The Implementation of An Object-Oriented Database to Analyze Concurrent Engineering

Amit Ahuja
Sr. Solutions Consultant, A G Convergence WLL, Doha (Qatar)

Abstract

In this paper, we discuss the requirements and design of a Database that can be used to support a Concurrent Engineering (CE) philosophy. We identify three fundamental support services that are of use in a concurrent engineering environment: distribution support, computer supported co-operative working support and database support. These requirements are modeled and applied in conjunction with the design and implementation of a concurrent engineering support Database.

Keywords

Concurrent Engineering, Support Environment, Object-Oriented, Distributed Systems, Computer Supported Co-operative Work, Database

I. Introduction

Concurrent Engineering (CE), or simultaneous engineering or integrated product development has been defined as: "... a systematic approach to the integrated concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset to consider all elements of the product life-cycle from conception through disposal, including quality, cost, scheduling and user requirements" [30]. The capturing of data requirements to support this philosophy is large and complex. It encompasses a large number of wide ranging and disparate objects. For example, consider a design object which may contain many hundreds or thousands of sub-components in a hierarchical fashion. The same design object may have multiple views, i.e. different aspects of the information may be required by different application programs in the design and manufacturing cycle, for example in the production of a Bill Of Materials (BOM), calculating tool paths and performing engineering analyses. It can be seen then, that the data pertaining to a single product is polymorphic. In addition to this type of data, a concurrent engineering philosophy requires an integration of all functional areas including for example: sales, marketing, design and process planning. So a database designed to support this philosophy also needs to cater for supplier information, customer information, stock levels and production schedules to name but a few. One method of facilitating a concurrent engineering philosophy is through the application of "the Virtual Team concept" [28]. A virtual team is an integrated project team linked through the use of information technology and which usually has one designated person to champion the project and lead or guide the team. Fig. 1 shows an example structure and composition of a virtual team in which it can be seen that the various functional areas involved in the development of a product are linked through the use of a common product information model. A common information model consists of product model data, resource information and process information all intrinsically linked, i.e. a change in one item of data in the model may have an effect on other data items.

Various attempts have been made at modeling product data, resources and processes in isolation and more recently in combination by Molina et al. who decomposes a common information model into a Product Model which: "captures information related to a product throughout its life cycle" and a Manufacturing Model which: "captures all data related to process capabilities, characteristics and resources within an Enterprise" [15]. The integration of the three model aspects described above (product, resources and processes) and the implementation of a model to support this through the use of an object- oriented database is the subject of this paper.

Section II of this paper describes the background to the work that is currently being undertaken by the authors and others and puts this in the context of previous work in this field. Section III, discusses a set of requirements that can be used to support concurrent engineering and then describes services that are being built to satisfy these requirements. In section IV, we propose a model for supporting concurrent engineering and show how this model can be applied to a real life engineering scenario. Finally, a description of the current implementation of these models and some conclusions are given in sections V and VI.

II. Background and Related Work

Many current environments that support the process of software development rely on a logically if not physically centralized data database. This is based on existing operating system kernel design and services. In this framework, the operating system consists of the nucleus, itself a minimal set of requirements, and then built on top of it are layers of specialized services for example a file service, directory service [22].

