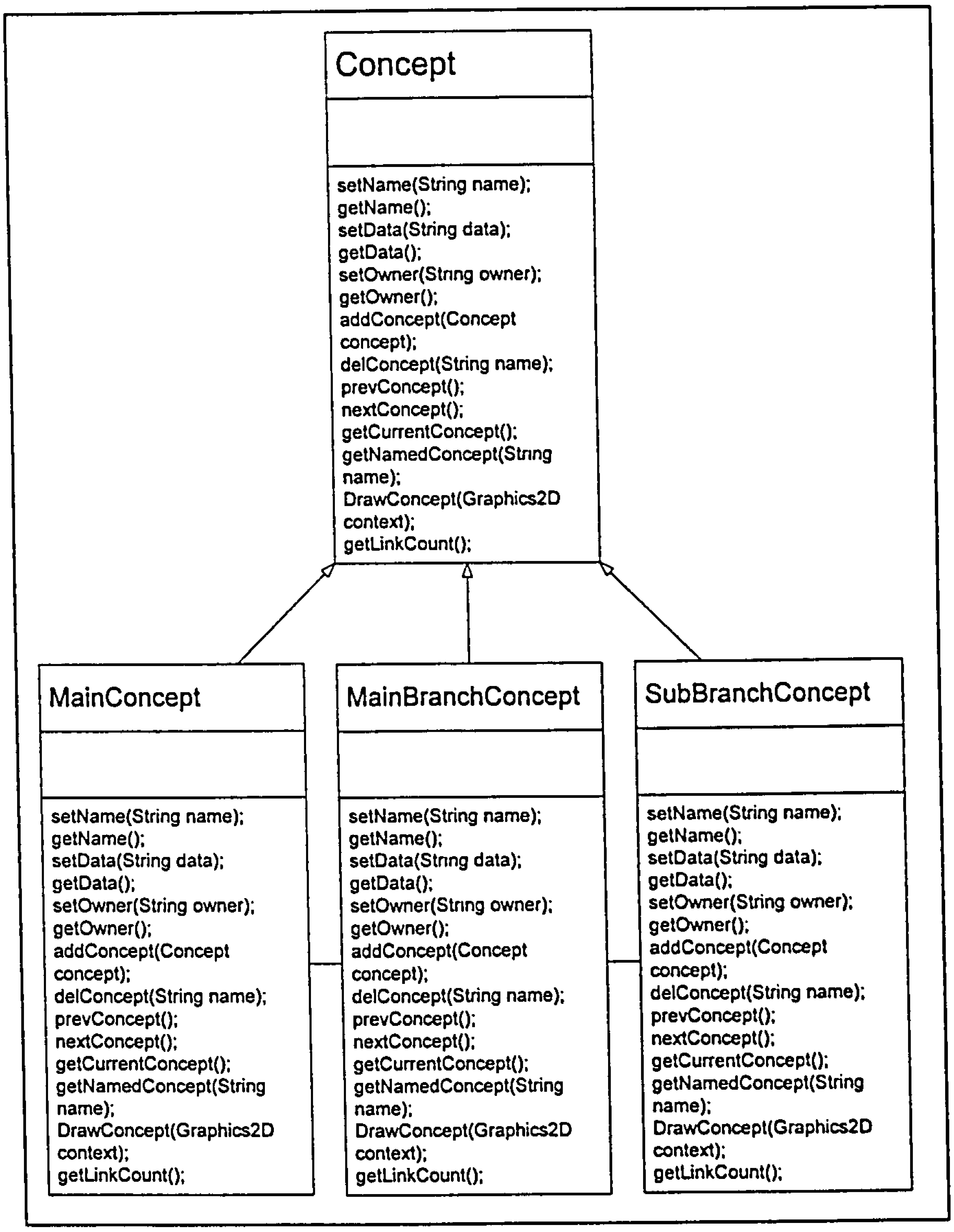


Figure (5-22) Concept Map Object Hierarchy

There is an abstract class that represents all possible types of concepts and the three main types of concepts override this, the main central concept, the main branches and the recursive sub branches.

Figure (5-23) through Figure (5-25) show the information representations from the concept map of Piston Parts through the XML structures of the map to the Java class representation of the example and thus the database structures as the database maintains the class relationships.

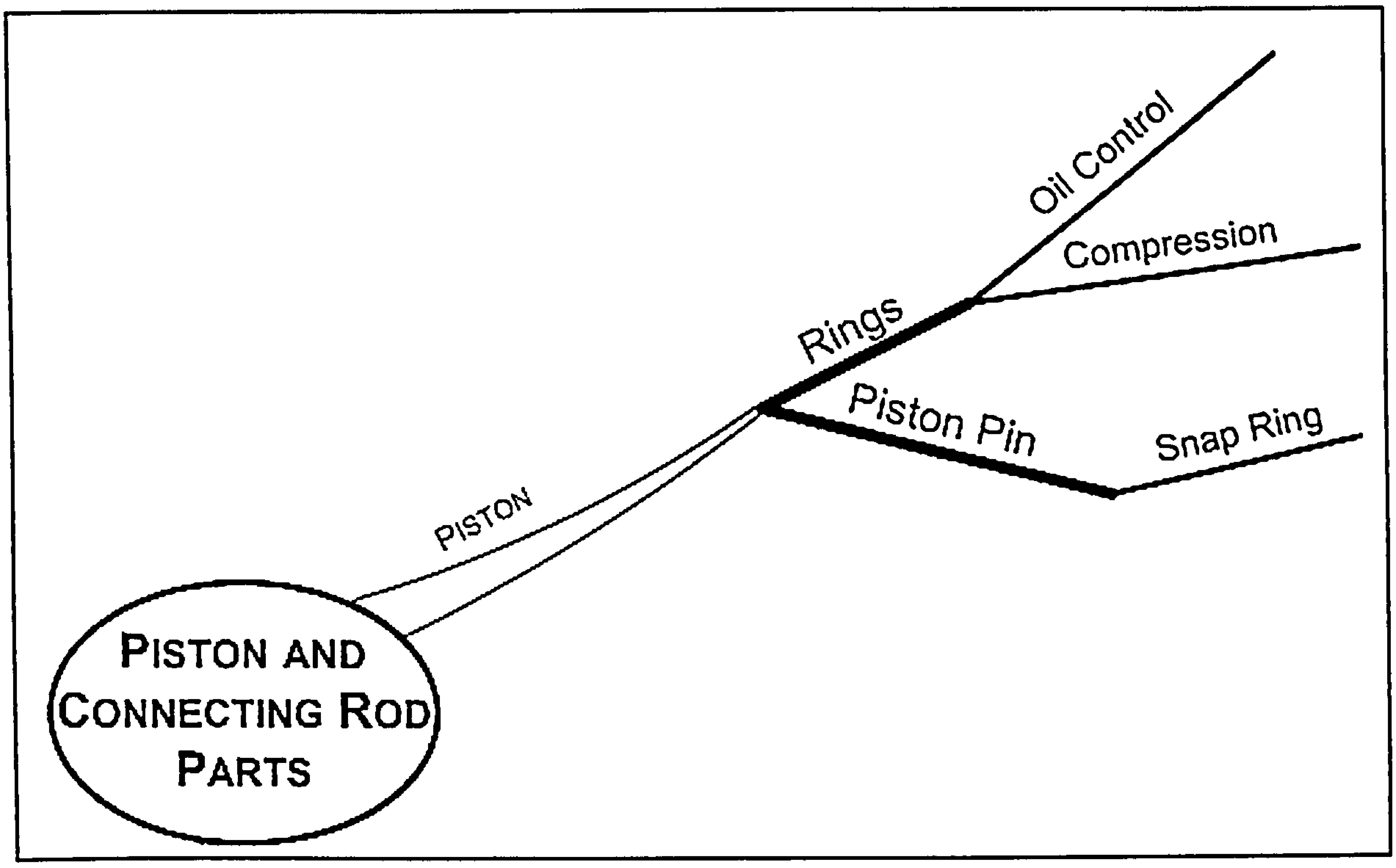


Figure (5-23) Example Piston Concepts

Figure (5-23) is a subsection of the example that can be found in Figure (5-12) and details the parts involved in the piston section of the concept map. Figure (5-24) details the XML representation of the piston concepts in Figure (5-23), each concept is represented and the relationships between concepts detailed by the enclosing of the end tags. Where a concept is connected to another then its parent is defined by an opening tag for example "<MAINCONCEPT>" then the details about that concept are defined in between the "<CONCEPTDETAILS>" tags and then the child concepts are defined such as "<MAINBRANCH>" until the closing tag of the concept such as "</MAINCONCEPT>".

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE CONCEPTMAP SYSTEM "conceptmap.dtd">
<CONCEPTMAP>

  <MAINCONCEPT>

    <CONCEPTDETAILS>
      <NAME>Piston and Connecting Rod Parts</NAME>
      <CREATOR NAME="Neil Anderson"
```

```
            POSITION="Software Engineer"
            ORGANISATION="DMU"></CREATOR>
  <CONCEPTDATE>15/5/2002</CONCEPTDATE>
  <CONCEPTLINKS></CONCEPTLINKS>
</CONCEPTDETAILS>

<MAINBRANCH>

  <CONCEPTDETAILS>
    <NAME>Piston</NAME>
    <CREATOR NAME="Neil Anderson"
             POSITION="Software Engineer"
             ORGANISATION="DMU"></CREATOR>
    <CONCEPTDATE>15/5/2002</CONCEPTDATE>
    <CONCEPTLINKS>
    </CONCEPTLINKS>
  </CONCEPTDETAILS>

  <CONCEPT>
    <BRANCH>

      <CONCEPTDETAILS>
        <NAME>Rings</NAME>
        <CREATOR NAME="Neil Anderson"
                 POSITION="Software Engineer"
                 ORGANISATION="DMU"></CREATOR>
        <CONCEPTDATE>15/5/2002</CONCEPTDATE>
        <CONCEPTLINKS></CONCEPTLINKS>
      </CONCEPTDETAILS>

      <CONCEPT>
        <BRANCH>

          <CONCEPTDETAILS>
            <NAME>Oil Control</NAME>
            <CREATOR NAME="Neil Anderson"
                     POSITION="Software Engineer"
                     ORGANISATION="DMU"></CREATOR>
            <CONCEPTDATE>15/5/2002</CONCEPTDATE>
            <CONCEPTLINKS></CONCEPTLINKS>
          </CONCEPTDETAILS>
```

```
  </BRANCH>
</CONCEPT>

<CONCEPT>
  <BRANCH>

    <CONCEPTDETAILS>
      <NAME>Compression</NAME>
      <CREATOR NAME="Neil Anderson"
               POSITION="Software Engineer"
               ORGANISATION="DMU"></CREATOR>
      <CONCEPTDATE>15/5/2002</CONCEPTDATE>
      <CONCEPTLINKS></CONCEPTLINKS>
    </CONCEPTDETAILS>

  </BRANCH>
</CONCEPT>

</BRANCH>

<BRANCH>

  <CONCEPTDETAILS>
    <NAME>Piston Pin</NAME>
    <CREATOR NAME="Neil Anderson"
             POSITION="Software Engineer"
             ORGANISATION="DMU"></CREATOR>
    <CONCEPTDATE>15/5/2002</CONCEPTDATE>
    <CONCEPTLINKS></CONCEPTLINKS>
  </CONCEPTDETAILS>

  <CONCEPT>
    <BRANCH>

      <CONCEPTDETAILS>
        <NAME>Snap Ring</NAME>
        <CREATOR NAME="Neil Anderson"
                 POSITION="Software Engineer"
                 ORGANISATION="DMU"></CREATOR>
        <CONCEPTDATE>15/5/2002</CONCEPTDATE>
```

