CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Concurrency Control (CC) is an important concept for proper transactions on objects to avoid any loss of data or to ensure proper updation of data in the database, Christos (1982). It is an important aspect in controlling and coordinating data reading and data modification especially in long time transaction database. In spite of sophisticated algorithms available for proper locking, Kyung et al. (1981), Eswaran et al. (1976), and unlocking of objects during transactions, intelligent methods with knowledgebase information have been proposed and implemented in this research work.

This research work focuses on implementation of novel concurrency control methods using Artificial Neural Networks (ANN) with applications to Computer Aided Design / Computer Aided Manufacturing (CAD/CAM) database that involves more and unspecified duration of time to complete a transaction, Lee et al. (1983). Most of the transactions in the CAD/CAM database include drawing entities to form a complete manufacturing drawing, with processing and inventory details. In this research work, focus is given on drawing entities by many designers. The transaction follows schema used to develop the CAD/CAM database for drawing. The schema generally is based on Standards for the Exchange of Product model data (STEP) format. STEP is the colloquial term for the International organization for standardization (ISO) 10303 Industrial systems and integration–Product data representation and exchange.
Read Object ID (O₁) to be write locked

Input Object ID to training module of the OL

Store the final weights after Training

To wait state for lock

a) Training ANN for learning locks in concurrency control

Read Object ID (Oᵢ) and Transaction Type

Input Object ID to training module of the OL

Process with final weights

Grant Write Lock to Tᵢ, if required

Is final Weights == Write Lock

NO

YES

Only Read Lock is possible for Tᵢ

b) Testing / Implementing ANN for granting / denying locks, ‘L’. in concurrency control

Fig.1.1 Schematic flow of the intelligent concurrency control using artificial neural networks
In this research work, the geometric entity of STEP is used as the input data for implementing intelligent concurrency control in the CAD database. Existing locking strategy for concurrency control uses standard techniques like look up table. These techniques require more memory to store the details of the transactions as rows of information. In the proposed artificial neural network algorithms, the memory required for storing the transactions and other details is minimal which is based on topology of algorithm that indicate the size of the memory used for providing locking and unlocking. The Functional Back Propagation Algorithm (FUBPA), Locally Weighted Projection Regression (LWPR) and Fuzzy Logic have been implemented to improve the locking strategies in concurrency control of CAD/CAM database.

The overall training and testing of ANN for implementation of concurrency control of long time CAD/CAM transaction database is given in Figure 1.1. The details of various blocks in Figure 1.1 (a) are as follows:

1. Read object id (‘O’)
2. Input to the training modules. During the training process, initial random weights are used for the network to converge and to obtain final weights which are stored in a file.

The steps involved in Figure 1.1 (b) are as follows:

1. Read object id (‘O’)
2. Input to the testing modules.
3. Process with already stored final weights
4. Check if it is write lock. If yes, only reading is allowed else write is allowed on a transaction, ‘T’.
1.2 RELATED WORK

Transaction is a process of executing actions based on requirements from the user, Xiaoying et al. (1992). Transaction is mainly based on the concept of concurrency. Traditional concurrency control algorithms, Theo et al. (1993), Wanlin et al. (1991), have been built around three basic algorithms: two-phase locking, timestamps, Sibsankar et al. (2002), and optimistic concurrency control. Bernstein et al. (1981), Rajeev et al. (1993), Bhavani et al. (1993), identified two main characteristics of synchronization algorithms, the synchronization time and the synchronization method. In two-phase locking, conflicts are detected at execution time, as they occur, and blocking is chosen as the synchronization method. Aborting a transaction is used only in case of deadlock. For timestamp algorithms, conflicts are detected at execution time, and rollback is immediately used in case of conflict. In optimistic concurrency control, Peter et al. (2000), conflicts are only detected at commit time, and rollback is used in case of conflict. Other algorithms can be devised through different combinations of synchronization time and method.

Agrawal et al. (1987), Mihalis (1981), Abdel et al. (1989), Widya et al. (1988), Malcolm et al. (1995), Shi et al. (2002), studied two-phase locking, timestamp, and optimistic concurrency control algorithms under various conditions and concluded that each algorithm can be the preferred choice under different circumstances. Factors that affect the efficiency of an algorithm are resource utilization, number of active transactions, and transaction duration. It is clear that there is no single best algorithm for all circumstances.