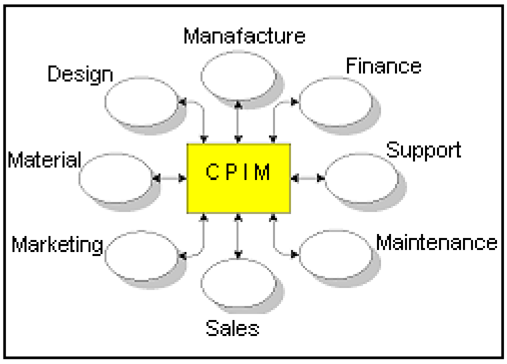

Fig. 1: The Virtual Team Concept

A user process would then interact with the uppermost layer for its processing needs. A user request is then processed as required, by each layer in turn until it is completed. Adhering to this scheme of support a number of tools have also been built as self contained packages which may span one or more layers. Furthermore, just like the operating systems layers, firewalls are built-in between each of these services. These restrictions make them less amenable to be used for co-operative working. In addition, many of these support environments [24], comprise of many discrete components, are so tightly coupled that the user is not able to choose components in order to perform a given computation, but rely on one large and complex entity. In this scheme there is a hierarchy of usage and privileges. To allow the concurrent access of files, a capability must be communicated to the operating system, if the environment and operating system allow for it. For example, in the UNIX operating system the access control attached to a singly owned file could be made to allow for a group of people to concurrently read it. However, the system does not provide for the management of the multiple access of this file which can lead to invalid data. In addition this shared access must be explicitly authorized by either the owner of the file or the system administrator which is a restriction on truly co-operative work where team members have a relationship of equals. These system restrictions make it quite hard for researchers and system developers to build co-operative support environments.

An alternative view of the support environments is to consider them as being made of a number of different components which are loosely connected as opposed to tightly connected to the kernel in a layered approach. Here again we see, rightly, an analogy with the evolution of operating systems design from a centralized to confederation of services which may be distributed or not, but all joined by a common theme for inter-working through message passing as a paradigm for communication. This method of communication which has been used extensively by the distributed systems community is being used as a method of integration for some support environments, for example the Software Factory project [6].

In this type of architecture number of components either integrated or self-contained are interconnected through a communication subsystem. The communication subsystem can be built as part of the component or more generally as communication manager as advocated by the widely accepted model for distributed systems [3, 16]. This type of architecture has been successfully applied in the development of the COMBINE architecture [14]. The COMBINE architecture consists of Service Providers and Application Entities which communicate via a Collaboration Management Object. This distributed method of interaction uses message passing as the basic unit of communication but both the service providers and the application entities also have an associated communication object which, in addition to the message passing paradigm, has an abstraction of the underlying communications architecture. This abstraction includes both the communications protocols and network architecture. This may be necessary for communication with shop floor objects such as conveyors and robots.

Within the last decade, object-oriented techniques have been used to manage the complexity that is required to accommodate a concurrent engineering philosophy. Efforts by the US Air Force Information Integration for Concurrent Engineering project (IICE) have resulted in an object-oriented design method, IDEF4, [13] that can be applied to model projects that follow a concurrent engineering philosophy. Object-oriented techniques have successfully been applied in the field of engineering in both the areas of modeling and implementation; Dori documents the use of object-oriented analysis to model engineering drawings [5] while Chen et al. describes the use of object-oriented techniques to produce a concurrent engineering system for features-based design [4]. In addition to this, the STEP (Standard for the Exchange of Product model data) protocol [11] has emerged as a useful component in implementing a common information model. STEP is an attempt to provide a way of representing and exchanging product data so that advanced application programs can interpret product model data directly without human intervention. It contains a set of generic resources for describing product models in terms including product description, product structure, geometrical and topological representation and shape representation amongst others. It also describes more specific definitions and constraints between the generic resources by means of an Application Protocol.

Object Database Management Systems (ODBMS) have been shown to provide good facilities for the storage of engineering data. The use of ODBMSs in the Computer Aided Design (CAD) field has been extensively documented (see for example [8, 25]) and it is the authors' opinion that OBDMSs serve the best method of providing a concurrent engineering database.

In the authors' current proposals for a concurrent engineering support environment, the CONCERT environment [9], we have identified three fundamental components that we consider important in the production and application of concurrent engineering support environments. These components relate to the areas of providing access to a central database, providing facilities for co-operative working and providing support for distribution. Although the efforts described above have addressed some of the problems, there has been relatively little effort applied to addressing an holistic view of supporting concurrent engineering. The following section will present such a model and further sections will describe a prototype implementation.