```
                <CONCEPTLINKS></CONCEPTLINKS>
              </CONCEPTDETAILS>

            </BRANCH>
          </CONCEPT>

        </BRANCH>
      </CONCEPT>

    </MAINBRANCH>

  </MAINCONCEPT>

</CONCEPTMAP>
```

Figure (5-24) XML for Piston Concepts

Figure (5-25) continues the representation of the XML in Figure (5-24) showing the UML class diagram for the Java code. This diagram however also shows the classes with partial data to maintain consistency with the examples in Figure (5-23) and Figure (5-24) thus it contains instance examples as well.

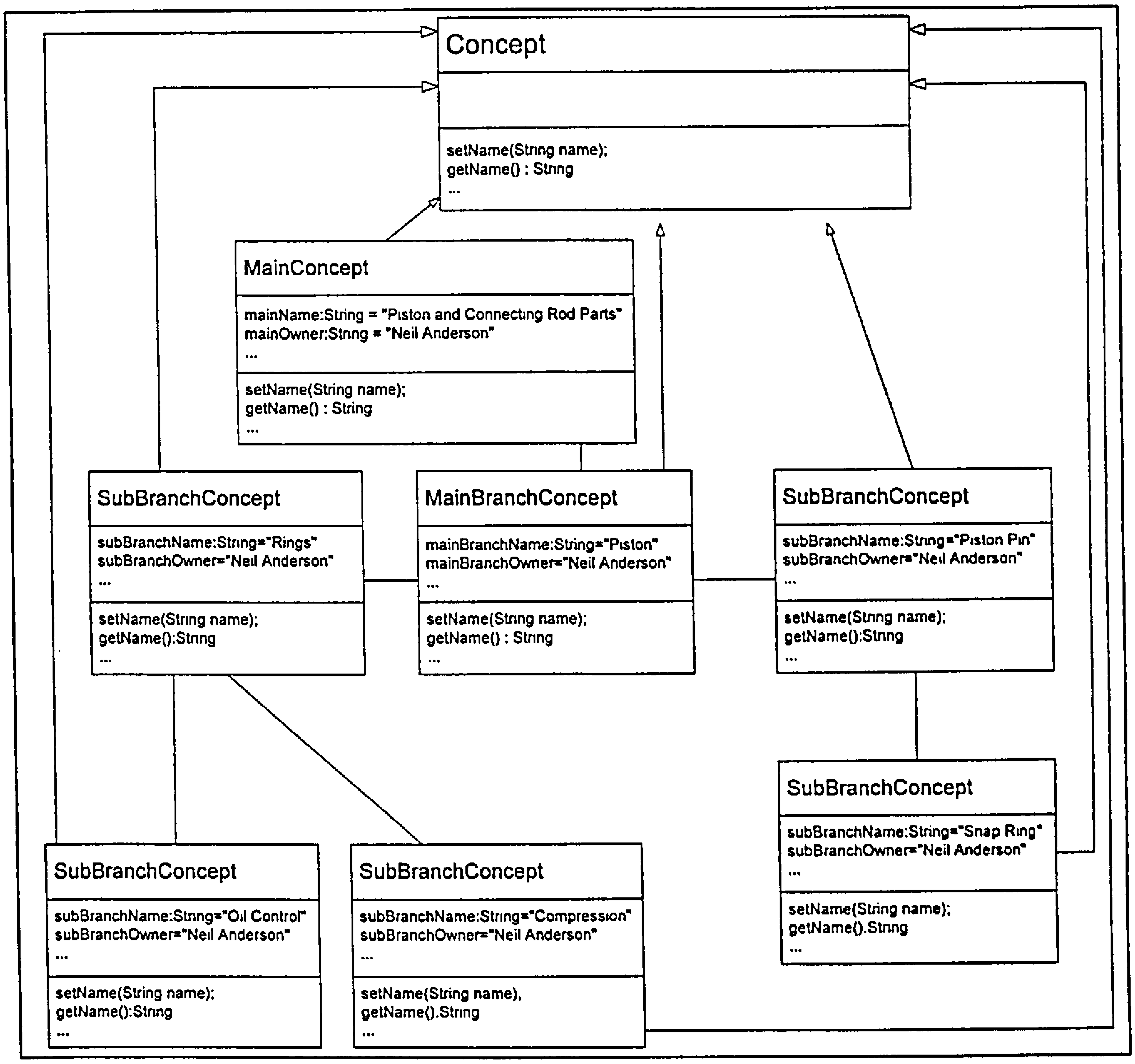


Figure (5-25) UML Class Diagram Showing Concept Relationships with Respect to the Java Class Organisation

The product database is facilitated through the use of STEP. As has been detailed in section 5.5.2 for the representation of product data under the STEP standard there is a requirement for a serious investment in the development technologies. In this research the product data has been represented at the granularity of a single model represented per file. With the utilisation of the full SDAI (Standard Data Access Interface) of the STEP standard the object database can be used to for representation down to attributes of the entity levels providing a much finer granularity and hence greater concurrency for a collaborative team.

The two databases are linked through the linking mechanism already discussed. The product data is reached through the concepts embedded within the concept maps via a URL that contains the query to return the product data from the database.

5.7. CAD Solid Modelling

Supporting systems for collaborative product development has been driven from two prime spheres, that of simple information sharing systems currently in use such as email and bulletin boards and from those wishing to share data between standalone applications. CAD systems used in collaboration often need to share information between them and hence the drive for the previously discussed STEP standard.

Collaborative systems however also have been driven from the CAD environment with the system of (Kim, et al. (1998)) building on the 3D model represented in that research by VRML. This system allows the user to attach text-based notes to a CAD model allowing the user to denote what the purpose was of a part in a short text message. This then leads to the development of systems that expand on the CAD model integrating them with other sources of knowledge.

In this research the CAD models can be represented through the WHIP browser plugin from AutoCAD, and translated into STEP AP203 through the use of the Unigraphics V16 STEP translator.

CAD solid modelling is often used in systems today as a starting point in product development building collaborative systems on this basis. This requires the product development team to already have an understanding of the product and its requirements from the team or risk mistakes being introduced in the first prototypes.

CAD modelling is rapid compared to building physical prototypes although systems have been in place based on stereo-lithographic techniques that allow these CAD models to be rapidly turned into physical models to be examined and tested. This is still a slow process compared to a conceptual driven view of the problem, where dimensional details are unknown or irrelevant at that stage. CAD modelling in the scope of this research is viewed as a process that takes place after much collaboration has taken place and many issues have already been resolved as once time has been invested in a complex model there is greater resistance to make large changes much in the same way as changing physical properties is more complex after time and effort has been spent in producing them.

5.8. Conflict Resolution

Collaborative product development is based on obtaining a consensus of opinion from a group of domain experts concerning a product. The purpose of this collaboration is to identify the differences in perspectives and to evolve new combined viewpoints. This allows the creation of more balanced products that satisfy a wide range of needs and criteria.

Facilitating the observation of conflicts should occur through a variety of mechanisms. Conflicts must first be identified before they can be resolved, how to actually quantify conflicts is a difficult problem, conflict on the low level aspects of product development can be alerted through constraint-based formulae however there has been less consideration towards the resolution of high level subjective conflicts. These are based on the problems early in the product development lifecycle before a comprehensive set of design variables can be determined. These early problems often have little in the way of tangible characteristics and can be subjective in nature. Going through the concept mapping process encourages the domain experts to reveal their opinions on the product at an early stage, each sees each others contributions in context and through a logical rational.

The system allow contributions to be made by the various domain experts but they must be entered within context, the user must add their contribution at its point of relevance increasing the rational thought that is undertaken before the idea is put forward assisting in conciseness of the issues. Conciseness in turn facilitates the observation of the core ideas of importance and thus a greater chance to detect inconsistencies in approaches.

With ownership attributed to a concept or an idea it is then possible to understand the perspective that the contribution was made from and place it not only within the context of the surrounding concepts but in the context of the creators own knowledge base. This also facilitates the notification of the creating party if another team member wishes to modify the concept. In a system where there is purely notification based upon tolerances and design variables conflict can be detected however the rational behind the conflict is often less obtainable, whether the new addition causing the conflict should be modified or the original values kept.

The concept mapping framework facilitates the resolution of conflicts at these more subjective and difficult to quantify levels of abstraction but also as the concepts become more formalised in later design phases with the focus switching to the concepts representing design variables and their associated tolerances it becomes possible to utilise the graph structure that underpins the concepts as a constraint network as shown in Figure (2-4). A constraint agent traverses this network of constraints producing a quantified series of design variables that fit the formulae defining the constraints between variables.

Figure (5-26) shows an example information flow for a conflict notification occurring in the collaboration for a new automobile example. The example shows the conflict is raised through the collaboration working on the Connecting Rod Features concept map. The maintenance department in the UK notice through their testing that excessive wear is occurring due to an inadequate flow of lubrication. They suggest increasing the size of the inlet. The designer of the inlet a member of the USA based design team is automatically notified by email that there is a problem with the inlet.

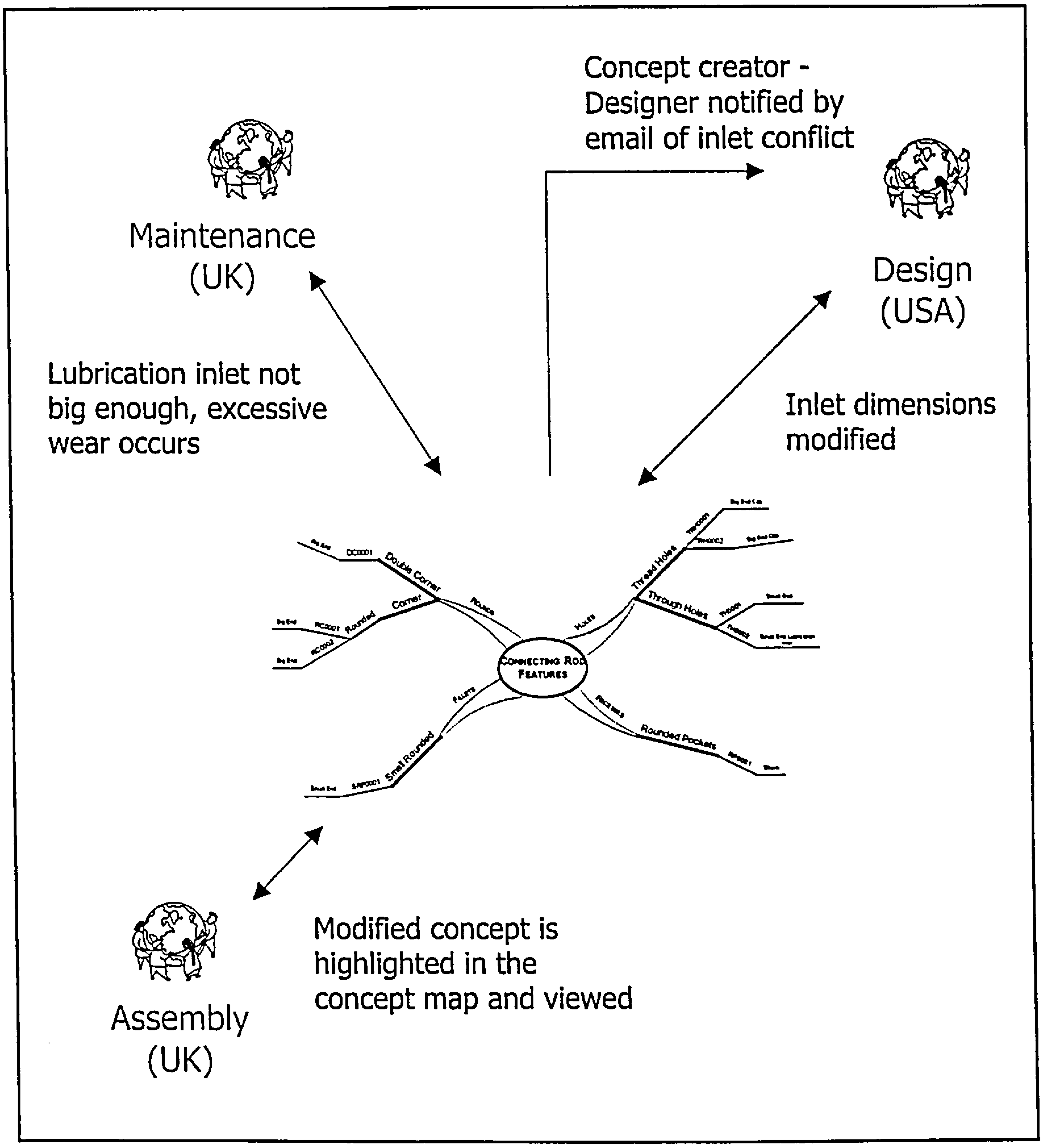


Figure (5-26) Conflict Notification & Resolution

5.8.1. Constraints Networks Derived from Concept Maps

A constraint network consists of a graph structure of nodes / variables connected through a set of constraints between them. See Figure (2-4) for an example network. As highlighted through the example in Figure (5-26) solving a constraint network or highlighting conflicts is essential in collaborative systems to integrate different requirements and provide the resolution of problems early in the product development lifecycle.

The integrated collaborative environment as put forward in this research necessitates that the constraints and the problems they will describe are relatively unknown. The constraint networks are to be constructed from the constraints entered in the concept dialogue boxes in the concept maps. These concepts are capable of representing assembly hierarchies for parts or features of parts as shown in Figure (6-4). Thus a constraint network can be modelled and therefore solved from the concept map information.

In keeping with this derivation of a constraint network from a concept map, the constraints solving the examples as follows are based on information derived from a concept map.

5.8.2. Solving the Constraints Network

A constraints problem consists of the following as proposed in (Barták (1998)):

- A set of variables
- Domain values for each variable
- Set of constraints between each pair of variables

The solution to a constraints problem can be found by the generation of all the combinations of the design variables and then these are tested to check against the constraints. However this method is computationally intensive and exponentially more complex with each additional design variable and constraint that is added.

Successful solution of a constraint network involves a multitude of steps each affording gains in efficiency towards the discovery of a solution or the knowledge that no solution with the current set of design variables exists. In clearly defined problems with specifically known design variables it is possible to choose algorithms that are finely tuned to the resolution of the constraint network with optimised look forward methods predicting which branches of a search space should be avoided.

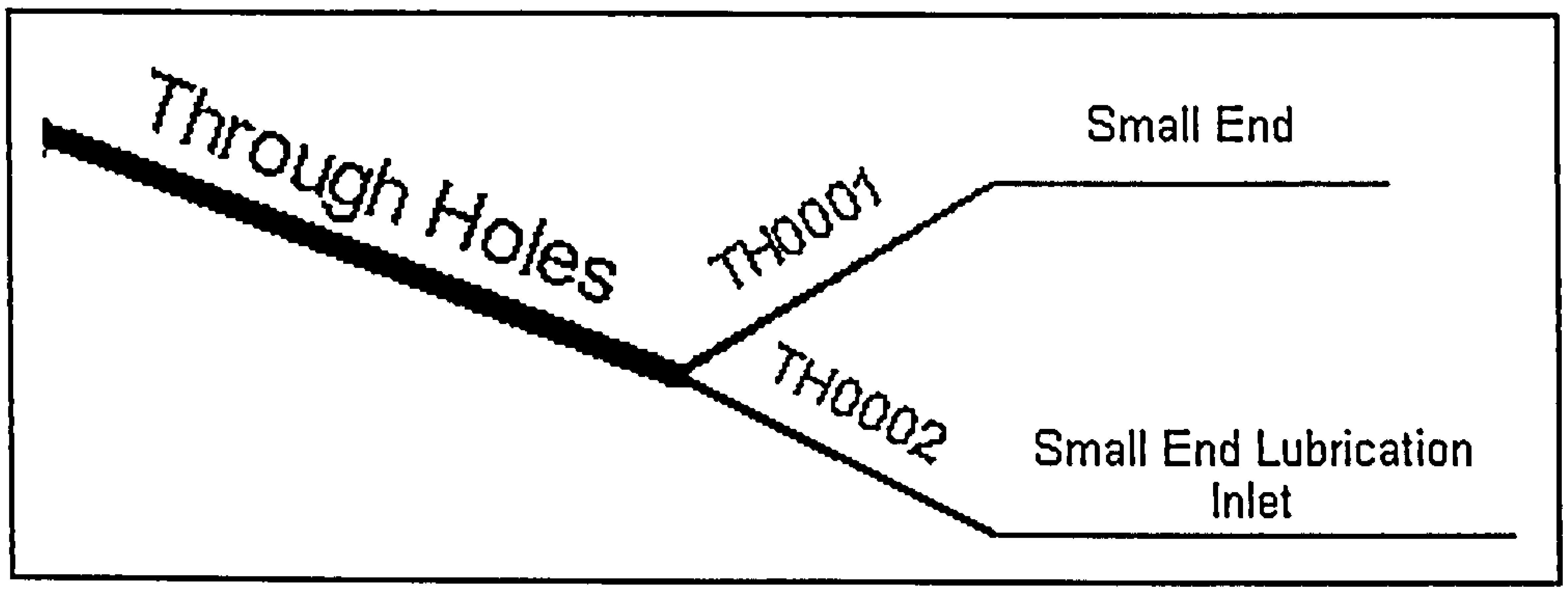


Figure (5-27) Features TH0001 and TH002

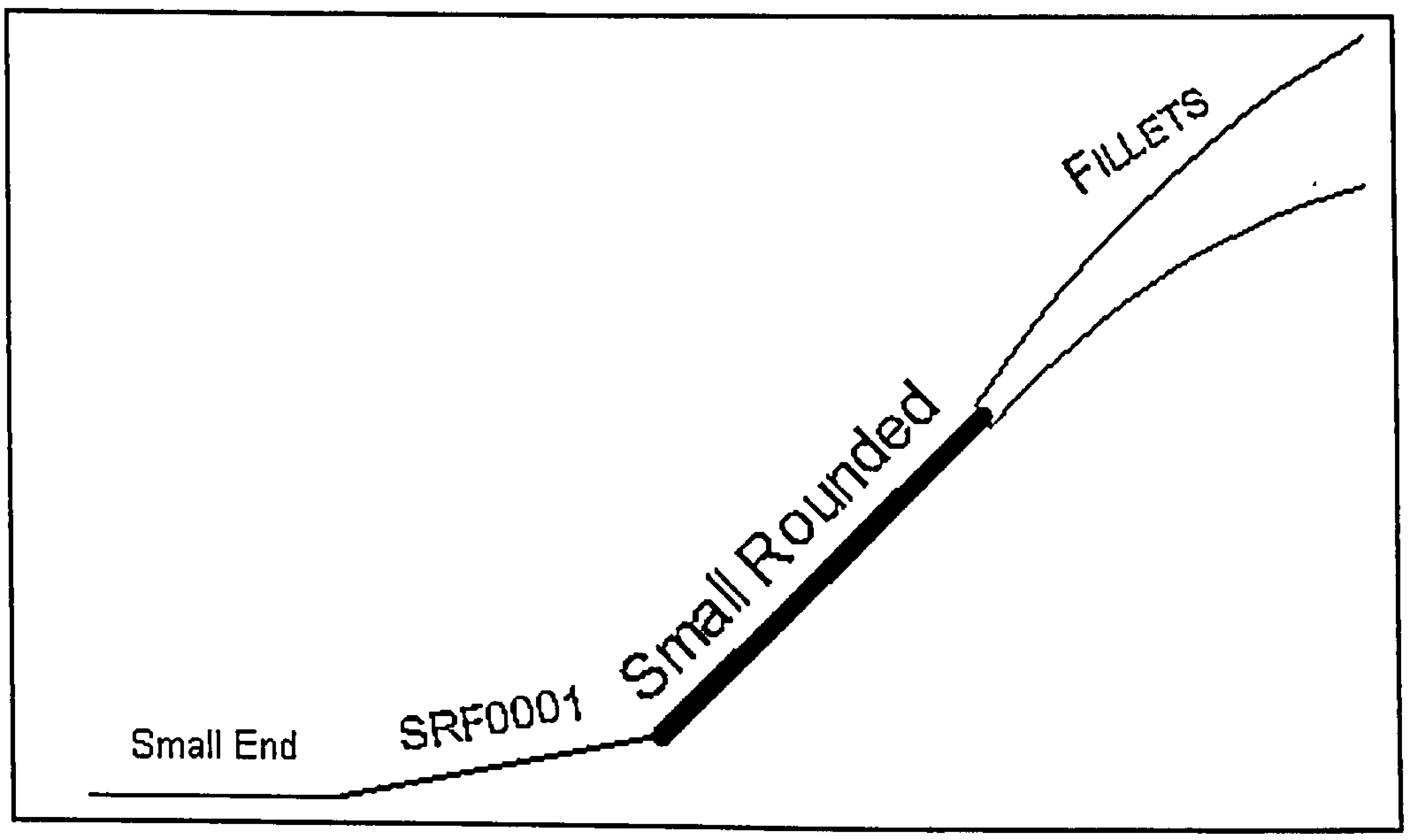


Figure (5-28) Feature SRF0001

In this example, several features have been extracted from the partial concept map in Figure (5-14) and further more precisely in Figure (5-27) and Figure (5-28).

SV = (TH0001, TH0002, SRF0001) is a set of design variables.
TH0001 = (0.000 ... X_n) is a set of positive values for TH0001.
TH0002 = (0.000 ... X_n) is a set of positive values for TH0002.
SRF0001 = (0.000 ... X_n) is a set of positive values for SRF0001.
DC = [(37 <= TH0001 <= 37.021), (4 <= TH0002 <= 4.070), (8 <= SRF0001 <= 8.500), ((TH0001 / TH0002) < 9.5), ((SRF0001 - TH0002) < 4)] is a domain of constraints between TH0001, TH0002 and SRF0001.

The overall methodology for solving a constraints network is to simplify the network through a series of mechanisms. This simplification results in a correspondingly reduced search space and thus faster computation.

The generate and test method for searching the potential solution space of the constraints network involves searching through a number of combinations equal to the Cartesian product of the design variables involved (TH0001 * TH0002 * SRF0001). Currently the design variables TH0001, TH0002, and SRF0001 are positive floating-point values, involving a huge potential search space limited in reality by the numerical accuracy of the system.

A unary constraint is one that involves a single design variable. Thus the first step in solving the constraints is to limit the potential search space involved. A practical mechanism is to reduce the domains of the variables to those dictated by the unary constraints. Thus the new status of the design variables after the removal of those values omitted by the unary constraints is as follows:

SV = (TH0001, TH0002, SRF0001) is a set of design variables.
TH0001 = (37 ... 37.021) is a set of values for TH0001.
TH0002 = (4 ... 4.070) is set of values for TH0002.
SRF0001 = (8 ... 8.500) is a set of positive values for SRF0001.
DC = [((TH0001 / TH0002) < 9.5), ((SRF0001 - TH0002) < 4)] is a set of constraints between TH0001, TH0002 and SRF0001.

The "generate and test" approach as it name suggests generates a combination of design variables and then tests the constraints with those variables to check for invalidity. In this relatively simple example the "generate and test" scenario would be capable of finding a solution to the network. However in more complex examples, even reducing the potential solution space in this manner, still results in large search spaces that grow exponentially with each additional design variable. Only partially generating a potential solution and then testing the applicable constraints can achieve an obvious gain in performance.

The backtracking algorithm takes this approach by generating the first design variable and then the second; the algorithm then tests these variables against the relevant constraints between them. If they satisfy the constraint, the next variable is added and consequently tested. This process continues until a variable does not satisfy a required constraint. Upon a failure the variable is changed to its next value and retested. If a domain for a variable is fully tested without satisfaction for the constraints then the algorithm backtracks to the previous variable instantiation and changes that to its next value and proceeds the testing once more.

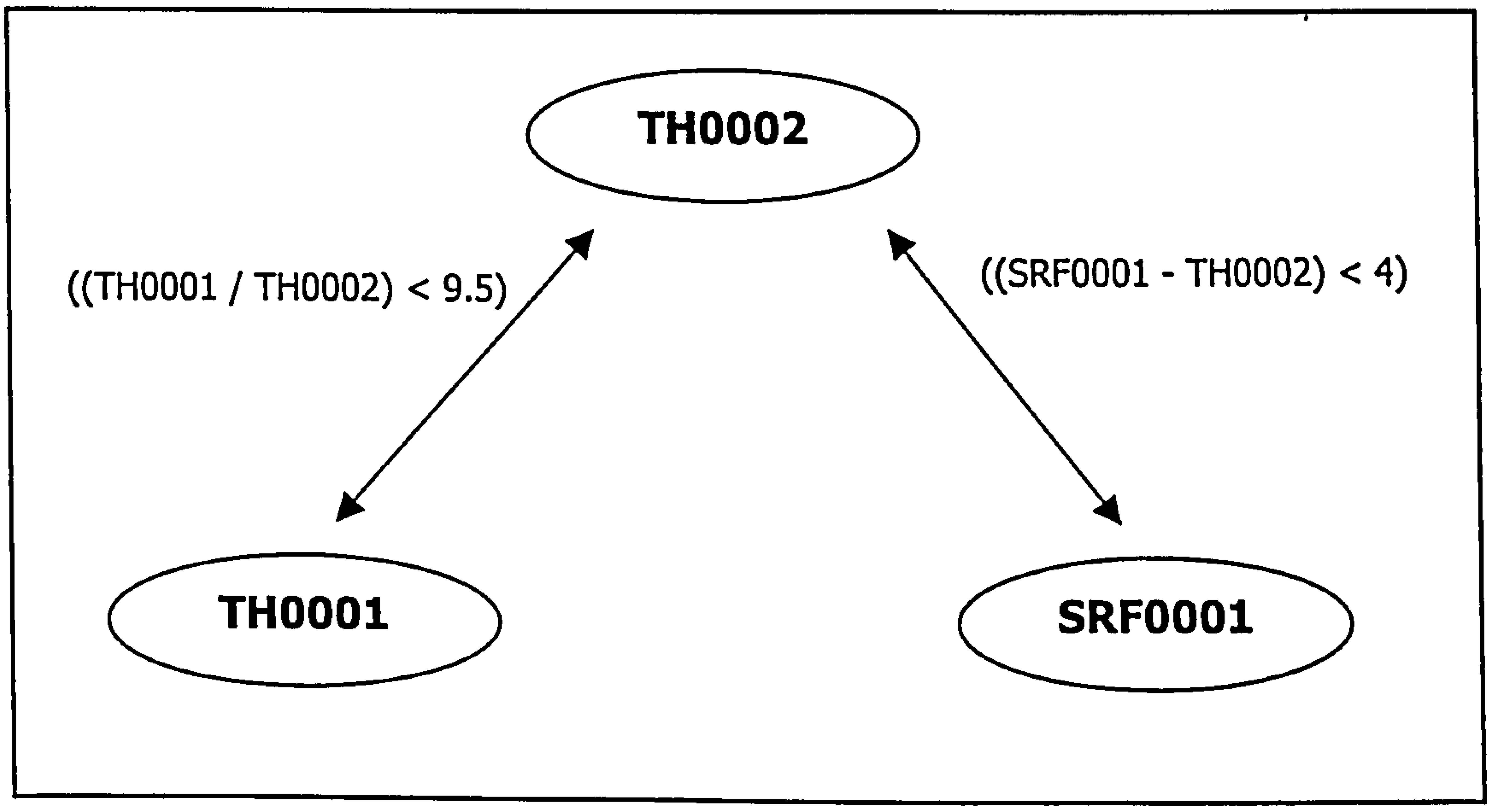


Figure (5-29) The Constraint Network after Unary Domain Reductions

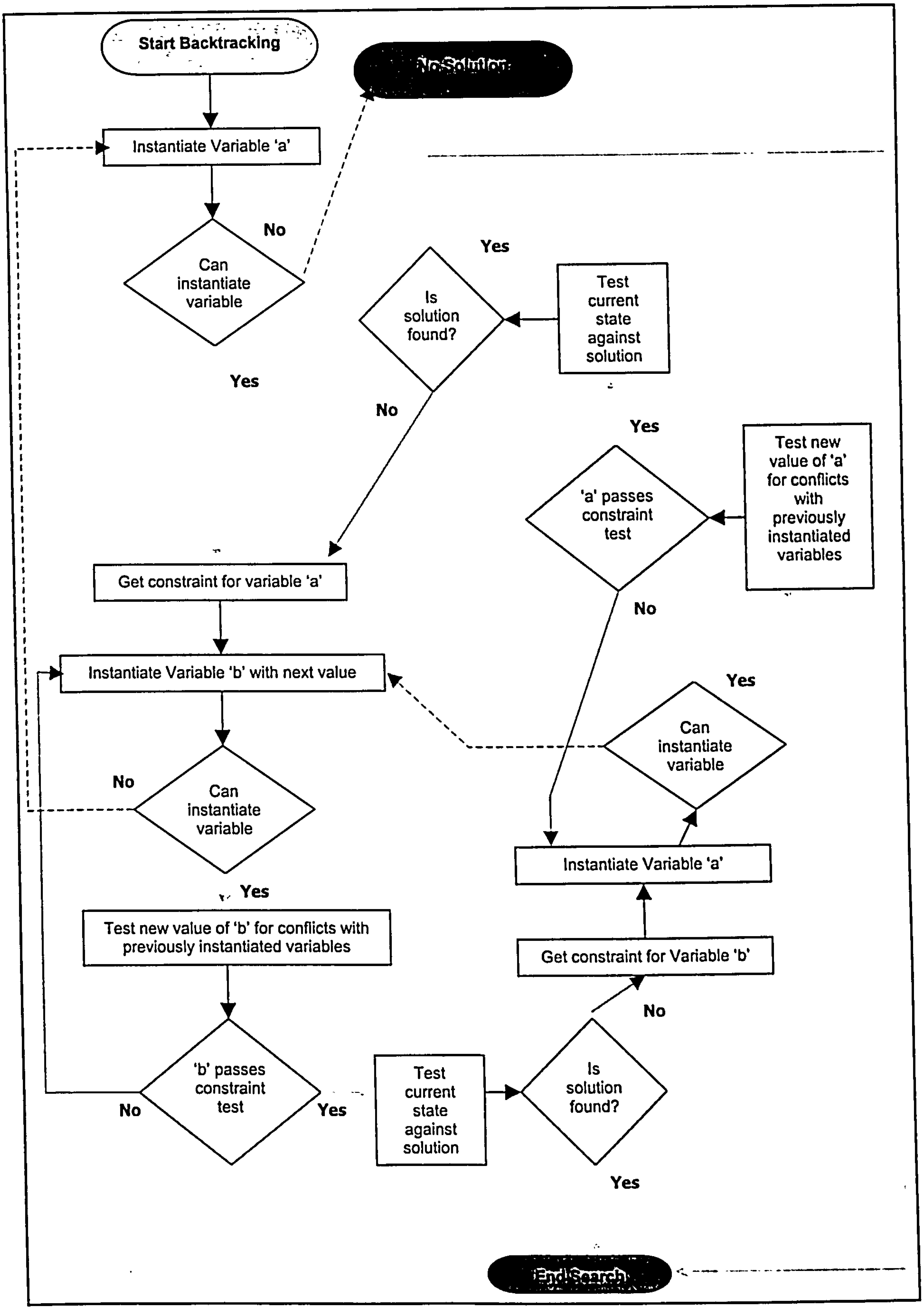


Figure (5-30) Backtracking Flowchart for Constraints Network Solving

Figure (5-30) show the backtracking method comparing variables that are instantiated in two temporary comparison slots, 'a' and 'b'. The flowchart shows that no solution can be found once all the variables have been exhausted and backtracked to the original value that in turn has no further values to explore.

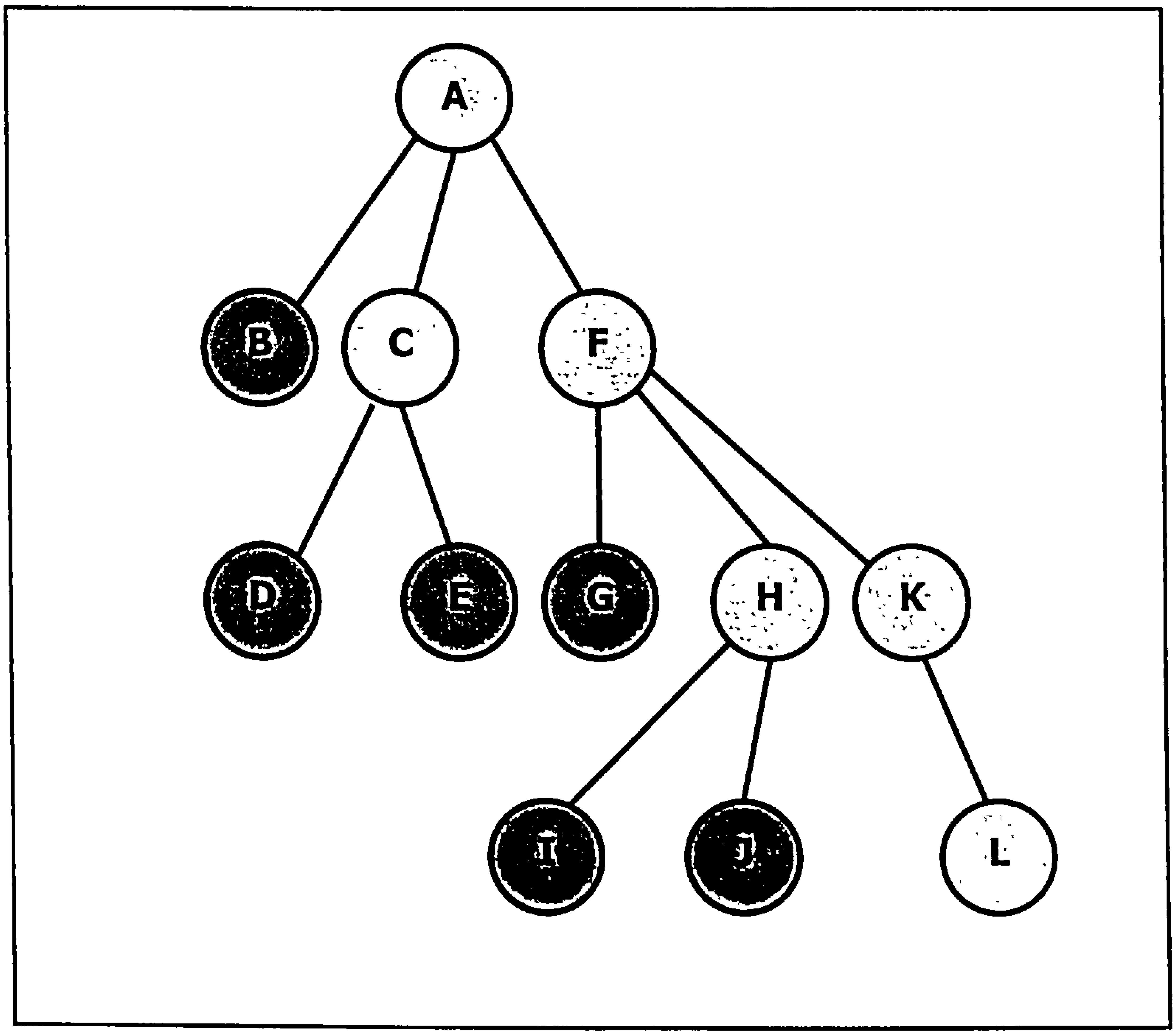


Figure (5-31) Backtracking Search Depth First Search Tree

Search State	Search Results
A	Check B, Check C, Check F
B	Fail
C	Check D, Check E, Fail
D	Fail
E	Fail
F	Check G, Check H, Check K
G	Fail
H	Check I, Check J, Fail
I	Fail
J	Fail
K	Check L
L	Success

Table (5-1) Search Results for Figure (5-31)

Figure (5-31) and Table (5-1) show an example depth first tree search sequence based that can be expected from the backtracking algorithm. Each node in the tree represents a unqiue state that is a set of the instantiated variables. States, B, D, E, G, I, J are all states where the state does not satisfy the required constraints and thus bracktracks to instantiate the next value for the variable. States C and H are also failed states as a result of all the child states returning as failed. State L represents a successful set of values to satify the constraints.

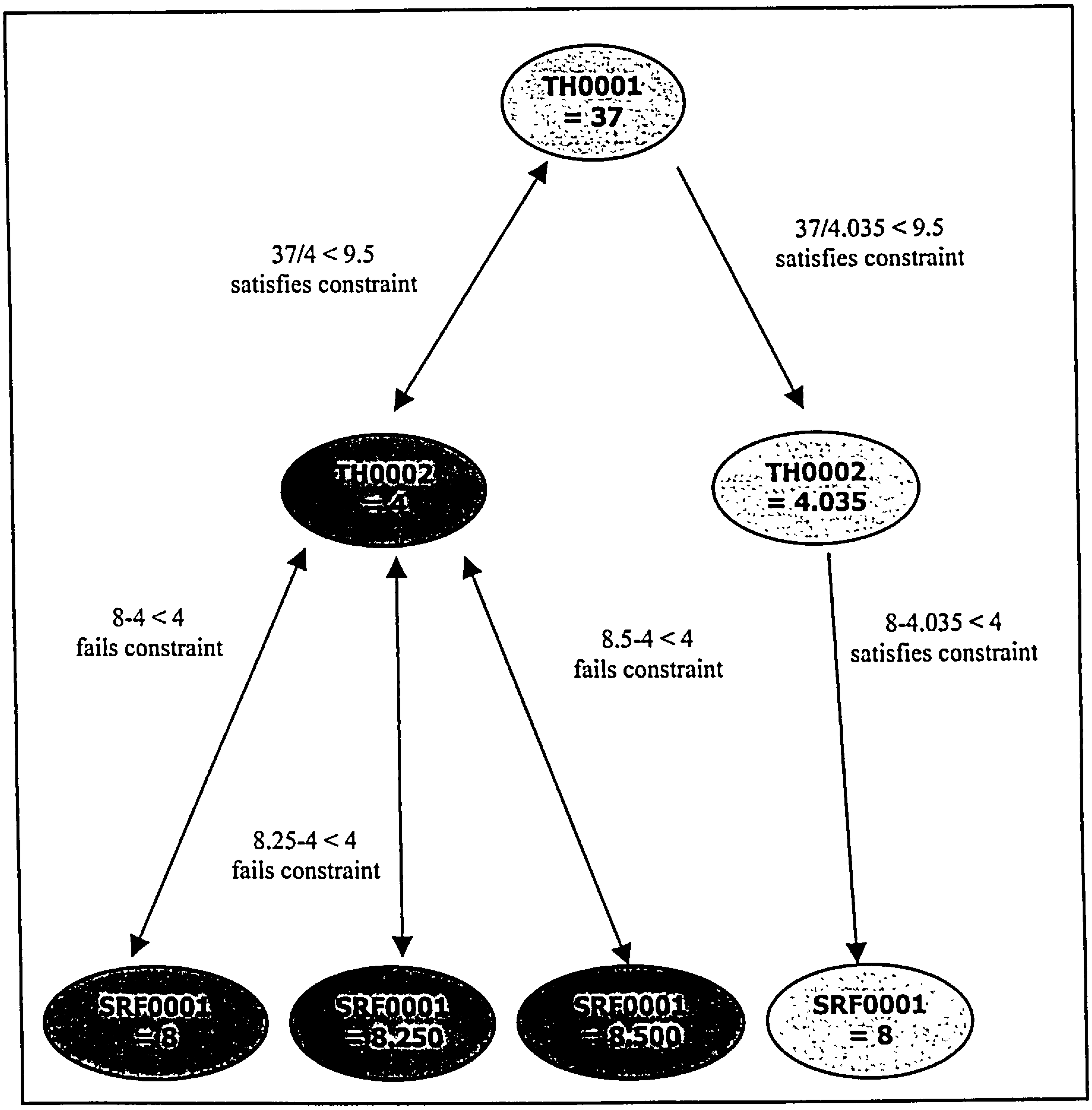


Figure (5-32) Search Tree for Constraint Network in Figure (5-29)

Figure (5-32) details the search tree for the constraint network in Figure (5-29) the constraint between TH0001 and TH0002 is not violated however for the constraint between SRF0001 and TH0002 to be satisfied the search backtracks and increments TH0002 to 4.035 and resets SRF0001 to the start value in its respective domain which provides satisfaction of all the constraints.

Backtracking is the basic mechanism for solving the constraints network and not the most effective however further optimisations of the search algorithm are outside the scope of this research.

5.9. HTTP Accessed Documents

These have been touched on in the other areas of the system, the HTTP accessed documents are the key to providing support for multiple and commonly unknown file formats. They are accessed in much the same manner as you would the ubiquitous web page through a URL.

A URL is not just a link to a web page or a picture; it can also be the construction of a database query provided this database is available through the Internet. This area also includes the submission of data through HTTP for further processing.

HTTP forms the foundation for access to other document types leaving the web browser technology to either display the file or to retrieve a plugin capable of extending the web browsers capabilities to display it.

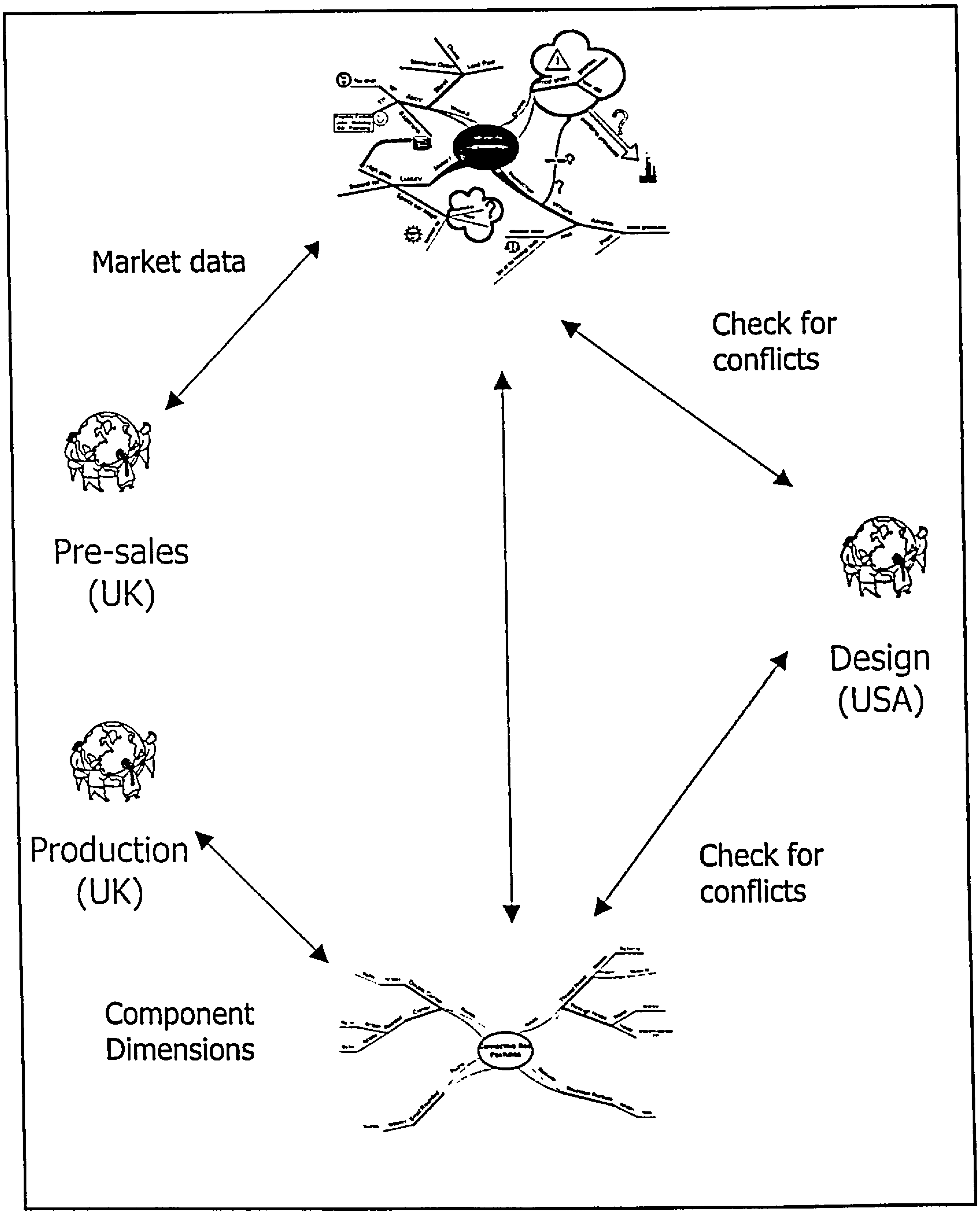


Figure (5-33) HTTP and Conflict Collaborations

5.10. Summary

The supportive elements to the core element of the concept mapping have been detailed within this chapter, demonstrating the importance of considering a wide range of aspects in the domain of collaboration. These aspects have been grouped within the categories; data sharing & structuring, data standardisation, object oriented database, CAD solid modelling, conflict resolution and HTTP accessed documents. The main point put forward in these sections were:

- The parsing of XML to provide support for multiple report types, natural language, BOM, and supplier lists;
- Linking of concept maps to themselves and external file types;
- External interfacing to other systems such as presentation packages, web calendars, workflow based project management tools, through the use of the concept map DTD as a formal definition of exchangeable data;
- The utilisation of standards to promote platform and application independence enhancing the longevity of system generated data;
- The STEP standard and how it can realistically be incorporated into a collaborative system;
- The representation of concepts within the object oriented database;
- The problems with CAD formats and how this can be overcome;
- How conflict resolution can be improved at differing levels of information complexity;
- How the mechanisms already in place on the WWW can be used to enhance an open system allowing extensibility in a variety of ways.

Chapter 6

CHAPTER 6. Validation of the Developed System

6.1. Overview

The purpose of this chapter is to validate and demonstrate the application of the system and its previously detailed aims and theories. This application of techniques details the representation of information from the initial development stages to the representation of complex fine-grained feature based data. This chapter brings together many of the wide-ranging information modelling procedures into a concrete example. It demonstrates the capability and potential for such a system.

6.2. Introduction

The system enables domain specialists to share information from the initial phases of the product development lifecycle providing continuous support for development throughout the subsequent phases. The key to this is providing a representation framework that is adaptable to ever increasing levels of detail.

The developed system was evaluated using products from the automotive sector. The initial level of concept mapping for a car is demonstrated. However due to the complexity of this product it is not feasible to represent it in its entirety thus a practical component has been selected for detailed investigation in this chapter. The detailed components of the car represented in this study were a piston and connecting rod assembly.

An example of how conceptual design information is made available to a distributed product development team and coherently represented in a suitable context is presented in this chapter.

6.3. Validation Objectives

The field of distributed collaboration comprises of a wide range of differing domains and facets. This validation concerns the foundation of the system; to what extent in terms of product development lifecycle phase and knowledge complexity can be represented and thus shared in collaboration by the developed environment.

There are other studies and validations that are future work for the system, for example the percentage costs saved through the resolution of conflicts. However these are areas require the capacity for of effective knowledge representation to be qualified first.

The objectives of the validation were to:

- Use the concept-mapping interface for collaboration to share knowledge for the ascertaining and clarification of the initial phase for development of a new automobile.
- Validate that the interface was capable of representing a high level conceptual scenario and could integrate multi-domain expertise in a single environment.
- Confirm that the developed concept mapping knowledge representation can facilitate detailed design knowledge representations as well as at the conceptual level.
- Verify that feature hierarchies concerning product development parts can be represented in the concept-mapping environment.
- Demonstrate the granularity and scope for conflict resolution of the environment through detailed storage of tolerances in the feature representation concepts.
- Validate the scalability potential for knowledge integration through multi-media linked VRML components.

6.4. Case Study Context

This case study is set in the domain of the automotive industry for the purpose of simulating the collaboration between the departments or companies of a large-scale organisation. A primary focus of this research project has been the integration of the product development lifecycle stages. Each lifecycle stage effectively is an increase in the complexity and detail of the shared knowledge. Thus this case study shows the representation of collaborative knowledge at increasing levels of complexity. To represent the entire rationale from initial concepts to component parts of an automobile is not feasible or necessary to explore this capability.

The case study starts with the initial investigation and examination of ideas for a new automobile. This then through design iterations results in individual concept maps for detailed facets of the development. In this study a piston and connecting rod assembly are the parts chosen for representation for their complexity of features. The connecting rod has been further explored in the case study and a feature hierarchy represented by the means of a collaborative concept map.

6.5. Information Sharing & Structuring Procedure

The procedure for the construction of a concept map is detailed in the Figure (6-1) and Figure (4-4) (see also section 4.5). The main concept defines the scope of the current concept map and thus is added first.

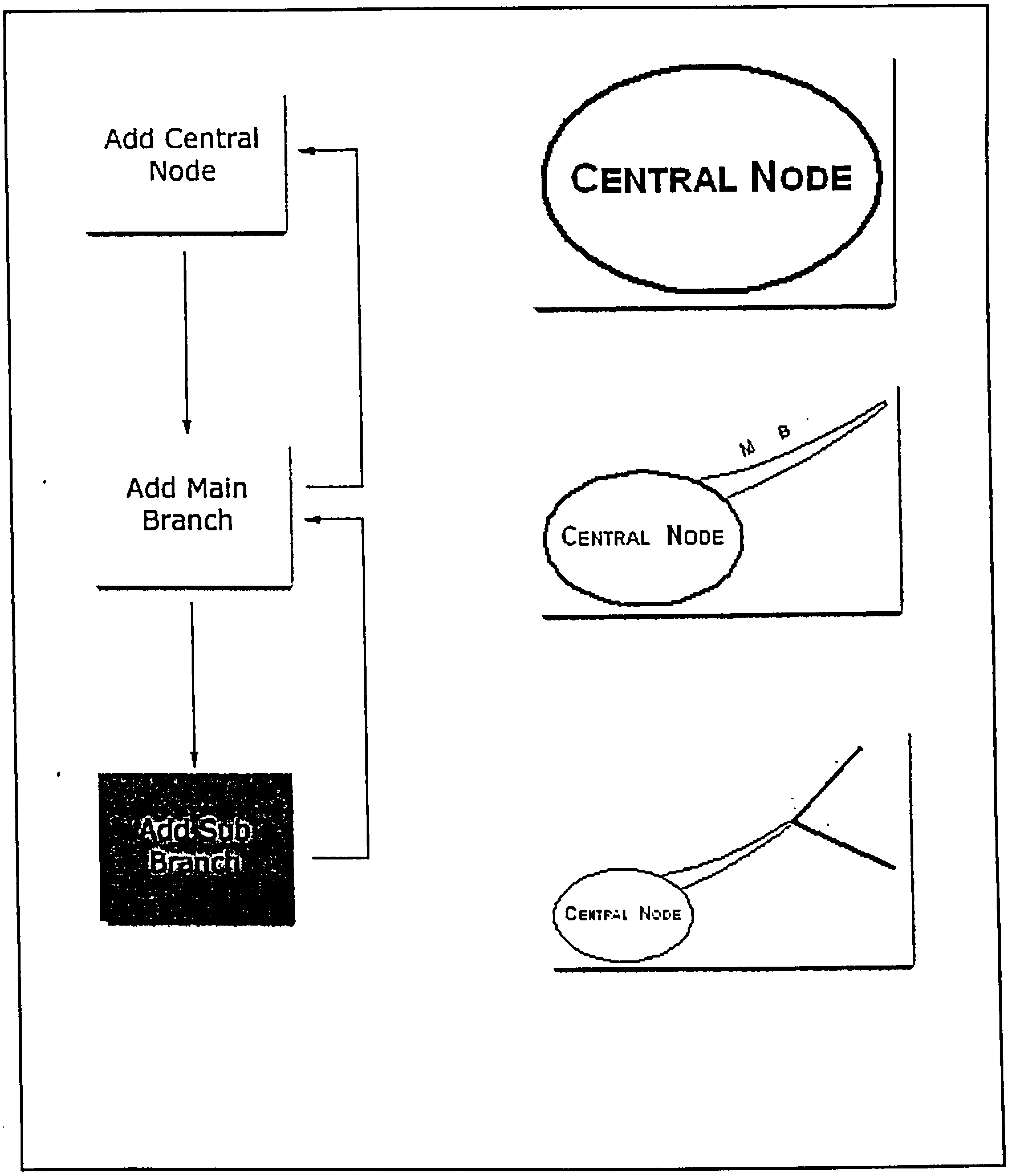


Figure (6-1) Information Structuring

This is followed by a number of main branches that define the general areas of the topic for discussion followed by the sub branches that provide the details of the concept map.

Sub branches are added recursively until the configuration limits of the map are reached. Each node in the map provides a combined series of dialogues structuring a collection of information concerning:

1. Concept Details
2. Description
3. Linked Files
4. Constraints
5. Features

The 'concept details' concern its represented name displayed in the map, contact points and the creation date along with the node's style. 'Description' contains a textual report of any comments concerning the concept. 'Linked files' provide a list of URL's, which are relevant to the concept and can be retrieved by clicking on them. 'Constraints' can be added, which form the basis for restrictions on other concepts in the hierarchy, and result in design variables that can in turn provide the basis for conversion to a STEP AP203 file. 'Features' refers to a list of features related to a specific component or a product. This list is linked in the same manner as the list of files and provides short cuts to concepts, which are related to sub assemblies or final form features.

6.5.1. Information Sharing Level 1

The concept-mapping module allows users of appropriate authority to register for a concept mapping based discussion. Figure (6-2) shows three team members using an integrated chat forum along side a concept map. The interface is based on Java Swing technology, which allows a greater range of interface components to be used. The toolbar docked on the right hand side can be removed and placed, for example, at the top or outside the browser. If the focus of the group is on the concept map, the panels drag bars that can be seen allows extra space as appropriate.

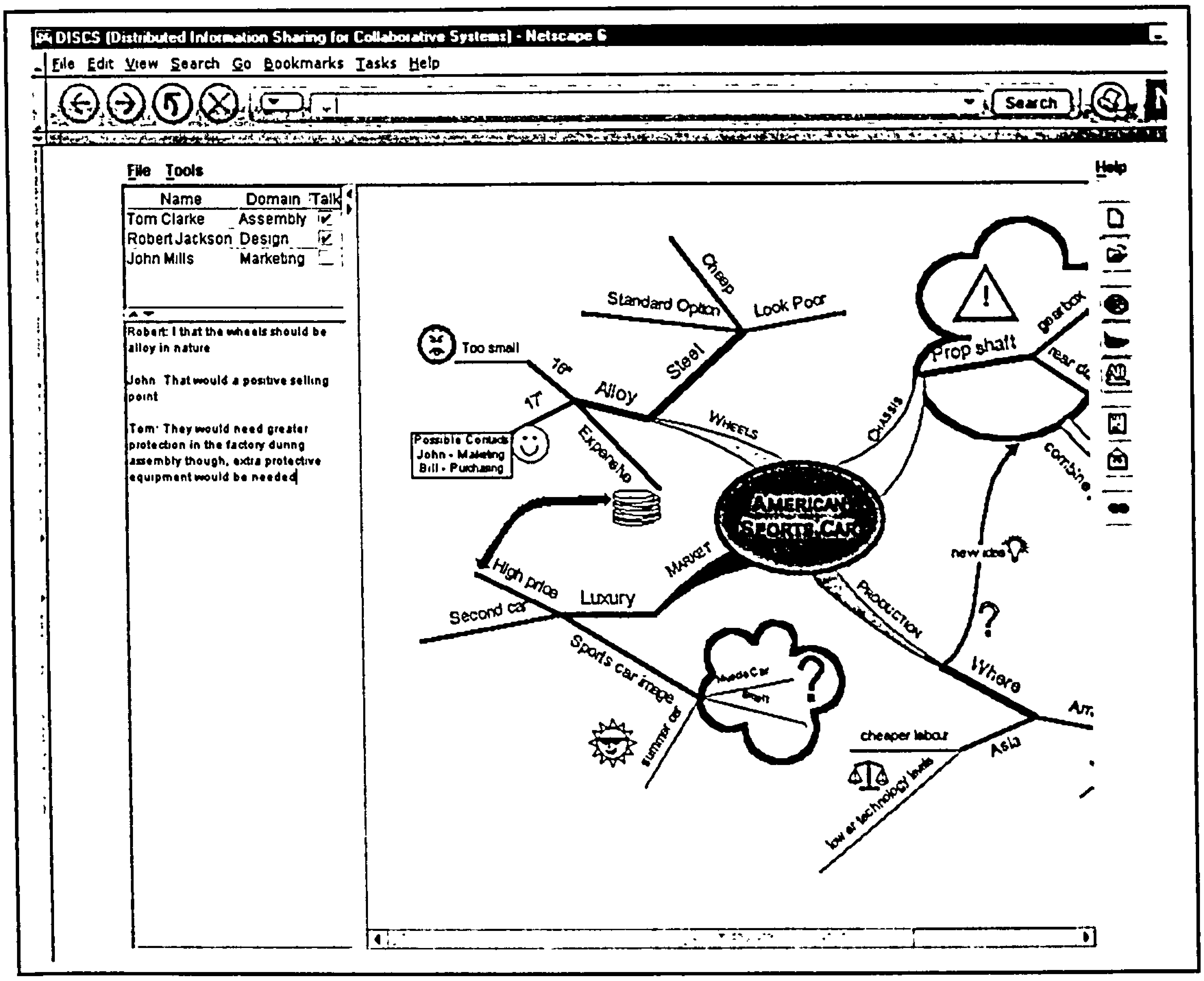


Figure (6-2) Web-based Concept Mapping Interface

6.5.2. Information Sharing Level 2

The piston and connecting rod assembly concept map in Figure (6-3) is the parent to the connecting rod features concept map shown in Figure (6-4). The map in Figure (6-3) describes the components for the complete piston and connecting rod assembly; only the sub branches for the connection rod are shown in this case. The circled branch in Figure (6-3) contains the link with which the child concept map is retrieved from the current context. The concept map in Figure (6-4) represents a feature-based hierarchy. Each concept in Figure (6-4) represents a feature that is in turn linked to a dialogue.

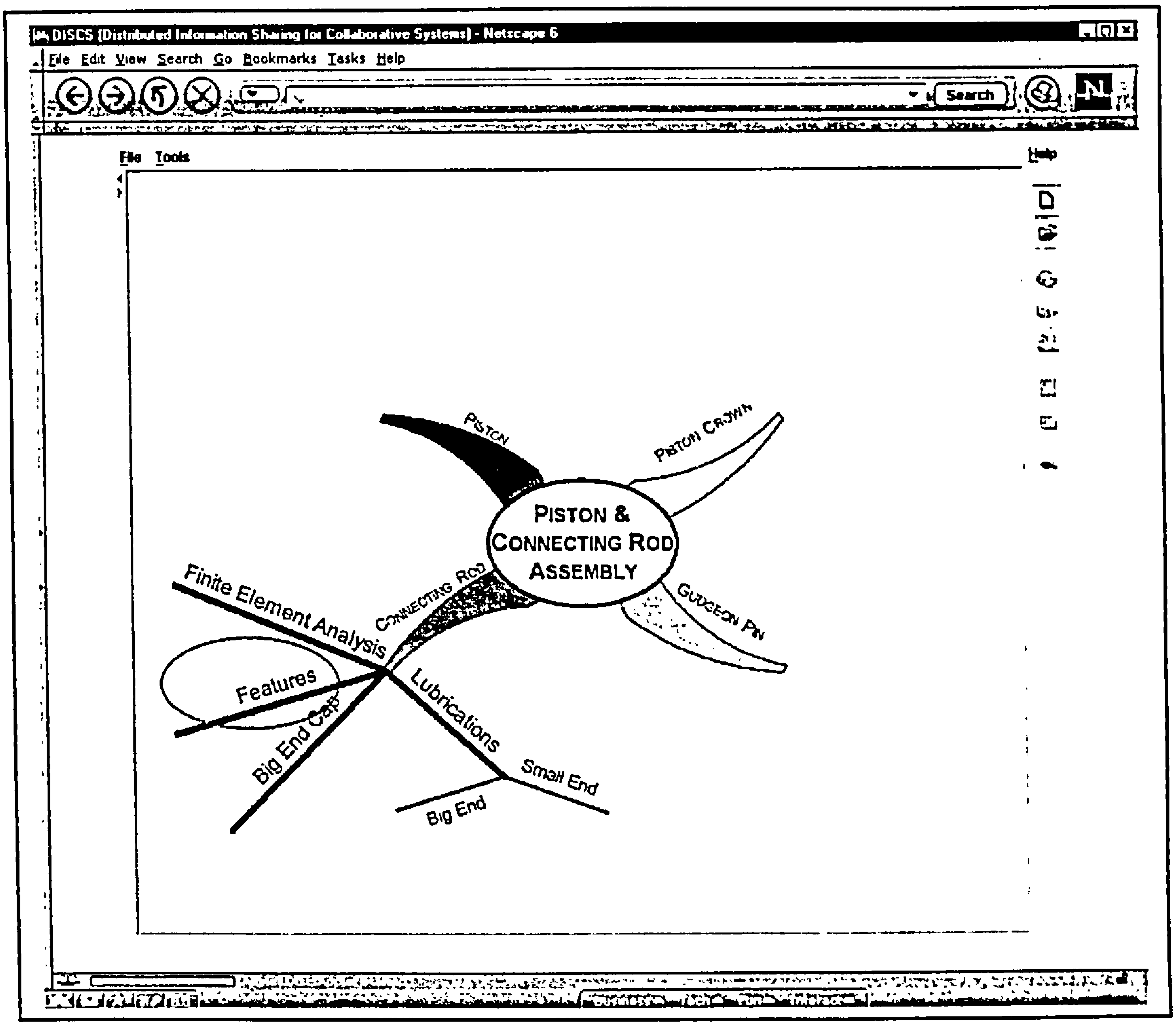


Figure (6-3) Parent Map for 'Connecting Rod Features

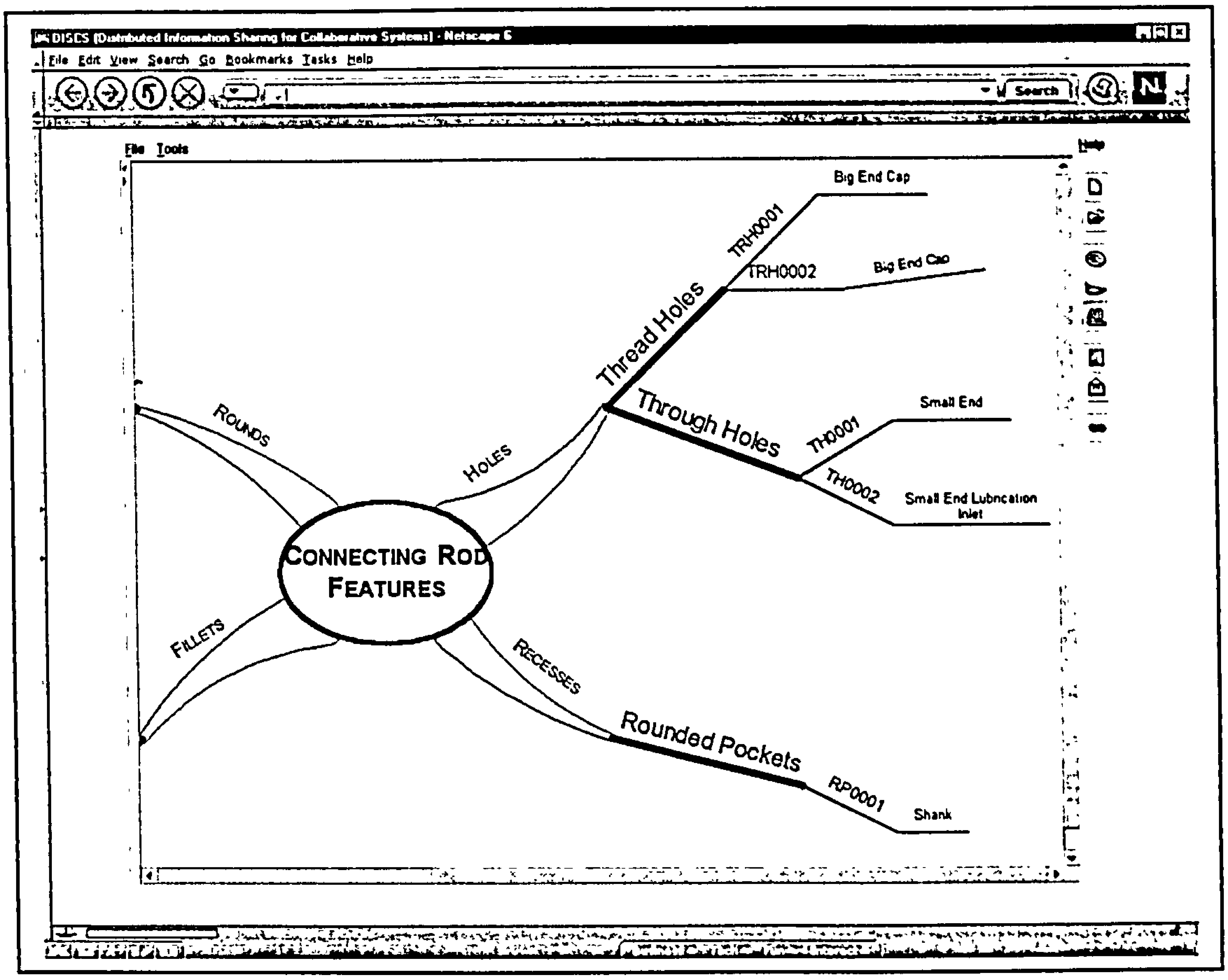


Figure (6-4) Child Map for 'Piston and Connecting Rod Assembly'

6.5.3. Information Sharing Level 3

The capability of the system is shown through the conceptual communication and storage of product data concerning a connecting rod. The component consists of a variety of features, holes, rounds, fillets, and pockets. These features describe the small end, shank, big end and big end cap of the connecting rod. A selected representation of these features is shown in Table (6-1) to Table (6-4).

Feature Name	Dimensions (mm)	Feature Code	Tolerance (mm)	Surface Finish (µm)
Through Hole	Diameter: 37 Depth: 26.27	TH0001	+ 0.021 - 0.000	0.6
Through Hole	Diameter: 4 Depth: 2.5	TH0002	+0.070 - 0.000	0.8
Fillet (Small Rounded)	Radius: 8	SRF0001	+0.500 - 0.000	6.3

Table (6-1) Small End

Feature Name	Dimensions (mm)	Feature Code	TOLERANCE (MM)	SURFACE FINISH (µM)
Rounded Pocket	Diameter: 12 Radius: 20	RP0001	+0.500 -0.000	6.3

Table (6-2) Shank

Feature Name	Dimensions (mm)	Feature Code	Tolerance (mm)	Surface Finish (µm)
Double Corner	Diameter: 62	DC0001	+ 0.190 - 0.000	0.6
Rounded Corner	Diameter: 80	RC0001		6.3
Rounded Corner	Diameter: 80	RC0001		6.3

Table (6-3) Big End

Feature Name	Dimensions (mm)	Feature Code	Tolerance (mm)	Surface Finish (µm)
Double Corner	Diameter: 62	DC0002	+ 0.190 - 0.000	0.6
Thread Hole	Diameter: 19 Depth: 42	TRH0001	+0.01 - 0.000	0.1
Thread Hole	Diameter: 19 Depth: 42	TRH0002	+0.01 - 0.000	0.1

Table (6-4) Big End Cap

Dialogues to facilitate standardised collaboration are associated with each node in a concept map. Depending on the context of the node various input fields are completed. The following series of dialogues (Figure (6-5) through Figure (6-8)) represent the concept detail stored in the node describing the attributes of the "Through Hole: TH0001" as seen previously in Table (6-1). The features dialogue is not required as TH0001 is a bottom level concept; there are no further sub features to include.

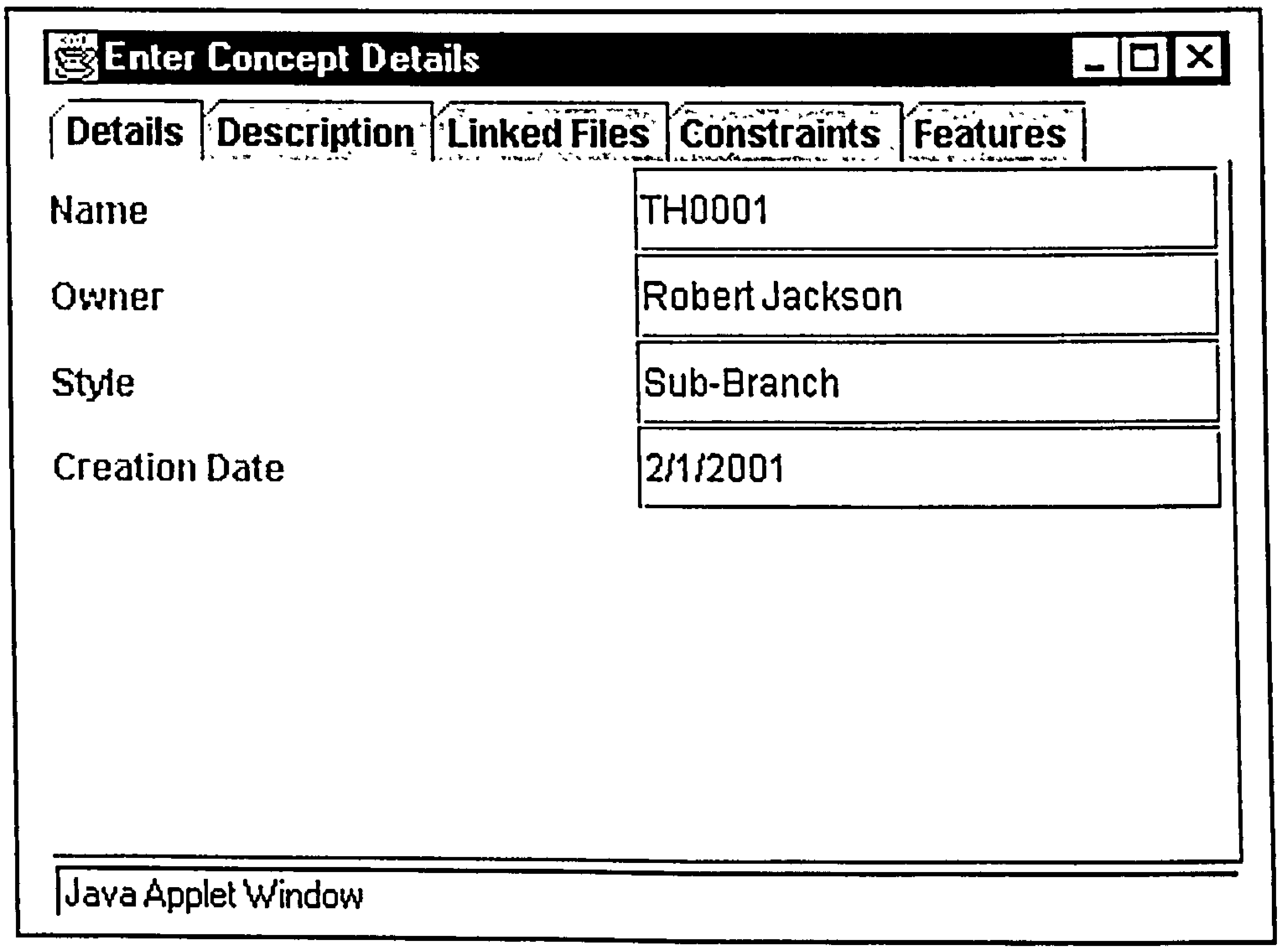


Figure (6-5) TH0001 Details

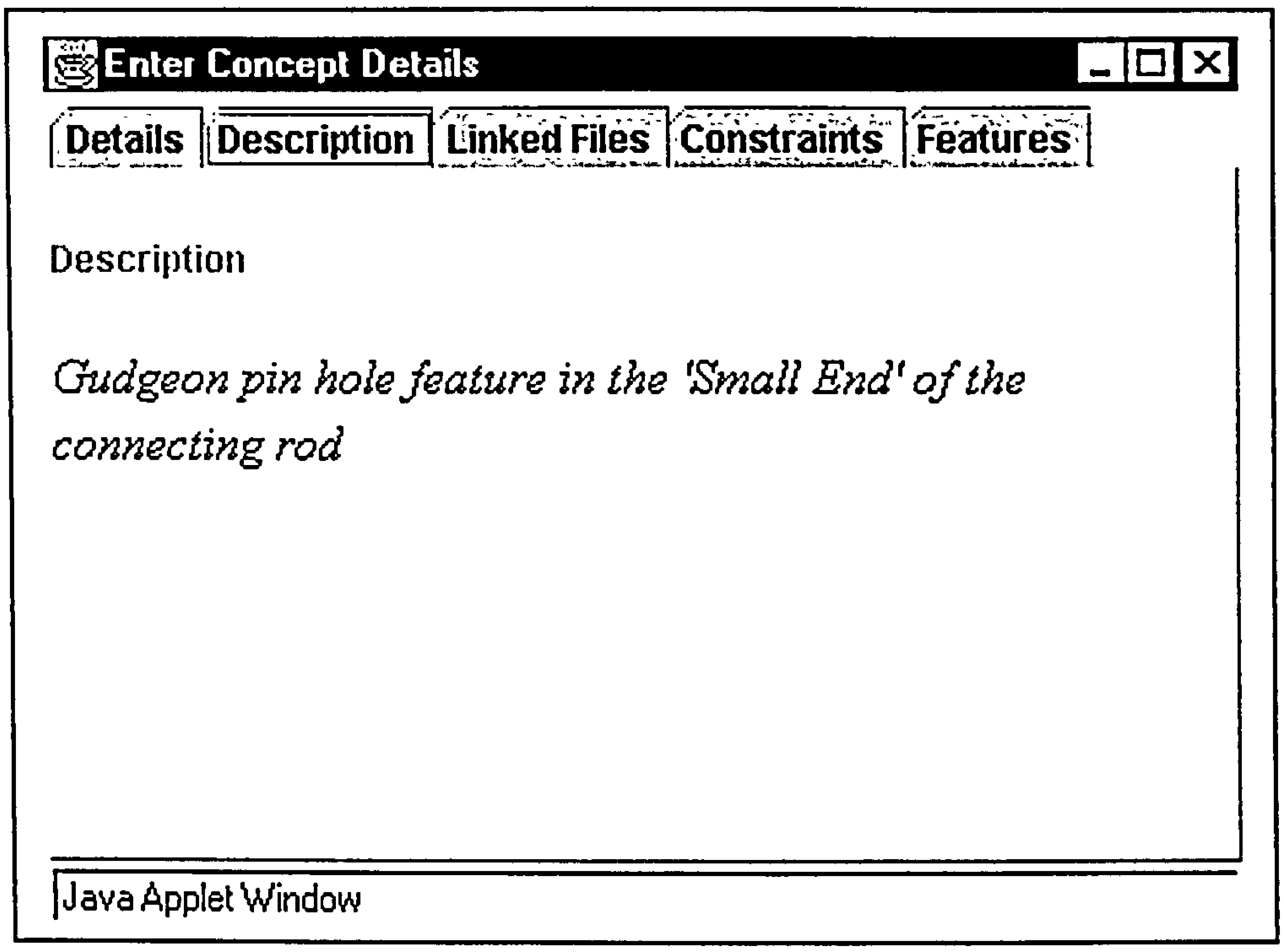


Figure (6-6) TH0001 Description

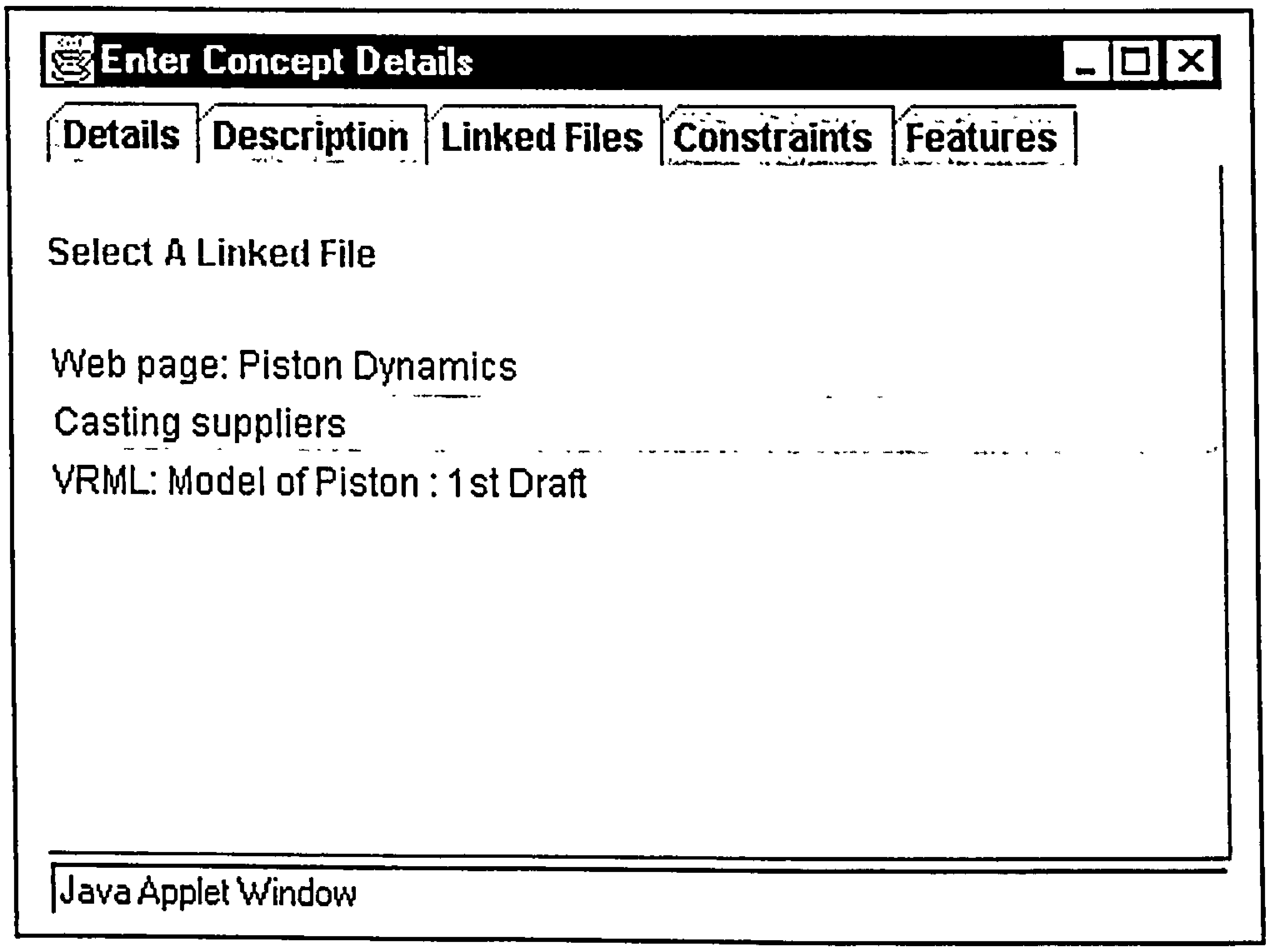


Figure (6-7) TH0001 Linked Files

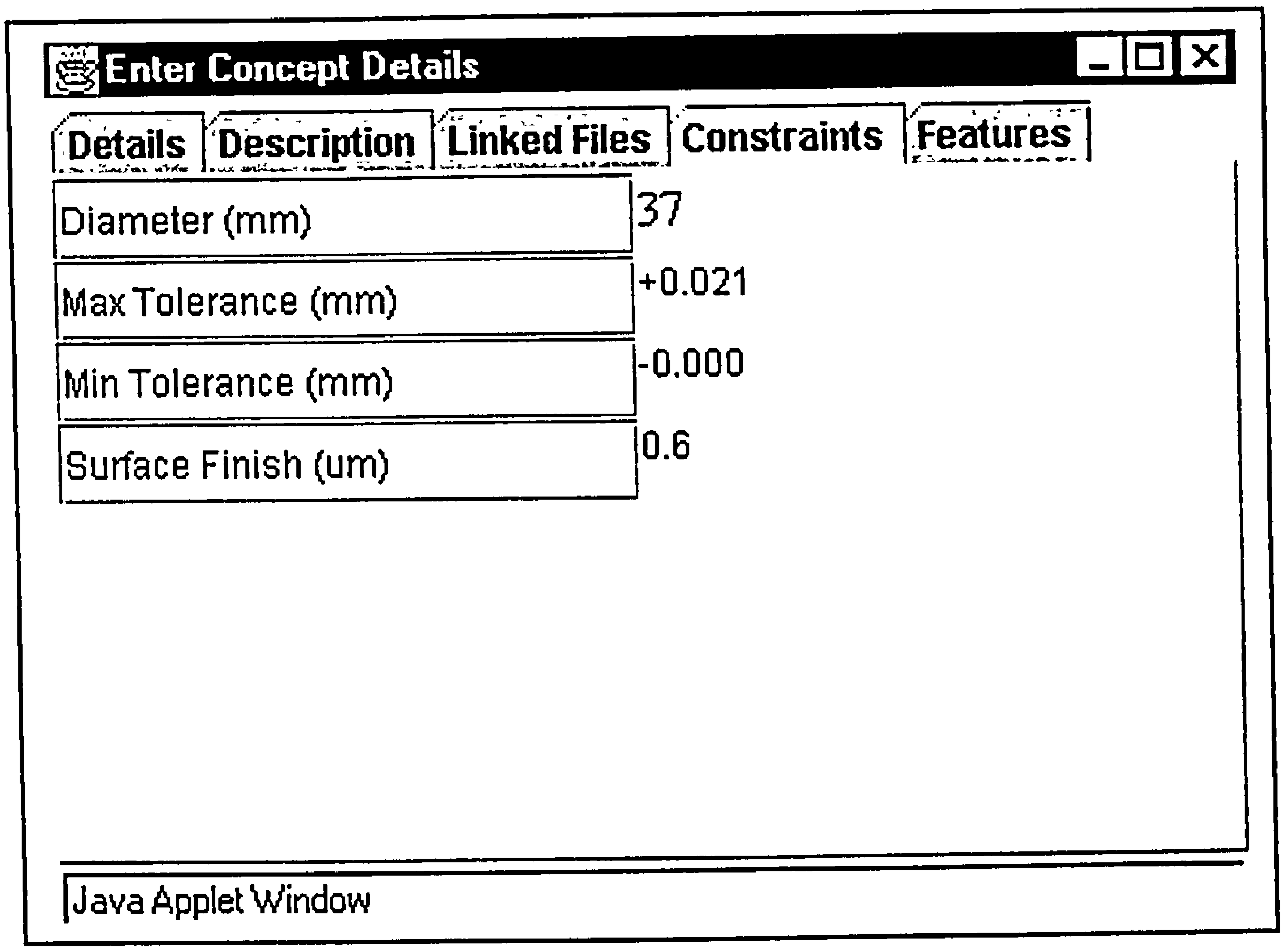


Figure (6-8) TH0001 Constraints

6.6. Results from the System

This case study has validated the approach to collaborative knowledge sharing undertaken in this research project. The integration of people, supply chain, lifecycle phases, and technology has been demonstrated.

The case study has validated the following features of this research project:

- The approach taken for knowledge sharing is suitable for the intended setting of a large organisation.
- The developed concept mapping is suitable for the representation of high-level initial conceptual stage knowledge representation.
- Detailed design iterations can be represented through the same concept mapping system facilitating the consistency of knowledge representation.

- Feature hierarchies can be embodied in the concept mapping representation. This facilitates tolerances to be set for a concept with specific design variables that in turn provides a platform for conflict resolution.
- The fine granularity of the knowledge shared demonstrates the potential for effective implementation of the Concurrent Engineering philosophy.
- The integration of cognitive psychology feature of cognitive mapping is shown; the knowledge shared is always in a context adding vital semantics information.
- The scalability of the knowledge sharing was shown through the HTTP linked VRML models and the linking to external web pages concerning piston dynamics.
- The linking of the concept maps of the increasing levels of knowledge complexity demonstrates the integration of product development lifecycle phases.
- Asynchronous and synchronous collaborations are facilitated; the concept-mapping environment provides facility for real time and serial knowledge input.
- The foundation for appropriate supply chain integration is demonstrated through the linking of a feature to a supplier's web site.

The resulting knowledge representation from the system represents a consensus, across the supply chain, on the details concerning the recommended component. The output illustrated, is viewable in a variety of forms, selecting from multi-media displays or focussed upon a single represented format such as a concept map.

Effective collaboration is facilitated through sharing of a linked framework of informative multimedia components. A complete multimedia perspective with concept map, concept details and a VRML models can be seen in Figure (6-9).

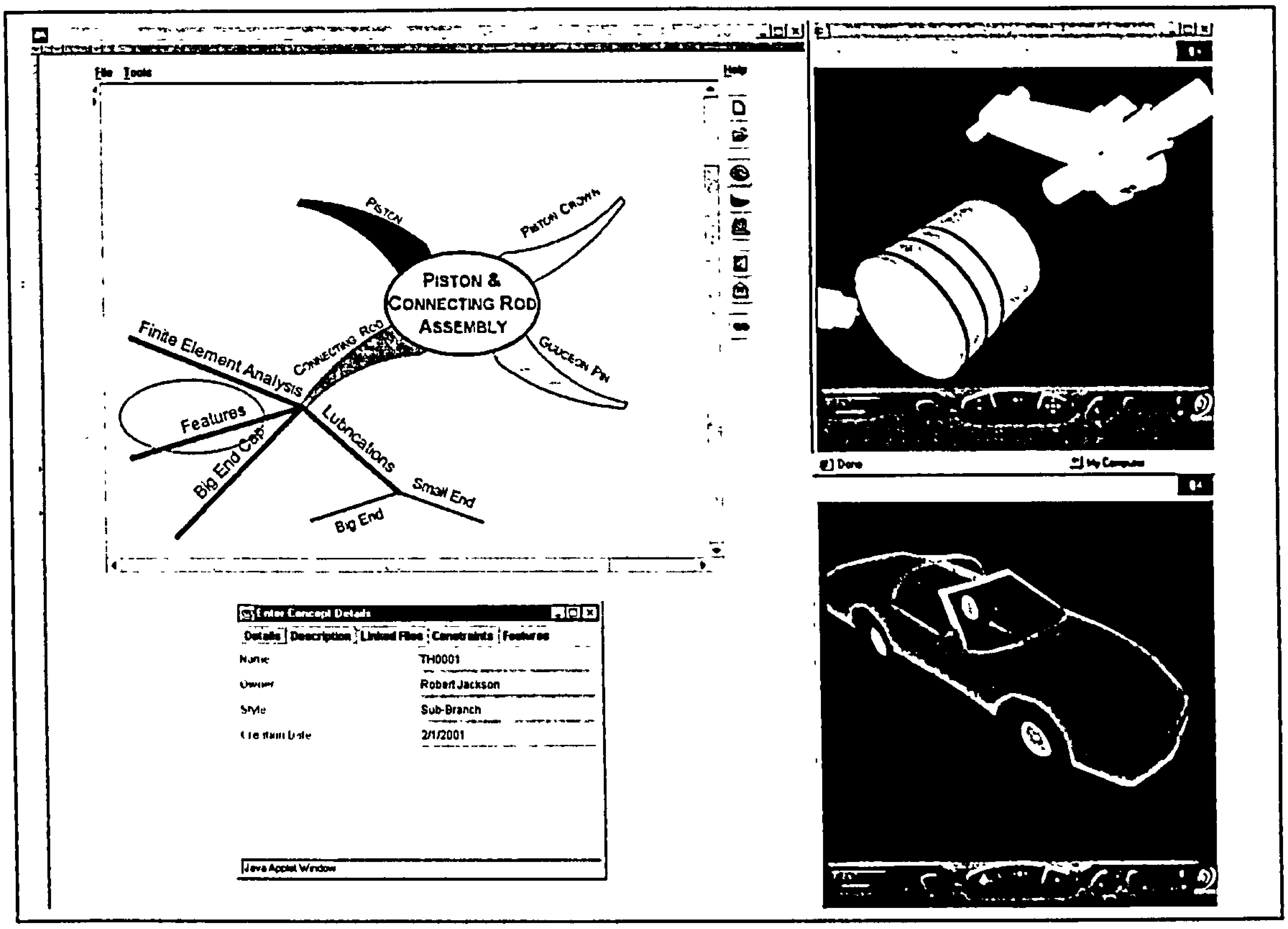


Figure (6-9) Snapshot of the System Results

6.7. Summary

The distributed collaborative environment has been demonstrated through a case study based on the automotive industry. The case study has demonstrated many of the knowledge sharing principles and developments of this research project. It has shown the innovation in the development of a common knowledge representation that has the capacity to bridge complexity and product development lifecycle phases. The approach taken has been successful in the integration of knowledge from multiple domains.

The case study has shown that the constructive elements from the fields of Concurrent Engineering, Cognitive Psychology, and Software Engineering have been integrated. The knowledge sharing system in the context of an automobile has demonstrated that the collaboration can be facilitated from the initial conceptual level and maintained in the same collaboration environment throughout successive product development phases.

Holistic conceptual phase collaborations for the early phases of the development of a new automobile product are made possible. This facilitates the Concurrent Engineering philosophy of integrating the product development team at the earliest lifecycle phase.

The facilitation of Concurrent Engineering through the reduction of problems and inconsistencies that occur later in the product development lifecycle in turn promotes the following benefits:

- Reduced overall product development costs.
- Faster time from product conception to delivery resulting in a reduced time to market.
- The improvement of product quality.
- The early resolution of potential conflicts provides opportunity for optimum design paths rather than ad-hoc modifications at a later stage.
- Early involvement of suppliers facilitates the consistency between the design and the process and product capabilities of the suppliers.
- The improvement of the products applicability for its intended market.
- Holistic perspective of the product development knowledge shared in turn facilitates further innovations.

The system has shown an innovative approach to collaborative product development addressing the facilitation of Concurrent Engineering from a holistic and integrated approach. The case study has shown that the concept mapping approach advances the knowledge representation capacity for the collaborative environment to integrate the following:

- Distributed team members.
- Differing knowledge domain specialisations.
- Product development lifecycle phases.
- The supply chain.

- Asynchronous and synchronous communications.

This chapter has through the undertaking of a case study based on the collaboration for a new automobile and a specific case of further detailed knowledge sharing and representation has covered the following aspects of the system validation:

- Objectives of validation.
- Automotive context for collaboration case study.
- Procedure for knowledge sharing.
- Conceptual information sharing for the initial design phase of a new automobile.
- Detailed design representation utilising the shared knowledge representation of concept mapping.
- Feature based concept mapping knowledge representation
- The successful results of the validation.

Chapter 7

CHAPTER 7. Overall Conclusions

The research work described in this thesis contributes to the area of collaborative product development by addressing the integration of people, product development processes and technology throughout the product development lifecycle.

The globalisation of competition has led to greater increases in product rivalry than ever before. Major factors in this globally competitive market are cost, functionality, and quality. To enable a product to compete on these facets of competition necessitates the collaboration of the entire product development team from the earliest possible stage of the product development lifecycle. Early distributed collaboration in the context of product development is essential in the identification of potential problems and design errors before they consume a significant investment of resources.

The developed collaborative environment incorporated the following novel features:

- The development of a conceptual knowledge system termed concept mapping.
- The expansion of concept mapping to form the knowledge framework for the integrating of the entire product development lifecycle in a holistic manner from product concept through design and product to disposal.
- The advanced integration of the supply chain at the earliest lifecycle phase.
- The facilitation of collaboration for a large number of differing domain specialists.
- The successful capability for the representation of the potential diversity of shared knowledge required in product development.
- The facility to be scalable to enable the facilitation for the unknown knowledge formats or future processing demands.

- The integration of both asynchronous and synchronous communications without inserting further new barriers to information sharing.

The literature review undertaken in Chapter 2 indicated that the current state of collaborative product development environments do not provide sufficient integration levels of people, processes and technology to harness the potential benefits obtainable from distributed collaboration. The area of distributed collaborative product development needs to be addressed from a broader perspective than previous undertakings have considered. Existing collaborative environments are limited in terms of consideration for the entire product development lifecycle. The research presented in this thesis addresses this lack of holistic integration and product development lifecycle consideration.

The developed work facilitates collaborative product development processes from the conceptual phase throughout the entire lifecycle. It provides a holistic view of the collaborative input for the product throughout development integrating the product development stages, the product development team, and data formats, overcoming both geographic and temporal barriers. The developed work enables consensus to be reached amongst distributed product development parties. It provides collaborating teams with an environment to contribute their ideas and perspectives and view those of other team members.

The developed shared concept mapping provides a facility for all the involved parties, allowing each to contribute from the earliest possible lifecycle phase. This provides a holistic viewpoint of all the elements involved in the product development process enabling a resolution of conflicts based on constraints upon requirements. The concept maps provide a scalable collaborative forum for the management of design contributions, facilitating the representation of collaborative interactions from the early conceptual stage through to detailed individual features of assemblies in the later development stages. This hierarchical data management framework provides facility for collaboration efforts to include other data formats as linked files to nodes within the concept maps.

This research has collated and defined a comprehensive set of requirements for distributed collaborative information sharing. It is upon these requirements that the developed system has been based. These requirements necessitate that collaboration needs to be facilitated at the earliest possible product development phase; facilitated in this work through the development of a shared integrated concept mapping environment.

This research project has addressed the following requirements for a successful collaborative system:

- The integration of the entire product development lifecycle.
- The improvement of the communication of domain knowledge between team members.
- The increase of a common holistic conceptual & semantic understanding.
- The facilitation of context linked support for multi-media information components, concept frameworks, CAD models, images, documents, video, and web pages.
- The provision of technical access through the consideration of portability in tackling multi-platform incompatibilities.
- The use of open standards.
- The facilitation for natural communication.
- The use of a database system capable of storing the required information.
- The importance of finding a common time slot for the meeting has been reduced.
- The necessity for travelling to a meeting has been minimised.
- The identification of the people involved in the problem solving process.
- Multipoint conferences, conferences between more than two people.

The integration of lifecycle phases necessitates the need for a contiguous information representation and management environment capable of representing not only the core aspects of the phase but also a range of unknown file formats.

The integration of people requires a collaboration environment that is consistent and integrated. It is important for collaborations to take place within a homogenous holistic setting that addresses temporal and geographic barriers as opposed to a range of separate systems. Integration of globally distributed people requires a single collaboration environment, sufficiently portable to be accessed from a wide range of the worlds computer systems and able to overcome the barriers that differences in time zones present. These temporal barriers require both synchronous real time and asynchronous sequential collaborations. Current systems deal with these problems through disparate incompatible technologies thus introducing further inconsistencies. This research work has presented a holistic collaboration environment that bridges the temporal barriers within an integrated concept mapping system.

Developing a computer-based environment suitable for globally distributed product development organisations to collaborate was a major purpose of this research. The system was tested with collaborative data from various stages in the product development lifecycle. The result of this work has shown that current systems and development paths due to narrow considerations have a limited potential long-term benefit and that a fully integrated solution is not only plausible but with appropriate resources achievable within the product development setting.

To provide effective access and the reduction of barriers to collaboration a technical solution has been presented that is portable, scalable, and adaptable. The utilisation of XML as a data-structuring tool provides a powerful information infrastructure that within a standard representation is flexible enough to accommodate a variety of possible future needs.

In today's large-scale organisations, global collaboration is a necessity; the effectiveness of these collaborations determines a large part of the final product's effectiveness in the marketplace.

The environment presented in this thesis shows both a solution to the problem of sharing information in a distributed product development teams and the work undertaken into making such a solution a reality. It is a solution requiring consideration and knowledge drawn from a variety of domains namely, Software Engineering, Psychology, and Engineering.

The problem of distributed information sharing needs to be tackled from a wider perspective than previous efforts have considered. Benefits can be gained through the limited scope of systems put forward in the past. Real gains can only be achieved through an expansive and fully integrated system. There is a strong need for Project Managers to employ an adequately considered I.T. solution. The implemented technologies chosen must take, suitability, portability, and ease of use, future change and expandability into account.

Real asynchronous and synchronous communications are the key, supported by a framework, which is closely tailored to our natural psychological infrastructure. Such a framework then allows the systems functions to be expanded beyond sharing information during development. The data is in a suitable structure for maintenance personnel to access, and for training purposes saving both time and money in collating data for these future tasks. The nature of the concept / constraint based framework essentially functions as an information gathering, collating, checking, formatting, structuring, disseminating, and storing tool.

This research work presents the facilities required to create knowledge repositories that have real world value, throughout the entire product development lifecycle.

The benefits from the implementation of the collaborative system developed are:

- The reduction of overall product development costs.
- The reduction in the product's time to market.
- The improvement of product quality.
- The early resolution of inconsistencies and conflicts avoiding the need for expensive reworking in the later stages of the product development lifecycle.
- The early involvement of suppliers facilitates the consistency between the design and process and product capabilities of the suppliers.
- The improvement of the products fitness for purpose for its intended market.

- Holistic perspective of the shared product development knowledge in turn facilitates further innovations.

Accurate enhanced collaboration information sharing repositories facilitate the reduction of problems and inconsistencies during the product development lifecycle through early detection and resolution of conflicts. Therefore this research contributes a collaborative environment capable of increasing an organisation's competitiveness through the lowering of costs, increased team-building opportunities, reductions in time to market and an increase in product applicability and quality.

Chapter 8

CHAPTER 8. Recommendations for Future Work

8.1. Overview

This research work has contributed much to facilitation of Distributed Concurrent Engineering from a global perspective. It has contributed bringing together previous research from three core domains, Software Engineering, Psychology, and Engineering. The perspectives from these fields has given rise to an innovative approach to the problem tackling the integration of the domain specialists knowledge at the earliest possible phase in the product development lifecycle.

However this approach has encompassed a wide variety of areas of specialisation. The solution presented is essentially a foundation for further research, a framework that more detailed and finer grained sub sections of the product development lifecycle can be integrated hence the focus upon flexibility and extensibility within the research.

The system is intended for the domain of collaborative product development, however due to its inherent holistic and flexible nature it is also applicable to other domains such as patient data in health care, and film / television production. The framework allows people to construct, find and understand the information they are looking for quickly, an ideal way to collate work on an upcoming operation for busy surgeons incorporating 3D models of MRI scans, or a film crew, distributed around the world, working on different scenes at the same time with relationships to computer rendered special effects models.

8.2. Introduction

The present research work has focused upon the foundations for purposeful information sharing platform building a solution to the wide range of problems with current information sharing systems in place today. The largest of these has been the omission for the consideration of the earliest phases in the product development lifecycle. Current systems have neglected this fully integrated macro approach in favour of complex micro approaches based on advanced modelling or simulation.

This integrated focus of the present work provides the platform for these micro approaches to support while maintaining a comprehensive solution to integration.

8.3. Technological Considerations

The approach taken in this research has been based upon the implementation of java, however there are other interface possibilities especially focused upon the use of multimedia, these include Flash and Shockwave formats. These both have high quality development tools and are portable on provision that a suitable plugin is installed within the web browser. These have both improved and continue to improve in recent years, ultimately they are likely to converge, with Shockwave likely to provide the superset with embedded Flash technologies.

Java has been used on the server and the client in this research but there are other possible candidates for the server side, namely ASP, JSP, and PHP. These could be used to connect to a variety of client implementations either in HTML & JavaScript, Java, or Flash.

8.4. STEP Implementation

STEP is a vast standard that necessitates serious investment for its implementation however the benefits of a platform independent product representation format warrant this investment, instead of a file based system as detailed in this research there should be full fine grained implementation allowing each entity to be locked at a time instead of a whole model essentially this in turn necessitates the use of a STEP developers toolkit to be able to take advantage of the progress within this domain of standardisation.

8.5. XML Scalability

With STEP being somewhat inaccessible to those without effective development tools investigation should be made in greater detail for the utilisation of XML to represent product data, preliminary research into this area had been under investigation at the time of writing, although mainly directed in providing translations from STEP's Application Protocols into XML based files.

XLL should also be considered for the linking mechanism that thus far in the research work has been based on the utilisation of the URL. XLL allows the development of an intermediate set of links to aid with maintenance issues. However XLL was only at the recommendation stage of its development during this research.

8.6. 3D Integration

A finer grained approach could be taken with the 3D implementation of model, should they be represented in Java 3D for example. This requires translators being built for various CAD file formats, but once achieved could be linked closely with the concept mapping hierarchy. This could be extended to a 3D conceptual environment in the future.

8.7. Conflict Realisation

Research into quantifying conflict realisation from subjective natural language rather than having to detail specific formulae is a research area by itself. However there is much discussion on the most appropriate conflict resolution algorithm. There is a balance in the computation time spent optimising the constraint network for searching and traversing the network for a solution. There are alternate emphases for increasing the computation spent optimising both the pre-search and the search itself. Research into the most appropriate approach is required.

8.8. Neutral Language

This research work has made the assumption that the development team speak a common language however in the global arena of collaborative development there is a strong likelihood that this is not the case, and hence research into a common language dictionary that the users choose terms from for concepts would be another area for further research.

8.9. Small Medium Enterprises (SME) Domain Consideration

Large-scale organisations and collaborations have been considered in this research work, however the rational applicable to that of the large concern is equally applicable to distributed clusters of SMEs. This application should be further explored, SME clusters provide an environment for the uptake of new technology to enable their competition with large-scale organisations and thus present an ideal opportunity for exploration of integrated collaborative systems deployment.

8.10. Summary

While the innovative research undertaken has resulted in a solution to the problem of Distributed Concurrent Engineering it can only be considered a partial solution and effectively any subsequent system can only ever be considered a partial solution.

The number of potential collaborators, their requirements, their IT infrastructure, the individual needs of the domain specialists dictates that research into this area is in its infancy, new technologies with which to base solutions on are becoming available bi-yearly. Information modelling research and improvements into the understanding of our own cognition systems all are factors to constantly try to integrate in such a system.

Further research requires in-depth multi domain knowledge in itself to be able to provide a system that begins to satisfy the requirements of distributed product development specialists. Development of such a system is taking place in a rapidly evolving software and hardware environment with equally rapidly evolving expectations.

References

References

Abdalla, H. S., "Concurrent engineering for global manufacturing." *International Journal of Production Economics*, no. 60-61, pp 251-260, 1999

Abdalla, H. S. and Knight, J., "An Expert System for Concurrent Product and Process Design of Mechanical Parts." *Proceedings Institute Mechanical Engineers*, no. 208, pp 167-172, 1994

Abdalla, S. A., "A Concurrent Engineering Constraint-Based System." *Computers Industrial Engineering*, no. 35, pp 459-462, 1998

Ahmed, A. M. and Abdalla, H. S., "The role of innovation process in crafting the vision of the future." *Computers & Industrial Engineering*, no. 37, pp 421-424, 1999

Ahmed, A. M. and Abdalla, H. S., "Beyond competition: a framework for the 21st century." *International Journal of Production Research*, no. 38, pp 3677-3709, 2000

Ahmed, A. M. and Abdalla, H. S., "Supertitive Strategy as a Portfolio for the Next Millennuim." *Human Factors and Ergonomics in Manufacturing*, no. 10, 2000

Al-Ashaab, A., Rodriguez, K., et al., "INTERNET Based Information System to Support Global Collaborative Manufacturing." *6th International Conference on Concurrent Enterprising* , Toulouse, pp 233-240, 2000

Al-Ashaab, A. H. S. and Ruiz, S. R., "Using a PDM as a Tool to Support A Concurrent Engineering Application in a Mexican Company." *CSIM* , ITESM Campus Monterrey, Mexico, 1998

An, D., Leep, H. R., et al., "A Product Data Exchange Integration Structure Using Pdes/Step For Automated Manufacturing Applications." *Computers & Industrial Engineering*, no. 29, pp 711-715, 1995

Anderson, N. and Abdalla, H., "Web Browser Based Collaborative Product Development for the 21st Century." *8th ISPE International Conference on Concurrent Engineering: Research and Applications - CE 2001* , Anaheim California, USA, pp 125-134, 2001

Anderson, N. and Abdalla, H., "A Distributed E-commerce System for Virtual Product Development Teams." *Proceedings Institute of Mechanical Engineers Part B: Journal Engineering Manufacturing*, no. 216, pp 251-264, 2002

Anderson, N. and Abdalla, H., "A Holistic Integrated Approach to Product Development." *International Journal of Advanced Manufacturing Technology*, no. (Submitted), 2002

Anderson, N. and Abdalla, H. S., "Distributed Information Sharing for Collaborative Systems (DISCS)." *Information Visualisation'99* , IEEE Computer Society, London, pp 476-481, 1999

Associates, D., *The Principles of Integrated Product Development*, [WWW] Available from: http://members.aol.com/drmassoc/principles.html [Accessed: 28/6/2001], 2001

Autodesk, *WHIP Plugin*, [WWW] Available from: http://www.autodesk.com [Accessed: 14/12/200], 2000

Barry, D. K., *The Object Database Handbook, How to Select, Implement, and Use Object-Oriented Databases*, Wiley & Sons Inc, 1996

Barták, R., *ON-LINE GUIDE TO CONSTRAINT PROGRAMMING*, [WWW] Available from: http://kti.ms.mff.cuni.cz_bartak\ [Accessed: 9/12/2002]

Bass, L. J. (1993). Human Computer Interaction, Third International Conference, EWHCI '93, Moscow, Russia, August 1993, Selected Papers. Berlin, Heidelberg, Springer-Verlag: 152-180.

Baxter, J. E., Henson, B. W., et al., "Multiple Viewpoint Support for the Product Data Management of Complex Assemblies." *The University of Leeds* , Leeds, England, 1998

Benzekri, A., Aoun, A., et al., "Integrating Web Based Training into Concurrent Engineering." *6th International Conference on Concurrent Enterprising* , Toulouse, pp 317-325, 2000

Berker, I. and Brown, C. D., "Conflicts and Negotiation in Single Function Agent Based Design Systems." *Concurrent Engineering:Research and Applications*, no. 4, pp 17-33, 1996

Bosak, J., *XML, Java, and the Future of the Web*, [WWW] Available from: http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.html [Accessed: 12/10/1998], 1998

Branki, C., "The Acts of Cooperative Design." *CE Concurrent Engineering: Research and Applications*, no. 3, 1995

Bray, T., *Beyond HTML: XML and Automated Web Processing*, [WWW] Available from: http://developer.netscape.com/viewsource/bray_xml.html [Accessed: 12/10/1998], 1998

Buzan, T. and Buzan, B., *The Mind Map Book, Radiant Thinking, The Major Evolution in Human Thought*, BBC Books, 1995

Caillaud, E., Gourc, D., et al., "A framework for a knowledge-based system for risk management in concurrent engineering." *Concurrent Engineering-Research and Applications*, no. 7, pp 257-267, 1999

Callaghan, M., *The RMI development process*, [WWW] Available from: http://www.cms.dmu.ac.uk/~jmc [Accessed: 26/10/1998], 1998

Calvin, *How Brains Think*, Orion Publishing, London, 1997, pp 184

Catarci, T., *Visual Interaction With Databases*, [WWW] Available from: http://www.it-link-demon.co.uk/IV99/Visual_Databases.htm [Accessed: 18/5/1999], 1999

Cattel, R. G. G., *The Object Database Standard, ODMG - 93*, Morgan Kaufmann Publishers, 1994, pp p167

CENTEX, *CENTEX, Concurrent Engineering Needs & Technologies Experimentation*, [WWW] Available from: http://esoce.pl.ecp.fr/ce-net/Information/cerelated/centex/Projectoverview.htm [Accessed: 16/3/1999], 1999

Chan, S. C. F., Lee, P. H. S., et al., "Synchronous collaborative development of UML models on the internet." *Concurrent Engineering-Research and Applications*, no. 9, pp 111-119, 2001

Chang, H. C., Lu, W. F., et al., "WWW-based collaborative system for integrated design and manufacturing." *Concurrent Engineering-Research and Applications*, no. 7, pp 319-334, 1999

Chaudri, A. B., *Akmal's Home Page*, [WWW] Available from: http://www.soi.city.ac.uk/~akmal/html.dir/home.html [Accessed: 15/12/1998], 1998

Chauvet, J. and Lerman, M., "Object Models and Java." *Dr Dobb's Journal*, , pp 20-23, 99-100, 1997

Chen, L. and Li, S., "Modeling concurrent product design: A multifunctional team approach." *Concurrent Engineering-Research and Applications*, no. 8, pp 183-198, 2000

Clarke, C., "ProductVision 1.0." *CADCAM*, , pp 14-16, 1999

Connolly, D., "The Evolution of Web Documents: The Ascent of XML." *World Wide Web Journal Special Issue on XML*, no. 2, pp 119-128, 1997

Connolly, D., *XML Activity*, [WWW] Available from: http://w3.org [Accessed: 28/10/1998], 1998

Danesh, M. R. and Jin, Y., "An agent-based decision network for concurrent engineering design." *Concurrent Engineering-Research and Applications*, no. 9, pp 37-47, 2001

de Groot, A. D., *Thought and Choice in Chess*, The Hague: Mouton, 1965

Delhaye, J. L., Bröcker, A., et al., "Multi-Site Concurrent Engineering for Large Aircraft Engineering and Support." *6th International Conference on Concurrent Enterprising* , Toulouse, pp 379-386, 2000

Design, O., *Introducing ObjectStore*, [WWW] Available from: http://rdp.cps.msu.edu:8000/~ODI/docs/OSJI/apiug/1_intro.htm [Accessed: 16/6/1999], 1999

Downs, R. M. and Stea, D., *Maps in Minds: Reflections on Cognitive Mapping*, Harper & Row, 1997

EDItEUR, "EDI for the Book World", 1995

Eodice, M. T., Leifer, L. J., et al., "Analyzing requirements - Evolution in engineering design using the method of problem-reduction." *Concurrent Engineering-Research and Applications*, no. 8, pp 104-114, 2000

Feiner, S., "Windows on the World: 2D Windows for 3D Augmented Reality." *ACM Symposium on User Interface Software and Technology* , pp 145-155, 1993

Feiner, S., MacIntyre, B., et al., "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment." *International Symposium on Wearable Computing* , Cambridge, M.A, USA., pp 74-81, 1997

Flanagan, D., *Java in a Nutshell*, O'Reilly & Associates Inc., 1997

Fowler, J., *STEP, For Data Management Exchange and Sharing*, Technology Appraisals Ltd., 1995

Frese, M., *Psychological Issues of Human-Computer Interation in the Work Place*, Elsevier Science Publishers B. V., 1987, pp 275-294

Gadient, A. J., Hines, L. E., et al., "Agility through information sharing: Results achieved in a production setting." *Concurrent Engineering-Research and Applications*, no. 5, pp 101-111, 1997

Gardoni, M., Spadoni, M., et al., "Harnessing non-structured information and knowledge and know- how capitalisation in integrated engineering: Case study at Aerospatiale Matra." *Concurrent Engineering-Research and Applications*, no. 8, pp 281-296, 2000

Goh, A., Hui, S. C., et al., "An integrated environment for product development using STEP/EXPRESS." *Computers in Industry*, no. 31, pp 305-313, 1996

Goh, A., Hui, S. C., et al., "A Study of Sdai Implementation On Object-Oriented Databases." *Computer Standards & Interfaces*, no. 16, pp 33-43, 1994

Gomes, R. C. P., "Reactivity and Pro-Activeness in Virtual Prototyping", 1998

Group, O. M., *CORBA FAQ*, [WWW] Available from: http://www.omg.org/gettingstarted/corbafaq.htm [Accessed: 21/11/2002]

Haas, R. E., "Corporate Intranets - Applications and Technologies." *6th International Conference on Concurrent Enterprising* , Toulouse, pp 269-276, 2000

Hague, M. J. and Taleb-Bendiab, A., "Tool for the Management of Concurrent Conceptual Engineering Design." *Concurrent Engineering: Research and Applications*, no. 6, pp 111-129, 1998

Hares, J. S., *Object Orientation: technology, techniques, management and migration*, Wiley, 1994

Hashemian, M. and Gu, P., "A Constraint-Based System for Product Design." *Concurrent Engineering: Research and Applications*, no. 3, pp 177-186, 1995

Heintz, T. J. and Acar, W., "Causal Modelling as a Tool for Problem Framing Within a Group Decision Support System: An Object Oriented Approach." *Info Systems Journal*, , pp 291-310, 1994

Horváth, I., Kuczogi, G., et al., "Spatial Behavioural Simulation of Mechanical Objects", 1998

Huang, G. Q., Huang, J., et al., "Early supplier involvement in new product development on the Internet: Implementation perspectives." *Concurrent Engineering-Research and Applications*, no. 8, pp 40-49, 2000

Huang, G. Q., Lee, S. W., et al., "Web-based Product and Process Data Modelling in Concurrent "Design for X"." *Robotics and Computer Integrated Manufacturing*, , pp 53-63, 1999

Jagannathan, V., "Information Sharing Systems", *Technical Report Series, Technical Memoranda*, 1992

Jagannathan, V., "Model-Based Information Access", *Technical Report Series, Technical Memoranda*, 1993

Jagannathan, V., "Collaborative Infrastructures Using the WWW and CORBA-Based Environments", *Technical Memoranda*, 1996

Johnson, M., *A beginner's guide to Enterprise JavaBeans*, [WWW] Available from: http://www.javaworld.com/javaworld/jw-10-1998-beans.html [Accessed: 27/11/1998], 1998

Johnson, P., *Human Computer Interaction, Psychology, Task Analysis and Software Engineering*, McGraw-Hill International (UK) Ltd., 1992, pp 44-59

Kim, C. Y., Kim, N., et al., "Distributed Concurrent Engineering: Internet-based interactive 3-D dynamic browsing and markup of STEP data." *Concurrent Engineering-Research and Applications*, no. 6, pp 53-70, 1998

Kim, N. K., Kim, Y., et al., "Subdivision methods of converting STEP into VRML on Web." *Computers & Industrial Engineering*, no. 33, pp 497-500, 1997

Kim, W., *Introduction to Object-Oriented Databases*, MIT Publishing, 1990

Klein, M., "iDCSS: Integrating Workflow, Conflict and Rationale-Based Concurrent Engineering Coordination Technologies." *Concurrent Engineering: Research and Applicattions*, no. 3, pp 21-27, 1994

Krause, I. and Doblies, M., "Global Product Data Management", 1998

Lai, W., Li, D. G., et al., "Designing effective user interfaces for Web courseware." *Concurrent Engineering-Research and Applications*, no. 9, pp 105-110, 2001

Lander, S. E. and Corkhill, D. D., "Designing Integrated Engineering Environments: Blackboard-Based Integration of Design and Analysis Tools." *Concurrent Engineering: Research and Applications*, no. 4, pp 59-71, 1996

Laszlo, E., *Changing Visions: Human Cognitive Maps: Past, Present and Future*, Adamantine, 1996

Lea, R., *Java for 3D and VRML Worlds*, New Riders, 1996

Lee, J. Y., Kim, H., et al., "A web-enabled approach to feature-based modeling in a distributed and collaborative design environment." *Concurrent Engineering-Research and Applications*, no. 9, pp 74-87, 2001

Leong, H. V., Ho, K. S., et al., "Web-based workflow framework with CORBA." *Concurrent Engineering-Research and Applications*, no. 9, pp 120-130, 2001

Lie, H. W. and Lilley, C., *Style Sheets Activity*, [WWW] Available from: http://w3.org [Accessed: 28/10/1998],

Lindeblad, M., Isaksson, O., et al., "Utilisation of common IT technology for Multidisciplinary Engineering Data Communication." *6th International Conference on Concurrent Enterprising* , Toulouse, 2000

Liu, T. H., Trappey, A. J. C., et al., "Development of SDAI-based common access interface for object- oriented DBMS." *Concurrent Engineering-Research and Applications*, no. 8, pp 230-241, 2000

Liu, X. Q. F., Noguchi, K., et al., "Requirement acquisition, analysis, and synthesis in quality function deployment." *Concurrent Engineering-Research and Applications*, no. 9, pp 24-36, 2001

LKSOFT, *SDAI Overview*, [WWW] Available from: http://www.lksoft.com/sdai_overview.html [Accessed: 8/6/1999], 1999

Loffredo, D. T. (1998). Efficient Database Implementation of EXPRESS Information Models. Department of Computer Science. Troy, NY, USA, Rensselaer Polytechnic Institute.