A system could be built such that two simultaneous transactions use different concurrency control algorithms. They illustrate the approach by
combining two synchronization methods, namely blocking and rollback, and two synchronization times, namely execution time and commit time. They prove that an integrated algorithm consisting of pairs of each synchronization method and synchronization time is correct. Their technique imposes some constraints on the algorithms. The locking algorithm cannot update-in-place. It requires a workspace for deferred-update to prevent interference with other commit time synchronization transactions.

The main issue in CAD/CAM database systems is a common database shared by CAD and CAM. The data model used in a CAD/CAM database system should be capable of modeling data which are often in both formatted and unformatted forms, and in both static and dynamic status. Many CAD systems have been developed, Beeby (1982), Voelcker et al. (1978). The modeling powers are not capable enough to model a more complex data domain such as CAD/CAM.

The basic characteristic of design objects is that they are composed of a number of components, each in turn made of many sub components. These components are mostly shared, Afsannanesh et al. (1985), Won et al.(1984), Atwood (1985), Batory et al.(1985), Narayanaswamy et al. (1988), Narayanan et al.(1994), Markos et al. (1997), Chou et al. (1986), Katz et al. (1987), Rieu et al. (1986), Banerjee et al. (1987), Beech et al. (1988). There should be a mechanism to identify some chosen or default version, such as the most recent version of the object, the one approved by most people.

The development of artificial neural networks was first reported in the early forties. In this model, a neuron fires if the sum of its excitatory inputs exceeds its threshold. This happens, as long as it receives no inhibitory
input. Using this model, it is possible to construct a network that can compute any logical function. Rosenblatt (1961) found that the McCulloh et al. (1943) model was unbiological. In order to overcome the deficiencies in the McCulloh-Pitts model, he found out a new model, namely, the perceptron model, which could be utilized to learn and generalize. He investigated several mathematical models, which included competitive learning or self-organization, and forced learning which is somewhat similar to reinforcement learning.

In addition to the above two types of learning, the concept of supervised learning was developed and incorporated in the adaptive linear element (ADALINE) model. The training of the multilayer network was first explained by Werbos (1974) as back-propagation algorithm (BPA). His work did not become popular. Rumelhart et al. (1986) published the parallel processing, a two-volume collection of studies on a broad variety of neural network configurations. Through these books, the concept of back-propagation algorithm became popular for training a multi layer network. Lippmann (1987) briefed the concept of different algorithms in his tutorial paper, and he still made neural networks more popular.

Much work has been carried out, with respect to the number of hidden layers, the number of hidden nodes in the hidden layer, methods of representing the patterns, training the network with initial random weights at different ranges, types of error criteria used and selection of patterns. Even though the training procedure for the neural network is unique and problem-oriented, it is sufficient to have one hidden layer for most of the problems solved by supervised training. Sietsma et al. (1991), has analyzed the various training strategies with more than one hidden layer and finally claimed that one hidden layer was sufficient. Chester (1990) has claimed
better performance for the network with two hidden layers. The number of nodes in a hidden layer should be, neither too many, nor too few. Too many nodes in the hidden layer will result in the oscillation of the mean squared error (MSE) around a particular value without any convergence; or sometimes the network converges to one of the local minima. Similarly, too few a number of nodes in the hidden layer will sometimes be just suitable, only to learn the training patterns, but generalization of the network is not possible. Therefore, it is necessary that there should be a way to find out the optimum number of nodes in a hidden layer. Hirose et al. (1991) have adapted a different approach, by using an algorithm based on MSE to estimate the same. To overcome the difficulty of analyzing the number of nodes in the hidden layer, Weymaere et al. (1991) have used Gaussian function in the hidden nodes and sigmoid function in the output nodes. Fujita has analyzed hidden unit function.