III. Supporting a Concurrent Engineering Philosophy

To apply a concurrent engineering philosophy sometimes requires dramatic changes in working practices. Traditionally, product development has taken the following route1: initial concept, preliminary design and prototyping, production planning, product manufacture and concluding with product disposal. The production planners may not even see the initial design until perhaps 50% of the total cost of product development has been committed. Changes at this stage must be passed back up the chain to the relevant function incurring additional costs. A concurrent engineering philosophy attempts to run these functions in parallel to achieve dramatic reductions in lead times and total manufacturing time and to discover design errors at an earlier stage. To enable this, computer networks are utilized to effectively bring distributed team members together. Integrated project teams are
formed that use information technology to concurrently access a common database. To accommodate this way of thinking, new models of the way in which engineers interact within a computer integrated manufacturing (CIM) are required.

A. Service Requirements

The authors' proposals for a concurrent engineering support environment are shown in fig. 2. The CONCERT environment [9] identifies three core support services that are considered important in the production and application of concurrent engineering support environments. These highly co-operative components are the database support service, the computer supported co-operative working support service and the distribution support service. People and resources are typically distributed within an organization.

A technology enabled team can overcome distribution problems through the use of computer networks. The distribution support service provides a transparent interface to applications so that they can work seamlessly in a distributed computing environment. Design and development is inherently a co-operative process. Facilities to pro-actively support collaborative design and development can aid a concurrent

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

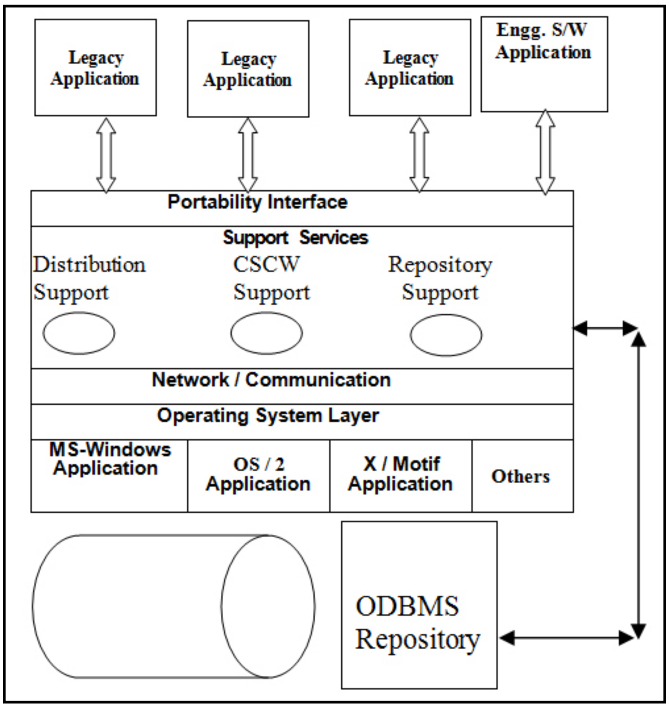

Fig. 2: CONCERT (Concurrent Engineering Support) Environment

development process. The CSCW (Computer Supported Co-operative Working) support service provides an interface for applications within a support environment so that they can use CSCW facilities to perform collaborative work. The Database is the single source 2 for persistent storage of all engineering data. This is not always limited to product designs but includes project tracking information, supplier and customer details and other information used throughout the life-cycle of the product. The database support service provides a consistent interface for applications within the support environment to store and retrieve persistent objects.

In modeling the requirements for these three services, we will build a framework on which engineering applications can be integrated to enable concurrent engineering. Each of the three support services is described in detail in the following section.