Londono, K. J., "Co-ordinating a Virtual Team", *Technical Report Series, Research Note*, 1992

MacIntyre, B. and Feiner, S., "Future Multimedia User Interfaces." *Multimedia Systems*, no. 4, pp 250-268, 1996

MacIntyre, B. and Feiner, S., "Language-Level Support for Exploratory Programming of Distributed Environments." *Symposium on User Interface Software and Technology* , pp 83-94, 1996

MacIntyre, B. and Hollerer, T., "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment." *International Symposium on Wearable Computing* , Cambridge, MA, USA, pp 74-81, 1997

MacKrell, J., "Review of CoCreates OneSpace "The Virtual Conference Room Where Global Design Teams Work Together"", 2001

Matta, N. and Corby, O., "A Generic Library of Knowledge Components to Manage Conflicts in CE Tasks." *Concurrent Engineering: Research and Applications*, no. 6, pp 274-287, 1998

Matthews, B., "XML and the World-Wide Web Consortium Leverage Action Project." *European Research Consortium for Informatics and Mathematics News*, , 1998

McKnight, C., *Hypertext, a psychological perspective*, Ellis Horwood Ltd., 1993, pp 51-69

Nakamura, I., "A CAD Database Based on STEP." *Industrial Systems for Production and Engineering (B-10), Elsevier Science*, , 1993

ObjectDesign, *Introducing ObjectStore*, [WWW] Available from: http://rdp.cps.msu.edu:8000/~ODI/docs/OSJI/apiug/1_intro.htm [Accessed: 16/6/1999], 1999

Olsen, G. R., Cutkosky, M., et al., "Collaborative Engineering Based on Knowledge Sharing Agreements." *Concurrent Engineering: Research and Applications*, no. 3, pp 145-159, 1995

Ottosson, S., "Virtual Reality in Product Development", 1998

Pallot, M., "EPICE: Realising the Virtual Project Office." *6th International Conference on Concurrent Engineering* , pp 57-70, 2000

Peng, T. K. and Trappey, A. J. C., "A step toward STEP-compatible engineering data management: the data models of product structure and engineering changes." *Robotics and Computer-Integrated Manufacturing*, no. 14, pp 89-109, 1998

Prasad, *Concurrent Engineering Fundamentals: Integrated Product and Process Organisation*, Prentice Hall International, 1996

Rees, L. C., *A Brief Introduction of the Mysteries of the eXtensible Markup Language*, [WWW] Available from: http://w3.org [Accessed: 28/10/1998], 1998

Rezayat, M., "The Enterprise-Web Portal for Life-cycle Support." *Computer-Aided Design*, , pp 85-96, 1999

Roucoules, L. and Tichkiewitch, S., "CoDE: A cooperative design environment - A new generation of CAD systems." *Concurrent Engineering-Research and Applications*, no. 8, pp 263-280, 2000

Roy, U., Bharadwaj, S., et al., "Product Development in a Collaborative Design Environment." *Concurrent Engineering: Research and Applications*, no. 5, pp 347-365, 1997

Sadeh, N. M., Hildum, D. W., et al., "A Blackboard Architecture for Integrating Process Planning and Production Scheduling." *Concurrent Engineering: Research and Applications*, no. 6, pp 88-99, 1998

Sky, R. W. E. and Buchal, R. O., "Modeling and implementing concurrent engineering in a virtual collaborative environment." *Concurrent Engineering-Research and Applications*, no. 7, pp 279-289, 1999

Smirnov, A. V. and Chandra, C., "E-Management of Scalable Supply Chain Configurations: Concept and Reusable Knowledge Management." *6th International Conference on Conurrent Enterprising* , Toulouse, 2000

Srinivas, S., "MONET: A multi-media System for Conferencing and Application Sharing in Distributed Systems", *Technical Report Series Research Note*, 1992

St. Laurent, S. and Cerami, S., *Building XML Applications*, McGraw-Hill, 1999

STEPTools, *STEP Software for World-Wide Manufacturing*, [WWW] Available from: http://www.steptools.com [Accessed: 22/11/1999], 1999

Struck, T., Ratchev, S., et al., "Management of Requirements Engineering Knowledge in Virtual Enterprises." *6th International Conference on Concurrent Enterprising* , Toulouse, 2000

Tangelder, J. W. H., Vergeest, J. S. M., et al., "Producing Physical Prototypes Using A Sculpting Robot." *Industrial Design Engineering* , Delft University of Technology, The Netherlands, 1998

Tay, F. E. H. and Ming, C., "A shared multi-media design environment for concurrent engineering over the internet." *Concurrent Engineering-Research and Applications*, no. 9, pp 55-63, 2001

Thomas, A., *Enterprise JavaBeans, Server Component Model for Java*, [WWW] Available from: http://java.sun.com/products/ejb/white_paper.html [Accessed: 27/11/1998], 1998

Udo, G. G. and Ebiefung, A. A., "Human factors affecting the success of advanced manufacturing systems." *Computers & Industrial Engineering*, no. 37, pp 297-300, 1999

Urban, S. D., "Engineering Data Management Achieving Integration Through Database Technology." *Computing and Control Journal*, no. June, pp 119-126, 1993

Vankan, J., Roth, N., et al., "Concurrent Engineering and Human Factors in the European project ENHANCE Enhanced Aeronautical Concurrent Engineering." *6th International Conceference on Concurrent Enterprising* , Toulouse, pp 309-315, 2000

VENTO, *Virtual Enterprise Organiser*, [WWW] Available from: http://www.prosoma.lu/cgi-bin/show.py?opt=flat&id=2581page=resume [Accessed: 10/8/1998], 1998

Vergeest, J. S. M. and Horvath, I., "GEOS-Based Analysis to Determine the Feasibility of Engineering Data Sharing", 1998

Vergeest, J. S. M. and Horvath, I., "Design model sharing in concurrent engineering: Theory and practice." *Concurrent Engineering-Research and Applications*, no. 7, pp 105-113, 1999

W3Apps, *Jeevan - The Object Database for Java Platform*, [WWW] Available from: http://www.w3apps.com/datasheet.html [Accessed: 10/5/1999], 1999

Walsh, N., *What Do XML Documents Look Like?*, [WWW] Available from: http://w3.org [Accessed: 29/10/1998], 1998

Web-3D-Consortium, [WWW] Available from: http://www.vrml.org [Accessed: 9/11/1999], 1999

Wu, T., Blackhurst, J., et al., "Integrated enterprise concurrent engineering: A framework and implementation." *Concurrent Engineering-Research and Applications*, no. 9, pp 211-222, 2001

XML, *XML for Managers, Evaluating SGML vs. XML from a Manager's Perspective*, [WWW] Available from: http://w3.org [Accessed: 28/10/1998], 1998

Yeh, S. C. and You, C. F., "Implementation of STEP-based product data exchange and sharing." *Concurrent Engineering-Research and Applications*, no. 8, pp 50-60, 2000

Yoo, S. B. and Suh, H. W., "Integrity Validation of Product Data in a Distributed Concurrent Engineering Environment." *Concurrent Engineering: Research and Applications*, no. 7, pp 201-213, 1999

Yoshimura, M. and Takahashi, K., "Collaborative design among different fields in mobile-agent environments." *Concurrent Engineering-Research and Applications*, no. 9, pp 146-154, 2001

Zhuang, Y., Chen, L., et al., "CyberEye: An Internet-enabled environment to support collaborative design." *Concurrent Engineering-Research and Applications*, no. 8, pp 213-229, 2000

Appendix A

Published Works From This Research

A distributed e-commerce system for virtual product development teams

N Anderson and H Abdalla*
Design Management and Communications, De Montfort University, Leicester, UK

Abstract: This paper presents a distributed information system that facilitates virtual multidisciplinary product development activities. A growing problem today for many companies is that of facing up to increasing global competition, increasing the importance of delivering the right product at the right time and at the least cost. To enable a company to provide competitive services, concurrent engineering methods are employed progressively more often. However, a truly effective concurrent engineering team requires specialists in a wide variety of occupations and skill bases. Effective communication and exchange of information between members usually require not only face-to-face meetings but also highly advanced tools. Various tools have been developed in the past with the premise of facilitating effective communication. However, these have realistically only covered a subsection of the total product development life-cycle. The work presented here displays an approach to the problem in terms of an Internet-based system, which provides a mechanism to allow integrated information sharing from the initial concepts through to the realization and the marketing of a product. The system effectively removes the barriers between people and incompatible computer systems and resulting from non-standardization.

Keywords: conceptual design, concept mapping, information sharing, concurrent engineering

NOTATION

AP203	application protocol 203
API	application programming interface
ASP	active server page
CAD	computer-aided design
CMS	Cambridge Material Selector
DISCS	distributed information sharing for collaborative systems
DOM	Document Object Model
DTD	Document Type Definition
MIME	multipurpose Internet mail extension
RMI	remote method invocation
SAX	Simple API for XML
SDAI	standard data access interface
STEP	Standard Exchange of Product
URL	uniform resource location
VRML	virtual reality mark-up language
XML	extensible mark-up language
XSL	extensible style language

The MS was received on 5 February 2001 and was accepted after revision for publication on 24 July 2001.
* *Corresponding author: Design Management and Communications, De Montfort University, The Gateway, Leicester LE1 9BH, UK.*

1 INTRODUCTION

Today's global marketplace requires organizations to obtain great levels of flexibility and responsiveness to change to stay competitive. E-business solutions provide a competitive advantage for those that choose to lever their benefits. However, these solutions need to expand beyond the currently existing, disparate and relatively restrictive tools to a fully integrated and flexible system.

An organization faced with a global marketplace often requires a global skills base. The focus of this work is the communication between the members of a virtual team. This is based on the premise that a multidisciplinary and multiskilled team functions most efficiently when there is a forum to group different perspectives together in a cohesive manner. The effective implementation of concurrent engineering hinges on all members of the product development team supplying their contribution at the earliest stage possible in the product development. Contributions should be made available for the appropriate members to view, understand, evaluate and build on for further development. Co-location is usually prohibitively expensive in terms of both cost and time. Notably skilled team members are certain to be in high demand and thus working on concurrent activities.

The work in this paper is focused on creating an enabling environment allowing the sharing and the assistive structuring of product information throughout a development cycle for globally distributed product development teams. An inherent expansion of the considered user base is incorporated by the earlier stages of data capture, where the input required can be at a relatively low technical level. This allows contributions from other domains such as sales, marketing and sample end users to be included, when conflict resolution is at the point of least cost and greatest overall benefit. The potential deployment and target area for an information-sharing environment, described within this paper, would be intended as a large-scale engineering establishment facing global competition.

Previous systems within the problem domain of distributed information sharing often ignore the aspects of integration and portability. In one of the most current contributions, CyberEye by Zhuang *et al.* [1] at the University of Toronto puts forward a system based on current technologies such as Java, servlets and active server pages. However, this and previous systems are today missing a tangible thread through their shared data stores. Other modern systems provide a narrow solution to the problem, such as the one presented by Chang *et al.* [2], which is a system for sharing constraint-based information but without complete framework integration. A framework is essential to provide a context for the retrieved data.

Abdalla [3] presented the IMS test case 3 global concurrent engineering and indicated that the following factors are required for success: effective communication, a systematic involvement of customers, suppliers and distributors, powerful information infrastructure and effective use of modern technology. Pallot *et al.* [4] stated that creativity and innovation are soon to become two of the major factors of competition in the global market and hence a system should considerably enhance these areas. The mind mapping technique [5] is a technique for information representation utilizing a natural cognitive framework, which provides a practical platform for innovation. Branki [6] stressed the importance of shared ontologies and their application.

Research work in the area of concurrent engineering with an emphasis on distributed information sharing was addressed by Kim *et al.* [7], Vergeest and Horváth [8], Srinivas *et al.* [9] and Jagannathan *et al.* [10]. The relevance of the conceptual model to adding semantic details to data is paramount within the work of Downs and Stea [11] and Jagannathan *et al.* [12]. Recent research work emphasized the development of platforms that enhance speed and efficiency. However, most of these systems do not consider a conceptual and macro-level information structure and representation. The current research work provides Internet-based information sharing that facilitates collaborative work throughout the entire product development cycle, from concept to manufacturing. The research undertaken is focused on improving communication where common understanding of that diction has been neglected in the tools available today. Much of the current/previous work undertaken has not addressed communication and data sharing of products at the conceptual level, which is an essential consideration of this work.

2 THE PROPOSED SYSTEM FOR DISTRIBUTED INFORMATION SHARING

The principal objective of this research is to develop a computer Web-based system to facilitate distributed information sharing across multidisciplinary teams. The focus is to improve the accuracy and consistency of communication during the early stages of product development for distributed teams. This communication forms a foundation framework with which information can be shared throughout the subsequent development cycle. Distributed information sharing for collaborative systems (DISCS) provides the facility for team members to utilize their early concepts at the start of the product development lifecycle. However, this approach also provides a logical and coherent flow of data from this stage into the rest of the lifecycle, where more concrete designs are produced.

2.1 System structure

The proposed system consists of a Web-based concept-mapping module, data sharing and structuring module, data standardization module, object oriented databases, computer aided design (CAD) solid modelling, a conflict resolution module and hypertext transfer protocol (HTTP) accessed documents including three-dimensional model viewing software. The proposed system architecture is shown in Fig. 1.

Fundamentally significant conceptual information is gathered at the start of the product development lifecycle. These valuable data are stored in a central concept database, which is structured using extensible mark-up language (XML), ensuring that formal and comprehensible information is shared. The purpose of the conceptual evaluation is to highlight conflicts and inconsistent requirements.

Conceptual evaluation forms the basis of CAD prototyping. For those without CAD solid modelling software, viewers based on both virtual reality mark-up language (VRML) and the AutoDesk™ WHIP™ plug-in [13] can be used to view the design. The STEP standard is used both to build a central product information database and an open standard data access interface (SDAI) and to provide export of AP203 files. The information in the conceptual and product databases is represented in the interface by concept maps and CAD solid models. A cohesive

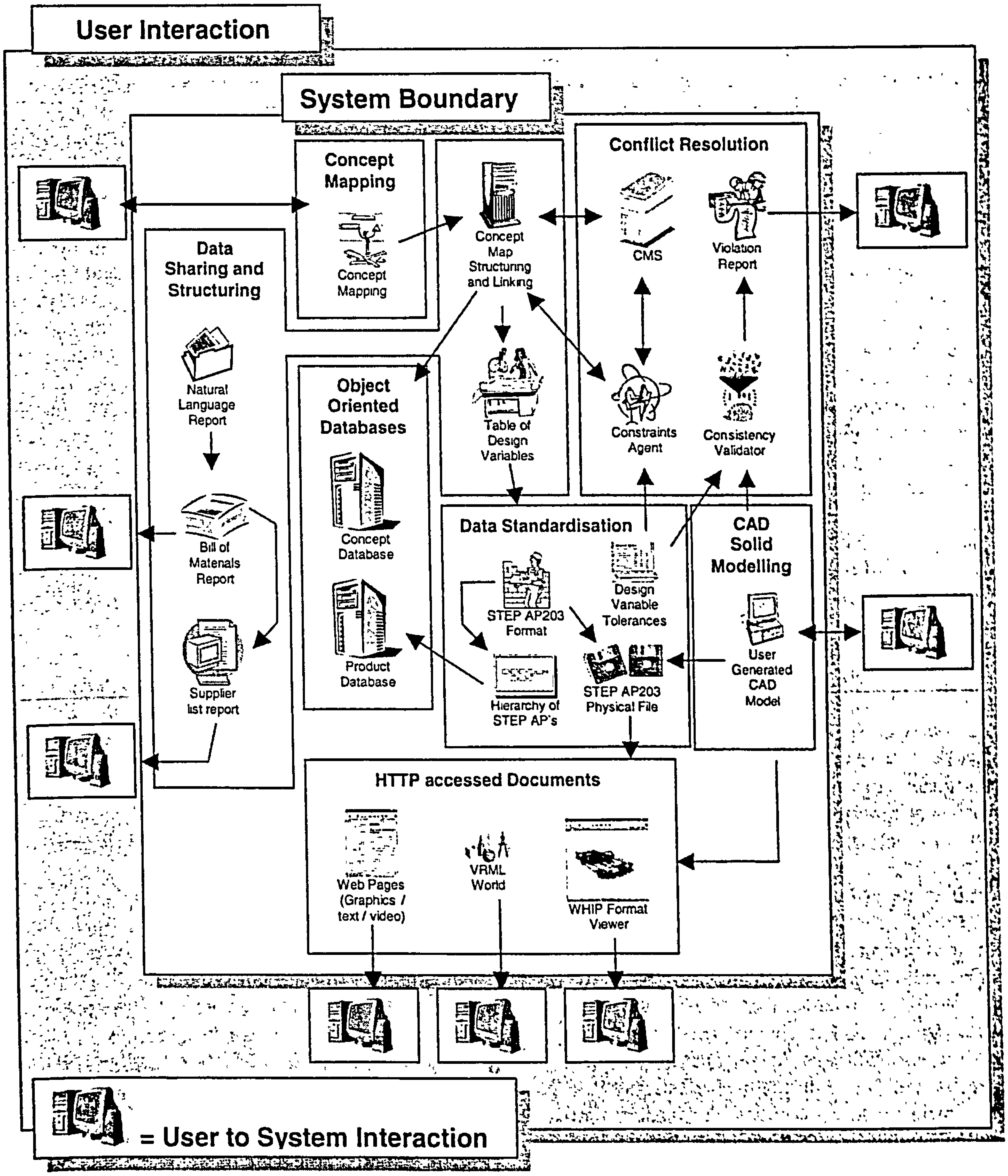


Fig. 1 Proposed system architecture

data store is built by linking concepts in the mapping stage with relevant information, which can concern Web sites, other concept maps or CAD models.

2.1.1 *Concept mapping module*

Concept mapping was designed to stimulate the generation of new and innovative ideas as well as for storing ideas. The research by Downs and Stea [11], from cognitive mapping, forms the basis of the system. Location within the data can be achieved via this mechanism. Without suitable information infrastructures, users lose track of where their information goal lies.

Conceptual mapping can be likened to mind mapping. However, it extends this useful information exchange or structuring mechanism in various manners. First is the linking system from each node/leaf in the structure, which can be likened to that of hyperlinks. These links take the follower to other conceptual maps, CAD models and Web pages. Second is the format that each node can be represented in, which in turn provides a formal basis for constraint notification and comparison.

Conceptual level driven communication benefits extend from those for example on three-dimensional models of pipe arrangements for chemical plants. Utilization of a three-dimensional model allows maintenance teams to be taken through the expected maintenance routine, in a virtual environment so that their input could be obtained. However, three-dimensional modelling is slow in comparison with an overview-driven conceptual analysis. Obviously deep intricate detail cannot be considered at the conceptual level. However, higher level thought trains, which otherwise would not be visible to the rest of the team, could be taken into account in the least amount of time. More detailed viewpoints such as the three-dimensional models can then be linked into the system for further analysis. DISCS offers various levels of detail for information to be presented; this helps new team members with a system which not only allows them to gain the essence of project requirements quickly but also allows them to delve into regions of greater detail if required. Much in the same way that a contents page in a book provides a framework for subsequent information, a concept map can offer an intuitive assistive cognitive medium for the comprehension of the various data types encountered in product development.

This research demonstrates the development of concept mapping to bridge geographical boundaries and time zones, hence addressing factors identified in research work by Zhuang *et al.* [1]. This previous research pinpoints the importance of a system which can deal with synchronous and asynchronous distributed interactions. Asynchronous communication is a vital factor, considering that globally distributed product development teams are very much affected by working in different time zones. Concept mapping, developed in the system and described in this paper, seamlessly provides both synchronous and asynchronous communication. Team members using the browser-based concept map tool are provided with the facility to construct a conceptual representation of a product. The greatest benefits are gained through the use of the system in real time, sharing access control during conceptual collaboration. However, the concepts can also be worked on off-line when the global distribution of the team members prohibits real-time interaction. Thus the system is in accordance with Zhuang *et al.*'s requirement for synchronous and asynchronous distributed interactions.

The developed concept map tool was built using advanced Java technologies, Swing Java RMI (Remote Method Invocation), SAX (Simple API for XML) and DOM (Document Object Model), providing a platform-independent creative framework tool. The tool allows the viewing of the XML based concept maps retrieved from the previously described concept database. Each user has access to a browser-based client, which provides access to the maps in the manner of a shared drawing tool, with control from menus, toolbar, dialogues and direct manipulation of the map area. These maps are viewable by the team as a whole and there is the facility provided to create links from added concepts to other relevant documents such as other concept maps, text, video, CAD models, etc. Storage of the concept map structure and the links embedded within the concept is implemented in XML.

The concept mapping technique has the ability to provide a precise organizational level, imparting a framework for subsequent media such as CAD solid models to be incorporated.

2.1.2 Data sharing and structuring module

Data have to be structured both logically and technologically to contribute to the success of a product's development. The data sharing and structuring module interacts with the concept mapping, providing the infrastructure for the sharing and structuring of data within the system.

Concept map structures and data related to material properties are integrated in the developed data sharing and structuring module. These concept maps can be parsed to produce a natural language report. Concept map data can be processed to allow configuration of the information for a specific report or user group, resulting in, for example, a bill of materials or a supplier list. Table 1 illustrates some of the wide range of the possible media, which can be context linked by the data structuring module. The processing of a concept map is capable of generating a table of design variables, which in turn forms the basis for building a CAD solid

Table 1 Linkable media types

Media	Linkable to
Concept maps	Design variable table, WHIP model, VRML, CAD solid model STEP AP203 file, video, graphics, Web pages, plain text
Design variable table	NA
WHIP .dwf (drawing web format)	WHIP model, video, graphics, Web pages
VRML world	VRML, video, graphics , Web pages
HTTP embedded graphics	Web pages, graphics, video, VRML
Web pages	Web pages, graphics, video, VRML
Plain text documents	NA

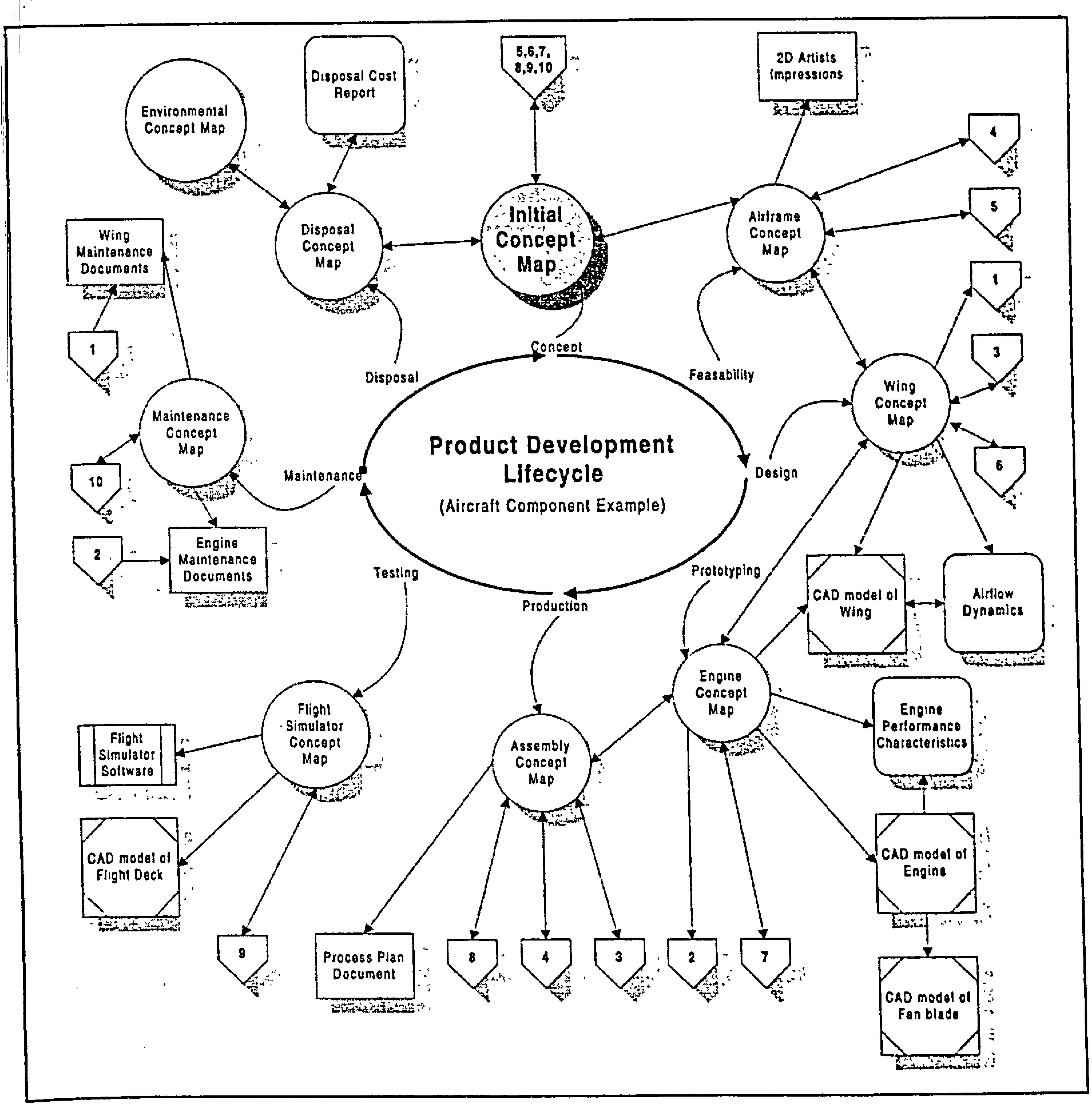


Fig. 2 Data structure connectivity within the product development life cycle

model or alternatively a mapping into a STEP application protocol (AP203).

The diagram in Fig. 2 illustrates a possible data structure that DISCS could accommodate, based on an example set of documents concerning a new aircraft concept. It highlights the structural connectivity and hence the navigation routes through which the information can be traversed; concept map structures can be connected to other purposeful documents such as maintenance documents. The diagram has been based on a more traditional lifecycle to show the data which would be created in a timeline fashion. However, the proposed system is more dynamic and connective in data retrieval.

The structural representation of concepts was carried out using the advanced information structuring language XML. XML is an ideal language to structure the concepts and their relationships. These relationships can exist between other concepts, other concept maps, VRML worlds, WHIP drawings for Web format models and Web pages. The extensible style language (XSL) in conjunction with XML provides a greater flexibility for the display of the data. A DTD (Document Type Definition) has been constructed with which XML can be checked for correctness and consistency. This mechanism is also implemented to exploit a level of control over the structure of the concept maps, should an industry, for example, find the concept maps spiralling out of control in terms of size and complexity. In this case a wide variety of limiting mechanisms can be put in place. This would force conciseness of the issue under discussion.

The data structuring module, together with its corresponding DTD, parses an XML file for structural

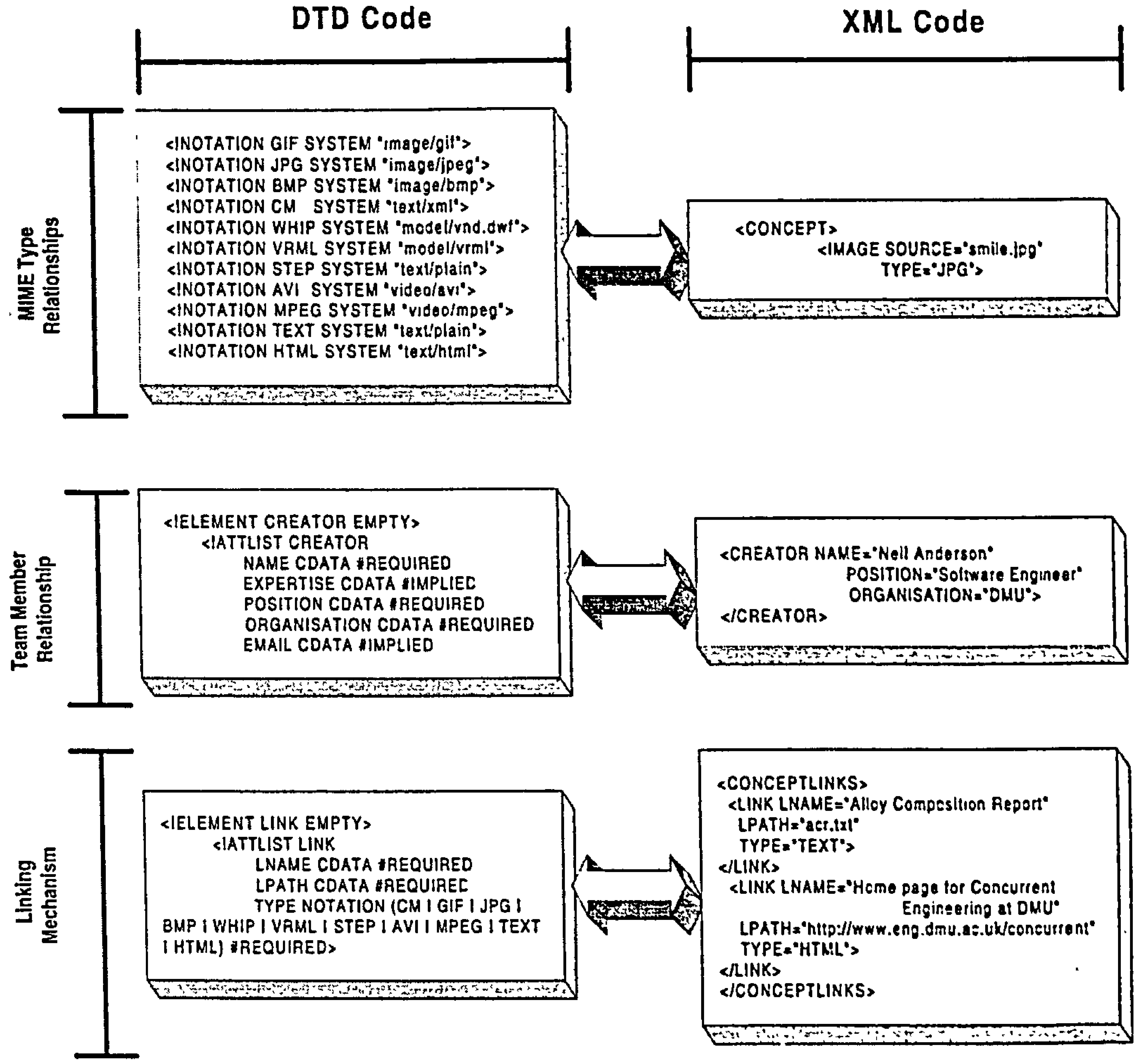


Fig. 3 DTD and XML relationships

integrity. XSL allows a different appearance to be placed on the concept maps. This can be utilized in the future to tailor the display to suit the user or media type. Interfacing to the system is achieved through use of the DTD. Other systems can interface to represent their information within a concept map structure defined by the DTD, which the system can then interpret and incorporate.

The relationship with the open DTD allows correctly formatted and well-formed XML data to be checked and validated as illustrated in Fig. 3. This shows examples of some of the supported MIME (Multipurpose Internet Mail Extensions) types, the team members and the linking support. MIME types are defined and are integral to the linking mechanism, facilitating the storage of related data together with the relevant concept. Concept tracing is facilitated through the structural ownership of a concept; a contact point can be found for each individual concept.

2.1.3 *Data standardization module*

The lifespan of product data is a vital issue to be tackled. The provision for product data to be effectively application independent is essential in today's climate of rapid software change. The utilization of standard technologies to develop the system greatly increases the lifespan of stored data. Structural meaning of the data is preserved outside the application's scope, preventing over time, as technologies change, the loss of access to legacy data.

Collaborative product development necessitates standardization of data using STEP. This research work has dealt with product data exchange to facilitate sharing of data at various complexity levels without conflict. There is a strong drive behind the philosophy to relate data to the products and processes that the data describe, as opposed to the computer systems, which create the data. Lifetimes of the application software, system software and the hardware, which were used to create the data, will be much shorter than that of the data. Application software has a typical lifecycle of 3–5 years, systems software 5–10 years and the hardware even less with cycles of less than 3 years [14]. Thus the integration of a standard for storage and interpretation within DISCS is a vital mechanism in overcoming the migration problems [15] which are rapidly occurring with legacy systems today. An *et al.* [16], Urban *et al.* [17], Peng and Trappey [18] and Loffredo [19] provide further details on STEP and its application.

Standardization applies not only to the data structures but also to the portability of the underlying modules. This is a highly important factor to consider where the deployment of such a system is likely to cross not only geographical boundaries but also operating environments. The technological foundations of the architecture have a solid basis in portable technologies. No matter how sophisticated a system is, it will provide little impact in the organizations concerned if it cannot work within the bounds of the lowest common technology denominator between organizations. Information technology investment in platforms cannot be disregarded but should be exploited [20].

2.1.4 Object-oriented database module

Object-oriented databases are ideally suited to the storage of complex product development data. A concept database and a product database are used within this research. The concept database stores the concept map XML data structures (Section 2.1.2) and the product database stores the STEP product data (Section 2.1.3). Each database is connected through links embedded within the concepts.

The embodiment of linking media types is further demonstrated in Fig. 4, which shows both the design iterations and the relationships to the object-oriented

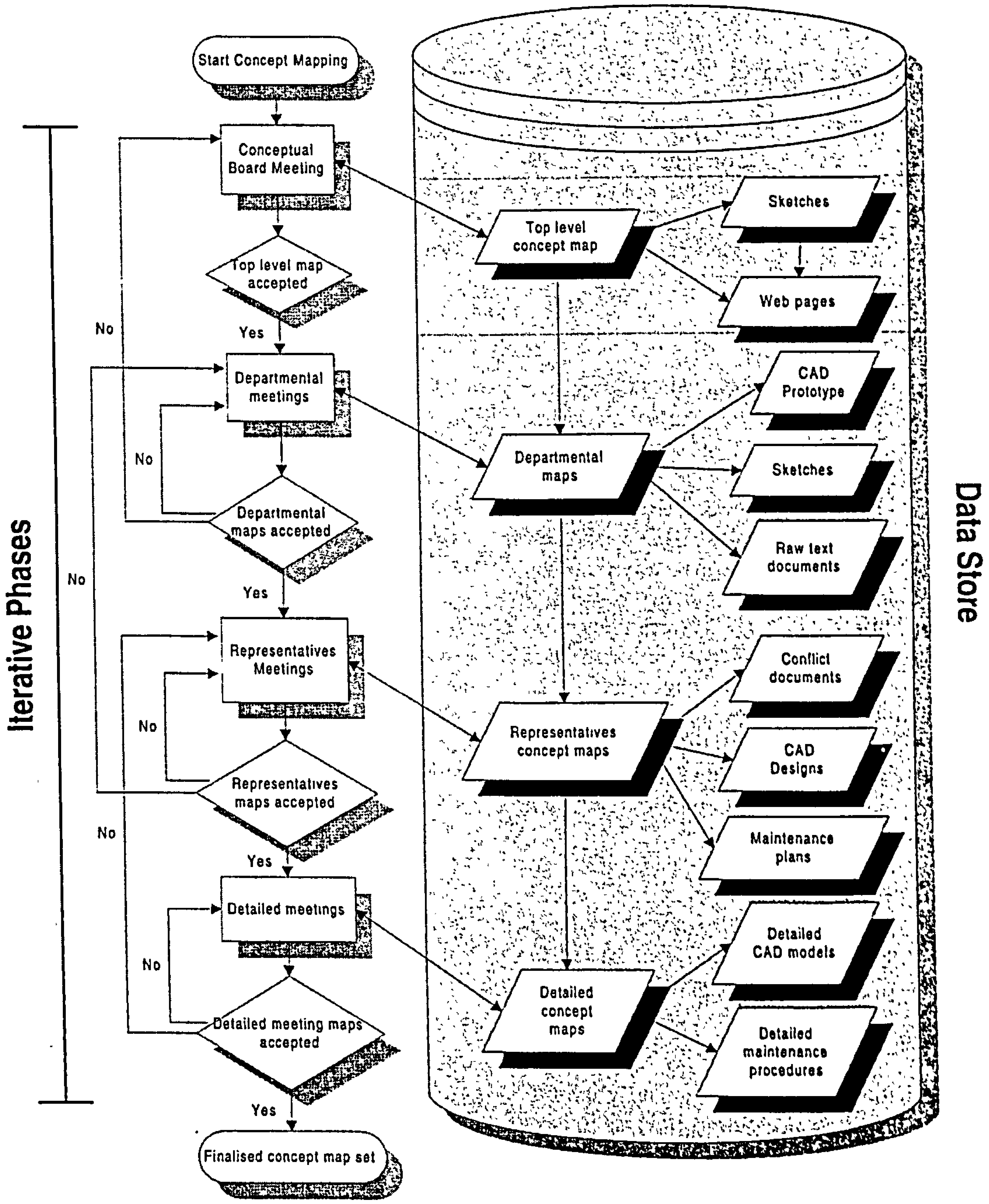


Fig. 4 Object-oriented database and information iteration flow

databases. It starts from the initial top level meeting through to a set of documents, taking into account the nested management infrastructures in today's organizations. The diagram is divided, showing the information flow on the left and the stored linked data model on the right.

A model of integrated knowledge which is capable of being shared through context-linked media is stored in the object-oriented databases. The information that is captured by various experts is information model driven. This allows navigation by users with a wide range of computer literacy levels and knowledge bases. The system allows all the information concerning the product during development and afterwards to be collated in a useful manner in a central repository for access by those concerned.

2.1.5 *CAD solid modelling module*

CAD solid models form an essential element in the data structure of a DISCS data framework. They provide the consolidation of data from iterations of concept mapping. To assist with the interface to STEP AP203, Unigraphics CAD solid modelling version 16 has been used in this research.

2.1.6 *Conflict resolution module*

Conflict resolution throughout product development is one of the purposes of shared communication. Conceptual analysis of different team member viewpoints allows conflicts to be identified at the earliest opportunity. It provides a basis with which to reach a consensus.

Concept map structures are checked by a constraint agent, to alert the creator of a concept to any conflicts within the constraints defined by it and other existing concepts. A link to CMS (Cambridge Material Selector) is provided to allow the inclusion of material properties as constraints. Concept maps form a hierarchy; those that define constraint parameters are analysed for conflict with parent concepts. Notification of a conflict occurs both at the time of a conflicting concept to the user creating the concept and via e-mail to the user who created the opposing concept.

Consistency validation compares a design variable tolerance consensus set with a user-generated CAD model. This provides conflict alert support for those without STEP capability for whom the design variables are propagated directly to an AP203 format.

2.1.7 *HTTP accessed documents*

The system has been developed to utilize the growing base of technologies for the Web. Within the previously mentioned linking mechanism is the facility to input uniform resource locations (URLs) to Web-based documents. The connection from concepts to other Web-based data utilizes the status of the Web today. Suppliers are likely to have a corporate Web site allowing their details to be incorporated in concepts that relate to their products. There is the facility to incorporate other Web-based systems. Within the developed system is the capability to display CAD solid models via a Web-enabled format (drawing web format, .dwf). This enables discussion for those without expensive CAD solid modelling software.

Utilization of URLs helps the wide range of developed data formats to be accessible through DISCS, increasing the scope for information structuring, which is usually limited by Web browser technology.

2.2 The concept mapping scenario

The developed system was evaluated using products from the automotive sector. The initial level of concept mapping for a car is demonstrated. However, because of the complexity of this product, a practical component has been selected for detailed demonstration in this paper. An example of how conceptual design information is made available to a distributed product development team and coherently represented in context is presented in this section.

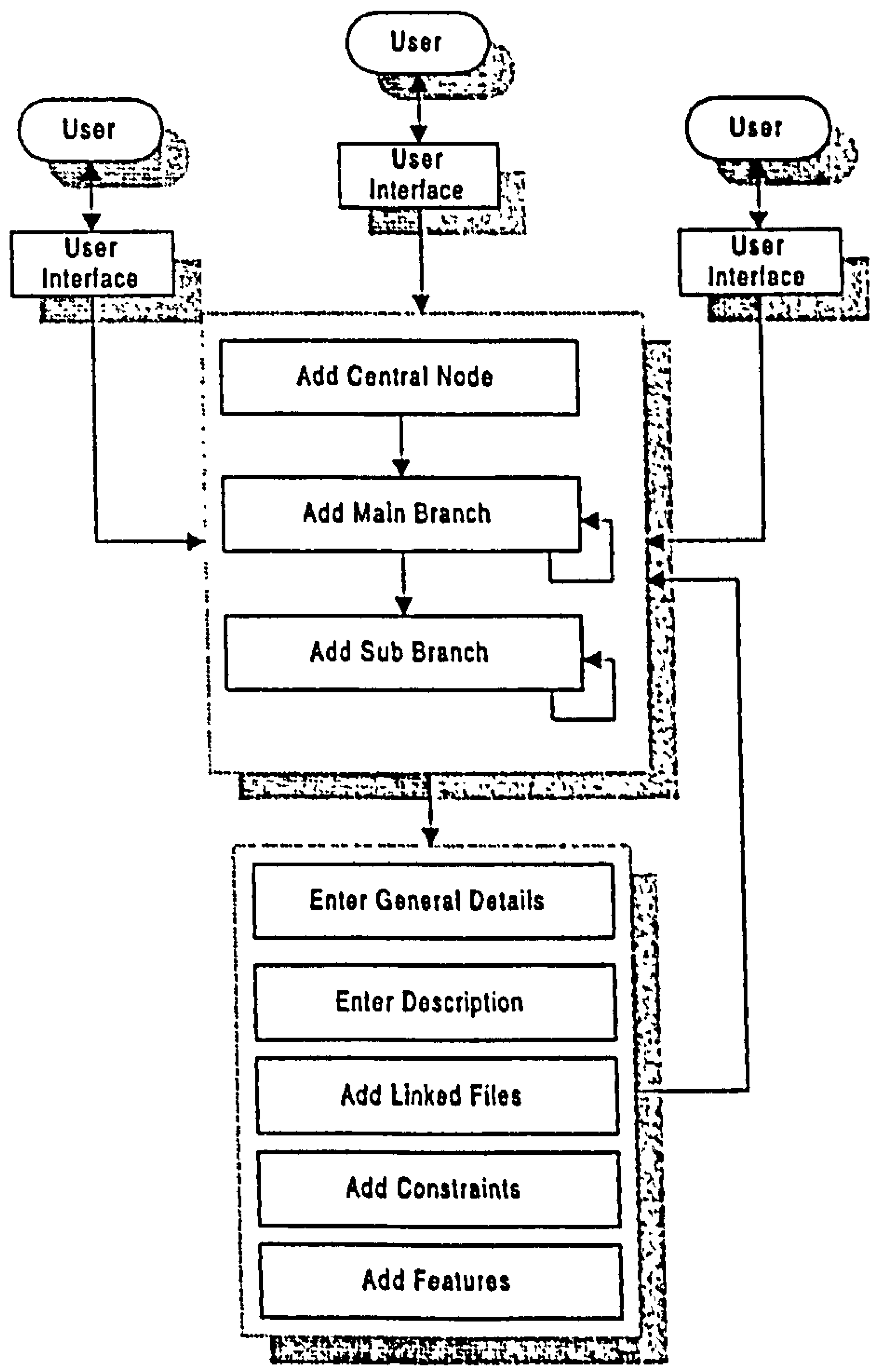


Fig. 5 Concept mapping procedures