In most of the supervised training methods, the patterns are presented in a pre-determined sequence in a cycle. Normally, the order of presentation of the patterns is maintained in all the cycles. Ridgway (1962) found in his thesis that cyclic presentation of patterns could lead to cyclic adaptation. These cycles would cause the weights of the entire network to cycle, by preventing convergence. Various error criteria have been tried by Zakai (1964), and by Walach et al. (1984), for better convergence of the network. Quantization of the weights and training BPA has been analyzed by Shoemaker et al. (1991). Analysis of BPA with respect to mean weight behavior was done by Bershad et al. (1993). In reality, most of the patterns are not linearly separable. Non-linear classifiers are used for pattern classification, in order to achieve good reparability. The multiplayer network is a non-linear classifier, since it uses hidden layer.
Neural networks are used for classify patterns by learning from samples. Different neural networks paradigms employ different learning rules. In some way, all these paradigms determine different pattern statistics from a set of training samples. Then, the network classifies new patterns on the basis of these statistics. The BPA uses steepest-descent method, which is slow and linear in convergences.

1.3 THE CAD/CAM DATABASE MANAGEMENT SYSTEM

Feature-based design is increasingly adopted in the development of CAD/CAM systems and in the integration of software systems, Yehuda (1985). Modeling of features using a formal language has become an essential task. This is particularly true, as STEP is emerging as an international standard and has already achieved as substantial set of technically sound data definitions covering the life cycle of products. The International Organization for Standardization STEP specifications aim to provide an effective means by which product information can be shared and exchanged between applications and enterprises. EXPRESS is a modeling language defined in STEP and used to present STEP specifications. It is also being used to describe product data by major Computer Integrated Manufacturing (CIM) projects, Richard (1983).

An EXPRESS model can be implemented on a database repository. Part 42 is a component of STEP that describes geometric and representation schema in EXPRESS. It has been prepared by the subcommittee 4 (industrial data and global manufacturing programming languages) of IS0 Technical Committee 184 (Industrial Automation Systems and Integration). This specification provides the integrated resources used for geometric and topological representations of shape and geometric form of a product model.
Fig. 1.2 General data structure in CAD database
Fig. 1.3 Different data structure
Part 42 is made up of three schemas, namely, a geometry schema, a topology schema and a geometric model schema.

The general data structure in a CAD database is shown in Figure 1.2, Qiang et al. (1997). It contains the data in files, the type of method of processing the data, a graphics display facility with input / output device configurations. A set of data items that are related to each other by a set of relations: The various arrangement of data in Tree, Linked list with Vertex based, edge based, face based data structure are given in Figure 1.3 (a-d). The advantages of data structure are: eliminate duplication, standard for maintaining data, security of the data, consistency of data and harmonize the conflicts at the time of processing the data.

The Requirements for CAD/CAM Database (DB) are Heterogeneous applications, Abraham et al.(1980), Dynamic modification, Tentative, iterative, evolutionary nature of the design process, Versions, levels of detail (LOD), Concurrent and multiple users, temporary DB, different designs require different design sequences, Easy access: should not require extensive knowledge on DB to extract the data.

1.3.1 The Database Management System (DBMS)

Database Management System (DBMS) is a layer of software between the physical DB and the users. Records in the DBMS provide an excellent tool for processing information that fits a certain pattern. Other kinds of information do not fit into record structures, William (1979). The record-based conventional data models have been criticized as not powerful enough to model a complex data domain of CAD/CAM, Ying (1988). A distributed database system involves lot of records and is one in which the database is spread among several sites and application programs and move from site to
site to access and update the data in the record they need, Daniel et al.(1978).

1.3.2 Communication Module

Communication Module can be two types. They are CIM (Computer Integrated Manufacturing) and IGES (Initial Graphics Exchange Specifications), STEP (Standard for the Exchange of Product model data)

1.3.3 EXPRESS: Overview

STEP is targeted at the exchange of data describing a product between Computer Aided X (X = CAD, CAM, etc.) systems, and also long term data retention of such data. Specifically, the exchangeable product data is defined in the application protocols. EXPRESS (Table 1.1) is the language used within STEP to formally define the semantics of the data, and the 20 series of Parts specify the standard data exchange mechanisms. Express defines a) Textual conceptual schema language, b) Object-Oriented flavor, c) STEP integrated resources, d) Human-readable and computer- process able and it is not a traditional programming language. The various EXPRESS elements are a) Main Elements (Schema, Type, Entity, and Rule) and other element (Constants, Functions and Procedures and Executable statements)