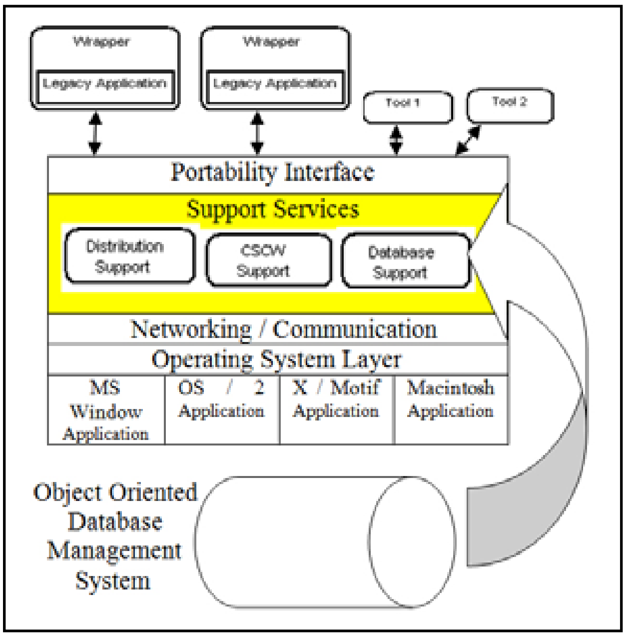

Fig. 2: The CONCERT (Concurrent Engineering Support Environment)

1. Database Support Service

The database support service provides a uniform method for applications to store and retrieve persistent objects. The client application can either issue a request for access to an object or to store a new object.

The client application may even request that a given object be destroyed in which case the database support service would be called upon to perform any necessary consistency and referential integrity checking. The database support service is also responsible for maintaining version control of objects within the database.3 Version control is needed when multiple versions of an object can exist. In an engineering environment, different versions of a design can exist during the development stage for example in the production of a product line based on one single common part with several product variations. An application of version control techniques in a concurrent engineering environment has been used to capture changes to a product model with the use of delta-files [10].

2. CSCW Support Service

The CSCW support service provides facilities to enable co-operative working between team members in a computer-based engineering environment. These facilities provide methods for both synchronous and asynchronous communication between users within the environment. It is also utilized by the environment itself to notify users of any system-wide messages such as the locking or freezing of components within the database (for example during system maintenance or design reviewing periods).

Typical requests from a client application are to obtain a list of users currently using the system or working on project, to send a message (or object) to a single user, to broadcast / multicast a message (or object) to a group / sub-group, to instantiate a multi-user conference and the sending of synchronous communication (see for example the UNIX talk and write commands) or asynchronous communication (e.g. electronic-mail or bulletin board). In the CONCERT environment we are applying the facilities of GroupKit [19] and utilising the CSCW models proposed by Greenberg and Marwood [7].

3. Distribution Service

Distributed systems have defined in terms of transparency and separation.

ANSA defines eight categories of transparency that are significant in the design of distributed systems: access transparency, location transparency, replication transparency, failure transparency, concurrency transparency, scaling transparency, performance transparency and migration transparency [2]. The distribution support service is responsible for the control of these aspects and the categories that are of importance to our design are access control, failure control and concurrency control.

Concurrency control is necessary in an environment where multiple users require simultaneous access to a given object. The ethos of concurrent engineering requires that team members have simultaneous access to objects throughout the product life-cycle. Therefore, in a concurrent engineering environment the goal is to achieve maximum concurrency. Andrews and Krieger state that a: "semantic concurrency model combined with notification locks is an effective mechanism that supports concurrent collaborative work environments." The concurrency control mechanism can use methods in the CSCW support service to notify users when conflicts arise [1].

Access control is used to determine whether objects within the database are being accessed by valid team members. The distribution support service will check that the user is authentic and that they have the appropriate privilege for the operation they wish to perform. Failure control is important in a multi-user environment. As we stated earlier, the complex data requirements for concurrent engineering mean that applications may require multiple views of the same object. If a software application fails whilst changing a crucial design object which has multiple views we either need to be able to notify all relevant parties currently using these views so that they can take appropriate action or attempt to transparently reconcile each affected user by rolling back a transaction to some designated point. A model of the three support services showing requests made of the service from tools within the environment is shown in fig. 3. The level of co-operation between the support services is very high. For example, when a request to store an object is made to the database support service, the database service will consult the distribution service to ensure access privileges are valid and check to see if the object is being accessed concurrently. If this is the case then it will use the CSCW service to notify all affected parties of any changes made to the working object.