2.2.1 *Concept mapping procedure*

The procedure for building a concept map is illustrated in Fig. 5. A concept map is built in the following order:

(a) central node (defines the intended scope of map);
(b) main branches (define relevant subtopics);
(c) subbranches (either textual or graphical).

Subbranches are added recursively until the configuration limits of the map are reached. Each node in the map provides a combined series of dialogues featuring a collection of information concerning the following:

(a) concept details,
(b) description,
(c) linked files,
(d) constraints,
(e) features.

The concept details concern its represented name displayed in the map, contact points and the creation date along with the node's style. 'Description' contains a textual report of any comments concerning the concept. 'Linked files' provide a list of URLs which are relevant to the concept and can be retrieved by clicking on them. Constraints can be added, which form the basis for restrictions on other concepts downstream in the hierarchy, and result in design variables for input into a STEP AP203 file. 'Features' refers to a list of features related to a specific component or a product. This list is linked in the same manner as the list of files and provides short cuts to concepts which are related to subassemblies or final form features.

2.2.2 *Concept mapping level 1*

The concept mapping module allows users of appropriate authority to register for a concept mapping based discussion. Figure 6 shows three team members using an integrated chat forum alongside a concept map. The interface is based on Java Swing technology, which allows a greater range of interface components to be used. The toolbar docked on the right-hand side can be removed and placed, for example, at the top or outside the browser. If the focus of the group is on the concept map, the panels drag bars that can be seen to allow extra space as appropriate.

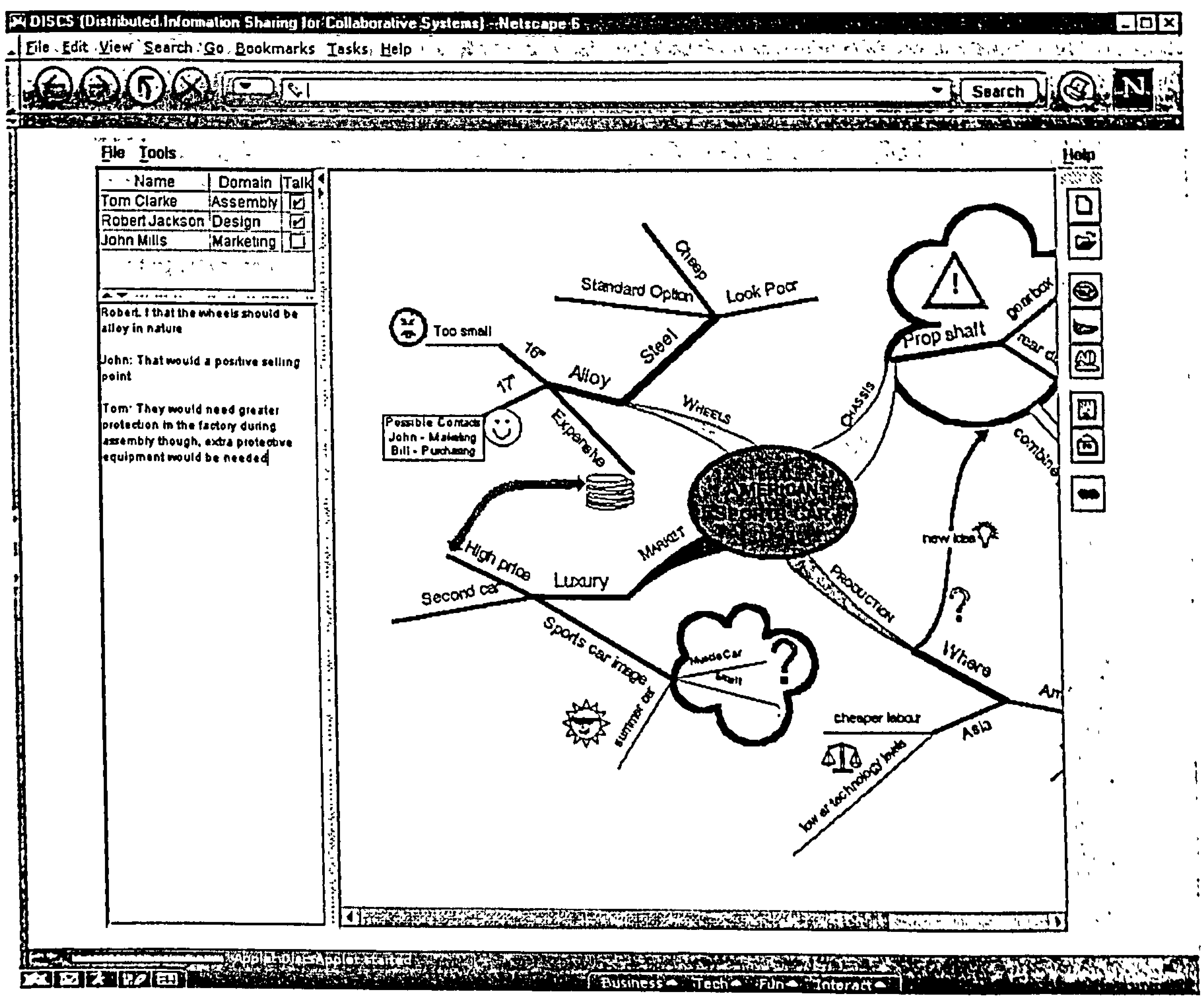


Fig. 6 Web-based concept mapping interface

mechanism can be seen later in Fig. 11. The concept map in Fig. 8 represents a feature-based hierarchy. Each concept in Fig. 8 represents a feature that is in turn linked to a dialogue.

2.2.4 *Concept mapping level 3*

The capability of the system is shown through the conceptual communication and storage of product data concerning a connecting rod. The component consists of a variety of features, holes, rounds, fillets and pockets. These features describe the small end, shank, big end and big end cap of the connecting rod. A selected representation of these features is shown in Tables 2 to 5.

Dialogues to facilitate standardized collaboration are associated with each node in a concept map. Depending on the context of the node various input fields are completed. The series of dialogues in Figs 9 to 12 represent the concept detail stored in the node describing the attributes of the 'through hole: TH0001' as seen previously in Table 2. The features dialogue is not required as TH0001 is a bottom level concept; there are no further subfeatures to include.

The conceptual dialogues allow both simple and detailed information to be included at the point of relevance. Navigation through the concept maps provides efficient and rapid location of data sets. Once the conceptual node of relevance has been located other non-concept map files can be retrieved as demonstrated in Fig. 11. Concept constraints, as shown in Fig. 12, allow a concept to represent a feature in a product assembly as in the example shown. These constraints can be processed as explained in Section 2.1.6.

2.2.5 *Results from the system*

The resulting output from the system represents a consensus, across the supply chain, on the details concerning the recommended component. The output illustrated can be viewed in a variety of forms, selecting from

Table 2 Small end

Feature name	Dimensions (mm)	Feature code	Tolerance (mm)	Surface finish (μm)
Through hole	Diameter: 37 Depth: 26.27	TH0001	+0.021 −0.000	0.6
Through hole	Diameter: 4 Depth: 2.5	TH0002	+0.070 −0.000	0.8
Fillet (small rounded)	Radius: 8	SRF0001	+0.500 −0.000	6.3

Table 3 Shank

Feature name	Dimensions (mm)	Feature code	Tolerance (mm)	Surface finish (μm)
Rounded pocket	Diameter: 12 Radius: 20	RP0001	+0.500 −0.000	6.3

Table 4 Big end

Feature name	Dimensions (mm)	Feature code	Tolerance (mm)	Surface finish (μm)
Double corner	Diameter: 62	DC0001	+0.190 −0.000	0.6
Rounded corner	Diameter: 80	RC0001		6.3
Rounded corner	Diameter: 80	RC0001		6.3

Table 5 Big end cap

Feature name	Dimensions (mm)	Feature code	Tolerance (mm)	Surface finish (μm)
Double corner	Diameter: 62	DC0002	+0.190 −0.000	0.6
Thread hole	Diameter: 19 Depth: 42	TRH0001	+0.01 −0.000	0.1
Thread hole	Diameter: 19 Depth: 42	TRH0002	+0.01 −0.000	0.1

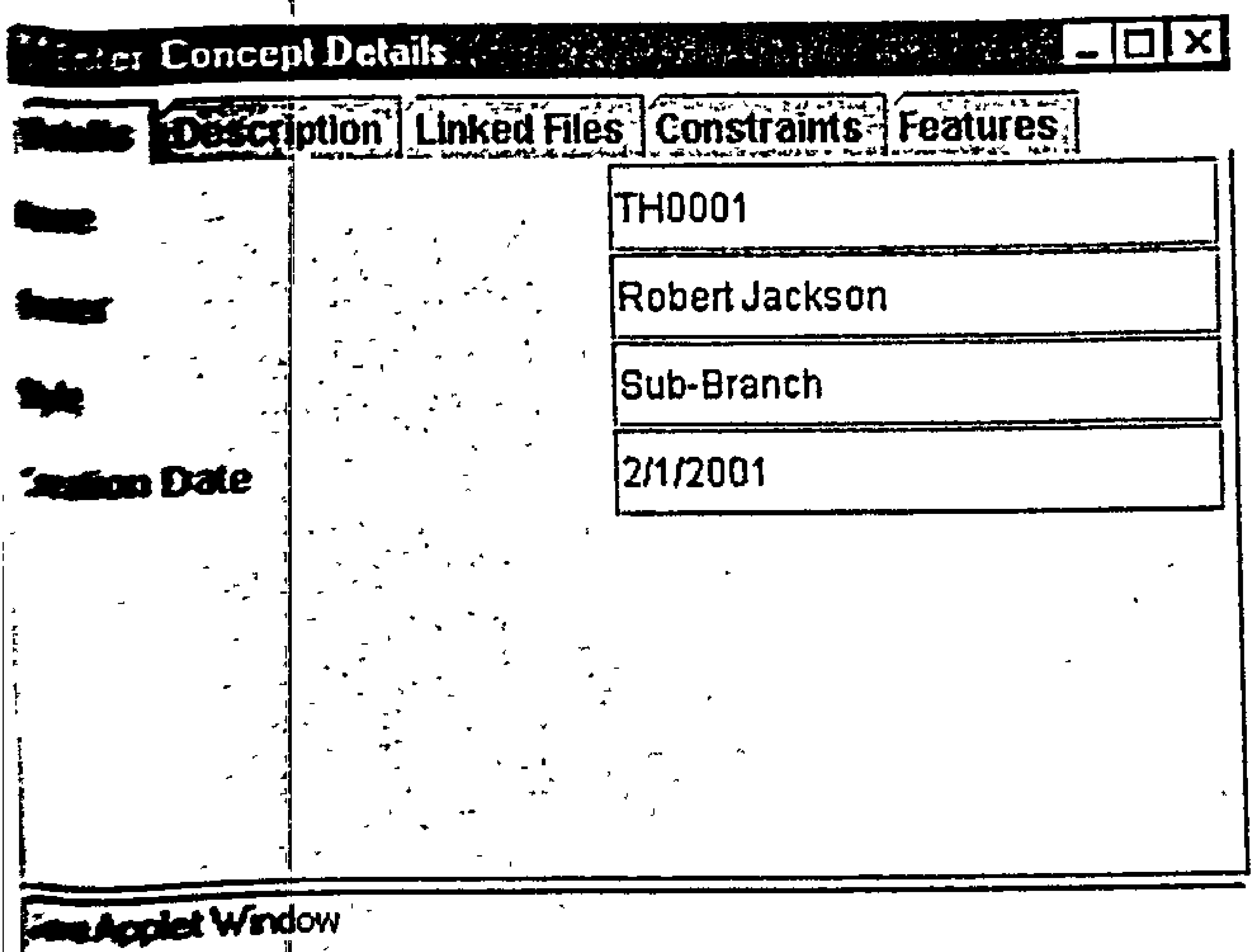


Fig. 9 TH0001 details

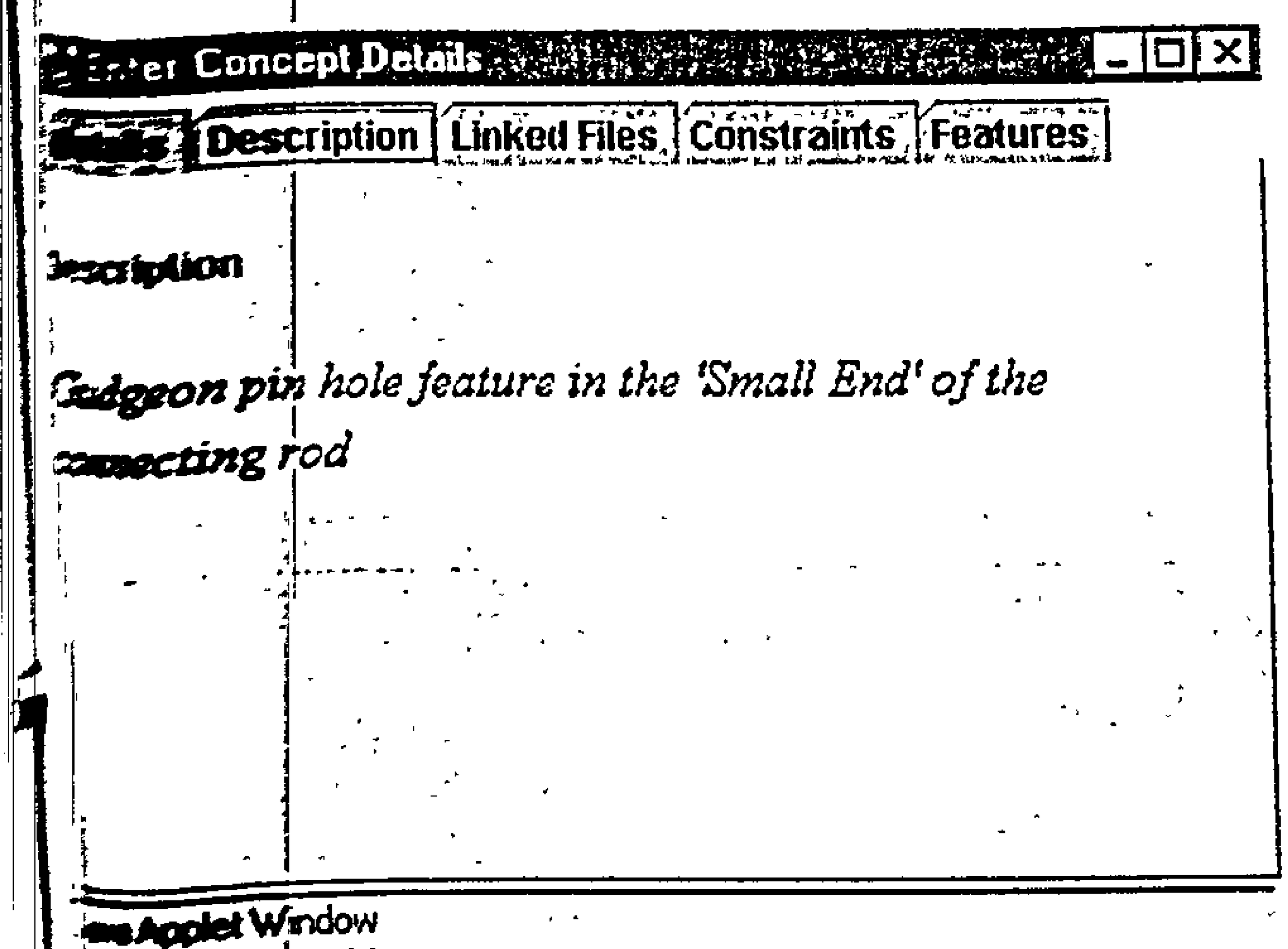


Fig. 10 TH0001 description

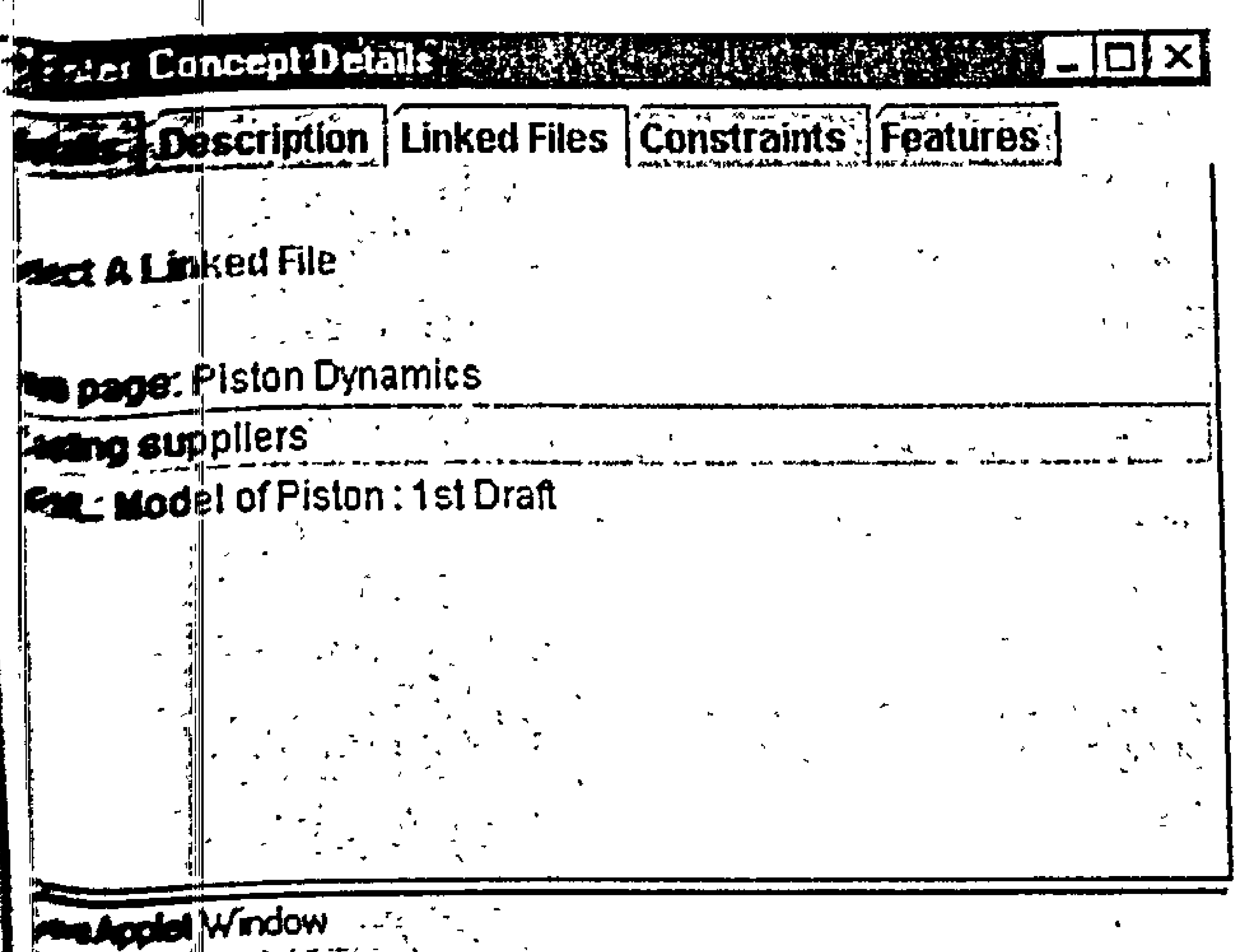


Fig. 11 TH0001 linked files

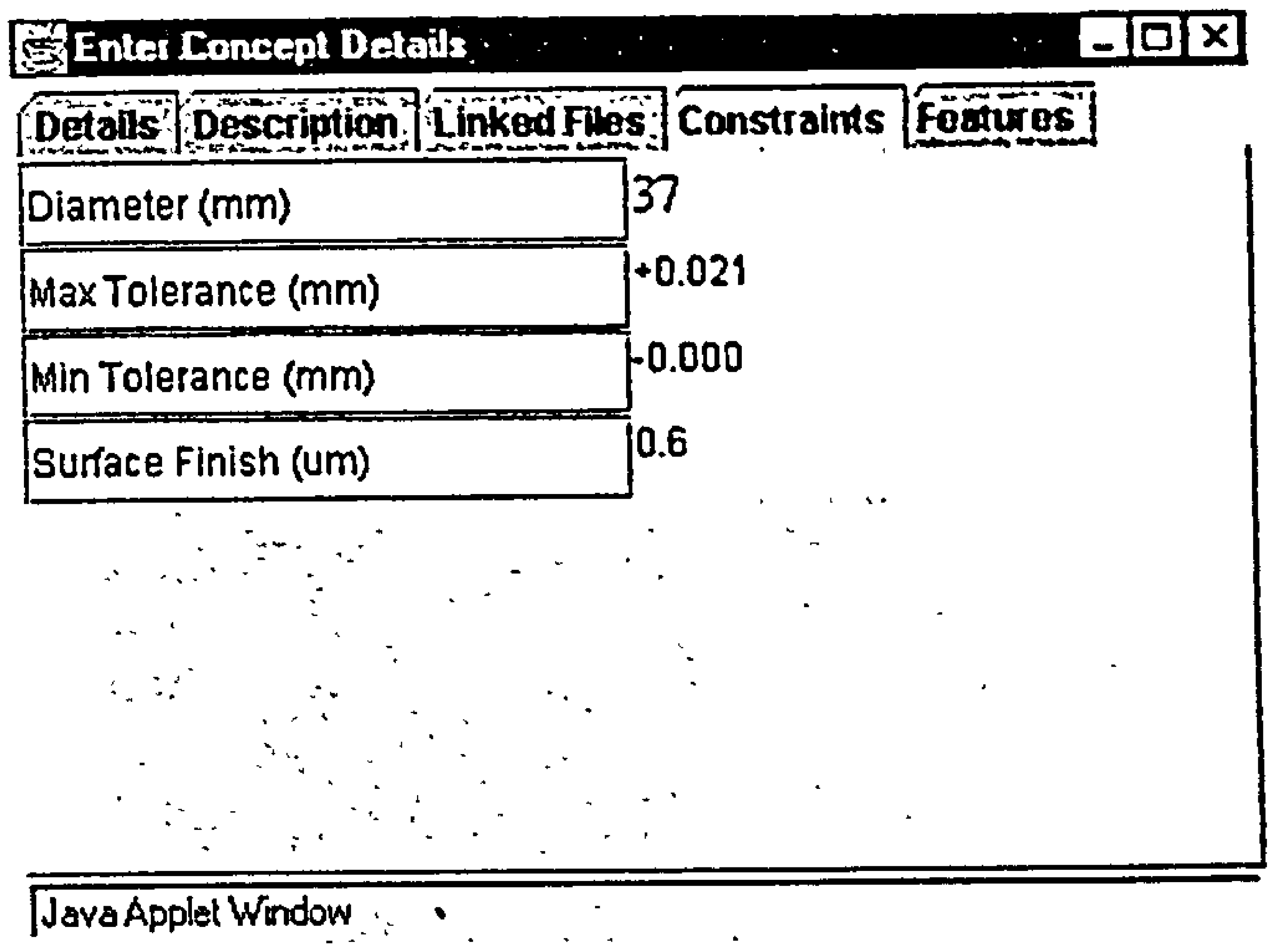


Fig. 12 TH0001 constraints

multimedia displays or focused on a single represented format such as a concept map.

Effective collaboration as demonstrated here is facilitated through sharing of a linked framework of information multimedia components. A complete multimedia perspective with concept map, concept details and a VRML models can be seen in Fig. 13.

3 CONCLUSIONS

The system presented in this paper shows both a solution to the problem of sharing information in a distributed product development team and the work undertaken in making such a solution a reality. It is a solution requiring consideration and knowledge drawn from a variety of domains. The problem of distributed information sharing needs to be tackled from a wider perspective than considered by previous efforts. Benefits can be gained through the limited scope of systems put forward in the past. Real gains can only be achieved through an expansive and fully integrated system. There is strong need for project managers to employ an adequately considered information technology solution. The technologies chosen must take suitability, portability, ease of use, future change and expandability into account.

Real asynchronous and synchronous communications are the key, supported by a framework which is closely tailored to human natural psychological infrastructure. Such a framework then allows the systems function to be expanded beyond sharing information during development. The data are in a suitable structure for maintenance personnel to access and for training purposes, saving both time and money in collating data for these future tasks. The nature of the concept/constraint based framework essentially functions properly as an information gathering, collating, checking, formatting, structuring, disseminating and storing tool.

This system offers the facilities to create information repositories that have real-world value, throughout the entire product development lifecycle. Through accurate enhanced information sharing these repositories offer the chance to drive out problems and inconsistencies

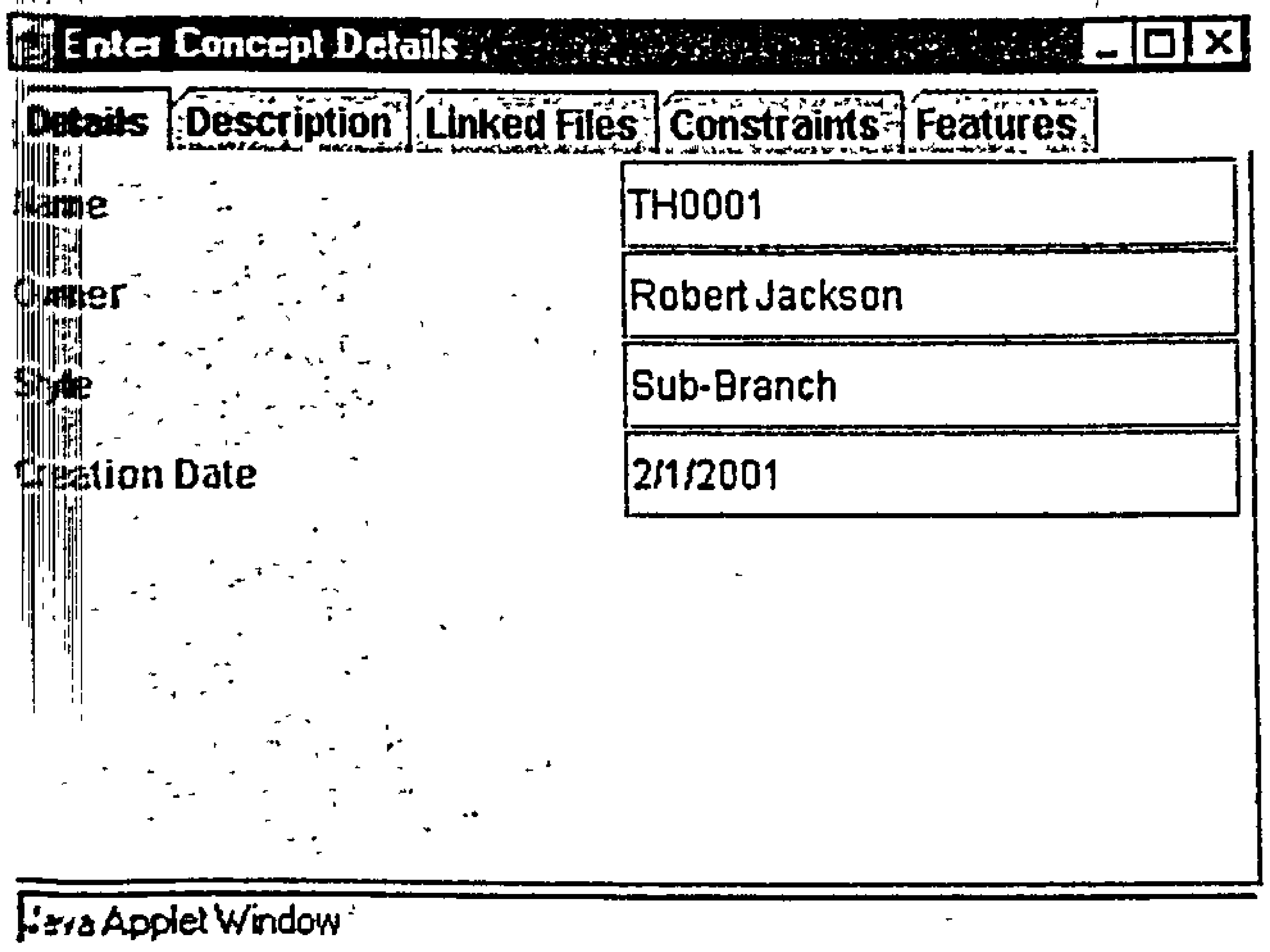


Fig. 9 TH0001 details

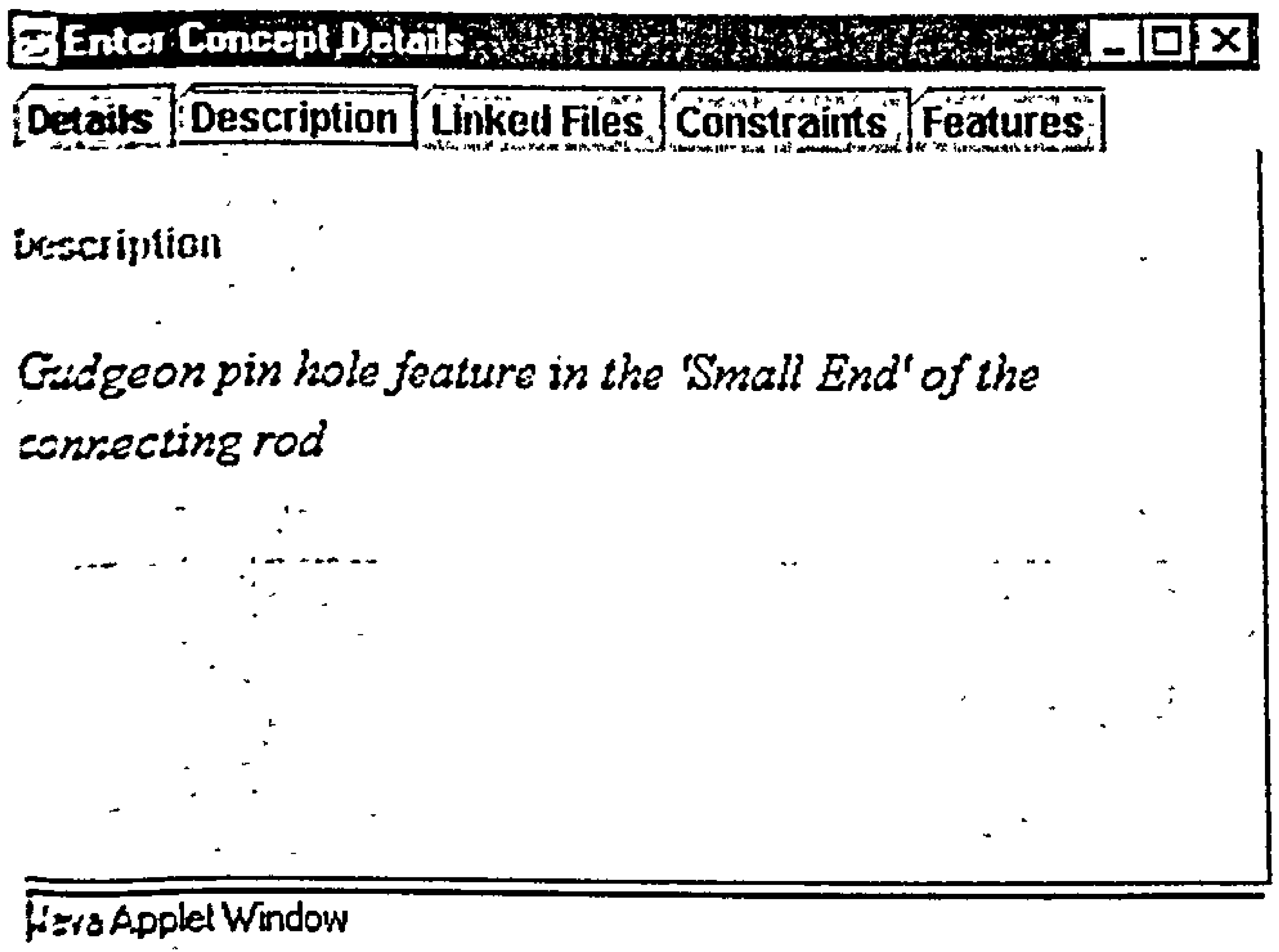


Fig. 10 TH0001 description

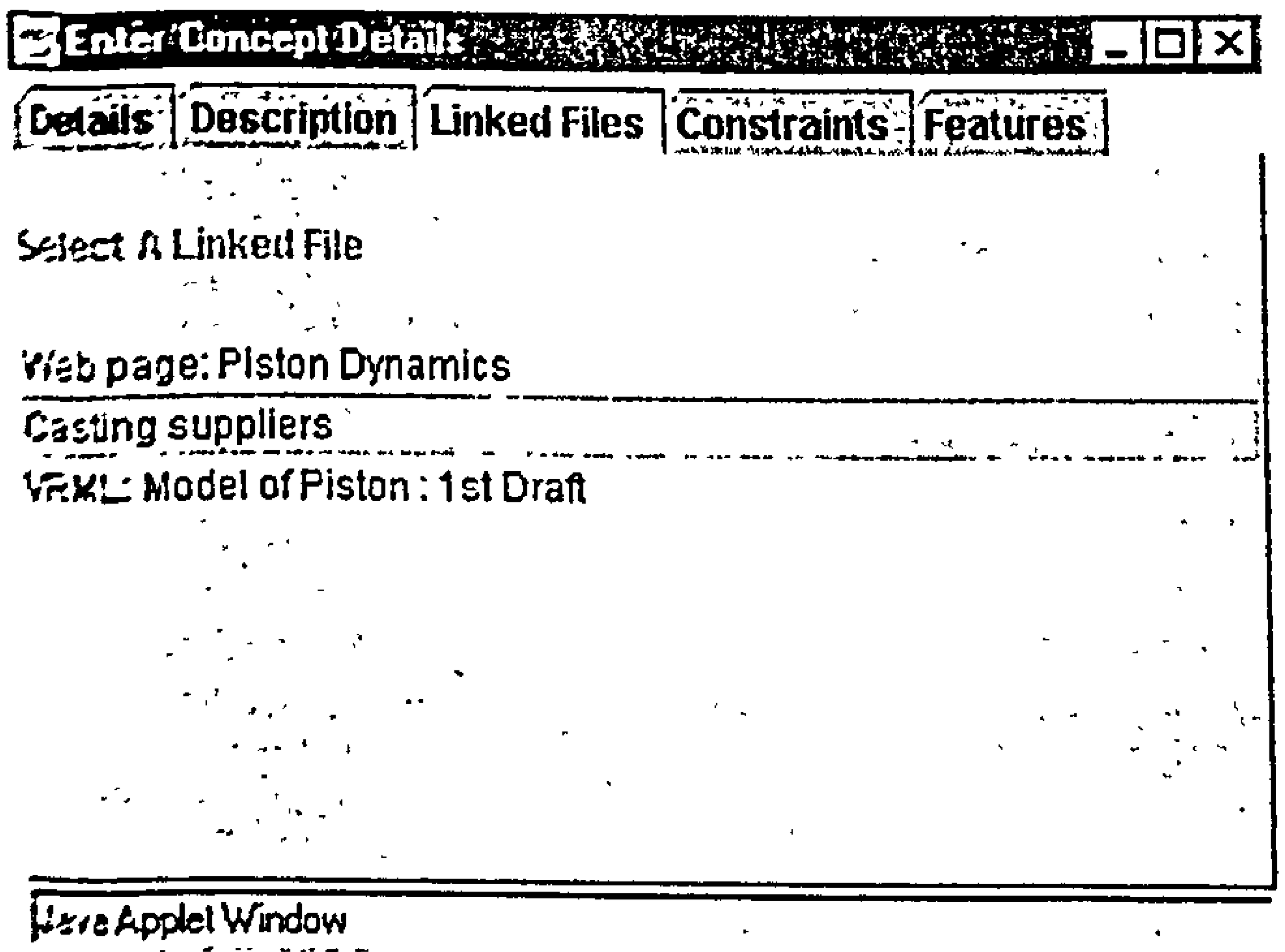


Fig. 11 TH0001 linked files

multimedia displays or focused on a single represented format such as a concept map.

Effective collaboration as demonstrated here is facilitated through sharing of a linked framework of informative multimedia components. A complete multimedia

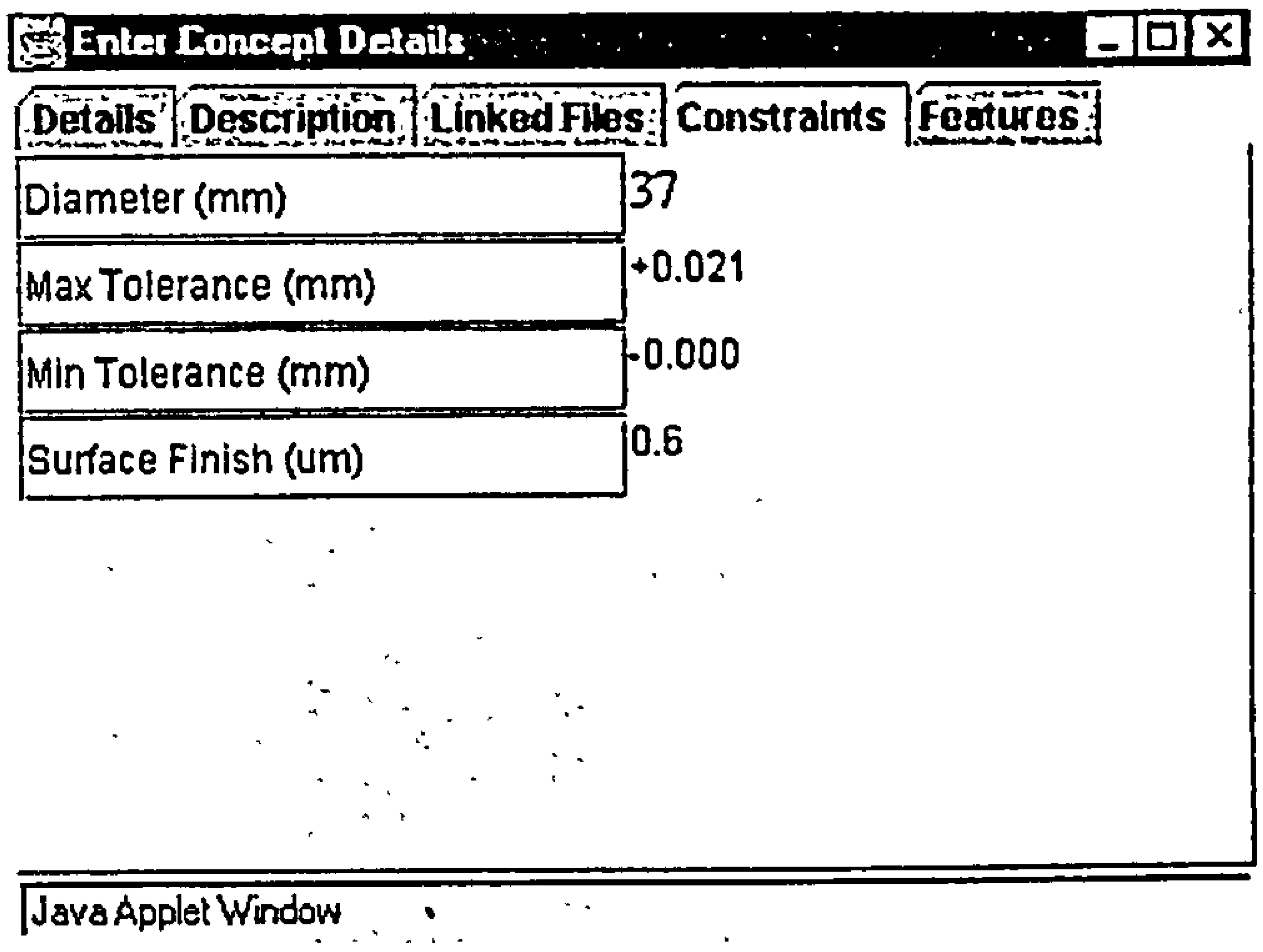


Fig. 12 TH0001 constraints

perspective with concept map, concept details and a VRML models can be seen in Fig. 13.

3 CONCLUSIONS

The system presented in this paper shows both a solution to the problem of sharing information in a distributed product development team and the work undertaken in making such a solution a reality. It is a solution requiring consideration and knowledge drawn from a variety of domains. The problem of distributed information sharing needs to be tackled from a wider perspective than considered by previous efforts. Benefits can be gained through the limited scope of systems put forward in the past. Real gains can only be achieved through an expansive and fully integrated system. There is strong need for project managers to employ an adequately considered information technology solution. The technologies chosen must take suitability, portability, ease of use, future change and expandability into account.

Real asynchronous and synchronous communications are the key, supported by a framework which is closely tailored to human natural psychological infrastructure. Such a framework then allows the systems function to be expanded beyond sharing information during development. The data are in a suitable structure for maintenance personnel to access and for training purposes, saving both time and money in collating data for these future tasks. The nature of the concept/constraint based framework essentially functions properly as an information gathering, collating, checking, formatting, structuring, disseminating and storing tool.

This system offers the facilities to create information repositories that have real-world value, throughout the entire product development lifecycle. Through accurate enhanced information sharing these repositories offer the chance to drive out problems and inconsistencies

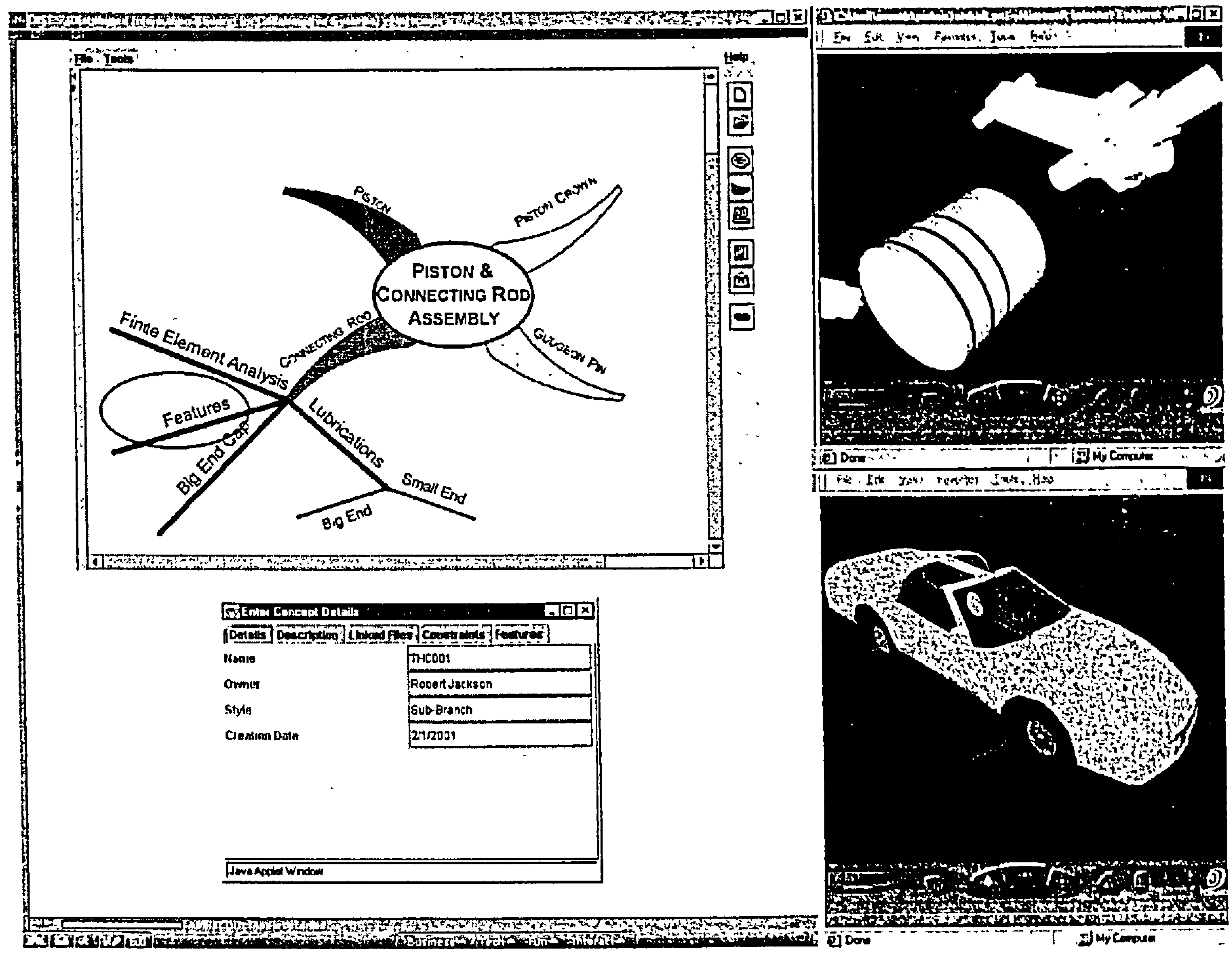


Fig. 13 Snapshot of the system results

during the product development lifecycle. Hence they lower costs, decrease turnaround times, increase team-building opportunities, reduce time to market and increase product applicability and quality.

Although the system was originally intended for product development, it is also applicable to other domains such as patient data in health care and film/television production. The framework allows people to construct, find and understand the information they are looking for quickly, an ideal way to collate work on an upcoming operation for busy surgeons incorporating three-dimensional models of magnetic resonance imaging scans or for a film crew, distributed around the world, working on different scenes at the same time with relationships to computer-rendered special effects models.

REFERENCES

1 **Zhuang, Y.**, **Chen, L.** and **Venter, R.** CyberEye: an internet-enabled environment to support collaborative design. *CE Concurrent Engng: Res. Applic.*, September 2000, 8(3), 213–229.

2 **Chang, H.**, **Lu, W. F.** and **Liu, X. F.** WWW-based collaborative system for integrated design and manufacturing. *CE Concurrent Engng: Res. Applic.*, December 1999, 7(4), 319–334.

3 **Abdalla, H. S.** Concurrent engineering for global manufacturing. *Int. J. Prod. Economics*, 1999, (60–61), 251–260.

4 **Pallot, M.**, **Maigret, J. P.**, **Boswell, J.** and **de Jong, E.** EPICE: realising the virtual project office. In 6th International Conference on *Concurrent Engineering*, June 2000, pp. 57–70.

5 **Buzan, T.** and **Buzan, B.** *The Mind Map Book, Radiant Thinking, The Major Evolution in Human Thought*, 1995 (BBC Books, London).

6 **Branki, C.** The acts of cooperative design. *CE Concurrent Engng: Res. Applic.*, September 1995, 3(3), 237–245.

7 **Kim, C.-Y.**, **Kim, N.**, **Kim, Y.**, **Kang, S. H.** and **O'Grady, P.** Distributed concurrent engineering: internet-based interactive 3-D dynamic browsing and markup of STEP data. *CE Concurrent Engng: Res. Applic.*, March 1998, 6(1), 53–70.

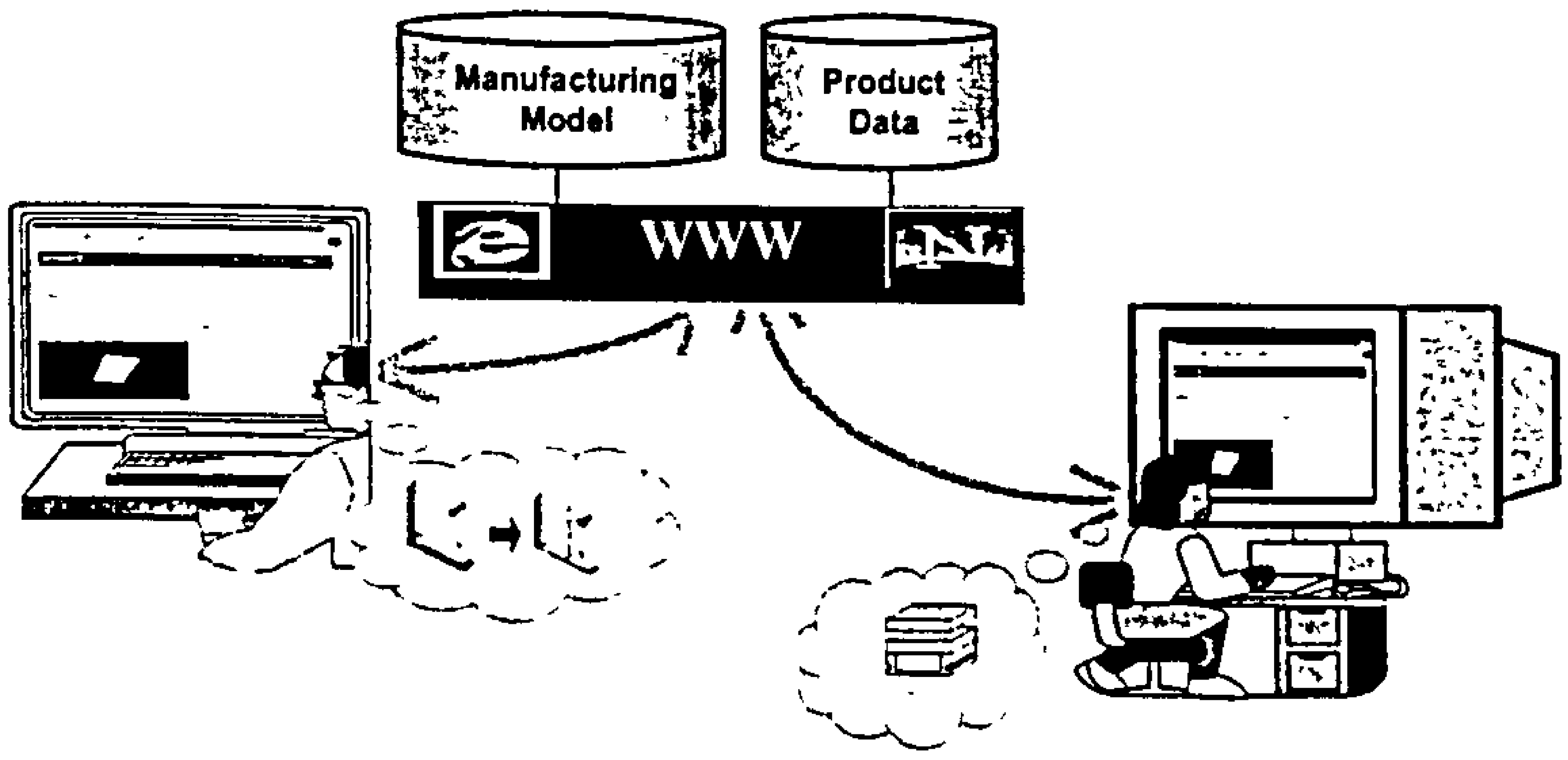
8 **Vergeest, J. S. M.** and **Horváth, I.** GEOS-based analysis to determine the feasibility of engineering data sharing. Subfaculty of Industrial Design Engineering, Delft University of Technology, Delft, The Netherlands, 1998.

9 **Srinivas, S.**, **Reddy, R.**, **Babadi, A.**, **Kamana, S.**, **Kumar, V.** and **Dai, Z.** MONET: a multi-media system for conferencing and application sharing in distributed systems. CERC Technical Report Series Research Note, February 1992.

10 **Jagannathan, V.**, **Karinthi, R.**, **Raman, R. S.**, **Montan, V. R.** and **Petro, J. J.** Information sharing system. CERC Technical Report Series Technical Memoranda, July 1992.

11 **Downs, R. M.** and **Stea, D.** *Maps in Minds: Reflections on Cognitive Mapping*, 1997 (Harper and Row, New York).
12 **Jagannathan, V., Karinthi, R., Sobolewski, M.** and **Almasi, G.** Model-based information access. CERC Technical Report Series Technical Memoranda, April 1993.
13 Autodesk WHIP Plugin^{TM}. Available from http://www.autodesk.com, accessed 14 December 2000.
14 **Fowler, J.** *STEP, for Data Management Exchange and Sharing*, 1995 (Technology Appraisals).
15 **Hares, J. S.** *Object Orientation: Technology, Techniques, Management and Migration*, 1994 (John Wiley, New York).
16 **An, D., Leep, H. R., Parsaei, H. R.** and **Nyaluke, A. O.** A product data exchange integration structure using PDES/STEP for automated manufacturing applications. *Computers Ind. Engng*, 1995, **29**(1–4), 711–715.
17 **Urban, S. D., Shah, J. J.** and **Rogers, M. T.** Engineering data management achieving integration through database technology. *Computing Control J.*, June 1993, **4**(3), 119–126.
18 **Peng, T.-K.** and **Trappey, A. J. C.** A step towards STEP compatible engineering data management: the data models of product structure and engineering changes. *Robotics Computer-Integrated Mfg*, 1998, **14**(2), 89–109.
19 **Loffredo, D. T.** Efficient database implementation of EXPRESS information models. PhD thesis, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, 1998.
20 **Anderson, N.** and **Abdalla, H. S.** Distributed information sharing for collaborative systems (DISCS). In *Information Visualisation '99*, 1999, pp. 476–481 (IEEE Computer Society).

Advances in CONCURRENT ENGINEERING



EDITED BY
R. ROY

SERIES EDITOR
B. PRASAD

PRODUCED BY
C. RUSH

ISBN: 09710461-0-7

www.ceconf.com

SPONSORED BY

CERA Journal

Journal of Production Planning and Control

CONCURRENT ENGINEERING: Research and Applications

CETEAM

EsoCE

IEE Manufacturing

International Journal of Engineering Simulation

Spec2Market Solutions

www.ceraj.com www.ceteam.com

Advances in CONCURRENT ENGINEERING CE2001

Project Management in CE
Collaborative Decision Making in CE
Internet-based Modeling in CE:
Collaborative Product Commerce in CE
Design Technologies in CE
Extended Enterprise and CE
Networking and Distribution in CE
Knowledge Engineering and Management of CE
Frameworks for Information Sharing
Cooperative Internet Computing
CE Education and Training
CE Techniques
Standards in CE
CE in Virtual Environment
CE in Construction
Virtual Prototyping in CE
Virtual Enterprise Engineering in CE

PRESENTED AT
EIGHTH ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING: RESEARCH AND APPLICATIONS

West Coast Anaheim Hotel, Anaheim, California, USA
July 28th – August 1st 2001

SPONSORED BY
THE INTERNATIONAL SOCIETY FOR PRODUCTIVITY ENHANCEMENT (ISPE)
CERA JOURNAL
THE IEE MANUFACTURING DIVISION
SPEC2MARKET SOLUTIONS
INTERNATIONAL JOURNAL OF ENGINEERING SIMULATION
JOURNAL OF PRODUCTION PLANNING AND CONTROL
EsoCE: EUROPEAN SOCIETY OF CONCURRENT ENGINEERING
CETEAM

ISBN: 09710461-0-7

Web Browser Based Collaborative Product Development for the 21st Century

N.Anderson BSc and H. Abdalla BSc, MSc, Ph.D., CEng, FIMechE
Department of Design Management and Communication, De Montfort University, Leicester, Leicestershire, LE1 9BH

Abstract

Organisations in the 21st century are facing increasing global competition. Previous geographical market boundaries are under increasing pressure as both suppliers and consumers have access to a greater range of potential options of an international nature for their custom. There is an increasing need for systems and information sharing techniques and technologies to be deployed within the collaborating frameworks of today's world-class global organisations. The focus is to enable them to interact with the effectiveness capable of providing product development benefits, which in turn ensure a competitive advantage. This paper presents such an approach to the problem of effective product development information sharing. The work undertaken at De Montfort University is aimed at providing a solution, which encompasses the entire product development lifecycle from the initial point of undertaking through to disposal.

Keywords: Concept mapping, virtual collaboration, STEP

1 Introduction

The harsh global market environment of today places an immense pressure on organisations to attain the higher echelons of agility and responsiveness to change. Current tools to assist in providing organisations with a mechanism to deal with such pressures fall short in terms of the key to the problem, integration. The point of greatest saving is as early as possible within the product development lifecycle. Problems discovered early in the design process within the first 20% of the cycle time are easier to solve than those in the later stages [1]. Much of the work undertaken elsewhere thus far deals with a restricted scope of the product development lifecycle usually concerned with later stages and is hence of reduced potential benefit. In essence a system is required which provides integration of the technologies and information that is shared from the earliest point in the product development lifecycle.

Concurrent Engineering is defined as a systematic approach to the integrated concurrent design of products and their processes, including manufacture and support. This approach is intended to cause the developers, from the outset to consider all elements of the product lifecycle from concept through to disposal, including quality, cost, schedule, and user requirements [2].

Harnessing the skills a distributed organisation has within its work force to the maximum of their potential is an inherently difficult task, the greater the skill set the employee has the greater the demands on their time and the less they can be afforded the time to travel to meetings. Trying to involve such essential team members in a greater number of projects therefore requires an enabling system for them to communicate effectively to their peers. The successful communication of information requires various factors to be in place, accuracy, context, qualitative content, richness, diversity, timeliness, availability and most importantly comprehensibility. A system is required which enables a wide spectrum of computer and technically literate people to contribute to the product development lifecycle. The work within this paper is focused upon a system which allows data to be built upon and shared in a cohesive integrated manner from the conceptual stages through evaluation and prototyping and design into production and through to disposal. The data can be accessed during any stage through a variety of systems in a multitude of mechanisms, providing the best possible access to and understanding of the information.

The core rationale of the work lays in the integration of technologies to provide solid information sharing support from the initial foundations and concepts of development through to the final disposal of the product. Data is more than just shared it is structured within a manner such as to enhance the comprehension and conception between team members.

Business orientation within the project is directed by a consequential reduction in costs, a reduction in time to market, and improvements in both business processes and product applicability. Costs are reduced via a variety of benefits to the utilising team; the enhanced communication brings about greater efficiency of understanding, speed of realisation and of improvements in processes. These assist in avoiding the problems, which often bring about modifications later on in the lifecycle. These changes are costly to make and would be reduced with effective early communication of ideas and requirements. The time to market is effectively reduced in much the same manner, with a reduction in time consuming changes later in the lifecycle the overall development period will be decreased.

Business processes are placed under increased scrutiny within such a communication medium, problems are highlighted with greater ease and there is sufficient means for discussion to resolve them.

There is void to be filled with software based tools, which provide the facilities to create information repositories that have real world value, throughout the entire concurrent engineering lifecycle, these information repositories offer benefits in terms of reduced costs, decreased turn around times, increased team-building qualities, a reduced time to market and an increase in product applicability and quality. Hence improving both product and business processes.

1.1 Background

Previous systems within the problem domain of distributed information sharing often ignore the aspects of integration and portability, which vastly increase the utility of both the system and its data. One of the more current works, CyberEye [3] at the University of Toronto puts forward a system based on current technologies such as Java, servlets, and ASP (Active Server Pages). However this and previous systems today are missing a tangible coherent thread through their data. CyberEye is an information sharing system with real-time features, however it is essentially still a shared repository, there is little innovation in increasing the understanding of the data and an improvement in the real qualitative communicational value of such shared data. There are no granularity considerations, the model is shared as a whole at any one time, there is little facilitation for people to work concurrently on sub sections of the representation, surely a primary goal of any information sharing product development tool.

Other modern systems provide a narrow solution to the problem such as [4], a system for sharing constraint based information within a product development team, which is essential within a tool but needs to be integrated within a complete framework describing the product. Without such a framework in whichever area the past projects have concentrated in, the data is frequently retrieved out of a context reducing the usefulness of the content. Tackling the whole product development lifecycle is a challenging problem but a fully integrated system has greater potential for impact on the desirable performance characteristics of a virtual product development team than disparate non-integrated sub systems.

The IMS-Test Case 3 [5], Global Concurrent Engineering the results from research into the best practices of an international consortium of companies from a wide range of business sectors including automotive, aerospace, telecommunications, shipbuilding, and information technology. The research indicated the following factors are required for success, effective communication, and a systematic involvement of customers, suppliers, distributors, powerful information infrastructure, and effective use of modern technology. The work undertaken is utilising modern technologies to provide a communication framework to enable advanced involvement of the organisations populous considered relevant to the product development. EPICE (Electronic commerce for sharing Program management Information in the virtual Concurrent Enterprise) [6], this work concerns the situation where trading partners are operating Program Management within the context of an Extended or Virtual Enterprise using a Virtual Project Office approach. Pallot et al., states that creativity and innovation are soon to become two of the major factors of competition in the global market and hence a system should considerably enhance these areas.

The Mind Mapping technique [7] of which a derivative, concept mapping is incorporated facilitates these prime factors at the core instigation of a project. The EPICE work puts forward the idea of exact common service availability to participating partners, which would be suitable for larger enterprises with budgets capable of ensuring this. However to be able to facilitate gradual or more accurately a flexible entry for SMEs (Small to Medium Enterprises) integrated yet functionally alternative elements must be available to allow those with small budgets to contribute and utilize the information sharing model. Although the expansive information infrastructure would suit the large-scale corporations, the facilitation for small organisations to interact must be taken into consideration, thus enabling smaller suppliers within the supply chain to cooperate in projects with the large-scale organisations.

Kim [8] introduces the term distributed concurrent engineering, which has the meaning of not just a dispersed tiger team but also a variety of computer systems, a reference to the portability issues discussed in [9]. Also within [8] a comprehensible set of five requirements for a distributed Concurrent Engineering implementation are put forward:

- The system should use open standards, to widen the participation of a variety of parties, including suppliers and customers.
- Natural communication between team members should be facilitated within such a system.
- A Database Management System (DBMS), which is capable of managing the storage and retrieval of the large amount of information typical for a reasonable sized company should be used.
- 3-D interactive graphical representations of parts and products should be available for users to manipulate.
- Effective conversions between the data stored in the DBMS, the standards used for data transfer, and the visualisation standards should be designed and developed.

The technological foundations of a distributed system should have a solid basis in portable technologies. This is an essential consideration where the deployment of such a system is not only likely to cross geographic boundaries but also operating environments. No matter how sophisticated a system it is, it will provide little impact in the organisations concerned if it cannot work within the bounds of the lowest common technology denominator between organisations. I.T investment in platforms cannot be disregarded; in fact it should be exploited, [10].

[11] States that Concurrent Engineering depends on the successful transfer of engineering information among the participants in a design project, highlighted is that there is little consideration in most modern research concerning the exchange of the semantics of information.

2 The Approach

The principle object has been to integrate a software based facility for team members to include their early thoughts and constraints at the start of the product development lifecycle when their resolution is at the point of least cost and greatest overall benefit. However this approach needs to provide a logical and coherent flow of data from this stage into the rest of the lifecycle where more concrete designs are produced. An important endeavor is to avoid isolated islands of data without semantics, which is common in many systems today without an assisting integrated framework. The system should provide a means for incremental capture of the data to be shared among the team members, a means to share this and provide extra cognition and semantic assistance concerning this information.

2.1 Situation Analysis

The first point of call within the information sharing system is to provide a conceptual mapping framework and discussion forum for the team members. The conceptual mapping can be likened to Mind Mapping [7] or for the probable majority, brainstorming. However it extends these useful information exchange or structuring mechanisms in various manners. First is the linking system from each node / leaf in the structure, this can be likened to that of hyper-links commonly found on the World Wide Web today, these links can take the follower to other conceptual maps, CAD models, and web pages. Secondly is the format that each node can be represented in, which in turn forms the basis for constraint notification and comparison. Figure 1 shows an example of the types of media and a possible linked configuration.

Constraint communication features in [6] for determining elements such as project resource acquirement, contracts between suppliers, and deliverable milestones. It is also stated that project resources should be expended only on communicating information, which contributes to success or whereby lack of communication can lead to failure. A forum is provided which can assist in both domains, enhancing successful projects with new avenues of exploration and enabling flaws to be alerted at the early stages preventing failure. Information should also be made available to the authorised team members in a timely manner.

The formatting of the concepts can be set in a variety of ways to convey suitably coherent and meaningful data. Formatting can take place in a material selection mode via integration with the Cambridge Material Selector. This in turn allows constraint related concepts to be set concerning the parameters available about the materials, such as tensile strength, and mass.

A highly valid point raised within [3] is one of providing a system, which can deal with synchronous and asynchronous distributed interaction. Extremely important when supporting a globally distributed product development team taking into account the problems of working from different time zones and work

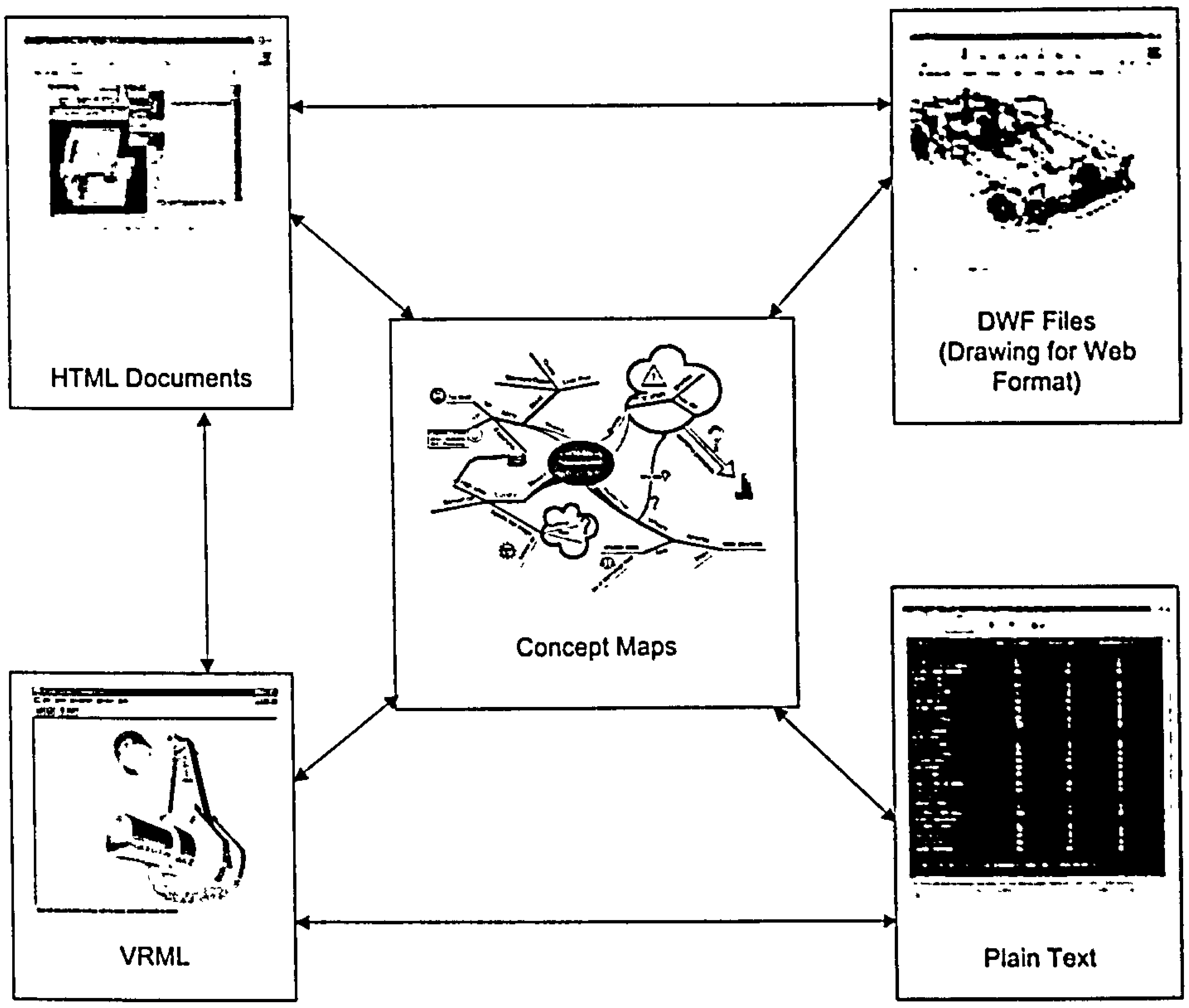


Figure 1 Example linked media types

commitments.

CyberEye [3] provides different mechanisms to enable this support in terms of web-based bulletin boards and chat rooms. However this is rather ineffective in terms of viewing the information at a later point in time, for example after the project is finished, and in terms of organisation the data can only be stored as files of the session at most, which is not conducive to understanding and conception. Mapping the concepts allows both synchronous and asynchronous communication, team members can build the maps together in real time showing visually and textually each train of thought, the maps can also be worked on individually providing both access in terms of fine granularity of information but also a natural concurrent working mechanism which forms a cognitively applicable framework of product development data. Figure 2 shows the iterative phases utilizing the conceptual mapping tool and the corresponding data store and information organisation. This shows how through the design process the end result is not only the successful product but also a coherent natural trail of applicable documentation.

Once all such data has been captured during the product development lifecycle, the system then can be used in alternative contexts. The information although captured by various experts is structured in such a manner as to be traversed by users with a wide range of technical skill levels and knowledge bases. The information when extracted from the product development team is built up from the first stages and hence is of a more comprehendible level, which can be drilled down for those with the expertise into more detailed information, however there is no set path through the information unless one is required for training purposes, the user can chose the contexts of information both relevant and understandable to them.

3 Architecture

The architecture is built from a series of web accessible interface components and object database repositories. The system provides an electronic canvas or sponge for product development data to be placed in without inherent deep restrictions on what that data should look like or how it should be navigated.