1.3.4 Database Schema

The key components of the benchmark database is the set of geometric objects, Gray (1978), Rakesh et al. (2005), Ari (2003), Hector (1983), Thomas (1980), Jayasri et al. (2007). Each object corresponds to a design primitive which in turn consists of other primitive objects that are selected from Part 42. Each geometric object is identified by an identifier (ID), a build-date and an associated graph of primitive objects.
<table>
<thead>
<tr>
<th>Table 1.1 Lines &amp; Points: Schema</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Express</th>
<th>Express - G</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHEMA example1;</td>
<td>REAL</td>
</tr>
<tr>
<td>ENTITY point;</td>
<td>point</td>
</tr>
<tr>
<td>x : REAL;</td>
<td></td>
</tr>
<tr>
<td>y : REAL;</td>
<td></td>
</tr>
<tr>
<td>END_ENTITY;</td>
<td></td>
</tr>
<tr>
<td>ENTITY line;</td>
<td></td>
</tr>
<tr>
<td>End1 : point;</td>
<td></td>
</tr>
<tr>
<td>End2 : point;</td>
<td></td>
</tr>
<tr>
<td>END_ENTITY;</td>
<td></td>
</tr>
<tr>
<td>END_SCHEMA;</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](image)
All the geometric classes were created using a group model storage manager. This allocates space for objects on a collection of fixed size pages, where each page contains objects of the same type. Applications which reference large numbers of related objects at a time would perform well using this storage model.

The aggregate classes used include unordered directories and lists. The unordered dictionaries use an efficient linear hashing algorithm which provides good access performance even at a high percentage of loading. This is utilized in the implementation of the queries. Lists, implemented as chains of buckets are mainly used for the traversals.

Real Time Database Systems (RTDB) and Active Database Systems have been discussed and implemented respectively to support non-traditional applications, Barghouti et al. (1991), Dewayne et al. (1991), Don (1986). Lot of work concentrate on the integration of active and real-time database systems and is very much used in Computer Integrated Manufacturing (CIM).

Transaction in CAD/CAM involves longer period of time when compared to other transaction applications, Costas et al. (1998), Eliot (1986), Gail (1988), Vinay et al. (1994). The CC concept should be developed suited to lower time transaction. New problems are evolved in concurrency control (CC) of real-time database systems, Skarra (1988). Conventional CC protocols are more concerned about the serializability. The situation is more complicated when real-time databases are integrated. RTDBs must not only respond to the external transactions but also the internal triggered events.
generated by the system itself. Due to the triggering structure in RTDBs, a
dynamic CC algorithm is needed suiting to different types of transactions.

The design and manufacture life cycle of a product involves many
distinct engineering disciplines, each with its own specialized computer
systems. These systems have massive data repositories handled by data
managers that cannot communicate with each other. As a result, there are
significant problems for data management and for the efficient coordination
of hundreds of complex systems. CAD/CAM data provides a basis for
integration and coordination of design and manufacturing. Data consists of
all types of product descriptions including requirements, drawings, parts
hierarchies, geometries, analyses, and manufacturing processes, and
administrative data for controlling, monitoring and planning. The same data
is frequently used in different forms by many systems.

A DBMS is used to integrate current and future CAD/CAM systems by
providing methods to store, manipulate, and manage all CAD/CAM data. The
concept of CAD/CAM DBMS (CCDBMS) provide means to solve such
problems as representing engineering data semantics, version control, data
exchange, distributed processing over heterogeneous databases, and
suitable interfaces, Klaus et al.(1988). It is continuously being designed and
developed to integrate all CAD/CAM systems. Each system will continue to
operate autonomously, but some measure of global control and access will
be imposed. The components of the CCDBMS architecture fall into three
functional groups.

1. First, a set of user interface components will provide uniform access to
all CCDBMS facilities, including those of the individual CAD/CAM
2. Second, a Global Data Manager will provide distributed processing for CCDBMS requests that require access to more than one system within the CCDBMS.