IV. An Object-Oriented Model of the Database

Given the requirements for the three support services, we now present a model of the objects needed to provide a database to support concurrent engineering. The model (see fig. 4) contains the objects necessary for project administration and tracking, distribution support and CSCW support. For the purpose of project administration we introduce a project object. The project object contains details pertaining to the project such as the project title, start date, and a list of users who will participate in the project; the project team is composed of a project leader and team members (both of these are instances of the object type user.) In order to facilitate the tracking of personnel during a project we introduce the concept of a task. A task is performed by users in the team and a log of this is recorded in the database. This enables the project leader to monitor the progress of the team and ensure that the project schedule is adhered to. Any changes made to this information are also logged into the database to enable changes in the team or project leader to be tracked over time. This is especially important in the development of products which have a long life-cycle.

To support the concept of distribution, we note that each user will access the concurrent engineering system from a computer workstation. Each workstation has a corresponding network-address that uniquely identifies it4. This address is used in the support of distributed processing (for example when a user runs an application program on a remote machine and views the output of the program in a window on their local workstation - the system needs to know the physical address of the user's workstation so that it can display the output on the correct VDU) and in the support of CSCW (for the transmission of messages and communication between team members.) To enable user authentication within the system, each user has a set of access-rights. These access rights determine the privileges that a user has to perform certain tasks5. The privileges we must cater for include the creation, modification, destruction and reading of objects (configuration items). For computer supported co-operative working we identify two elemental objects: the conference and the message. Users can participate in real-time conferences where ideas are exchanged and results are immediate, or they can communicate via synchronous or asynchronous messages.

In order to perform a task, the user requires the use of a toll. The tool is an application program that performs a specific task at some stage of the product life-cycle. The tool will access configuration-items and produce new configuration items (reports are an example of this). The configuration items will have an associated access-protection which again is denoted by a set of privileges. This is necessary, for example, when design objects need to be frozen during a design reviewing stage. Note the concept of versions of configuration items and the fact that we can have versions of versions. When a tool requests a configuration item From the database, the correct version has to be delivered and a new version built when the item is returned.

A. Applying the Model to Support Manufacturing Data Management

The increasingly competitive global marketplace is leading UK manufacturing not only to utilize computer based technology but also to address systemic issues such as technology integration, operations streamlining, product quality and cost, information management and system development. A typical "technology enabled" product development environment could include Computer Aided Design (CAD), Computer Aided Process Planning (CAPP), Computer Aided Production Management (CAPM), and Computer Aided Manufacture (CAM) systems which are used in conjunction to design, to plan manufacturing and finally to manufacture a product.

A CAPM system such as Manufacturing Resource Planning (MRPII) is a large manufacturing database providing high level planning, capacity planning, and scheduling, purchasing and financial functions. MRPII systems hold master 'part', product structure and process route information that is used or created by CAD, CAPP or CAM systems. The component geometry created by CAD systems is input data for Finite Element Analysis (FEA) systems and tool path generation in CAM systems. Design for Manufacture (DFM), Design for Assembly (DFA) [27], Reliability and Maintainability [20] are tools used to support the product design process. CAPP systems provide tooling details for tool path generation in CAM systems and process routes for shop floor scheduling systems. The STEP protocol attempts to resolve some of these criticisms. STEP includes a data modeling language, Express, which is based on "entities" that are defined "in terms of data and behavior" [12] so STEP Express models exhibit object-oriented characteristics.

Fig. 5, shows a selection of the applications used during a typical product cycle for manufacturing and the inputs and outputs needed to fulfill their function. It can be seen that each of these applications produces its output based on a fundamental input: the product data model (in this case the product data model is based on the STEP protocol.) The product model data is accessed through a data access interface (the STEP Data Access Interface) to ensure that only relevant data is provided to the requesting application7. Note that in many cases it is also necessary to include resource information (i.e. machine, tooling, and lead time) as for example in tool path generation and process planning. In our model (fig. 4) we define the processes shown in fig. 5 as instances of the object type, tool.