The system's conceptual data structures and formats are intended to be openly available as XML Document Type Descriptors (DTD's). This consideration facilitates future integration of application modules such as workflow management tools, which can export to the conceptual maps allowing information to be structured as to the process ordering for example. This mechanism also assists greatly with the lifespan of data. Data stored in the system is therefore ensured that it can always be retrieved with meaning, preventing over time as technologies mature and systems evolve, the loss of legacy data. Through a flexible data interface, legacy data can still be utilized in the modern tools and can be incorporated as a basis in further development work. Users build a series of concept maps, which can combine data from the Cambridge Material Selector (CMS) these can then be checked against a constraint agent detecting inconsistencies in the values selected. These concept maps can be parsed to produce either a natural language report, which to a certain extent produces textual readable content from the maps. This report can be continued into an exportation format for a bill of materials or for example a list of known suppliers, essentially these in effect act as filters for specific subsets of the concept map data.

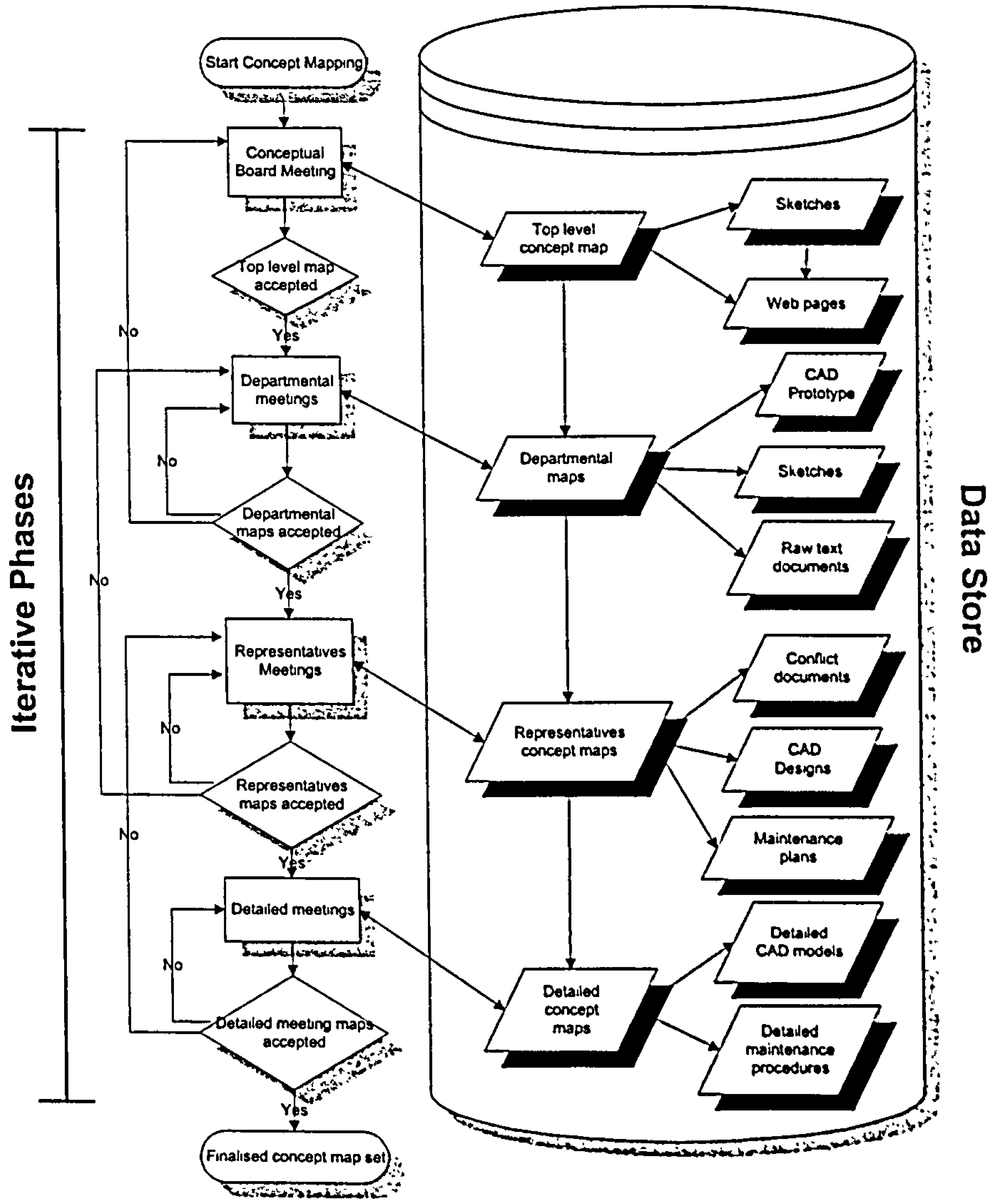


Figure 2 Information iteration flow

Hence tailoring the information for a specific subset of the potential spectrum of employee categories. Alternatively the concept mapping can lead into and produce a table of design variables, which in turn form the basis for building a model. Thus providing a solid base for the prototyping stage, input has been gathered from the relevant parties with currently foreseen conflicts already resolved. The conceptual stage presented in this work can be thought of in much the same manner as prototyping but with a greater capacity to work concurrently both in terms of a team coherency and on multiple facets of the product at a time. Change can be applied more swiftly and increase the turn around time of new ideas and input. From the table of design variables there are a couple of possible routes the design could take, with the design variables mapped into a STEP Application Protocol for example AP 203 / AP 214.

However instead for those without STEP capability there should be a facility to compare a user generated CAD model against a series of design variable tolerances producing a violation report for those values in the model, which exceed the boundaries of those tolerances. To provide a forum for viewing and discussion of such models there is the capability to display in a web enabled format (Drawing Web Format [.dwf])

A STEP Application Protocol route allows either the model to be published in a chosen STEP AP file, which can then be passed to other team members directly, as the file is a platform independent format. Alternatively combined into a hierarchy of STEP AP's for Application Protocols that concentrate on a single part. These can then be stored within the repository, which is a set of linked Object Databases containing other document formats or their references and the XML representations concerning the Concept Maps. The STEP physical files can be extracted from the repository via the SDAI and a web enabled and viewable VRML world built from the geometric data contained.

Figure 3 shows the connections within the data structure in the data model. This diagram illustrates a possible data model that the system could accommodate based upon a sample set of documents concerning a new aircraft concept. It highlights the connectivity and hence the navigation routes the information can be traversed in showing how concept map information can be related to other purposeful documents such as maintenance documents. The data model has been centred on a more traditional lifecycle to show the data, which although in reality would be created in a time line fashion, is a dynamic and connective reiterative approach.

4 Concept Mapping

Apart from being used to store ideas the concept map platform is designed to stimulate the generation of new and innovative ideas. The ideas within cognitive mapping come into play when using the system. A sense of a location within the data can be achieved via this mechanism as opposed to a faceless stance often found on complex web sites and databases where the user loses track of where they came from and where their information goal lies. It is of greater use when trying to trace back through ideas when the chosen one proved to be inadequate for the task or alternatively proof of a correct solution is sought. The ownership of an idea can be attributed as deemed suitable, thus providing a point of contact, should at some point that concept be judged of relevance. However knowing that the ownership of an idea would be recoverable can prevent the placing of ideas out in the open for consideration for fear of criticism. This feature could be altered via a toggle for the team in question to utilise. The concept mapping builds on the idea of information frameworks but also includes extra facilities; allowing the user to drill down into greater detail from a given node from a choice of related information, and the facility for restrictions and constraints to be imposed upon the values of design variables incorporated into the maps effecting subsequently constructed nodes.

4.1 Why is a Conceptual Level of Data Store Required?

The benefits of the insight into a problem before construction, has been shown before, for example on 3D models of pipe arrangements for chemical plants. Maintenance teams can be taken through the expected maintenance routine, in a virtual environment so that their input could be obtained. However 3D modeling although compared to real world construction is extremely rapid, compared to an overview driven conceptual analysis it is not. Obviously deep intricate detail cannot be considered at this level but the higher level thought trains, which previously would not be visible to the rest of the team, could be taken into account in the least amount of time. The system offers various planes of detail for information to be presented; this facilitates new team members with a system, which allows them to gain the essence of a project

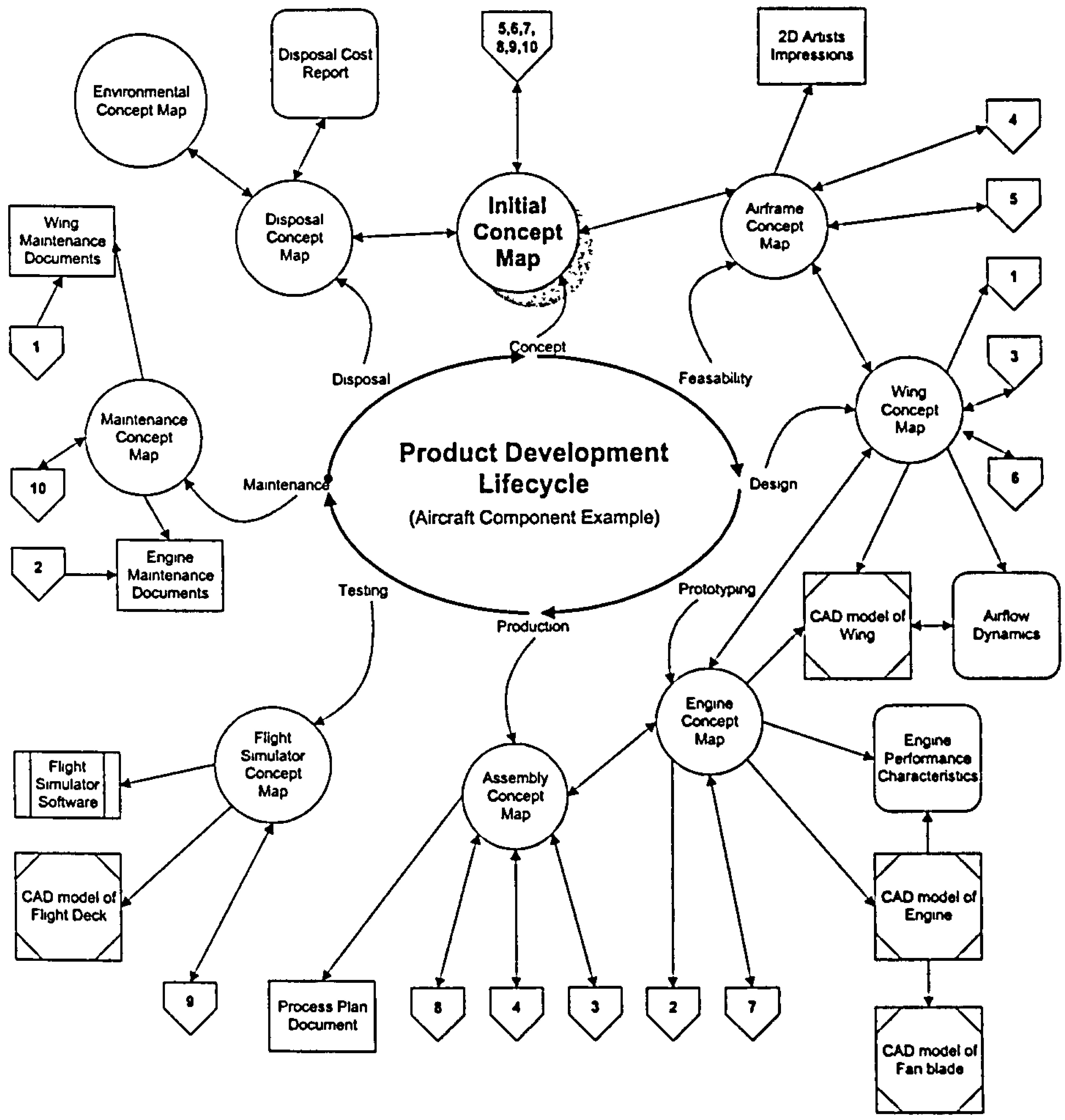


Figure 3 Data structure connectivity within the product development lifecycle

quickly, but also allows them to delve into regions of greater detail if required. A concept map, like comments in software programs can offer an intuitive assistive cognitive medium for the comprehension of the various media types available.

4.2 Concept Map Representation

Due to the flexible nature of concept maps their representation format has to be able to handle a variety of situations. The eXtensible Markup Language (XML) is an ideal choice. XML can be used to model the concepts and their relationships. These relationships can exist between other concepts, other concept maps, VRML worlds, WHIP drawing for web format models, and web pages. The eXtensible Style Language (XSL) provides a mechanism to enable a basic model and view approach to the visual representation of the data. The style information concerning the concept maps can be kept separate from the map structuring data. This allows for greater flexibility within the system when defining a set up for a known product development team.

The concept map tool is written in Java, providing a platform independent creative framework tool. The tool allows the viewing of the XML based concept maps retrieved from the database. However the main strength of this tool is to enable a distributed team to construct concept maps in real time interaction. These maps are viewable by the team as a whole and there is the facility provided to create links from added concepts to other relevant documents such as other concept maps, text, video, CAD models, etc.

An XML file stores the structure of the map and the links embedded within the concepts. The concept map tool along with its corresponding DTD parses this XML file for structural integrity. A standard DTD has been constructed with which XML files can be checked for consistency. This can also act as a data interface for extended application module development. XML files which contain the populated data constructs for the concepts maps are checked against a DTD for validity, the maps would usually be generated by the system, but by having an accessible DTD other systems can interface or represent their information within a concept map which the system can interpret and incorporate into the document structures. The XML files are accessed through Remote Method Invocation (RMI) calls through a Document Object Model, which is turn in constructed by an XML API in this case the more portable SAX (Simple API for XML) implementation, which thus through populating a DOM (Document Object Model) makes the XML data available to the concept map tool for display and manipulation. Figure 4 shows three team members using an integrated chat forum along side a concept map. The interface is based on the Java Swing technology allowing the toolbar docked on the right hand side to be removed and placed for example at the top or outside the browser. If the focus of the group is on the concept map the panels drag bars, which can be seen, can allow extra space or as is appropriate. The Java applet is controlled by the use of JavaScript to extend to fill the available browser window space, hence incorporating as much flexibility as possible to take into account the various limitations of the users desktops.

Figure 5 shows the relationships between the open DTD, which allows correctly formatted and well formed XML data to be checked and validated against

5 STEP

Life times of the application software, system software and the hardware, which are all used to create the data, will have much shorter lifecycles than the data itself [12]. Application software has a typical lifecycle of 3 to 5 years if that, systems software, 5 to 10 years and the hardware even less with cycles of less than 3 years. Thus the integration of such a standard for storage and interpretation within an information sharing system can be a vital mechanism in overcoming the migration problems [13] increasingly a problem of today. There is a strong drive behind the philosophy to relate data to the products and processes that the data describes, as opposed to the computer systems, which create the data. There are three main approaches to the implementation of STEP [13, 14].

- Entire model, off-line batch transfer - File Upload / Download SDAI Binding.
- Entire model, on-line batch transfer - Cached SDAI Binding.
- Individual values, on-line transfer - Direct SDAI Binding.

Essentially a file upload, will use a SDAI (Standard Data Access Interface) to extract a STEP model from a database and write to a file. A Cached implementation works in a similar fashion but extracts to memory for a lower access latency period. It is the third implementation, which draws the most attention, a direct binding, where the SDAI can manipulate individual values at a time, a vast improvement in the access granularity of the data. The potential for concurrent working is greatly increased with corresponding increases in the granularity of the data.

A Java based effort exists [15] where they confirm that "The International Standard Organization has published part 27 of STEP as a Technical Specification: Implementation methods: Java programming language binding to the standard data access interface with Internet/Intranet extensions". An SDAI such as this should be integrated into a portable system as opposed to many of the systems, which do not consider the leveragability of the Internet.

6 Object Database

Implementations such as the recent [3] are still basing solutions upon relational databases, which have the benefits of providing sophisticated tools to aid their creation, however there is strong evidence to suggest that an implementation of an SDAI should in fact be centred on an OODBMS as in [16]. Work concentrated around the development of the SDAI is often based on an OODBMS, however work intended to integrate other applications and functions into the information-sharing domain tends to avoid them. It is time that these technologies and approaches were integrated. An object database deals with highly complex data very competently [17]. Complex data can be modeled in a relational database however this is difficult to do. Data often needs to be deconstructed upon entry into the database and reassembled on retrieval.

When there are lots of relationships between the data, the multiple joins required, reconstructing it become severely

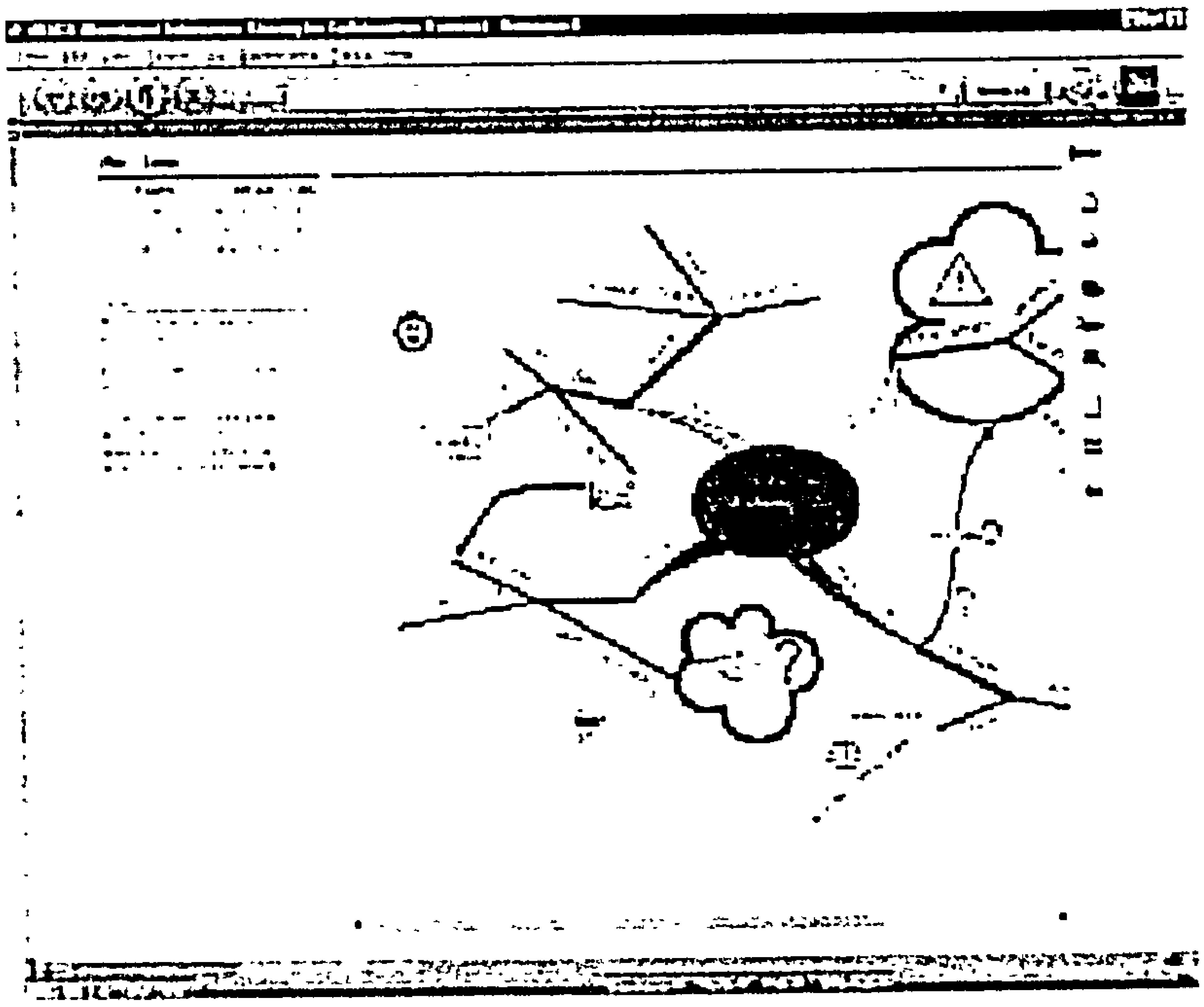


Figure 4 Web Based Concept Mapping Interface

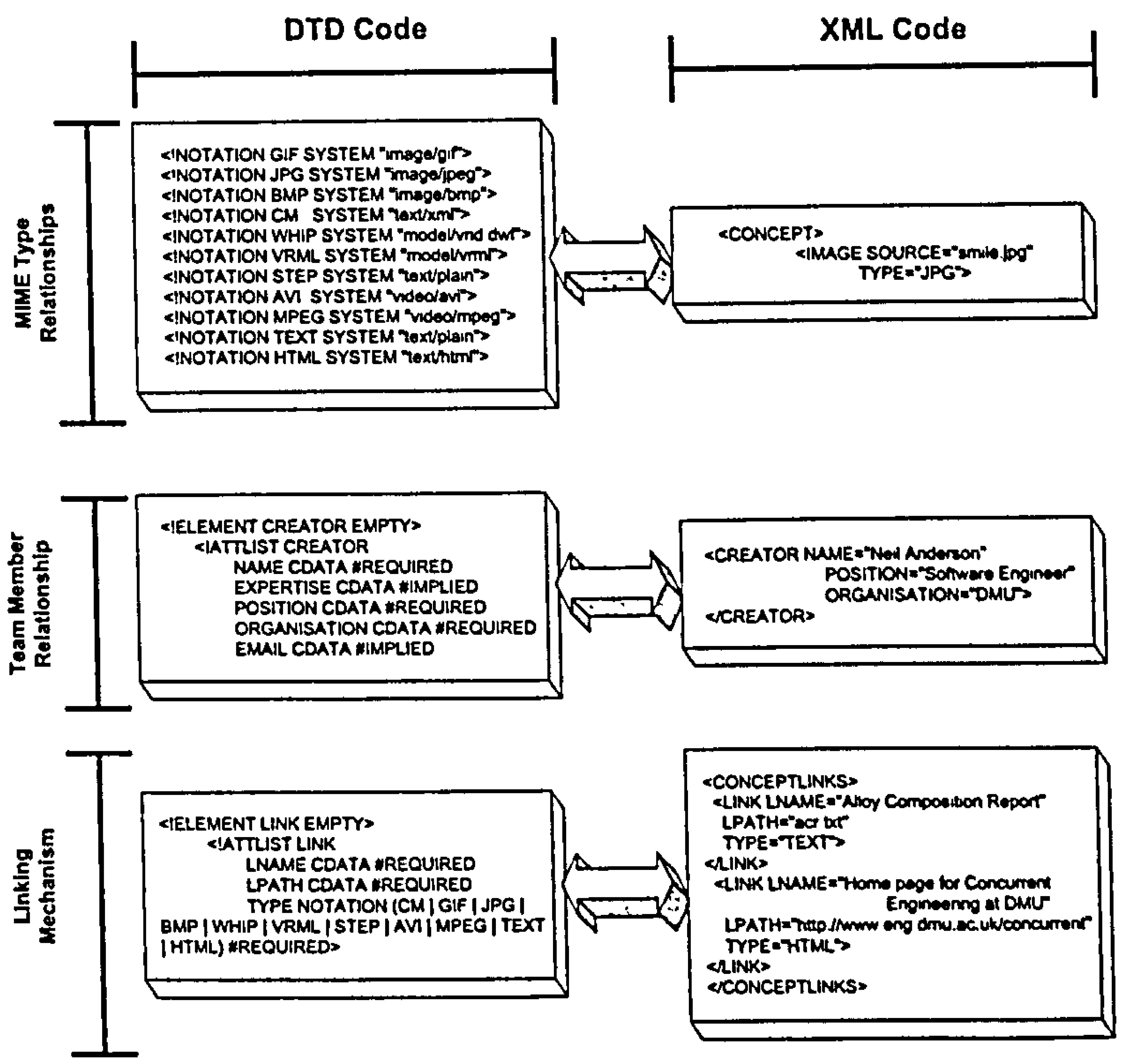


Figure 5 DTD and XML relationships

degrading to the systems performance. An object database perfectly suits both complex data and the XML of the concept maps.

ObjectStore has become the database of choice as it supports; applications that interface with databases and servers on local or remote machines, multiple concurrent users, collections with indexed look-ups and queries, databases consisting of multiple segments, which are variable-sized regions of disk space that ObjectStore uses to cluster objects stored in a database, operating on multiple databases in a transaction and cross-database and cross-segment references with online backup, fail over, archive logging.

7 Conclusions

An ambitious solution has been presented requiring consideration and knowledge drawn from a variety of domains, the problem is much larger than previous efforts consider, benefits can be gained through the narrow considerations put forward in the past however real gains can only be achieved through an expansive and fully integrated system.

Real asynchronous and synchronous communication is the key, supported by a framework, which is closely tailored to our natural neurological infrastructure. Such a framework then allows the systems function to be expanded beyond sharing information during development. The nature of the concept / constraint based framework essentially functions properly as an information gathering, collating, checking, formatting, structuring, disseminating, and storing, tool.

In conclusion the system offers the facilities to create information repositories that have real world value, throughout the entire product development lifecycle. These repositories through accurate enhanced information sharing offer the chance to really drive out problems and inconsistencies during the product development lifecycle, hence lower costs, decreased turn around times, increased team building opportunities, reduced time to market and an increased product applicability and quality.

Applicability spans also to other domains just as patient data in health care, and film / television production. The framework is an ideal way to collate work on an upcoming operation for busy surgeons incorporating 3D models of MRI scans, or a distributed film crew with relationships to casting data, clips and computer rendered special effects models.

References

[1] B. Prasad, Concurrent Engineering Fundamentals, Vol. 1: Integrated Product and Process Organisation, Prentice Hall, 1996.

[2] Institute for Defence Report R-338 (1988).

[3] Zhuang, Y. et al. "CyberEye:An Internet-Enabled Environment to Support Collaborative Design", *CE Concurrent Engineering: Research and Applications*, vol. 8, no 3, September 2000.

[4] Chang, H., et al. "WWW-Based Collaborative System for Integrated Design and Manufacturing", *CE Concurrent Engineering: Research and Applications*, vol. 7, no 4, December 1999.

[5] Abdalla, H., S., "Concurrent Engineering for Global Manufacturing", *International Journal of Production Economics*, Elsevier Science Ltd. Nos. 60 – 61, pp 251 – 260, 1999.

[6] Pallot, M. et al. "EPICE: Realising the Virtual Project Office", *6th International Conference on Concurrent Engineering*, pp57 – 70, June 2000.

[7] Buzan, T., and Buzan, B. The Mind Map Book, Radiant Thinking, The Major Evolution in Human Thought. BBC Books, 1995.

[8] Kim, C-Y., et al. *"Distributed Concurrent Engineering: Internet-Based Interactive 3-D Dynamic Browsing and Markup of STEP Data", CE Concurrent Engineering: Research and Applications*, vol. 6, no 1, March 1998.

[9] Jagannathan, V., et al. *"Collaborative Infrastructures Using the WWW and CORBA-Based Environments", CERC Technical Report Series Technical Memoranda*, July 1996.

[10] Anderson, N., and Abdalla, H., S., *"Distributed Information Sharing for Collaborative Systems (DISCS)", Information Visualisation '99*, IEEE Computer Society, pp 476 - 481, 1999.

[11] Vergeest, J., S., M., and Horvàth. *"GEOS-Based Analysis to Determine the Feasibility of Engineering Data Sharing", Subfaculty of Industrial Design Engineering, Delft University of Technology, Delft, The Netherlands.* 1998.

[12] Fowler, J., STEP, For Data Management Exchange and Sharing, Technology Appraisals Ltd., 1995.

[13] Hares, J., S., Object Orientation: technology, techniques, management and migration. Wiley, 1994.

[14] Loffredo, D.T. 1998. *"Efficient Database Implementation of EXPRESS information Models"*, Ph.D. Thesis, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA.

[15] LKSOFT, *SDAI Overview*, [WWW] Available from: http://www.lksoft.com/sdai_overview.html [Accessed 10th January 2001], 2001.

[16] Liu, T.H., et al. *"Development of SDAI-Based Common Access Interface for Object-Oriented DBMS", CE Concurrent Engineering: Research and Applications*, vol. 8, no 3, September 2000.

[17] Barry, D., K. The Object Database Handbook, How to Select, Implement, and Use Object-Oriented Databases. Wiley & Sons Inc., 1996.

INFORMATION VISUALIZATION IV'99

Edited by
E. Banissi, F. Khosrowshahi, M. Sarfraz, E. Tatham, and A. Ursyn

Convened by
GraphicsLink

Supported by
Visualisation & Graphics Research Unit (VGRU), South Bank University, UK
ICAD - Intelligent CAD Laboratory, Pontificia Universidade Catolica do Rio De Janeiro, Brazil
Computer Graphics & Modelling Group - De Montfort University, UK
Information and Computer Science Department, KFUPM, SA
The Augmented Reality Research Group, The Open University, UK
School of Construction, South Bank University, UK
Faculty of Architecture and Town Planning, Technion, Israel
Department of Visual Art, University of Northern Colorado, USA

Proceedings

1999 IEEE International Conference on
Information Visualization

An International Conference on Computer Visualization & Graphics

14-16 July 1999
London, England

Edited by
E. Banissi, F. Khosrowshahi, M. Sarfraz, E. Tatham, and A. Ursyn

Convened by
GraphicsLink

Supported by
Visualisation & Graphics Research Unit (VGRU), South Bank University, UK
Intelligent CAD Laboratory, Pontificia Universidade Catolica do Rio De Janeiro, Brazil
Computer Graphics & Modelling Group, DMU Milton Keynes, UK
Information and Computer Science Department, KFUPM, SA
The Augmented Reality Research Group, The Open University, UK
School of Construction, South Bank University, UK
Faculty of Architecture and Town Planning, Technion, Israel
Department of Visual Art, University of Northern Colorado, USA

IEEE
COMPUTER
SOCIETY

Los Alamitos, California
Washington • Brussels • Tokyo

Proceedings

1999 IEEE International Conference on

Information Visualization

An International Conference on Computer Visualization & Graphics

Distributed Information Sharing for Collaborative Systems (DISCS)

Anderson N, Abdalla H S

Department of Mechanical and Manufacturing Engineering
Faculty of Computer Sciences and Engineering
De Montfort University, Leicester, UK.

Email: nell@dmu.ac.uk / ha@dmu.ac.uk
http://www.eng.dmu.ac.uk/concurrent/

Abstract

With the advent of global competition, greater numbers of organisations require collaborations with others. Concurrent engineering can provide a highly effective framework for this collaboration. A Distributed Object Oriented Concurrent Engineering System (DISCS) focuses on the communication of data between the members of a distributed project team. Conceptually the communication is the enabling factor; technologically the use of the WWW and object oriented technologies. A portfolio of collaboration support tools facilitates practical mechanisms for information sharing within the distributed enterprise, communication / conferencing for project teams, configuration / version control and workflow structures. DISCS effectively facilitates the virtual team, currently within an engineering context.

Keywords

Concurrent Engineering, Object-Oriented Programming, Java, XML, STEP, Object-Oriented Database, Concept Mapping, VRML, Internet, WWW, Virtual Team.

Introduction

The purpose of DISCS is to enable distributed project teams who for which co-location is either very expensive or impossible to share the benefits experienced by co-located teams. The increase in collaborating organisations sharing expertise to stay competitive leaves a void where previous facilitating tools are minimal in function. Currently each tool only provides part of the solution; they lack the integrated completeness, which is a prime enabling factor. Projects such as MONET [1] and Cyberview [2], are where some of the founding concepts for DISCS are drawn. These only concentrate on the sharing of data such as CAD models. Assuming the sharing to be the enabling factor for the communication of a distributed project team. DISCS expands this to place retrieved data within a concept framework [3, pp. 132-138], hence giving each piece of data a context and increasing the coherence of the information. Thus improving the communication between members of the project team. This framework also assists the generation of new concepts not just their retrieval, tackling a conclusion of [4, p 11]; "Teams should encourage the sharing of new ideas, brainstorming to solve a problem."

The benefits which arrive from an improvement in the communication of a distributed concurrent engineering team are a reduction in costs, a reduction in time to market, an improvement in business processes throughout the product lifecycle, and an improvement in the final applicability to market for the product. The entire lifecycle is considered leading to these improvements.

The technologies chosen for the development of DISCS are VRML [5], STEP [6], Java [7], XML [8, 9], and an object oriented database [10]. These offer the potential for a highly portable system. Distributed access introduces a high likelihood of different platform usage throughout the

0-7695-0210-5/99 $10.00 © 1999 IEEE

distribution. IT investment in platforms cannot be disregarded, in fact it should be exploited. An enabling tool must bridge this gap to be of any real world use. Another valid reason for their selection is the industries activity and support for them, they are each the current technology in their domain. Hence the functionality is still increasing allowing a closer integration of the components.

Previous Background Work

As previously mentioned DISCS overcomes what is considered the shortcomings of previous systems within this problem domain. It is the integration and portability of the sub system components, which increases the utility of such a system. Previous systems [11, 12, 2, 4, 1, 13] have a reduced integration of their component parts and a reduced portability. This results in a lack of a context for the information retrieved.

DISCS has relationships to projects, which in the past have provided only part of solution to the communication problem of the virtual team. However it also provides benefits identified as desirable in recent / currently undergoing projects such as CENTEX (Concurrent Engineering Needs & Technologies Experimentation).

MONET

Facilitates the creation and operation of the virtual team, via an application sharing mechanism. Technologically MONET [1] uses UNIX-BSD sockets and TCP/IP protocols to provide the communications infrastructure. Thus allowing X-Windows applications to be shared by the members of a virtual team such as CAD, art packages, and text windows. The information exchange possible in this manner is relatively limited. The scope of the system is restricted to a UNIX network; the portability is hence limited to within this operating domain. Control is achieved through floor control or chalk passing mechanisms.

CENTEX

CENTEX [13] is the evaluation of the impacts of the concurrent design process on product quality, costs and time to market. A new generation radio system is the target project to be evaluated. A combination of virtual co-simulation and workflow / process flow technologies are used to create an Integrated Collaborative Working Environment.

CYBERVIEW

Cyberview [2] was undertaken by the Department of Industrial Engineering at Seoul National University and the Department of Industrial Engineering at the University of Iowa. It is an enabling system for distributed concurrent engineering. It allows the distributed viewing of CAD models with some simple textual mark-up information. It is in essence a database of CAD models with the facility to view them over the Internet.

The proposed DISCS will provide a far greater access scope, instead of being limited to the UNIX network as in MONET, the access is provided via a connection to the Internet. This scope requires the platform portability of DISCS, which is not present within MONET. In previous work [11, 12, 2, 4, 1, 13] the connectivity between data created in the various applications has little semantic linking between them. DISCS allows a wide variety of connection possibilities directly within the information, at the strongest points of cognitive mapping.

CENTEX used the Mentor Graphics WorkXpert tool. After user feedback, WorkXpert was implemented with the introduction of a new graphical design process view based on a hierarchical representation. DISCS emulates both this structure and expands upon it to permit any linked structure of the data. Conceptual ideas can be mapped / linked into a process flow management display within DISCS, also linkable, hence increasing the entry points into the system. CENTEX identified a problem with the motivation of the team to adopt the new methods. DISCS tackles this in terms of simplicity of use, and cognitive understanding of the retrieved information.

Cyberview's mark-up information is of a limited scope, a short textual annotation is all that is permitted. There is no support for the initial phases of a concurrent engineering lifecycle. DISCS takes this much further with an assisting concept generation and viewing mechanism which can be linked into any part of the CAD model. This enables the initial phases of a concurrent engineering lifecycle. The design is then linked into the initial ideas, creating a cognitive map of the flow of concepts and designs within the minds of the team members, greatly increasing the understanding of the project among the team.

The Proposed System (DISCS)

Conceptual Overview

Through the linking of conceptual trains of thought to a variety of information mediums ranging from sound, video, images through to CAD data, the understanding of designs can be enhanced. Virtually every component in the system can be linked to another, hence providing a context with which the retrieved information can be interpreted. Another important aspect is the expansion of potential users. Currently a sound knowledge in computer systems is required to operate most facilitating tools. However with DISCS a cognitively understandable concept mapping system is used, derived from the Mind Mapping Technique [3]. This makes the understanding of the data second nature, the information and its relationships can be very easily visualised.

If the system is easier to use for the target user audience, it then allows a much greater flow of ideas into the system. The variety of team members can be expanded from the highly technically skilled group to form an IPD (Integrated Product Development) Team [11, p102]. This includes end product users, assembly teams, sales, marketing, i.e. categories of people whose input makes a big difference if captured early on in the product lifecycle. With the more usual "over the wall" [11, p102] approach this input would arrive at a much later stage when redesign is costly and thus possibly omitted, resulting in a substandard product. DISCS facilitates this early retrieval of data with a tool, which provides a cognitively applicable interface allowing the expansion of the possible user base.

The anonymity of many workers and a sense of lack of importance in larger corporations are often due to their lack of opportunities to contribute. This would also be assisted as a side effect of involving more people early on in the design stage of a project. The sense of ownership of the project would be higher increasing motivation aspects. The insight from views other than high level management can easily be obtained; the value of this contribution should not be under estimated. The daily issues, which lack consideration at the initial stages then, represent barriers and problems later on in the lifecycle. These are now identified at a stage where their rectification is of a lower cost.

The benefits of the insight into a problem before construction, has been shown before for example on 3D models of pipe arrangements for chemical plants. Maintenance teams can be taken through the expected maintenance routine, virtually so that their input could be obtained. However 3D modelling although compared to a real world construction is extremely rapid, compared to an overview driven conceptual analysis it is not. Obviously detail cannot be considered at this level but the higher level thought which may not usually be visible to the rest of the team can be taken into account in the least amount of time. More detailed viewpoints such as the 3D models can then be linked into the system for review purposes. DISCS offer various planes of detail for information to be presented; this facilitates new team members with a system, which allows them to gain the essence of project quickly, but also allows them to delve into regions of greater detail if required.

Architectural Overview

The system is centred on the Internet protocol. This is the primary link between components. This protocol of communication was chosen for the target audience, which ranges from the global distributed collaboration of organisations, to the internal Internet solution within a multi-site company.

The system has five main interconnected sub-systems. At the heart of the system is an object oriented database, a concept creation / viewing tool, CAD model creation, VRML viewing of the CAD data, and a suite of supplementary communication tools. Figure 1. below shows the relationships between these components.

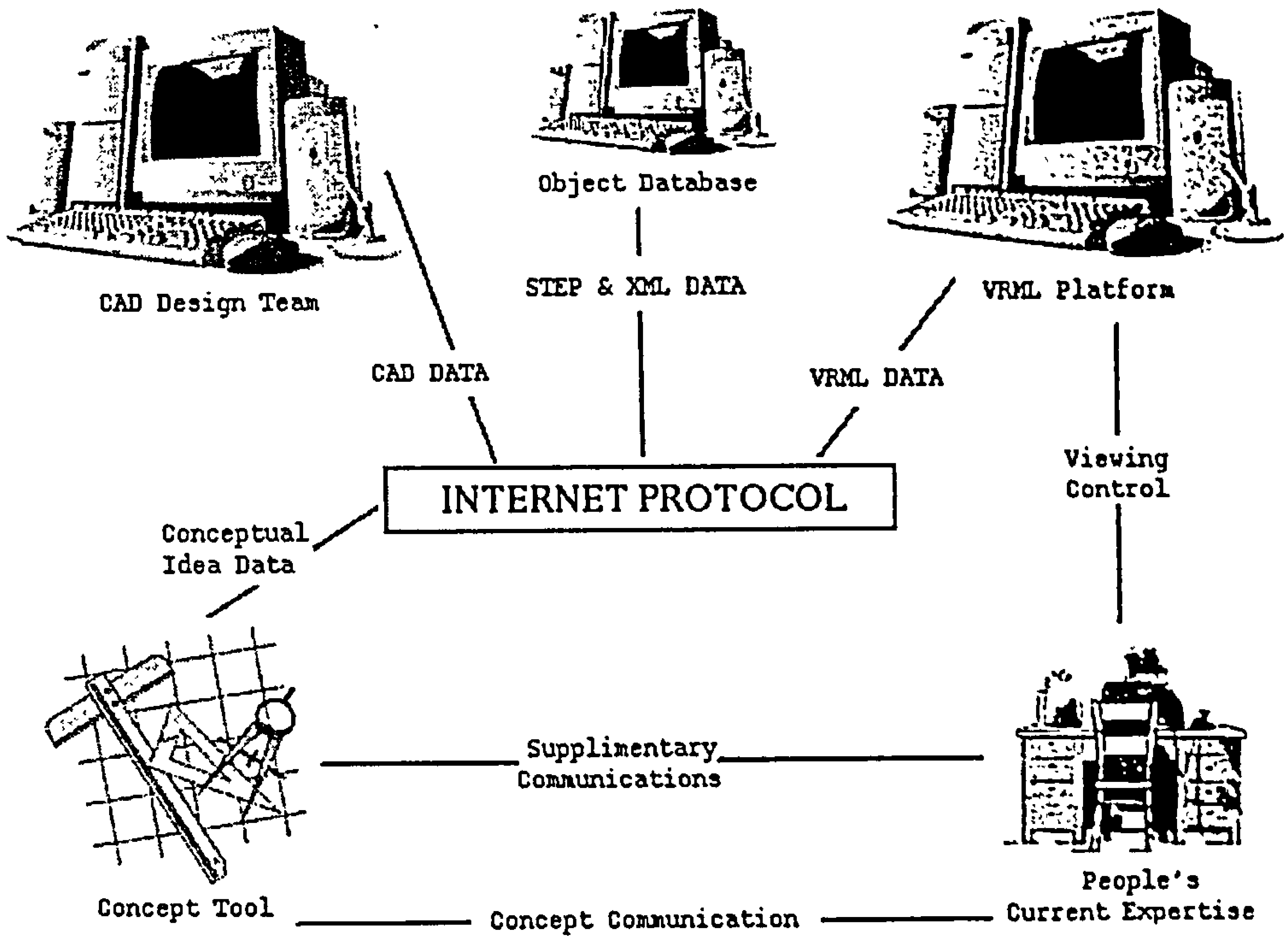


FIGURE 1: Architecture of the DISCS System

Database

The obvious database choice is between relational and object oriented, due to the complexity of CAD data plus the additional linking framework the choice was made to use an object oriented database. The security aspects can also be implemented in a more logical fashion with various levels of locking. This flexibility is well suited to the multi-user format the system will support.

The database essentially stores the CAD data to be shared with a host of linked in supplementary information in a wide variety of formats, from the concept maps produced within the concept map tool, through to images, video, audio, and even HTML documents. The possible information experience is limited only by the imagination of the creator.

Sharing of CAD models - VRML

Information at the conceptual level is essential for guiding a project in the correct direction during the early phases and providing an overview of the work required. This conceptual level can be used to browse through the model, an essential element for an information server [12]. However this is lacking the technical detail and hence communication of detail needed. The sharing of CAD models poses a problem for viewing. The translators which are required to translate between CAD formats increase in number exponentially. A model needs to be used as a medium; chosen for this task is STEP. CAD models are translated into STEP then into VRML. As the support for the STEP standard increases this will increase the future scope and possibilities for the system. STEP is a platform independent format aiding the portability aspect of the system.

The intended VR mechanism is through VRML. The source for the VRML worlds will be translated CAD models; VRML is the chosen technology for a variety of enabling factors [2]. The components are primarily being based on their adaptation to the distributed aspects of the system. I.e. for software the portability factors rank very highly. VRML through common web-browser support is highly portable. Various plug-ins can be downloaded for most browsers on most platforms. This support allows the CAD models to be viewed anywhere in the world where there is access to the Internet. VRML also allows the linking of nodes in a model to other URLs. These URLs can be other VRML models allowing the nesting of models. Concept maps can also be linked to, along with any other commonly found web documents, text, graphics, sound etc. Existing databases could be connected via an ODBC (Open Database Connectivity) connection / bridge. Allowing a change over from a legacy of stored models / file of a project.

VRML is also flexible enough to accommodate the various CAD formats once translated. Currently perceived as a purely desktop based viewer. There is no reason why specialist nodes / sites in the communicating team couldn't be set up to allow a more realistic and immersive experience in terms of virtual reality coves / caves. These allow the system to be used for after design applications. Instead of storing a lot of technical inaccessible data, the system is encouraging previously input data to be reused. The reality level of VRML is sufficient for the system to be used to provide a walk through of designs either for iterative modification and review or even to provide the first steps of training for future users of the product. This would be cheaper than building real world models / simulators, also somewhat more portable and modifiable.

VRML also allows the multi-participants existence in a single world. Again increasing the concurrency and the communication. If you want to show a person an aspect of the world for discussion the ability to follow someone around the world is invaluable. The use of more standard natural communication is suggested to accompany this sort of procedure. This approach can be termed the tour guide and can be used to help new team members become accustomed to the viewed product noting important features. A basic mannequin represents each user.

Other Components Overview

Translators

The system requires two points of translation from CAD data to STEP data, which can then be stored. Then for the viewing i.e. VRML this STEP data must be translated into VRML worlds. This is opposed to the more direct CAD format to format translation, which increases exponentially with each new configuration. For each CAD application a single translator to the STEP model is all that is required as the basis for conversion to VRML. Hence increasing the real world applicability and acceptance criteria.

STEP

STEP is an expanding technology, platform independent, with industry backing. Apart from the translation and transport medium within DISCS, the expansion of STEP compliant applications means that data stored in this way will be available to communicate this information during the next generation of systems. Product lifecycles often extend far beyond the operational life of the applications, systems software and hardware. Its important that a storage and translation medium should be able to from the outset bridge these changes when they occur rather than cause disruption upon change.

Conclusion

DISCS offers a wide range of benefits based on vastly improving communication between team members. This improvement in communication is derived through a linked repository of globally accessible information. Related information, concerning conceptual analysis, design, and workflow processes can all be retrieved from each other. This reinforces the understanding of the material retrieved and the hence the communication of ideas between team members. If the understanding of team members can all be increased an extra emphasis on involving a wide range of people at the earliest opportunity allows a variety of common problems to be communicated to the rest of the project team and dealt with at the point of lowest cost. A lifecycle of problems can be identified at the beginning stages, improving the number of change requests, marketing, assembly, testing, usability, through to ecological disposal of the product.

However DISCS offers the facilities to reuse this information, basic training can be performed via the VRML interface, maintenance teams can be given a run through of the expected servicing routine before they have to encounter it for real. Service manuals could be easily linked in to the repository of information used to build the product ensuring the accuracy of the data.

Unexpected changes to the project team later on in the lifecycle can be more suitably dealt with; the concept maps provide a clear overview of any product.

In conclusion DISCS offer the facilities to create information repositories that have real world value, throughout the entire concurrent engineering lifecycle, these information repositories offer benefits in terms of reduced costs, decreased turn around times, increased team-building qualities, a reduced time to market and an increase in applicability and quality. Hence improving both product and business processes.

Future Work

DISCS is an ongoing project and hence has various outlets for the expansion of other considerations. The virtual reality aspects can be taken to a higher level of realism, this would reflect the general improvements which could be made to a virtual world such as sound, nasal and haptic (tactile) feedback.

In terms of other domains, DISCS could replace the CAD model components with a software component viewing system, this could be initially textual expanding to a graphical representation. Hence reusing the linking possibilities within DISCS.

References

[1] SRINISVAS K., et al. *MONET: A multi-media System for Conferencing and Application Sharing in Distributed Systems*, CERC-TR-RN-91-009, CERC Technical Report Series, West Virginia University. Feb. 1992

[2] KIM C-Y., et al. Distributed Concurrent Engineering: Internet-Based Interactive 3-D Dynamic Browsing and Mark-up of STEP Data. *Concurrent Engineering: Research and Applications*, March 1998, vol. 6 (pt. 1). pp. 53-70.

[3] BUZAN, T. *The Mind Map Book*. 2nd ed. London, BBC Books, 1995.

[4] LONDONO F. et al. *Co-ordinating a Virtual Team*, Morgan Town West Virginia, CERC Concurrent Engineering Research Center Drawer 2000, West Virginia University. [No Date].

[5] Web 3D Consortium. *Web 3D Consortium*. [WWW]. Available from: http://www.vrml.org/ [Accessed March 1999].

[6] FOWLER, J. *STEP For Data Management Exchange and Sharing*. Twickenham, Technology Appraisals, 1995. pp. 53-67.

[7] FLANAGAN, D. *Java in a Nutshell*. 2nd ed. Sebastapol, O'Reilly & Associates, Inc., 1997.

[8] CONNOLLY D., et al. *The Evolution of Web Documents: The Ascent of XML*. World Wide Web Special Issue on XML, vol. 2 (pt. 4), Fall 1997. pp 119-128.

[9] REES, L. *A brief Introduction the Mysteries of the Extensible Mark-up Language*. [WWW, Printed October 28th 1998]. pp. 1-8.

[10] BARRY, D. *The Object Database Handbook*. United States of America , Wiley Computer Publishing, 1996. pp. 29-45.

[11] GAIENT A. Agility through Information Sharing: Results Achieved in a Production Setting. *Concurrent Engineering: Research and Applications*, June 1997, vol. 5 (pt. 2). pp. 101-111.

[12] JAGANNATHAN, V., et al. *Model Based Information Access*. In: *Proceedings of the Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise*. CERC-TM-93-007, IEEE Computer Press, April 1993. pp. 198-212.

[13] CENTEX, *Concurrent Engineering Needs & Technologies Experimentation*. [WWW]. Available from: http://esoce.pl.ecp.fr/ce-net/Information/cerelated/centex/Projectoverview.htm [Accessed 16th March 1999].

Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Order Number PR00210
Library of Congress Number 99-61704
ISSN 1093-9547
ISBN 0-7695-0210-5

ISBN 0-7695-0210-5
9 780769 502106 90000

Anderson, N. and Abdalla, H., "A Holistic Integrated Approach to Product Development." *International Journal of Advanced Manufacturing Technology*, (Submitted), 2002

A Holistic Integrated Approach to Product Development

N Anderson BSc., and H Abdalla BSc. MSc. PhD. CEng. FIMechE

Department of Design Management and Communication

De Montfort University, Leicester, UK.

Email: neilanderson@arczip.com / ha@dmu.ac.uk

http://www.eng.dmu.ac.uk/productdesign

Abstract

This paper presents a distributed collaborative approach that facilitates virtual multi-disciplinary collaborative product development activities. Thus enabling organisations to compete within an environment of increasing global competition to deliver the right product at the right time and at the least cost. This necessitates the development and utilisation of effective collaborative computer systems that support the entire product development lifecycle. The problem of distributed collaboration caused by ineffective knowledge sharing leads to the late detection of discrepancies and conflicts in the product development lifecycle when it is costly in terms of time and money to correct. The developed work comprises of an information sharing protocol and a set of modules; concept mapping, data sharing and structuring, data standardisation, object oriented database, conflict resolution, and Hypertext Transfer Protocol (HTTP) accessed document modules.

Key Words: E-Commerce, Collaborative Product Development, Information Sharing, Concurrent Engineering, Virtual Teams.

1. Introduction

Novel product development systems facilitate high quality, flexible, cost effective, timely to market products, these are crucial elements required by an organisation to be competitive in a global market. The domain of distributed product development team communications is the prime sphere requiring improvements through the use of supportive IT solutions. Widely available technologies such as the telephone, email, and more recently the advent of instant messengers provide too generic a solution to facilitate an effective distributed environment for successful information sharing. Successful utilisation of distributed product development teams requires advanced communication tools to support the Concurrent Engineering (CE) techniques.

CE used successfully as an approach to product development facilitates competitive advantage through the integration of lifecycle requirements early on in the design phase such as concept generation resulting in the faster and cheaper manufacture of superior quality products. The key to success is accurate, timely and effective communication from a range of distributed parties within the conceptual / design phase thus reducing potential errors and therefore positively impacting on the reduction of costs.

Design information needs to be available to all concerned parties from the outset. Accurate and timely information is essential for superlative product development. Collation and communication of multifaceted requirements is fundamental to ensure that a proposed design is as accurate and correct as the available information resources allow.

It is vital that the information gathering process be undertaken at the earliest stages in the product development lifecycle to ensure that the impact of accurate design decisions is maximised. Problems discovered early in the design process (particularly during the first 20% of the cycle time) are easier to solve than those discovered later (Prasad (1996)).

Important in accurate distributed communication is the group creation of collective comprehensible models. Conventional systems use complex data collation models that place a restricted domain on potential users of the technologies. The shared model needed to facilitate effective collaborative benefits requires a basis in natural human cognition avoiding unnecessary depiction complexities yet flexible enough to represent complex data (Anderson and Abdalla (2002)).

The increase in collaborating organisations sharing expertise to stay competitive leaves a void where previous facilitating tools are minimal in function. Currently supporting tools only provide part of the solution; they lack the integrated completeness, which is in itself a prime enabling factor (Anderson and Abdalla (1999)).

The need for distributed team formations arises from today's highly competitive global markets. Product development teams often require expertise from around the globe; previously co-location was the only mechanism available to achieve performance from such teams. However co-location can prove very expensive or even impossible. The more skilled a team member is the greater likelihood their skills are in demand; commonly they are working concurrently, on multiple projects at a time (CENTEX (1999)). Hence making co-location impractical.

The actual design costs are only a small proportion of the overall product development cost, however the design phase determines the majority of the overall production costs. The benefits that consequently derive from an improvement in the communication of a distributed concurrent engineering team are a reduction in costs, a reduction in time to market, an improvement in business processes throughout the product lifecycle, and an improvement in the final applicability to market for the product.

Recent research work has emphasised the development of platforms that enhance speed and efficiency. However, most of these systems are lacking the consideration of a conceptual and macro-level framework structure and representation (Anderson and Abdalla (2001)).

This current research work provides a web based supportive IT solution that facilitates collaborative distributed product development teams throughout the entire product development cycle, from concept to manufacturing. The research undertaken is focused on improving communication where common understanding of that diction has been neglected in the tools available today. Much of the current / previous work undertaken has not addressed communication and data sharing of products at conceptual level, which is an essential consideration of this work.

This paper is primarily concerned with the integration of the differing facets of the supply chain within a distributed collaboration scenario. Integration covers a multitude of aspects; the technologies, product development lifecycle phases, and information representation to the people involved. Collaboration requires a common information-sharing denominator, an environment that is consistent and available to all parties involved. This has been developed in this research in the form of concept mapping; it is important for there to be facility for this representation to be parsed and processed into specific forms that each increase the suitability for a domain of the integrated supply chain.

2. The Proposed Collaborative Environment

The proposed collaborative environment comprises of, concept mapping, data management and standardisation, conflict resolution, an object oriented database, and a scalable HTTP interface. This research work focuses on the integration of people, processes and product development phases. Collaborative product development necessitates team input from the earliest possible phases as addressed by the developed concept-mapping environment. It is the parsing of concept maps (Anderson and Abdalla (2002)) into domain specific representations that this paper presents.

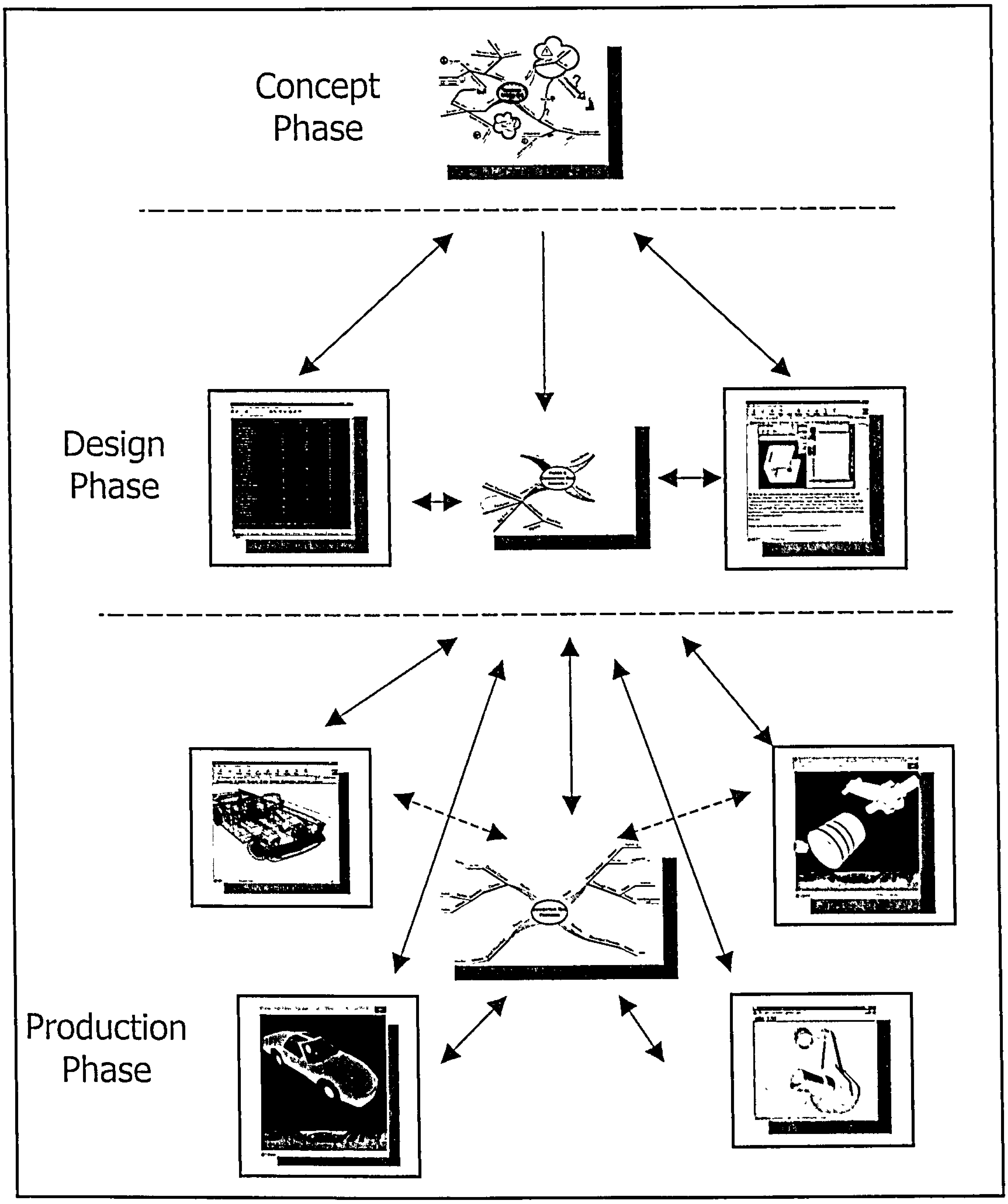
The philosophy of Concurrent Engineering does not dictate a specific methodology or technology but puts forward the perspective that as many potential factors of product development should be available for analysis in the design phase. This has been addressed in part by the concept mapping facility, however to lever extended benefits the collaboration has to extend beyond concepts and basic information sharing to detailed design and management representations.

2.1. Integration

Throughout the product development lifecycle the depth and diversity of the information in a collaborative effort continually expands. It is necessary to propagate the results of design decisions from early design stages effectively throughout the product development lifecycle. There is a necessity for the integration of the data generated by collaboration not only within each phase but also between development phases facilitating their transition. An example model demonstrating this centred on the concept maps is shown in (Figure 1).

A conceptual representation addresses the needs at the beginning of product development however this representation needs to maintain continuity through subsequent development phases. In the product development collaboration the concept maps are the hubs for other information representations. They provide a framework not only for a single development phase but also to facilitate the building of continuity between phases.

It is the continual focus on the integration of the elements of collaboration that facilitates the required holistic product development environment. Unless distributed collaboration is addressed as a whole rather than piecemeal, the omitted facets of integration indirectly affect the potential of the collaboration and introduce new barriers. System portability is an area that is usually neglected in favour of the introduction of more features and easier system development. This restricts potential users to single (often incompatible) operating environments making global collaboration increasingly difficult where previous IT investment in different systems becomes a strong barrier to distributed collaboration.



(Figure 1) Collaboration Relationships

2.2. Concept Mapping

The core purpose for the development of concept mapping is to facilitate the sharing of information to product development teams at an essential early phase in the product development lifecycle. Concept mapping provides a unique medium with which to convey ideas and logical trains of thought. It aims to tackle a range of the problems with information sharing in the product development lifecycle.

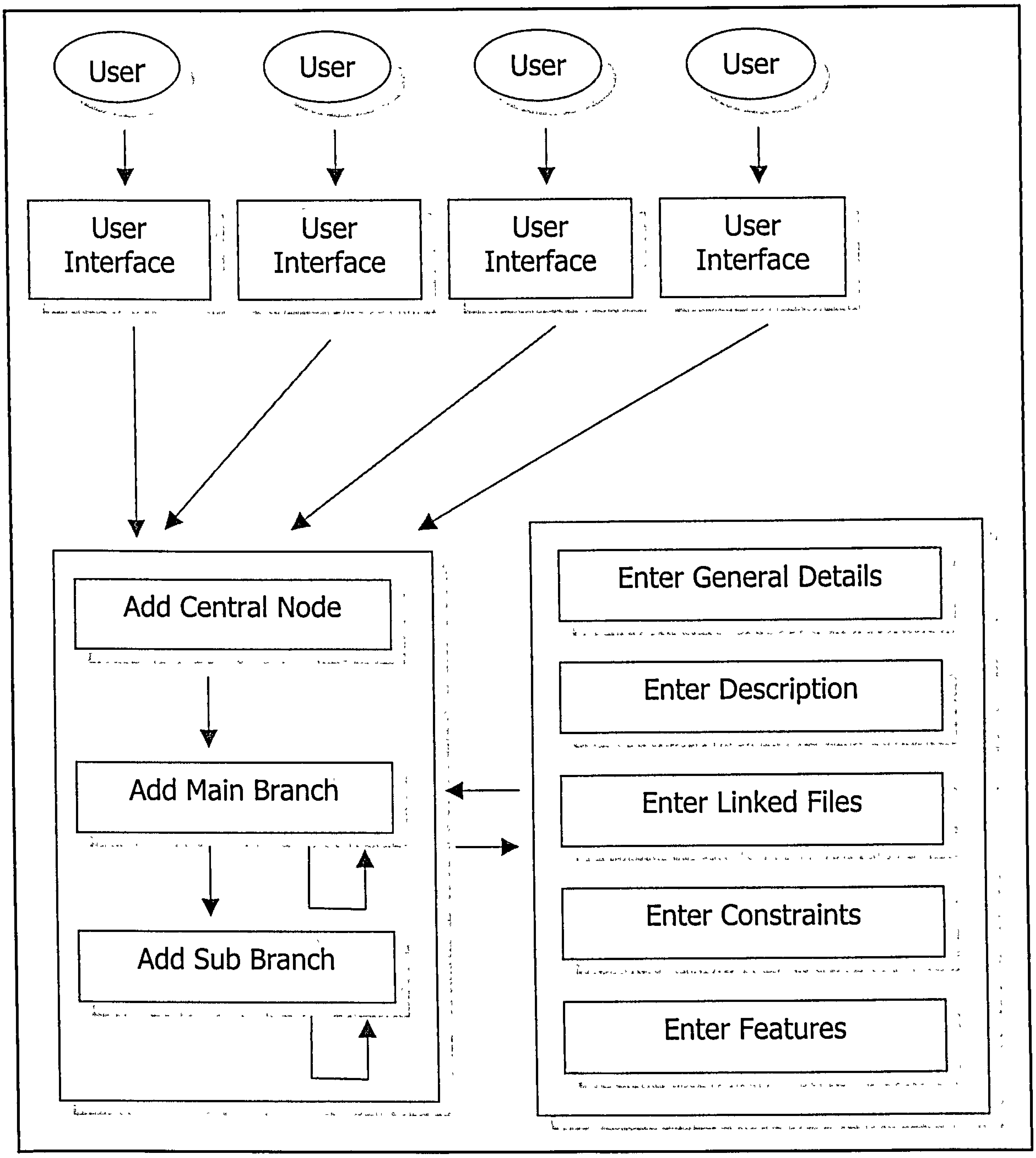
The fundamental theories concept mapping is based upon lay in that of Tony Buzan's successful mind mapping techniques (Buzan and Buzan (1995)). These have been designed to utilise the basic cognitive functions of the brain for thought processing, recall, and structuring.

The procedure for building a concept map is illustrated in (Figure 2). A concept map is built in the following order:

1. Central node (defines the intended scope of map);
2. Main branches (define relevant sub topics);
3. Sub branches (either textual or graphical);
4. Additional concept details.

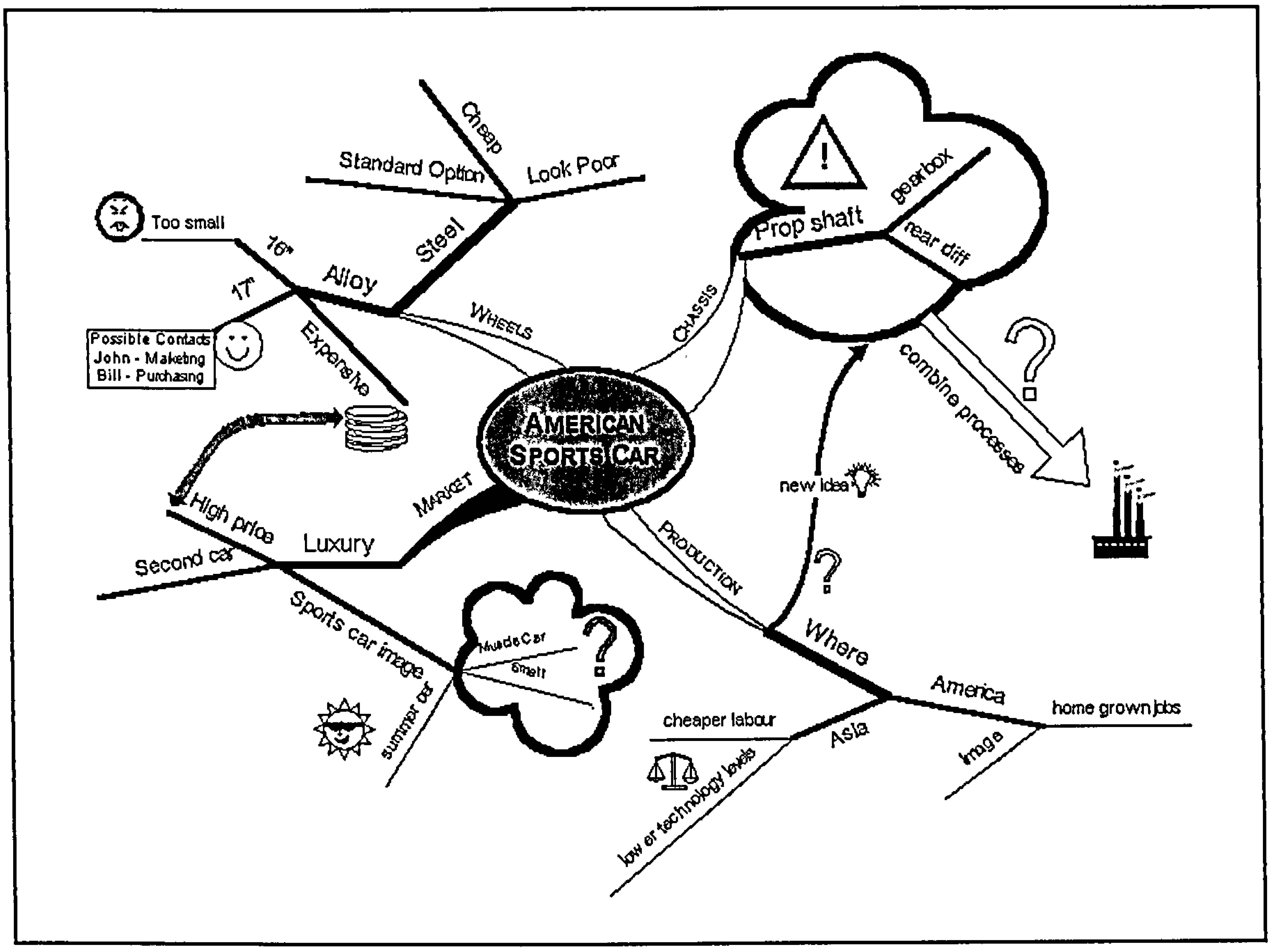
The central node of the map defines the topic and the scope that the concept map should encompass or relate to. A large central point such as this is a constant reminder to those working on the structure what the focus is. Each new area of discussion within this topic is built upon a main branch that constantly brings the users back to the main topic each time one is added rather than the end of the previous topic as in a linear representation. Sub branches are added to convey further details.

Each sub branch is also a placeholder for a variety of further information. A sub branch, via a dialogue provides; general details about the concept, an additional plain text description of its purpose, links to other locations (such as files, web pages), constraints in relation to other concepts, and if the concept is representing the actual features for an object relationships then also links to other features can be added.



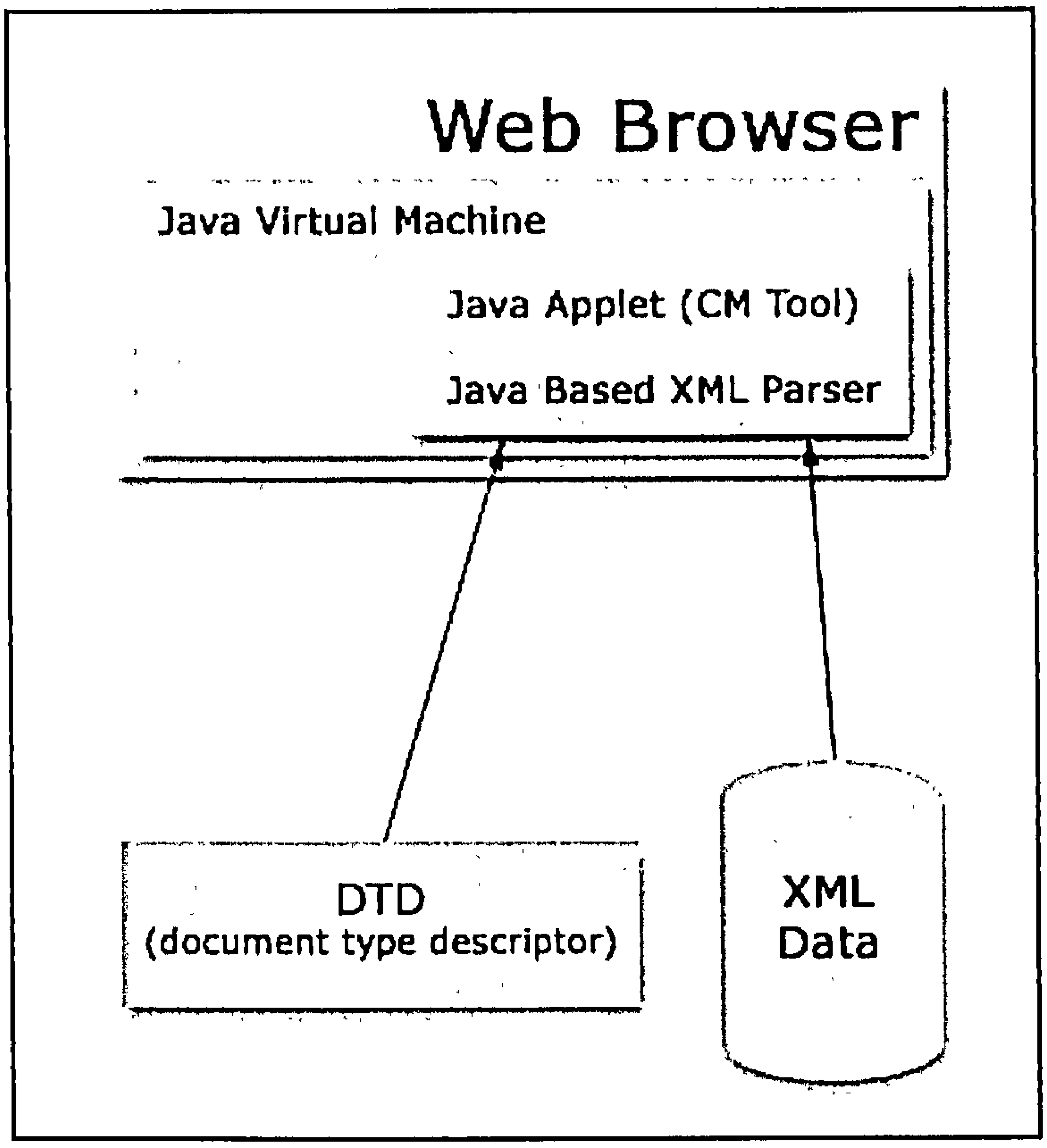
(Figure 2) Concept Mapping Procedure

An example concept map is shown in (Figure 3). The central node, main branches and numerous sub branches can easily be seen.



(Figure 3) Example Concept Map

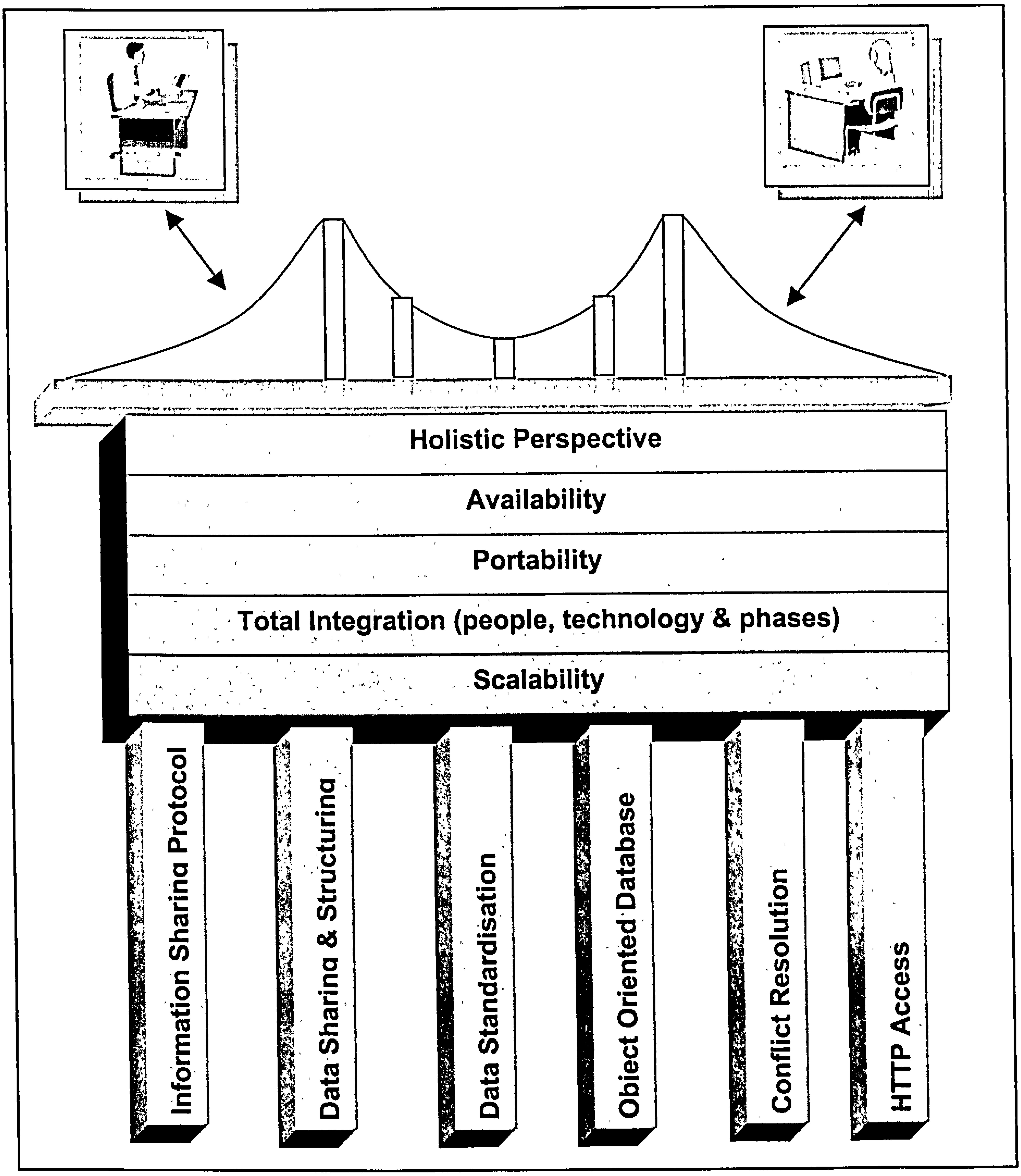
The concept maps form a suitable cognitive framework for the comprehension and sharing of product development information. Concept mapping is underpinned by the use of XML to represent the data structures involved. The Java Virtual machine runs within the web browser running an applet that is the concept-mapping tool, this in turn contains an embedded Java based XML parser (Figure 4).



(Figure 4) XML Parser Integration

2.3. Bridging Collaboration

The collaboration bridge in (Figure 5) represents the horizontal bridging tenets for a system necessary to maximise the collaboration between organisations. The supporting pillars of the bridge are effectively the sub sections of the collaborative system developed to address these issues.



(Figure 5) Collaboration Bridge

2.3.1. Holistic Perspective

A holistic perspective can only be gained through the input of all the parties involved in the collaboration. A distributed collaborative system must encourage and facilitate the contributions from all team members. This information must be presented in such a manner such as to assist in the resolution of conflicts, facilitate creativity through the connection of previously unforeseen relationships, and provide a central resource for all the information products of the collaboration. Unless a holistic approach is undertaken, the result of collaboration produces islands of unrelated data and thus reduced in their potential effectiveness. Data mining and comprehension of information is increased in difficulty without a holistic approach due to the lack of relationships between data entities.

2.3.2. Availability

Full time access to retrieve or contribute to a collaboration database within a globally distributed environment is essential. The global environment necessitates that the collaboration can be contributed to at any hour of the day or night, it should not need to be necessary to arrange for resources to be free to access them. It is important for stored data to be as finely grained as possible. Without relatively fine-grained information, i.e. storing information in large chunks it is difficult to provide a sufficient environment for team members to work concurrently. The result of this is a collaborative environment that loses its concurrent properties. Thus the availability of the system and its data addresses not only geographic boundaries but also the barriers to concurrent engineering.

2.3.3. Portability

Portability in the context of the general computing environment is seen as a beneficial but not essential requirement, however in the domain of collaborative product development, the portability of a supporting system is of a far greater importance. Supply chain integration necessitates the ease of compatibility between computer systems. Increasingly supplier contracts are put out to tender, if the requesting organisation or the supplier necessitates a requirement for a single operating environment to be able to allow the collaboration to continue, this will introduce barriers between potential candidate suppliers.

Portability in the collaboration domain is essential in the modern computing era; there is an ever-increasing diversity among platforms despite efforts for standardisation and consistency. A platform in the context of this research corresponds to the operating system and the web browser of the client. Many systems are developed on a proprietary platform; clients are specially designed to run under Windows for example. This often provides faster and easier software development, greater possible features and reduces the complexity of the system using well-defined core libraries with many years of testing behind them. Single platform systems present new barriers to collaboration; each team member must be supplied with a suitable computing environment to be able to provide their contribution. This is usually a prohibitive task, purchasing and configuring new environments is costly both in time and money.

The end results of such problems are that the contributions from non-technologically capable partners are submitted in other non-integrated manners or worse still are omitted altogether from the collaborative process.

2.3.4. Integration

The core purpose of a collaborative system is to reduce or remove the barriers between the people processes and the technology deployed; effectively the purpose of the integration of these factors is to provide emergent benefits greater than the sum of the parts. Full integration is the key to collaboration efforts for the reduction of costly product development barriers. Strong integration brings with it benefits in terms of relationships between previously disparate data and more powerful data management. However a balance needs to be struck between this support for breaking down the barriers between sets of data and the prevention of the development of a system so tightly integrated that it in turn through lack of scalability then disregards the future accommodation of technological changes.

2.3.5. Scalability

With the constant advance of product development and information technologies a collaborative environment designed and developed for today will provide short lived benefits unless it is based upon scalable technologies that can further the existing integration with as yet unknown data representations. The developed research in this thesis utilises an open system approach whose data format in this case the DTD is not hidden or buried in inaccessible complexity, it is represented in the format of an internationally recognised data standard. The system also facilitates scalability through the provision of direct access to external data representations through utilising HTTP linking. A scalable collaborative environment allows the collaborating organisations the flexibility to produce structures that provide greater relevance to their industrial domain without constricting them unnecessarily. Scalability also involves the created datasets independence from the system that created it. The framework is based on XML that allows the data to be processed independently of the system. It is only through the advent of such information standards like XML that tight system integration alongside data independence is achievable.

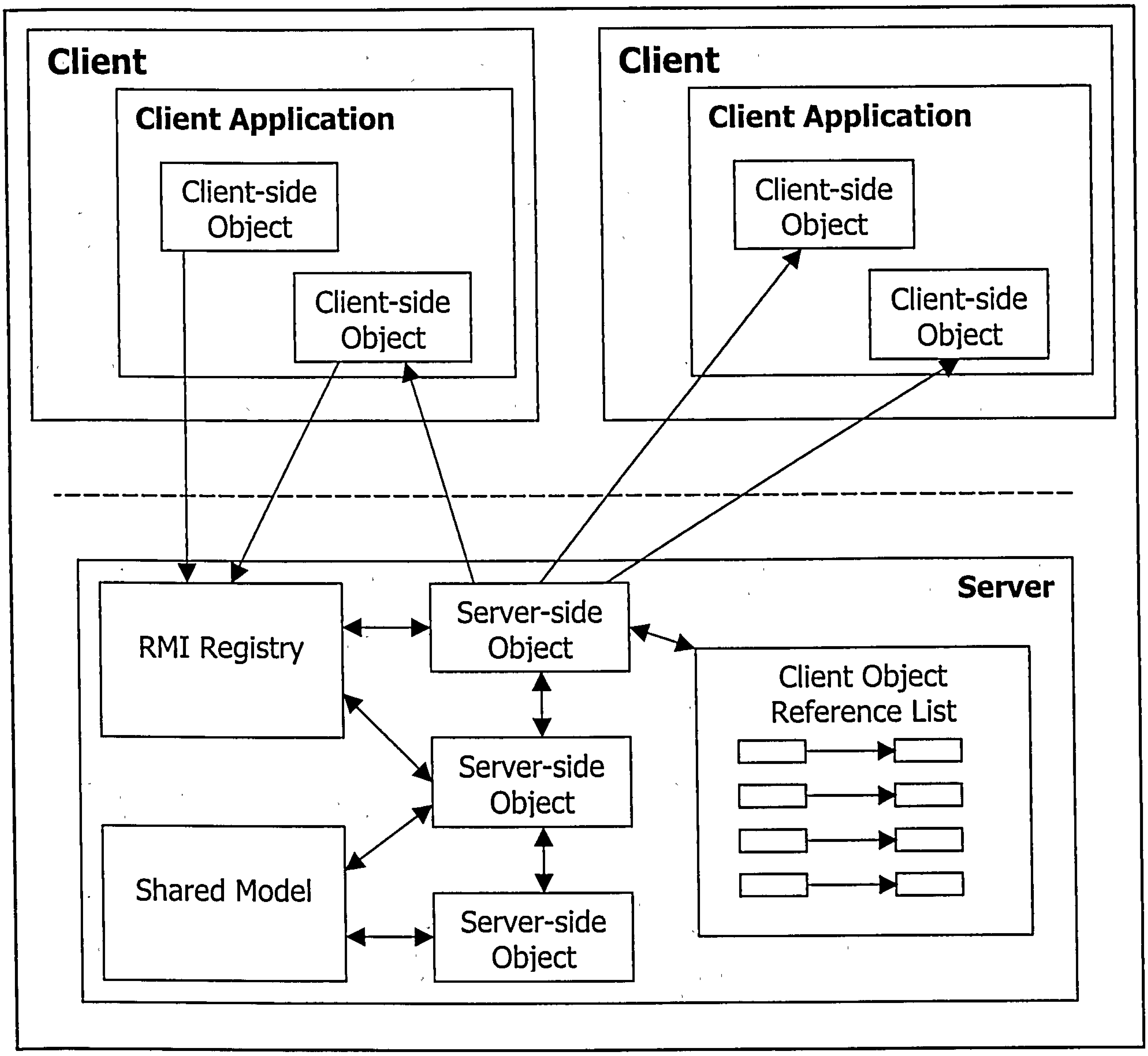
2.4. Information Sharing Protocol

Central to a collaborative system is the underlying information protocol that provides the data transfers and management of the information flow. This usually is a factor that restricts the potential of many of the facets of a collaborative system. Providing a collaborative environment based around the multiple clients and server architecture generally restricts system development to a single language or operating environment. However with the advent of web-based technologies and the increased portability of the clients there is now the potential for only the server side software to be restricted. This restriction can also potentially be reduced through the use of the portable Java run time environment.

2.4.1. RMI (Remote Method Invocation)

An RMI application is usually comprised of at least two programs, a client and a server. A server application creates the remote objects and generates references for them. These references are available from potential clients to locate and invoke methods upon the objects that they denote. Java's RMI provides the mechanism that enables the client and the server to pass data to each other.

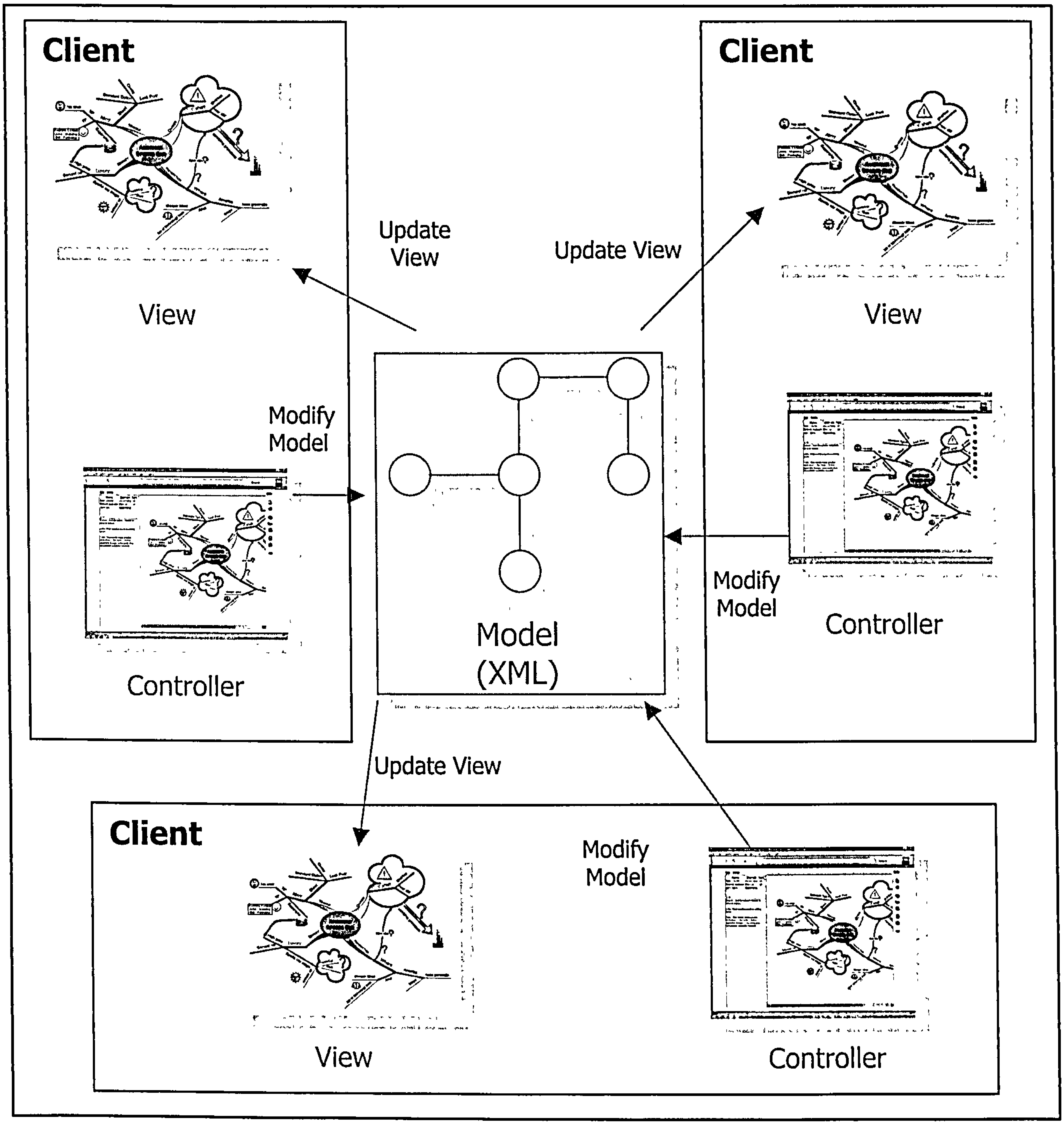
There are two mechanisms available within RMI to obtain references to remote objects. Applications can pass and return remote object references as part of its operation or the RMI registry can be utilised. A combination of methods has been used in this research. The RMI registry provides a centralised location for applications to register remote objects. In this case the server creates objects and registers them with the RMI registry that then binds a name to the object. The client then connects to the server and looks up the name in the server's registry and invokes methods upon it.



(Figure 6) Server Maintained Client Object Reference List and RMI Model

This method for client server communications allows many clients to connect to a single server application. However in this research it is necessary for bi-directional communications to take place. A server also needs to be able to communicate to a client. To accomplish this a reference to the client object is passed when the client invokes a method on the server, this is then stored providing the server with a means to in turn invoke methods on the client.

(Figure 6) combines the two RMI approaches, the client with the facility to access the server objects through the RMI Registry and conversely a server maintained object reference list to the client objects. This configuration allows an effective implementation of the MVC (Model View Controller) design pattern. The server maintains a shared model that is updated by server objects with methods called via RMI from a client. When a change to the model is made the server iterates through its maintained client object reference list and in turn invokes methods on the client to reflect the modifications on the multiple views.



(Figure 7) Model View Controller Model

This bi-directional RMI mechanism effectively implements a broadcast system for the synchronous update of clients. Concerning the event that multiple users are viewing the same concept map (considered as the model in (Figure 7) the server invokes methods on those clients to update the changes to them. Considering the case when a single user is working with a model, an individual client populates the client object reference list on its own and thus both synchronous and asynchronous collaborations are encompassed in the one mechanism.

2.5. Data Sharing & Structuring

Supply chain integration requires the effective structuring and sharing of product development data; this is essential to the facilitation of successful collaborations. The process of concept mapping requires suitable structuring of the data in order to facilitate the manipulation and granularity of access required. The concept maps themselves are represented in XML with their corresponding DTD. The concept maps themselves can be used, for alternative uses such as presentations however the underlying format can extend much further than that.