The outputs from these processes are defined as configuration items of which the sub-classes, report and STEP product data model are examples. All of the objects visible in fig. 5, (databases, process outputs and the processes themselves) should ideally be stored in a common data repository. It is possible that the various

 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

outputs (tool path, process plan, BOM, etc.) could be derived from a product data model as and when they are needed. This product data model as and when they are needed. This however would be a very computation-intensive task even with today's fastest workstations. We therefore propose the use of a "rich repository" to store both product model data and the computed (process) outputs along with a mechanism to determine whether or not the computed output is up-to-date (based on changes being made to the product model). The repository could either present the requesting application with the data or compute it on-the-fly and update itself automatically. This type of caching mechanism will increase the storage requirements of the repository, however the speed increases that can be gained from this approach could justify the extra storage.

V. Implementation

Our initial prototype was implemented using a combination of C++ [21] and Tcl / Tk [18]. This proved a workable solution but we felt that it would be of greater value if we were able to scale-up the size and spread of users that could access the information stored in the repository and communicate with each other. The Internet already provides us with an infrastructure to do this and so became the focus for our current implementation.

We are currently developing a prototype World-Wide Web (WWW) interface to the CONCERT environment which we have called The Concurrent Engineering Web Workbench.

We have now chosen to use the Java [26] object oriented language and environment for development of the necessary classes and web pages since this has far greater flexibility than standard HTML and provides application developers with a method of creating truly dynamic objects capable of updating themselves in real-time. Java and the Hot Java WWW browser are currently available for the Solaris and Windows NT operating systems and it is currently being ported to Macintosh and Windows95 based platforms amongst others. It therefore gives us a good platform for prototyping and allows us to create truly portable applications. In this sense, we use Java / Hot Java to implement the portability interface within the environment.

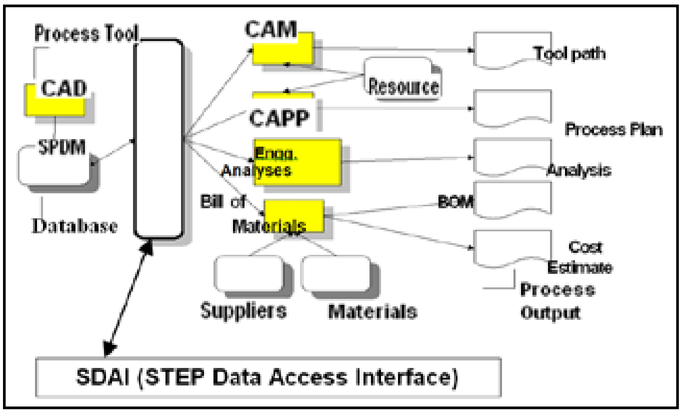

Fig. 5: An Engineering Environment

The Web server machine, a Sun Microsystems SPARC station 4 operating under Solaris 2.4, is running NCSA's HTTP web server software. The Java applications and applets are currently executed on Sun workstations operating under the Solaris operating system. The web server has been configured to restrict access to the information contained in the repository to a group of known usernames through the use of passwords. Increasing levels of access can be catered for in this way - the highest level of access is available to a user known as the CONCERT Administrator. This person has privileges to create and remove users and change user passwords and database contents. Some examples of the Java classes can be found in an Appendix at the end of this paper.

VI. Conclusions

For the repository we are using the freely available OBST object database management system [23]. We have not yet had the opportunity to evaluate the performance of this ODBMS or to determine the performance gains that can achieved by fully implementing the rich repository against the number of commercial packages that are available.

Both the distribution support service and CSCW support service utilize a Client-Server Programming technique which draws on our experiences with previous work undertaken during the development of the COMBINE architecture [14]. Future work will concentrate on the issues of CSCW and distribution control and in particular on the effect that enforcing concurrency control has on the working practices of users within a concurrent engineering environment. We have observed that there is a distinct lack of Application Objects (see Object Management Architecture Guide [17]) for manufacturing / engineering, e.g. tools such as DFA, DFM, FEA, CAM and CAD based on an underlying STEP product model. Engineering software vendors seem reluctant to develop generic components that can be used to build modular applications, instead preferring to provide complete systems. If software development activities were concentrated on developing open components that could be linked coherently into an environment (much like the current trend with compound documents) we feel that this would greatly facilitate concurrent engineering software integration. We stress that integration is not simply the interfacing of separate engineering tools but rather the bringing together of these tools so that they are perceived, and behave, as one.