The integration of a supply chain necessitates the dynamic representation of information in a variety of formats and filtrations. It is important for the information stored to be both human and computer readable so as to enable automated processing to output custom displays for the diverse members of the supply chain.

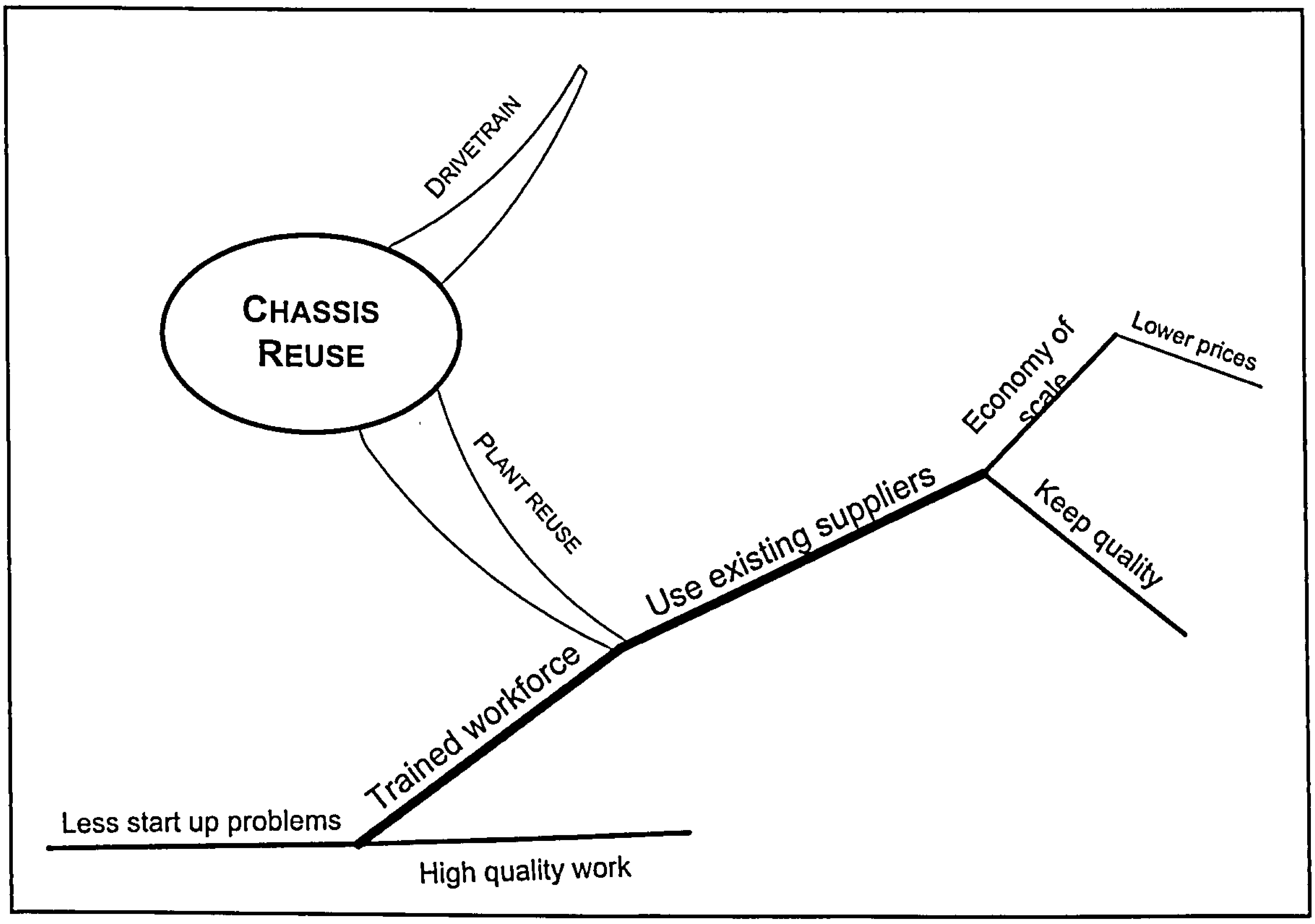
The shared information formed through collaborations is represented by XML data structures that can be processed in a manner to effectively filter out subsets of relevant data and thus produce more suitable traditional formatted information mediums for the collaborating supply chain parties. This allows the extraction of other useful user oriented data view ports such as:

- Natural language based reports;
- BOM (Bill of Materials);
- Supplier list;
- Design variable tolerances.

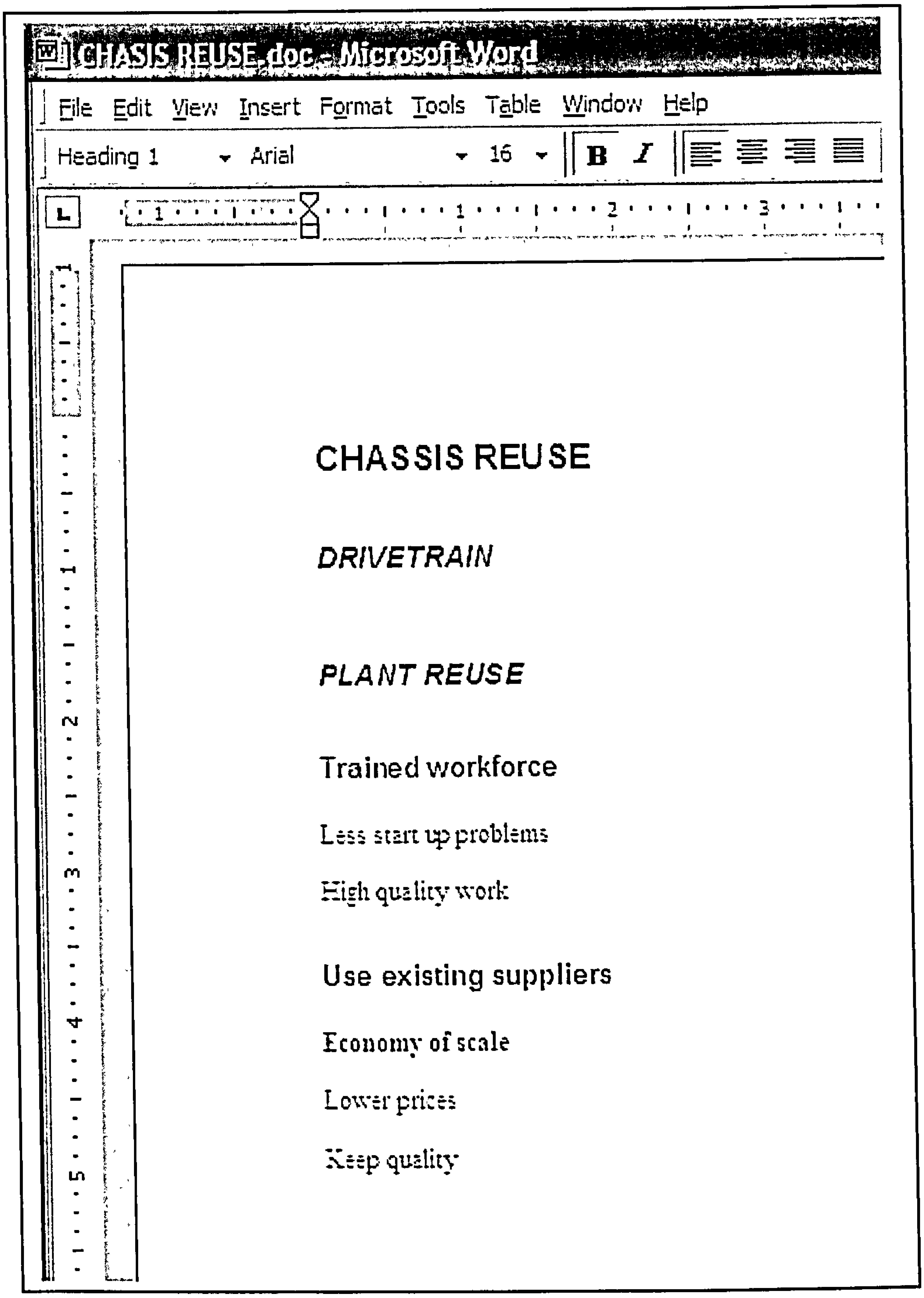
2.5.1. Natural Language Reports

A natural language report can be generated like a table of contents based on the branches of the concept maps that have text files linked to the concepts. The result of this processing would be a document framework for milestone / progress reports or even user guides or maintenance handbooks to be created. A complex language processor is required with a real knowledge for the semantics of a language to produce a more comprehensive document. However in the same manner that a software case tool can process UML (Unified Modelling Language) to provide the programmer with function headers and the outline of the code, exporting to a natural language document can provide an author with the outline of a document. Reducing the time from collaboration to written documentation.

(Figure 8) and (Figure 9) show a report from a concept map, for example taking the form of an early industry press release; the outline of the situation can be exported, reducing the time an expert needs to spend on assisting more menial tasks.



(Figure 8) Partial Concept Map – Chassis Reuse



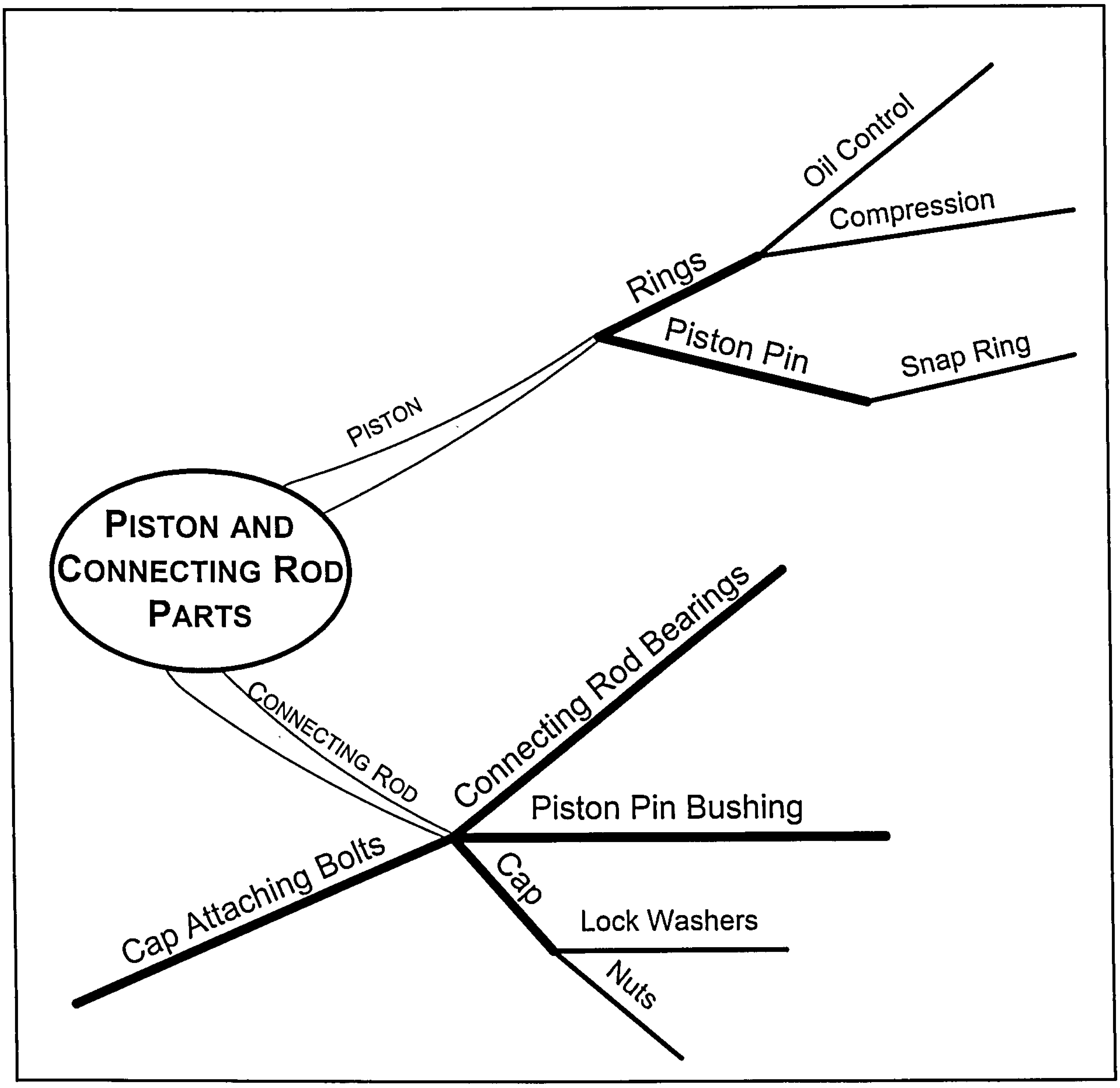
(Figure 9) Natural Language Document Output

2.5.2. BOM (Bill of Materials) & Supplier List

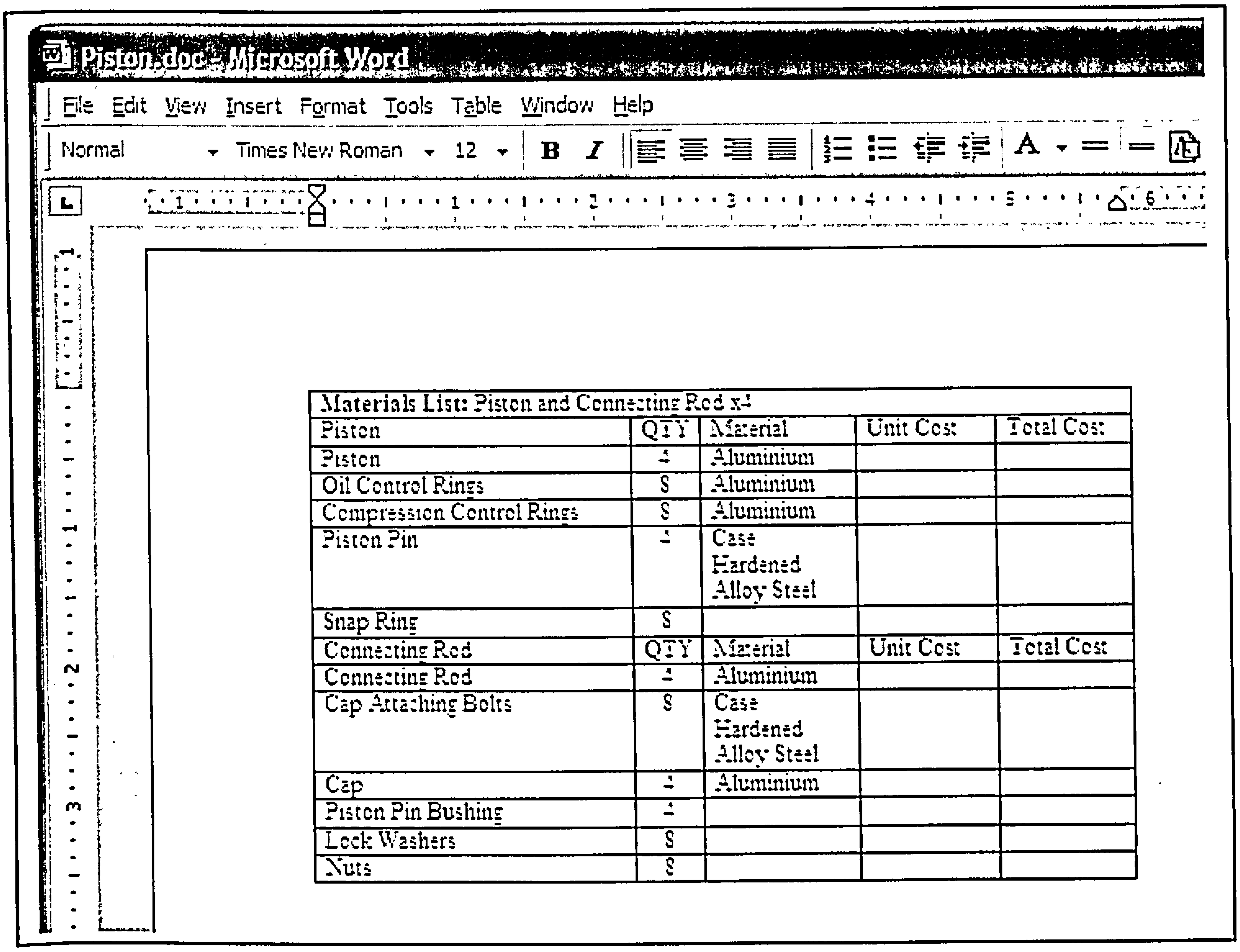
The information stored throughout collaboration represents high-level early concepts that are iteratively broken down into increasingly complex components. These different levels of complexity are capable of representation by the concept maps. Thus facilitating formats to be represented such as the parts and sub assemblies along with their associated materials.

Supply chain integration can be further improved by greater completeness at the time of concept definition. At more complex levels of representation where concept maps are representing the actual product parts and features, each feature provides the facility to associate the materials required to produce it.

Collaboration based on the Concurrent Engineering philosophy requires the earliest possible involvement of the collaborating parties; further to this these parties must contribute as early as possible. It is this early contribution of necessary information that provides greater completeness of concepts at creation. When information is of greater completeness at the time of creation it makes the retrieval and processing of information for later lifecycle tasks much more simple. The association of a supplier to a concept representing a part, either for the part itself or the required materials allows a supplier list to be generated for general contact purposes such as a company address book. Collaborating part and material suppliers linked to concepts thus have the facility to be made aware at an early stage of potential material requirements to assist in turn their own projected material and process planning. This information once approved by the collaboration can then be automatically processed into a more formally structured BOM for official parts ordering see (Figure 10) and (Figure 11)



(Figure 10) Partial Concept Map – Piston and Connecting Rod Parts



Materials List: Piston and Connecting Rod x4				
Piston	QTY	Material	Unit Cost	Total Cost
Piston	4	Aluminium		
Oil Control Rings	8	Aluminium		
Compression Control Rings	8	Aluminium		
Piston Pin	4	Case Hardened Alloy Steel		
Snap Ring	8			
Connecting Rod	QTY	Material	Unit Cost	Total Cost
Connecting Rod	4	Aluminium		
Cap Attaching Bolts	8	Case Hardened Alloy Steel		
Cap	4	Aluminium		
Piston Pin Bushing	4			
Lock Washers	8			
Nuts	8			

(Figure 11) BOM (Bill of Materials) Document Output

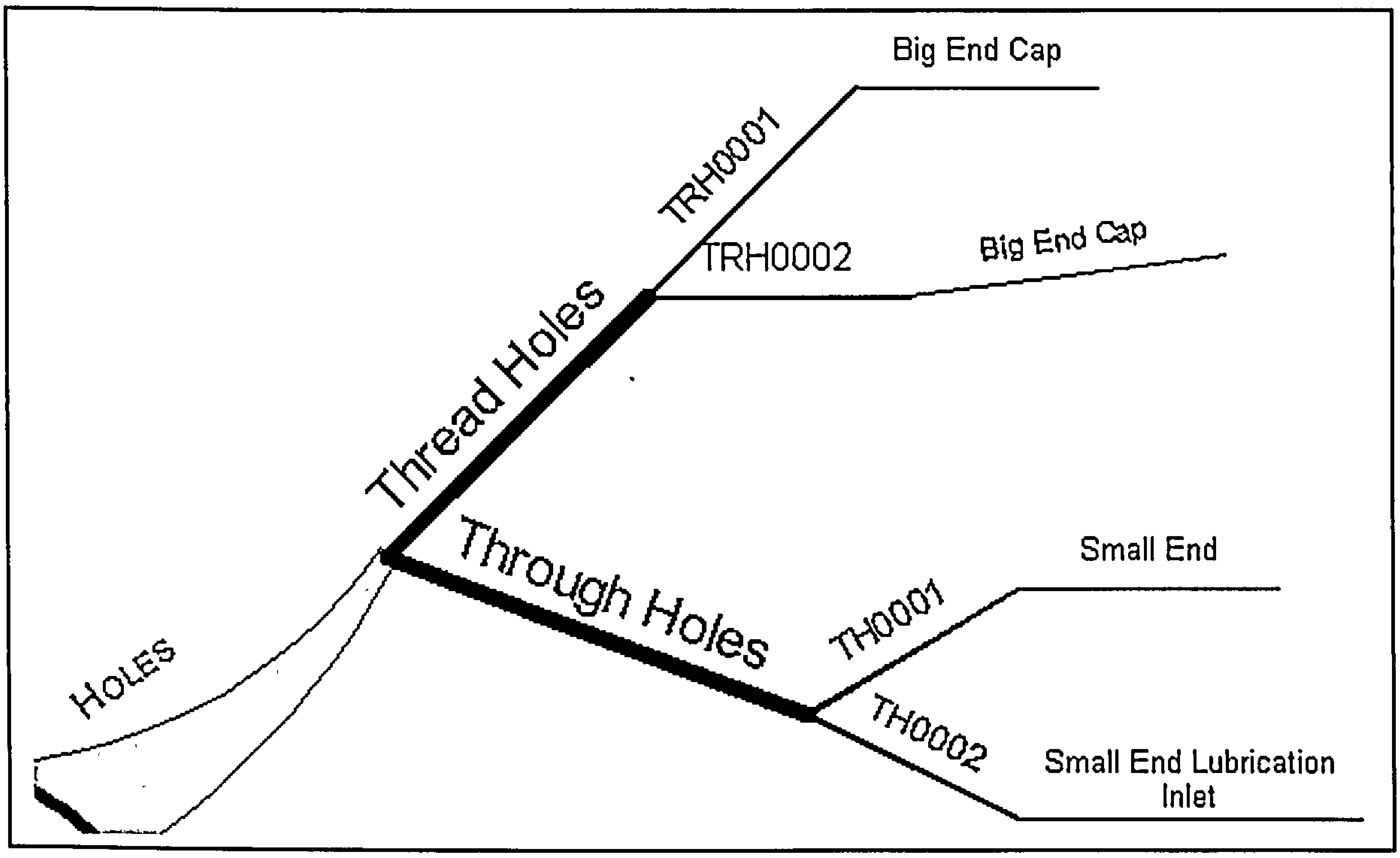
2.5.3. Design Variable Tolerances

Integration of other activities required in the early design stages facilitates the lowering of costs and the reduction of unforeseen problems occurring later in the product development lifecycle. A main purpose of collaboration is to identify requirements and raise conflicts based upon those requirements.

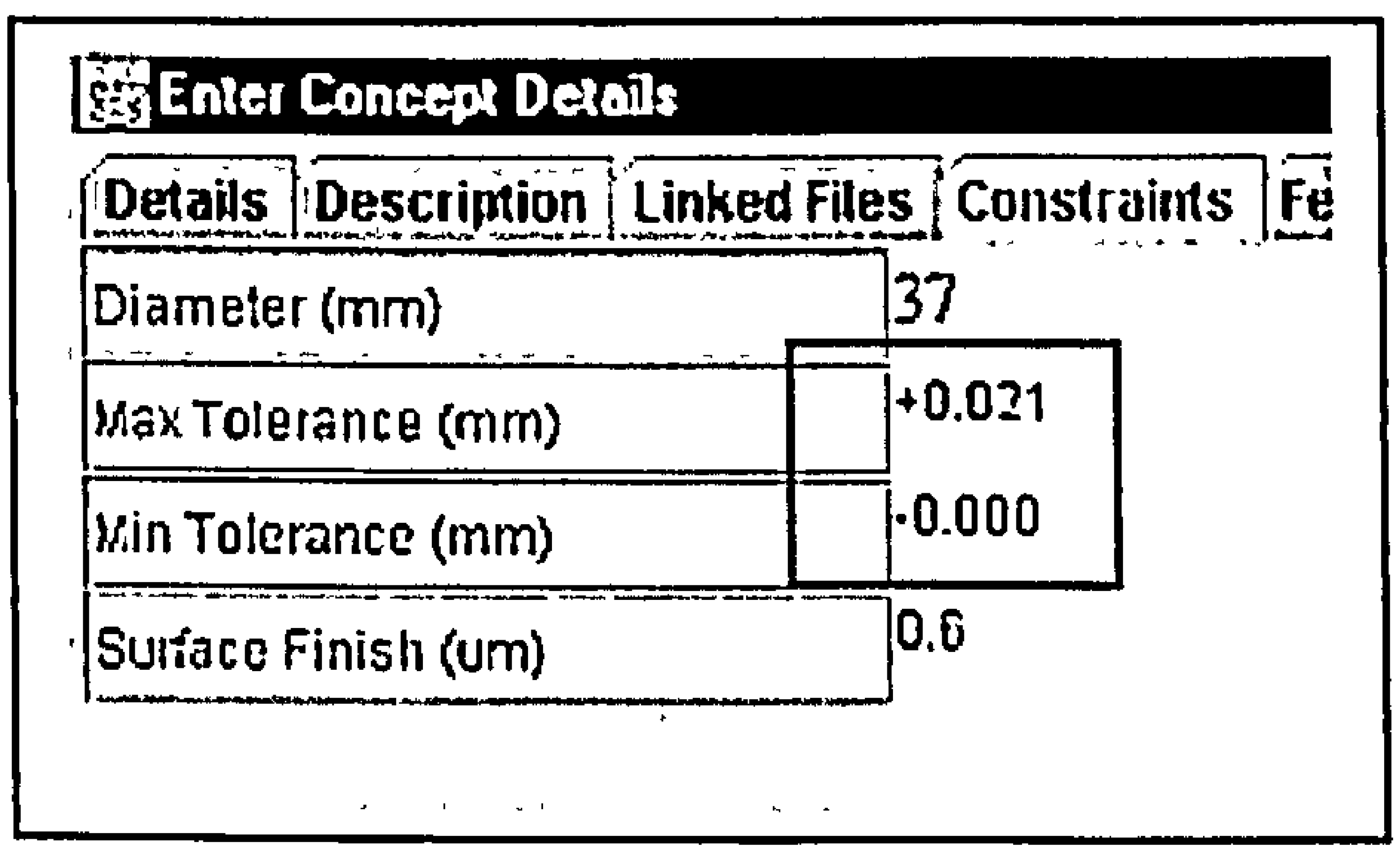
Complex requirements can be specified as design variables and associated with a concept. These design variables represent the tolerances of agreed constraints upon aspects of the product under development, a consensus reached by the collaboration.

The set of design variables that represent a product can be extracted in the same manner as the materials and supplier information. This effectively results in a checklist of concurred tolerances that can be utilised in prototyping or in later quality control development phases.

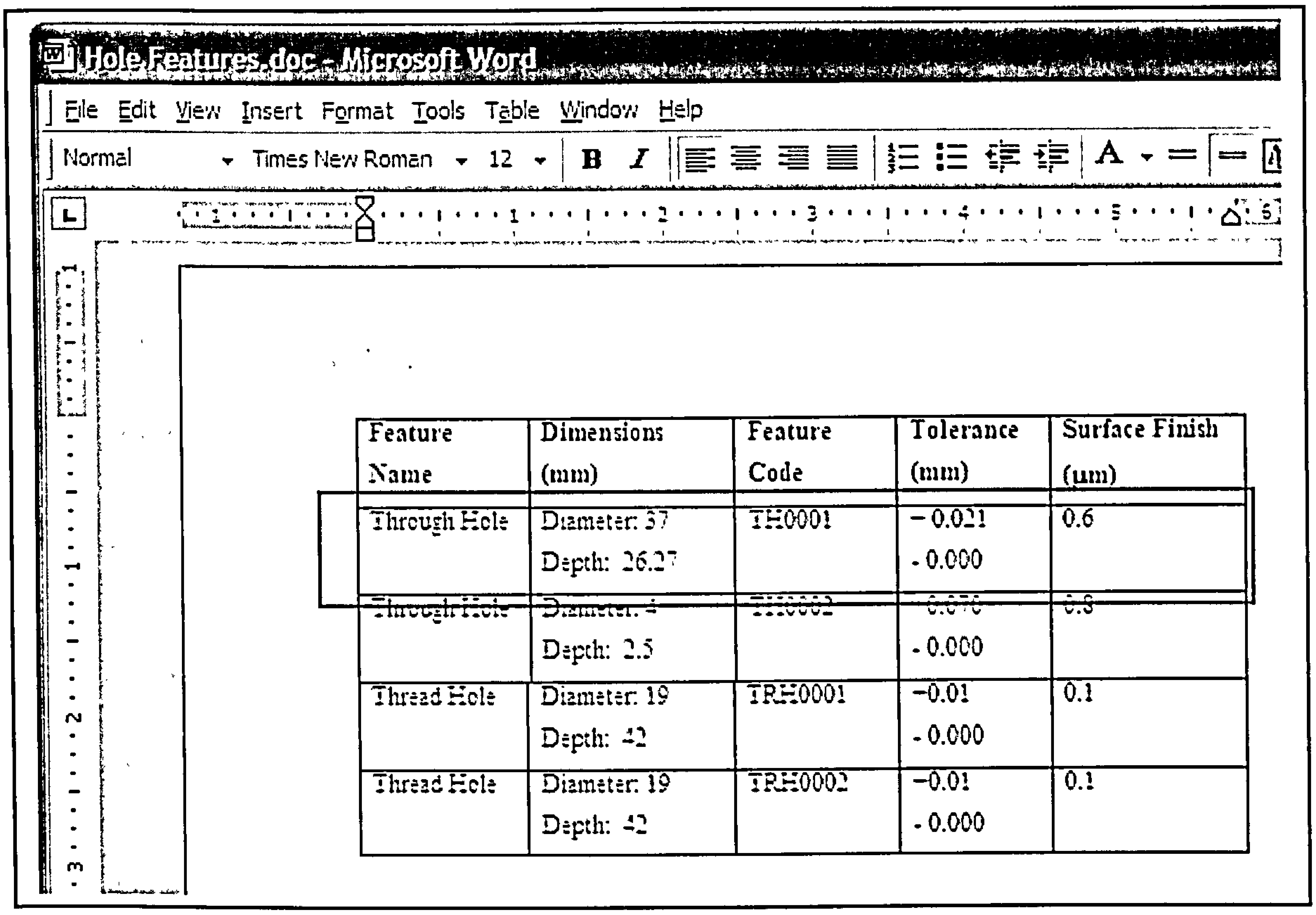
(Figure 12) highlights the tolerances in the associated concept information and these are again highlighted in the filtered document in (Figure 13) and (Figure 15).



(Figure 12) Partial Concept Map – Hole Features



(Figure 13) Feature TH0001 Highlighted Tolerances

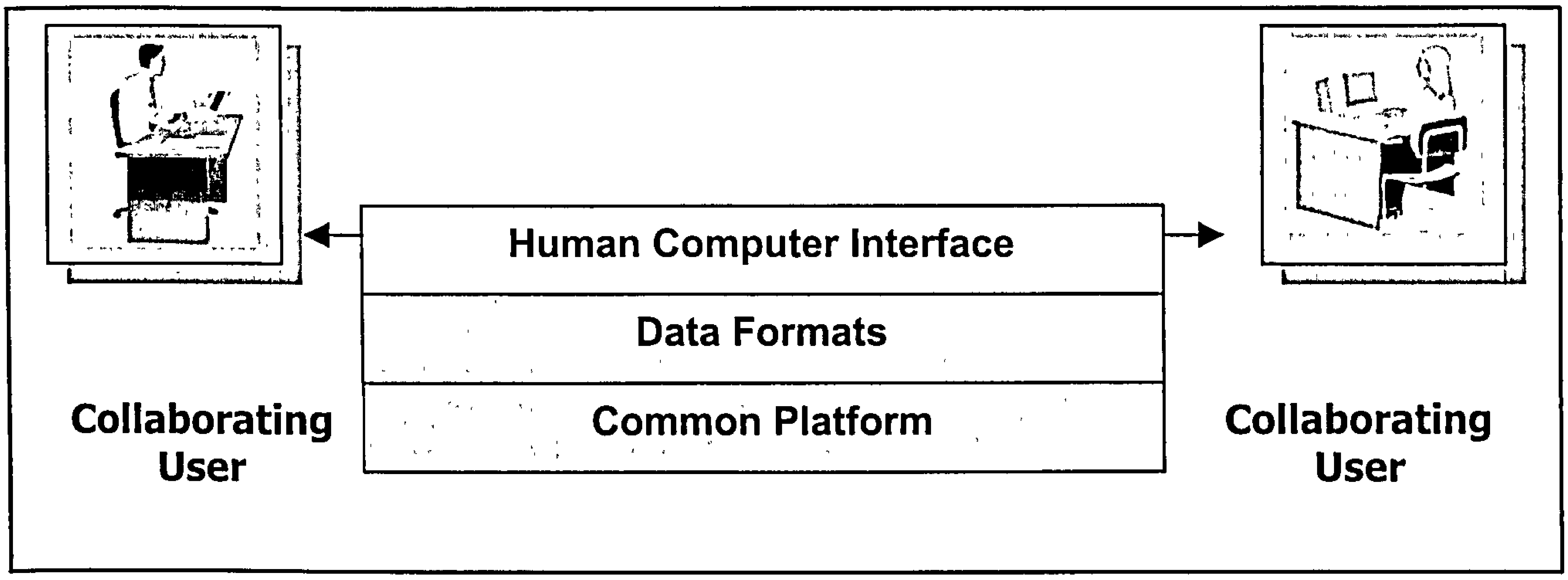


Feature Name	Dimensions (mm)	Feature Code	Tolerance (mm)	Surface Finish (µm)
Through Hole	Diameter: 37 Depth: 26.27	TH0001	− 0.021 - 0.000	0.6
Through Hole	Diameter: 4 Depth: 2.5	TH0002	-0.070 - 0.000	0.8
Thread Hole	Diameter: 19 Depth: 42	TRH0001	−0.01 - 0.000	0.1
Thread Hole	Diameter: 19 Depth: 42	TRH0002	−0.01 - 0.000	0.1

(Figure 14) Hole Design Tolerances Document Output

2.6. Data Standardisation

Integration of the supply chain necessitates standardisation at varying levels. Standardisation is essential as part of the protocol with which organisations collaborate. In an environment that supports multiple organisations there must be strong standard elements to provide a basis for collaboration. Standardisation must consider the common platform, data formats, and human computer interface see (Figure 15).



(Figure 15) Data Standardisation Elements in Collaboration

2.6.1. Standardisation in the Human Computer Interface

Distributed product development collaboration is essentially founded on the accurate sharing and conveyance of the collaborating parties viewpoints. All facets of standardisation are important but it is the human computer interface that the users will be confronted with as their guide or barrier to the other parties.

There are generally accepted user interface de-facto standards, such as the placement of screen furniture, the menu bar, toolbar, and document body. Each of these can be further decomposed into standard items, for example the file menu on the left, and the help on the right. Style guides for interface design are common, however it is usually the knowledge that is represented that is far from common.

The human computer interface has the responsibility to deliver the collaborative efforts of others; the concept mapping described provides a consistent manner for this. Standardising the way collaboration takes place is multi-faceted. The onus for representation of knowledge in this work is placed upon the concept maps. However there is a requirement to standardise the access methods to the collaborative formats.

Globally distributed collaborative product development crosses not only geographic but also temporal boundaries. Software interface development utilises de-facto interface standards to facilitate the ease of use of the interface. In the same way that interface elements should remain consistent from application to application, the same interface should be used for access to the collaborative environment regardless of temporal and geographic factors.

Geographic factors should be overcome through the use of web-based technologies, providing access to the same material regardless of locale. This material consists of the user interface embedded within a web page thus providing exactly the same experience. This also provides a suitable method for updates to the system. These can be undertaken at a single location and reflected upon all the users systems without the need to apply a large number of patches to installed software.

The integration of people in the product development process necessitates that the issue of common meeting slots be addressed. The Concurrent Engineering philosophy requires that the parties involved in the product development lifecycle are involved from the earliest possible stage. Domain specialists are generally the people involved, experts in their own fields, these are generally people whose skills are in high demand and thus finding a common meeting slot for the number of people required is difficult. Integration of these people in the global setting is even more difficult when combined with the world time zone differentials.

The majority of systems today provide mechanisms for these different communication types as previously termed, synchronous and asynchronous collaborations. For example chat room style facilities are provided for real time communications or video conferencing. These facilitate synchronous collaboration and require a common meeting slot to take place. Asynchronous or sequential collaboration is easily facilitated by email for example. The problem with using such mediums to address synchronous and asynchronous collaboration as previously discussed is their lack of integration, the information itself is dispersed, but also the lack of uniformity in the interface. Information needs to be represented in the same manner and not separated as a means of solving the synchronous / asynchronous problem.

2.6.2. Standard Data Formats

Integration of the supply chain necessitates that the collaboration utilises a common repository for the sharing of information. It has been stressed the importance of the comprehension of information, however to underpin this requires the employment of independent data standards.

Just as human communication requires common factors to enable comprehension, the systems involved in the collaboration also need certain standards to facilitate a common base to exchange data.

The worldwide access and scalability of a web-based system have made it the obvious environment for a global collaborative system from the view of functional capability. However the benefits that can be leveraged from global connectivity also are the cause of problems in the development of common standards for information exchange. There are a number of organisations pushing forward their versions of browsers, client and server programming languages and platforms, each looking to gain competitive advantage from the others. There is no de-facto method for web based systems development at present; each technology has its own advantages and disadvantages.

Standards for data formats envelop the Internet community at present, the W3C (World Wide Web Commission) generally spur the advance of these standards and recommendations. However many of these standards are corrupted through extensions of browsers by vendors and organisations to attract a greater number of users, this in turn provides a testing ground for ideas and whether they are taken up by the development community as a whole or not. This situation leads to an unstable development environment for web based applications, ever evolving standards and technologies with constant modifications make it difficult to adhere to a set format for information representation and hence collaboration.

There are however in this diverse range of technologies several that have widespread industry backing, XML and Java are those used in this research. XML is utilised in this research as a mean of providing a stable and portable collaboration data format.

XML is a relatively stable standard within this dynamic development environment and thus was chosen for the representation of the concept maps ensuring their compatibility and comprehension for at least the medium term future. XML has been used to facilitate information sharing across the Internet far beyond the now ubiquitous web page.

Collaboration efforts require serious investment in technology and the data that is stored in supportive repositories. The parties involved in the supply chain are committing product development data to a specific collaboration environment and thus this data must be independent from the application that created it. XML provides the means to create a data format that can be processed by a wide range of software tools. A simple web browser for example with a style sheet could format the data. This separation is usually not necessary in the general computing environment however product development data has a long life that usually far exceeds that of a software system regardless of how scalable it is.

Concept mapping provides a very powerful mechanism for collaboration, however there is a need to provide greater facility for the complex product detail involved in the collaboration of a supply chain. This requires the representation of the product development data itself. It is possible for XML to be used in this realm, however the developments in this area have been limited and relatively primitive.

2.6.3. Common Platform Standardisation

Product development collaboration is facilitated through standardisation of the human computer interface and in the data formats used and it is essential that these are supported by effective standardisation of the common platform. The common platform in this research is facilitated by Java and the HTTP protocol.

Adequate human computer interfaces and data formats can be developed on platforms using different technologies. Utilisation of HTTP and the web facilitates a common platform based on Java to be available to all parties without dependence on specific operating systems or environments.

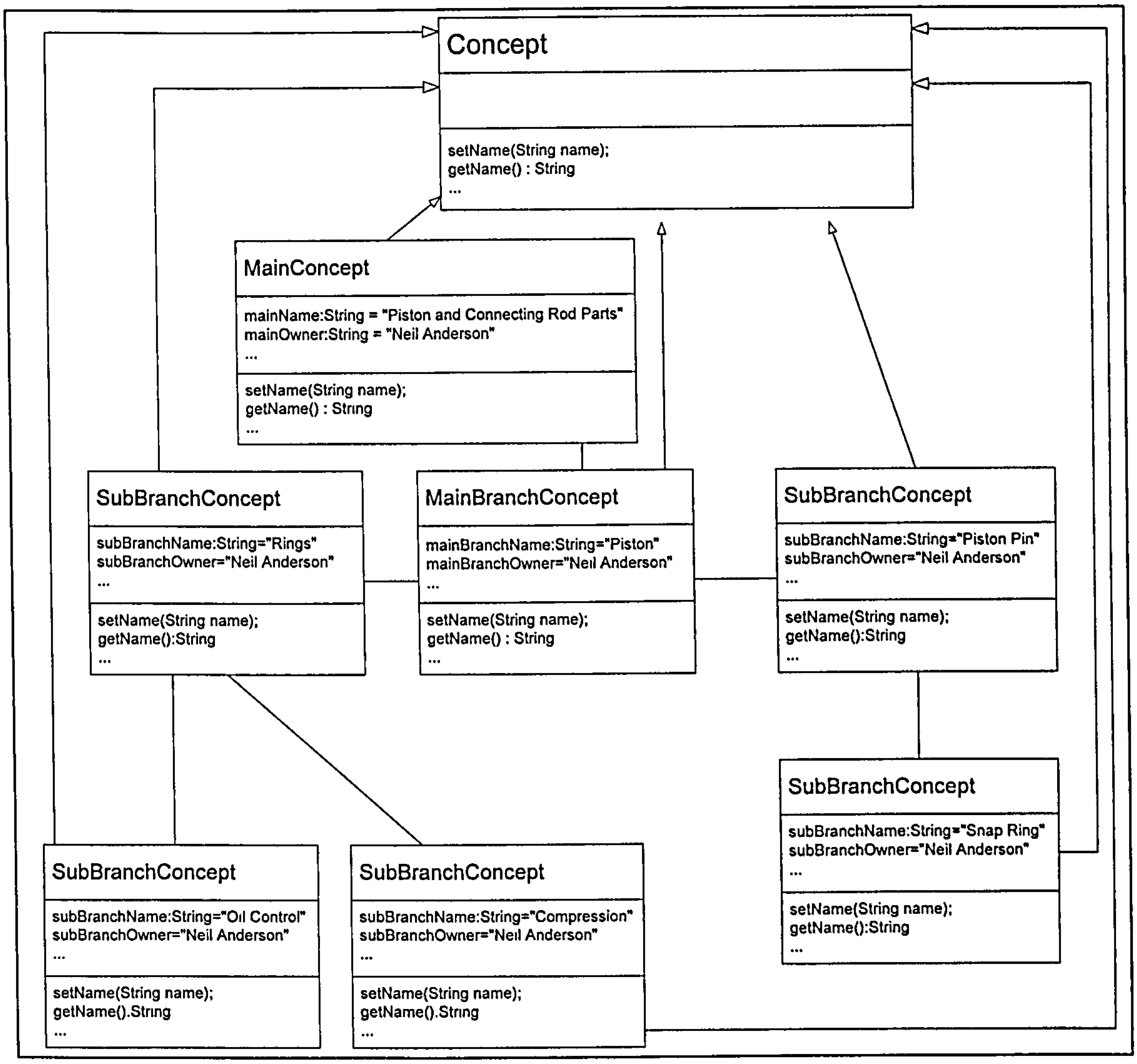
A common platform in this research is centred on an applet that provides the application interface for the collaborating parties. This communicates with a central server via the RMI methods.

2.7. Object Oriented Database

Integration of the supply chain during collaborative product development necessitates the maintaining of a large number of complex relationships between data entities.

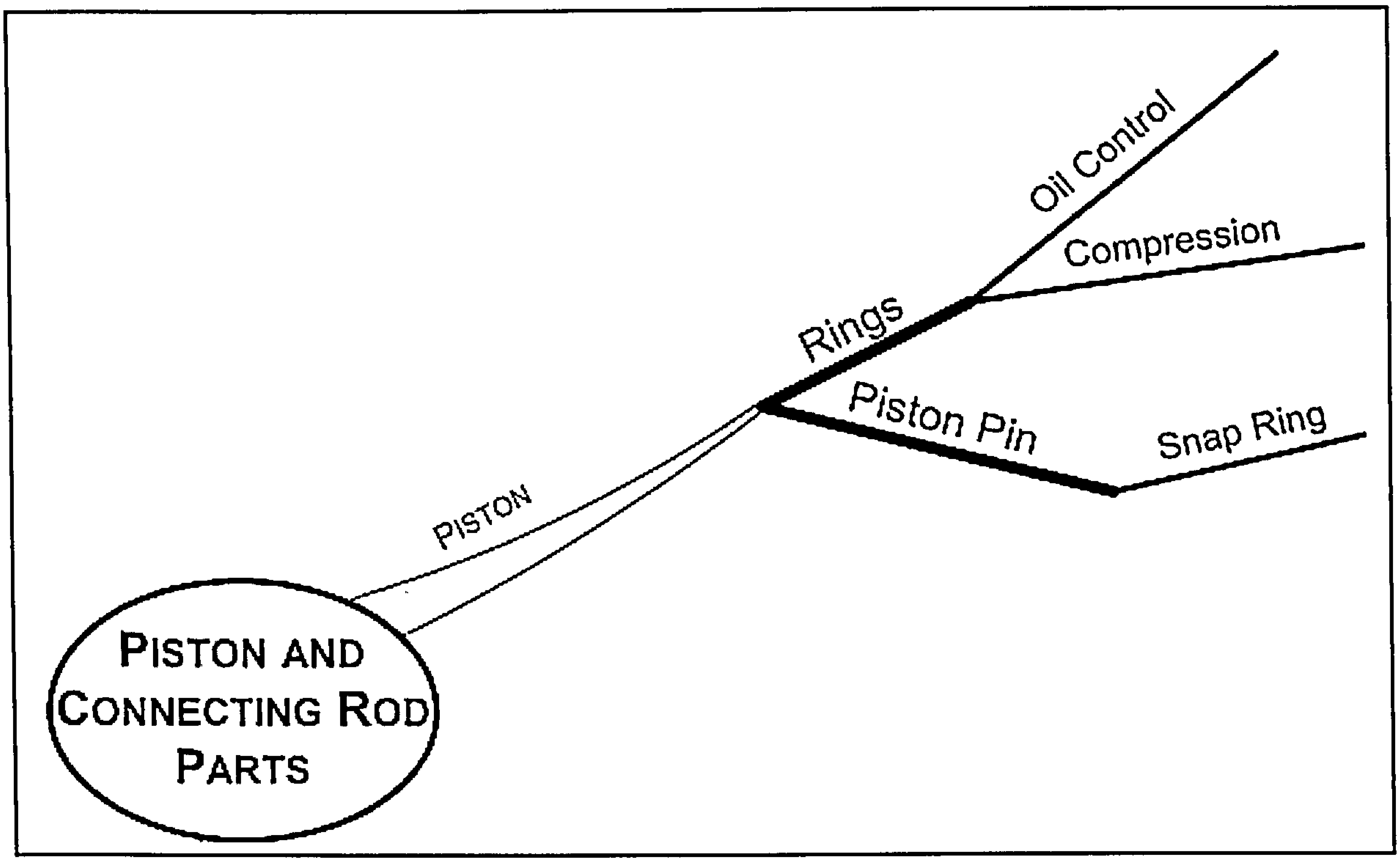
Barry (1996), details the choices between relational and object oriented databases. An object-oriented database has been chosen for this research work due to the inherent capability to represent complex relationships present in the product development data. There are effectively two databases one for the storage of the concept map structures, and their relationships and one for the actual product data.

The concept map class structure is represented in UML and the class structure developed in Java and is stored as such in the Object Database, (Figure 16).



(Figure 16) UML Class Diagram Showing Concept Relationships with Respect to the Java Class Organisation

There is an abstract class that represents all possible types of concepts and the three main types of concepts override this, the main central concept, the main branches and the recursive sub branches.



(Figure 17) Example Piston Concepts

(Figure 17) details the parts involved in the piston section of the concept map, each concept is represented and the relationships between concepts detailed by the enclosing of the end tags. Where a concept is connected to another then its parent is defined by an opening tag for example "<MAINCONCEPT>" then the details about that concept are defined in between the "<CONCEPTDETAILS>" tags and then the child concepts are defined such as "<MAINBRANCH>" until the closing tag of the concept such as "</MAINCONCEPT>".

The product database is facilitated through the use of STEP Fowler (1995). There is a requirement for a serious investment in the development technologies. In this research the product data has been represented at the granularity of a single model represented per file. With the utilisation of the full SDAI (Standard Data Access Interface) of the STEP standard the object database can be used to for representation down to attributes of the entity levels providing a much finer granularity and hence greater concurrency for a collaborative team.

The two databases are linked through the linking mechanism already discussed. The product data is reached through the concepts embedded within the concept maps via a URL that contains the query to return the product data from the database.

2.8. Conflict Resolution

Collaborative product development is based on obtaining a consensus of opinion from a group of domain experts concerning a product. The purpose of this collaboration is to identify the differences in perspectives and to evolve new combined viewpoints. This allows the creation of more balanced products that satisfy a wide range of needs and criteria.

Facilitating the observation of conflicts should occur through a variety of mechanisms. Conflicts must first be identified before they can be resolved, how to actually quantify conflicts is a difficult problem, conflict on the low levels aspects of product development can be alerted through constraint-based formulae however there has been less consideration towards the resolution of high level subjective conflicts. These are based on the problems early in the product development lifecycle before a comprehensive set of design variables can be determined. These early problems often have little in the way of tangible characteristics and can be subjective in nature. Proceeding through the concept mapping process encourages the domain experts to reveal their opinions on the product at an early stage, each sees each others contributions in context and through a logical rational.

The system allows contributions to be made by the various domain experts but they must be entered within context, the user must add their contribution at its point of relevance increasing the rational thought that is undertaken before the idea is put forward assisting in conciseness of the issues. Conciseness in turn facilitates the observation of the core ideas of importance and thus a greater chance to detect inconsistencies in approaches. With ownership attributed to a concept or an idea it is then possible to understand the perspective that the contribution was made from and place it not only within the context of the surrounding concepts but in the context of the creators own knowledge base. This also facilitates the notification of the creating party if another team member wishes to modify the concept. In a system where there is purely notification based upon tolerances and design variables conflict can be detected however the rational behind the conflict is often less obtainable, whether the new addition causing the conflict should be modified or the original values kept.

The concept mapping framework facilitates the resolution of conflicts at these more subjective and difficult to quantify levels of abstraction but also as the concepts become more formalised in later design phases with the focus switching to the concepts representing design variables and their associated tolerances it becomes possible to utilise the graph structure that underpins the concepts as a constraint network. A constraint agent traverses this network of constraints producing a quantified series of design variables that fit the formulae defining the constraints between variables.

2.8.1. Constraints Networks Derived from Concept Maps

A constraint network consists of a graph structure of nodes / variables connected through a set of constraints between them. Solving a constraint network or highlighting conflicts is essential in collaborative systems to integrate different requirements and provide the resolution of problems early in the product development lifecycle.

The integrated collaborative environment as put forward in this research necessitates that the constraints and the problems they will describe are relatively unknown. The constraint networks are to be constructed from the constraints entered in the concept dialogue boxes in the concept maps. These concepts are capable of representing assembly hierarchies for parts or features of parts as shown in (Figure 18) and (Figure 19). Thus a constraint network can be modelled and therefore solved from the concept map information.

In keeping with this derivation of a constraint network from a concept map, the constraints solving the examples as follows are based on information derived from a concept map.

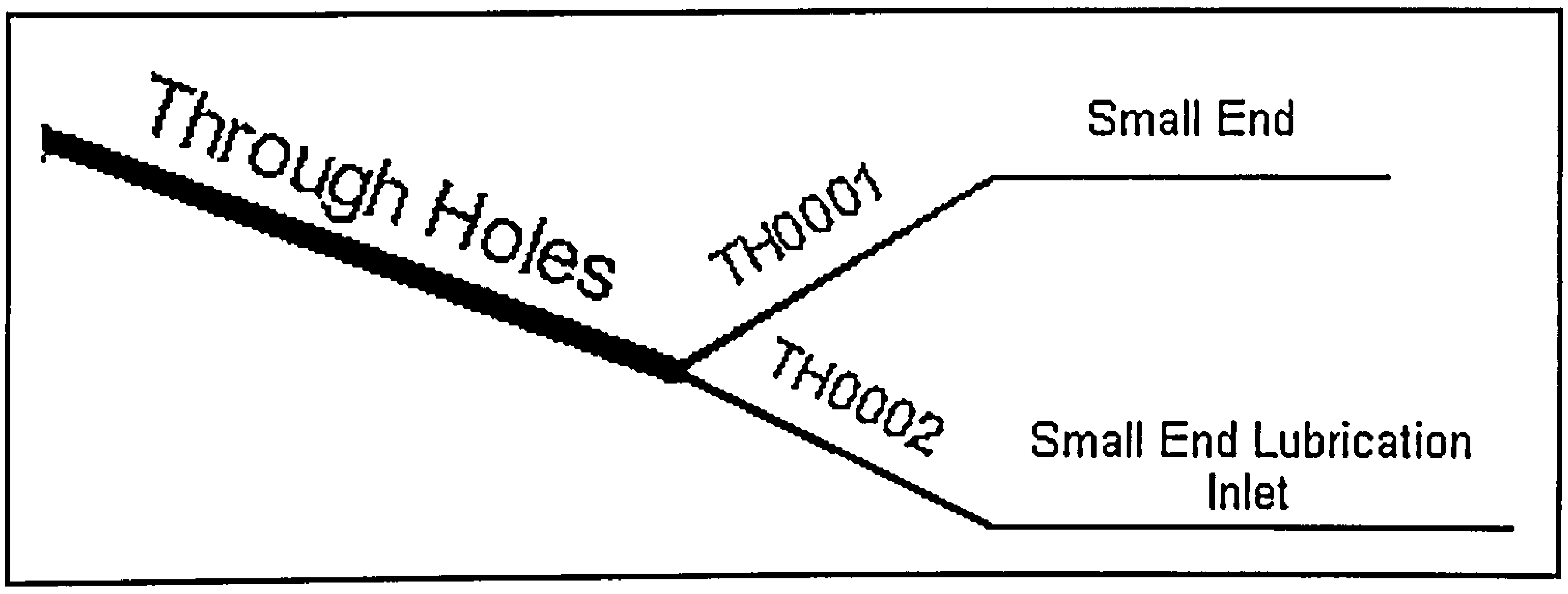
2.8.2. Solving the Constraints Network

A constraints problem consists of the following Barták (1998):

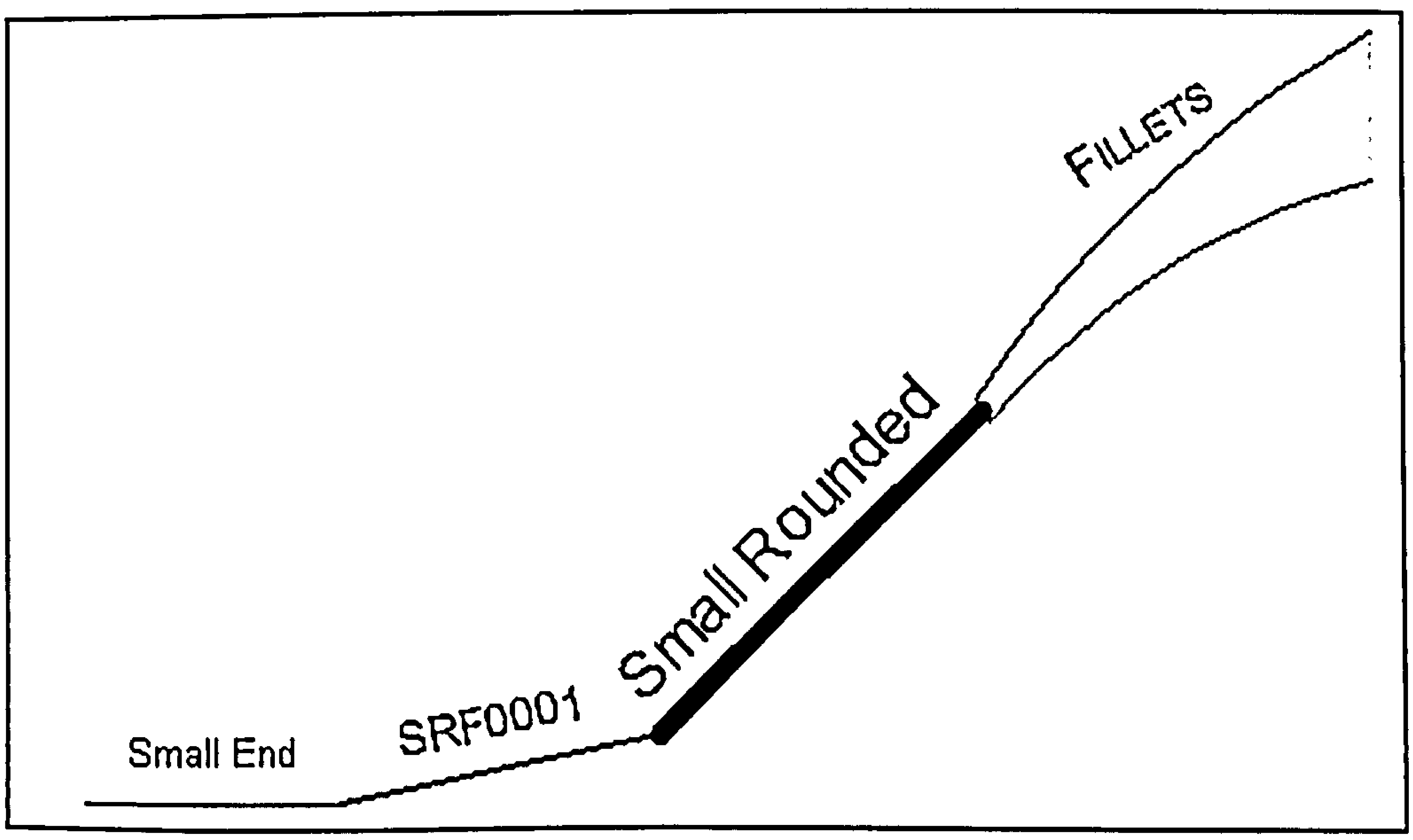
- A set of variables
- Domain values for each variable
- Set of constraints between each pair of variables

The solution to a constraints problem can be found by the generation of all the combinations of the design variables and then these are tested to check against the constraints. However this method is computationally intensive and exponentially more complex with each additional design variable and constraint that is added.

Successful solution of a constraint network involves a multitude of steps each affording gains in efficiency towards the discovery of a solution or the knowledge that no solution with the current set of design variables exists. In clearly defined problems with specifically known design variables it is possible to choose algorithms that are finely tuned to the resolution of the constraint network with optimised look forward methods predicting which branches of a search space should be avoided.



(Figure 18) Features TH0001 and TH002



(Figure 19) Feature SRF0001

In this example, several features have been extracted from the partial concept map:

SV = (TH0001, TH0002, SRF0001) is a set of design variables.
TH0001 = (0.000 … X_n) is a set of positive values for TH0001.
TH0002 = (0.000 … X_n) is a set of positive values for TH0002.
SRF0001 = (0.000 … X_n) is a set of positive values for SRF0001.

DC = [(37 <= TH0001 <= 37.021), (4 <= TH0002 <= 4.070), (8 <= SRF0001 <= 8.500), ((TH0001 / TH0002) < 9.5), ((SRF0001 – TH0002) < 4)] is a domain of constraints between TH0001, TH0002 and SRF0001.

The overall methodology for solving a constraints network is to simplify the network through a series of mechanisms. This simplification results in a correspondingly reduced search space and thus faster computation.

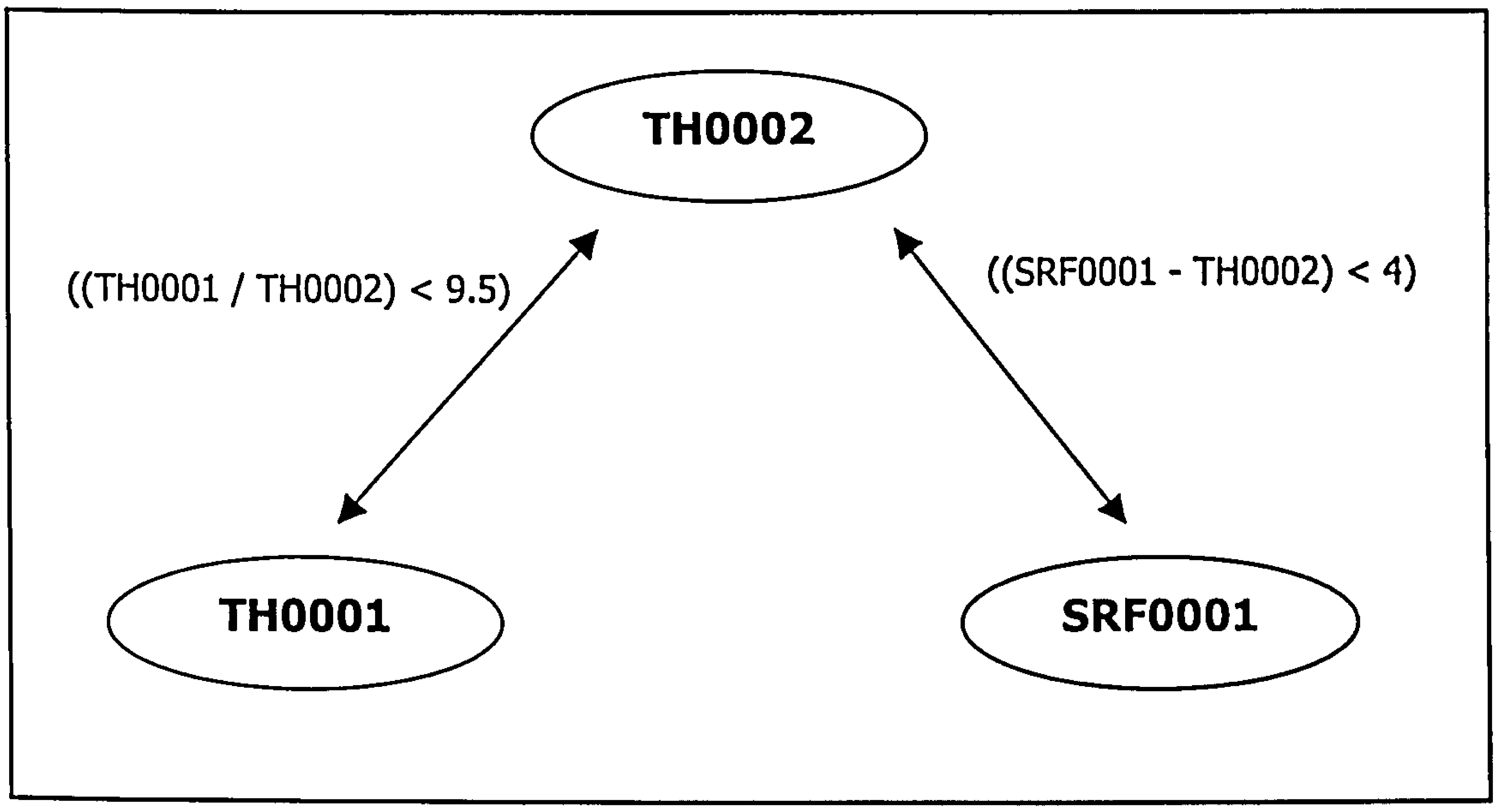
The generate and test method for searching the potential solution space of the constraints network involves searching through a number of combinations equal to the Cartesian product of the design variables involved (TH0001 * TH0002 * SRF0001). Currently the design variables TH0001, TH0002, and SRF0001 are positive floating-point values, involving a huge potential search space limited in reality by the numerical accuracy of the system.

A unary constraint is one that involves a single design variable. Thus the first step in solving the constraints is to limit the potential search space involved (Figure 20). A practical mechanism is to reduce the domains of the variables to those dictated by the unary constraints. Thus the new status of the design variables after the removal of those values omitted by the unary constraints is as follows:

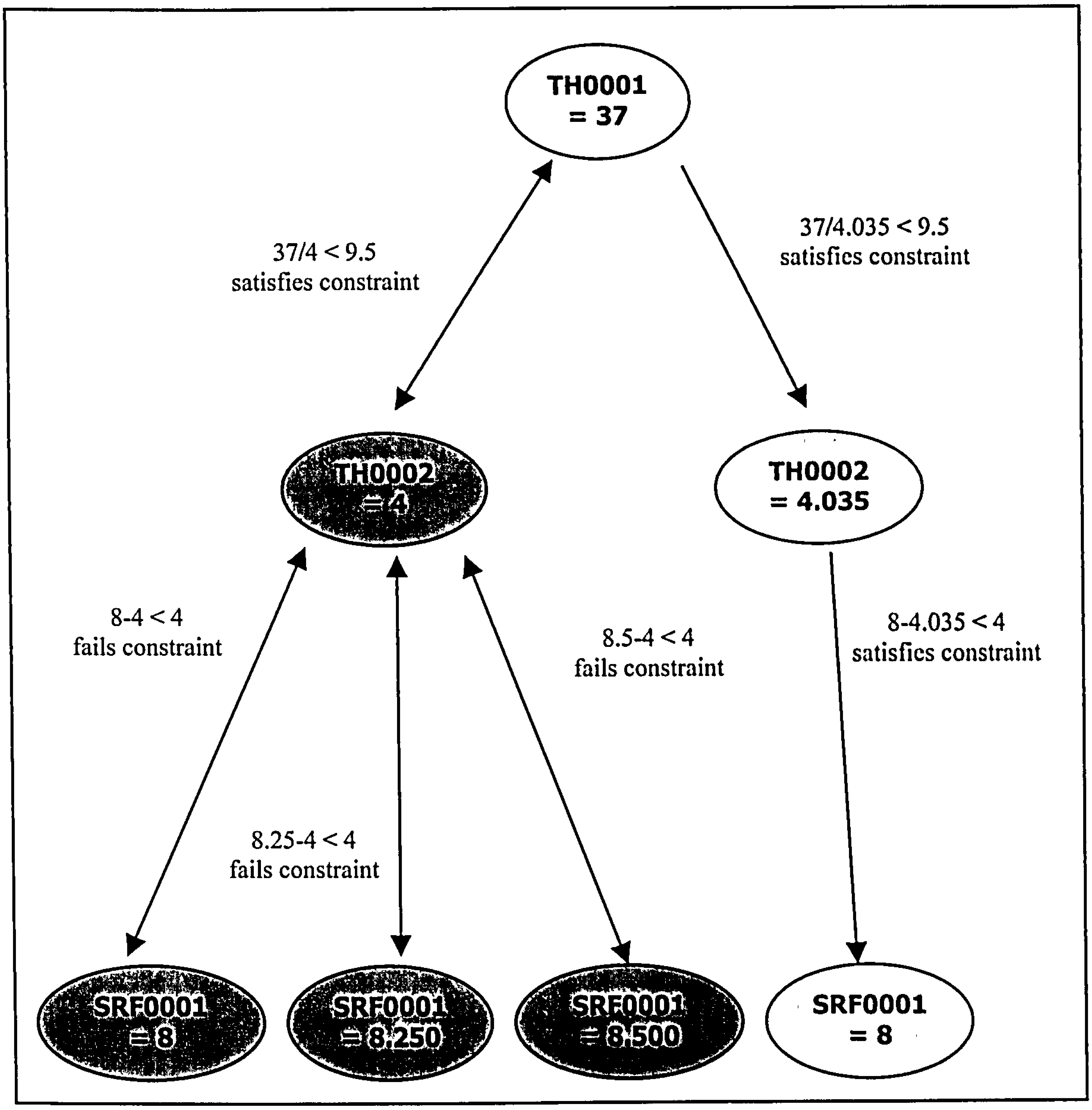
SV = (TH0001, TH0002, SRF0001) is a set of design variables.
TH0001 = (37 … 37.021) is a set of values for TH0001.
TH0002 = (4 … 4.070) is set of values for TH0002.
SRF0001 = (8 … 8.500) is a set of positive values for SRF0001.
DC = [((TH0001 / TH0002) < 9.5), ((SRF0001 – TH0002) < 4)] is a set of constraints between TH0001, TH0002 and SRF0001.

The "generate and test" approach as it name suggests generates a combination of design variables and then tests the constraints with those variables to check for invalidity. In this relatively simple example the "generate and test" scenario would be capable of finding a solution to the network. However in more complex examples, even reducing the potential solution space in this manner, still results in large search spaces that grow exponentially with each additional design variable. Only partially generating a potential solution and then testing the applicable constraints can achieve an obvious gain in performance.

The backtracking algorithm takes this approach by generating the first design variable and then the second; the algorithm then tests these variables against the relevant constraints between them. If they satisfy the constraint, the next variable is added and consequently tested. This process continues until a variable does not satisfy a required constraint. Upon a failure the variable is changed to its next value and retested. If a domain for a variable is fully tested without satisfaction for the constraints then the algorithm backtracks to the previous variable instantiation and changes that to its next value and proceeds the testing once more.



(Figure 20) The Constraint Network after Unary Domain Reductions



(Figure 21) Search Tree for Constraint Network

(Figure 21) details the search tree for the constraint network. The constraint between TH0001 and TH0002 is not violated however for the constraint between SRF0001 and TH0002 to be satisfied the search backtracks and increments TH0002 to 4.035 and resets SRF0001 to the start value in its respective domain which provides satisfaction of all the constraints.

Backtracking is the basic mechanism for solving the constraints network and not the most effective however further optimisations of the search algorithm are for further integration in research.

2.9. HTTP Accessed Documents

These have been touched on in the other areas of the system, the HTTP accessed documents are the key to providing support for multiple and commonly unknown file formats. They are accessed in much the same manner as you would the ubiquitous web page through a URL.

A URL is not just a link to a web page or a picture; it can also be the construction of a database query provided this database is available through the Internet. This area also includes the submission of data through HTTP for further processing.

HTTP forms the foundation for access to other document types leaving the web browser technology to either display the file or to retrieve a plugin capable of extending the web browsers capabilities to display it.

3. Conclusions

The research work described in this paper contributes to the area of collaborative product development by addressing the integration of people, product development processes and technology throughout the product development lifecycle.

The globalisation of competition has led to greater increases in product rivalry than ever before. Major factors in this globally competitive market are cost, functionality, and quality. To enable a product to compete on these facets of competition necessitates the collaboration of the entire product development team from the earliest possible stage of the product development lifecycle. Early distributed collaboration in the context of product development is essential in the identification of potential problems and design errors before they consume a significant investment of resources.

This research has involved the development of a novel concept-mapping framework to address the lack of consideration for integration of the main aspects of the product development lifecycle. This paper has described a continuation of earlier work demonstrating the capacity for the concept maps application to specific domain requirements. Demonstrating the importance for the consideration of a wide range of aspects in the domain of collaboration. These aspects have been grouped within the categories; data sharing & structuring, data standardisation, object oriented database, conflict resolution and HTTP accessed documents. The main points put forward in this publication of the work were:

- The parsing of XML to provide support for multiple report types, natural language, BOM, and supplier lists;
- Linking of concept maps to themselves and external file types;
- The utilisation of standards to promote platform and application independence enhancing the longevity of system generated data;
- The representation of concepts within the object oriented database;
- How conflict resolution can be improved at differing levels of information complexity;
- How the mechanisms already in place on the WWW can be used to enhance an open system allowing extensibility in a variety of ways.

Accurate enhanced collaboration information sharing repositories facilitate the reduction of problems and inconsistencies during the product development lifecycle through early detection and resolution of conflicts. Therefore this research contributes a collaborative environment capable of increasing an organisation's competitiveness through the lowering of costs, increased team-building opportunities, reductions in time to market and an increase in product applicability and quality.

4. References

Anderson, N. and Abdalla, H., "Web Browser Based Collaborative Product Development for the 21st Century." *8th ISPE International Conference on Concurrent Engineering: Research and Applications - CE 2001* , Anaheim California, USA, pp 125-134, 2001

Anderson, N. and Abdalla, H., "A Distributed E-commerce System for Virtual Product Development Teams." *Proceedings Institute of Mechanical Engineers Part B: Journal Engineering Manufacturing*, no. 216, pp 251-264, 2002

Anderson, N. and Abdalla, H. S., "Distributed Information Sharing for Collaborative Systems (DISCS)." *Information Visualisation'99* , IEEE Computer Society, London, pp 476-481, 1999

Barry, D. K., *The Object Database Handbook, How to Select, Implement, and Use Object-Oriented Databases*, Wiley & Sons Inc, 1996

Barták, R., *ON-LINE GUIDE TO CONSTRAINT PROGRAMMING*, [WWW] Available from: http://kti.ms.mff.cuni.cz_bartak\ [Accessed: 9/12/2002]

Buzan, T. and Buzan, B., *The Mind Map Book, Radiant Thinking, The Major Evolution in Human Thought*, BBC Books, 1995

CENTEX, *CENTEX, Concurrent Engineering Needs & Technologies Experimentation*, [WWW] Available from: http://esoce.pl.ecp.fr/ce-net/Information/cerelated/centex/Projectoverview.htm [Accessed: 16/3/1999], 1999

Fowler, J., *STEP, For Data Management Exchange and Sharing*, Technology Appraisals Ltd., 1995

Prasad, *Concurrent Engineering Fundamentals: Integrated Product and Process Organisation*, Prentice Hall International, 1996

Appendix B

RMI Development Process

Remote Method Invocation (RMI) Development Process

- Define a remote (server) interface.
- Implement the remote interface.
- Compile your server class as normal, using javac.
- Run the stub compiler, rmic.
- Start the RMI registry on your server machine by executing the program rmiresgistry.
- Load and run your server code.
- Write the client code.
- Compile and run the client class.