This paper has described the authors' attempt to address a holistic view of supporting Concurrent engineering. The CONCERT environment [9] addresses three services that are key components in the production and application of concurrent engineering support environments: repository support, CSCW support and distribution support. We believe that by considering these components in combination, the synergy that will be achieved will be more effective than in the case when the components are applied without integration.

References

[1] Andrews, T.A., Krieger, D., 1993, Concurrency Control for Workgroups, Object Magazine, Vol. 3, No. 2, pp. 38-45, July- August.

[2] ANSA, 1987, ANSA Reference Manual Release 0.03 (Draft), Alvey Advanced Network ystems Architecture Project, 24 Hills Road, Cambridge, CB2 1JP, UK.

[3] Champine, G.A., 1991, MIT Project Athena: A Model for Distributed Campus Computing, Digital Press.

[4] Chen, C., Swift, F., Lee, S., Ege, R., Shen, Q., 1994, Development of a Feature-Based and Object-Oriented Concurrent Engineering System, Journal of Intelligent Manufacturing, Vol. 5, No. 1, pp. 23- 31.

[5] Dori, D., 1994, Automated Understanding of Engineering Drawings: An Object-Oriented Analysis, Journal of Object-Oriented Programming, Vol. 7, No. 5, September, pp. 35-43.

[6] Fernström, C., 1991, "An ESF Pilot Factory for Real-Time Software," in Software Engineering Environments, Long, F. (ed.), Ellis Horwood, Chichester, pp. 305-316.

[7] Greenberg, S., Marwood, D., 1994, Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface, Research Report 94/534/03, Department of Computer Science, University of Calgary, Alberta, Canada.

[8] Gupta, Rajiv, Gupta, Rajesh, Breuer, M., Cheng, W.H., Hardonag, I., 1991, 'The Development of a Framework for VLSI CAD', In Object-Oriented Databases With Applications To CASE, Networks and VLSI CAD, Rajiv Gupta & Ellis Horowitz (eds.), Prentice Hall, pp.237-260.

[9] Hanneghan, M., Colquhoun, G., Merabti, M., 1995, Concurrent Engineering Support Environments, poster session presented at 11th International Conference on Computer Aided Production Engineering, 20-21 September, IMechE Headquarters, London, UK.

[10] Hardwick, M., Downie, B.R., Kutcher, M., Spooner, D.L., 1995, Concurrent Engineering with Delta Files, IEEE Computer Graphics and Applications, Vol. 15, No. 1, pp. 62-68, January.

[11] International Standard 10303, 'Industrial Automation Systems - Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles', ISO Central Secretariat.

[12] International Standard 10303, 'Industrial Automation Systems - Exchange of Product Model Data - Part 11: Descriptive Methods: The Express Language Reference Model', ISO Central Secretariat.

[13] Mayer, R.J., Keen, A., Wells, M.S., 1992, Information Integration for Concurrent Engineering (IICE) IDEF4 Object-Oriented Design Method Report, Report Number AL-TR-1992-0056, Armstrong Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 45433-6573, USA.

[14] Merabti, M., Carew, M., 1994, "The COMBINE: A Distributed Support Environment for Concurrent Engineering", Proceedings of the 1994 Engineering Systems Design and Analysis Conference (ESDA), Tanik, M.M., Ertas, A., and Esat, I.I. (eds.), PD-vol. 64-5, pp. 89-96, London, July.

[15] Molina, A., Ellis, T.I.A., Marr Young, R.I., Bell, R., 1995, Modelling Manufacturing Capability to Support Concurrent Engineering, Concurrent Engineering: Research and Applications, Vol. 3, No. 1, March.

[16] The Common Object Request Broker: Architecture and Specification (CORBA), Report 91.12.1, The Object Management Group, Framingham, MA, USA.

[17] Object Management Architecture Guide, The Object Management Group, Framingham, MA, USA.

[18] Ousterhout, J.K., 1994, Tcl and the Tk Toolkit, Addison Wesley Publishing Company.

[19] Roseman, M., Greenberg, S., 1992, GroupKit: A Groupware Tool For Building Real-Time Conferencing Applications, in Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW' 92).

[20] Santos, J.L.T., Cardoso, J.B., 1993, Computer Aided Mechanical Engineering Design Environment for Concurrent Design Process, Simulation in Concurrent Engineering, Proceedings of the 1993 ITEC Workshop, 4-6 May, pp. 71-83.

[21] Stroustrup, B., 1991, The C++ Programming Language (second edition), Addison Wesley Publishing Company.

[22] Saxena, A.R., Bredt, T.H., 1975, "A Structured Specification of a Hierarchical Operating System," Proceedings of the ACM International Conference on Reliable Software, pp. 310-318.

[23] Schiefer, B., Theobald, D., Uhl, J., Casais, E., Freyberg, A., 1994, User's Guide - OBST Release 3-4, Document FZI.032.7, Forschungszentrum Informatik (FZI), Haid-und-Neu-Straße 10-14, D-76131 Karlsruhe, Germany, June.

[24] Sommerville, I., 1989, "Integrated Project Support Environments," Microprocessors and Microsystems, Vol. 13, No. 4, pp. 254-262.

[25] Spooner, D.L., 1991, 'Towards an Object-Oriented Data Model for a Mechanical CAD Database System', in On Object-Oriented Database Systems, K.R. Dittrich, U. Dayal, A.P. Buchmann (Eds.), Springer-Verlag, pp.189-205.

[26] Sun Microsystems, HotJava Home Page, URL= http://java.sun.com [Syan and Swift, 1994 a] Syan, C.S., Swift, K.G., 1994, 'Design for Manufacture', in Concurrent Engineering: Concepts, Implementation and Practice, Syan, C.S. and Menon, U. (eds.), Chapman & Hall.

[27] Syan, C.S., Swift, K.G., 1994, 'Design for Assembly', in Concurrent Engineering: Concepts, Implementation and Practice, Syan, C.S. and Menon, U. (eds.), Chapman & Hall.

[28] Trapp, G., Lawson, M., Burkett B., 1992, Workshop on Concurrent Engineering, Proceedings of the CAD and Engineering Workstations '92 and Business Graphics '92 Conference and Exposition, pp. 766- 792.

[29] Wilson, P., 1990, Modeling Languages Compared: EXPRESS, IDEF1x, NIAM and Shlaer-Mellor, Rensselaer Design Research Center, May.

[30] [Winner et al., 1988] Winner, K.E., Pennell, J.P., Bertrand, H.E., Slusarzuk, M.M.G., 1988, The Role of Concurrent Engineering in Weapons System Acquisition, Report R-338, Institute for Defense Analysis, Alexandria, VA, USA.

Amit Ahuja received his Bachelor of Science (Maths) degree from M.J.P.Rohilkhand University in 2003, Master of Computer Application (M.C.A) degree from U.P.Technical University, Lucknow in 2006. He completed Oracle certification (DBA, Oracle-9i) from SQL Star, Lodhi Road, New Delhi in 2006. He also completed 2 years of DNIIT course from NIIT Bareilly in 2005, same year he also completed Web Designing & hardware Course from Informatics Bareilly. He has a very good knowledge of different Technologies at national and international level. He has a 6 years corporate experience. He Started his career in 2006 as Data Analyst with Seth Services for 6 months, then he joined a Manufacturing Firm Unitech Machine Limited as Jr. DBA, He worked with that around one & half year. After that he joined Advance Ventilation Private Limited as Ass. Manager IT, His work with that company around two years. Has been to Sudan for on-site project as "TECHNICAL TRAINER" and has trained the employees of STC (SUDAN TAXATION CHAMBER, A UNIT OF MINISTRTY OF FINANCE-SUDAN). After that he joined Argon Global WLL (Doha, Qatar) as Sr. Solution Consultant.