3. Third, a new DBMS will provide a global view of all data. Needed to support queries against the whole CCDBMS, distributed processing, and version control. The global view will include an abstract or extract of all data in the CCDBMS as well as a global dictionary and directory. A centralized database for all CAD/CAM data would make the integration of current and future CAD/CAM systems economically infeasible. The problem of determining one or more standard data representations for all CAD/CAM data is currently intractable due to mathematical problems with translation between representations and the real need for specialized representations.

An ideal design environment should be both collaborative and intelligent. Many of today’s design projects have to be undertaken by groups of designers distributed across computer networks. The designers differ from each other in expertise, problem understanding, experience, or even design tools. As they are loosely linked by network connections, their design results are closely coupled in logic and should always be consistent in terms of design rationale. To support the designers’ collaborative design actions, a software platform must be established. In either software protocols or hardware facilities, the networks with which designers are connected are not dedicated to design.

Design is knowledge intensive in nature. A platform to support distributed design should have two categories of functionalities. One is the capability of cooperative work support, such as group awareness, multi-user interfaces, concurrency control, communication and coordination within the
group. The other concerns knowledge management and intelligence support. For analyzing, designing, and implementing software systems, agent technology has brought us a lot of successful examples in recent years and is contributing to diverse domains.

Agent technology is a facility to update information, replying to queries in an effective way using a specialized module. For the collaboration behaviors of the designers in a distributed collaborative design process, are similar to those of the agents in a distributed agent system. It is possible to make a collaborative design system work more efficiently by developing it in accordance with the distributed agent architecture.

Geometric information is generated and displayed on workstations. The CCDBMS will store this information centrally and provide access to parts information via geometric conditions. In considering a design change, all parts close to the point of change can be found and their properties accessed. The DBMS works with approximations to actual shapes.

1.4 CAD

A practical approach to advanced CAD environments should incorporate the advantages of both database (DB) and Artificial Intelligence (AI) techniques. The approach that is described here is centered around and is called Knowledge Base Management System (KBMS) that integrates AI and DB techniques in an effective way.

1.5 ANN

Artificial Neural Networks are computing methods which does not require any mapping equations. The outputs are model independent of
inputs. The various entities of an object in CAD can be used to train ANN. The trained information can be used for manipulation of the entities. Knowledge Base System is incorporated in Fork used in a two wheeler front structure, Bolted connection and Bearing. Some of the following are

1. Allotting dimensions for features of Fork used in a two wheeler front structure, Bolted connection and Bearing.
2. Linking the various entities resulting into a standard shape.
3. Changing one feature results in changing the dimensions of other features of bolt based on whichever features are associated.
4. Facilities like zooming, rotating, maintaining similar shape for the Fork used in a two wheeler front structure, Bolted connection and Bearing based on the information stored in the file.
5. Storing the Fork used in a two wheeler front structure, Bolted connection and Bearing information in a standard format IGES.

1.6 TRANSACTION AND CONCURRENCY CONTROL

Transaction is series of actions, carried out by user or application, which accesses or changes contents of database. It is a logical unit of work on the database. It transforms database from one consistent state to another, although consistency may be violated during transaction, Salem et al. (1987) and Klahold et al. (1986). Concurrency is the process of managing simultaneous operations on the database without having them interfere with one another. It prevents interference when two or more users are accessing database simultaneously and one is updating data. Although two transactions may be correct in themselves, interleaving of operations may produce an incorrect result. Three potential problems caused by concurrency are lost update, uncommitted dependency and inconsistent analysis. Executions of transactions guaranteed to ensure consistency is identified by the concept of serializability with those schedules of read / write. Serial
schedule is where operations of each transaction are executed consecutively without any interleaved operations from other transactions. Non serial schedule operations are from a set of concurrent transactions which are interleaved techniques used for concurrency control are locking and time stamping. Both are conservative approaches when delay transactions in case they conflict with other transactions. Optimistic methods assume conflict is rare and only check for conflicts at commit.

Transaction uses locks to deny access to other transactions and so prevent incorrect updates. A transaction must claim a shared (read) or exclusive (write) lock on a data item before read or write. Lock prevents another transaction from modifying item or even reading it, in the case of a write lock. Rules of locking are, if transaction has shared lock on item it can read but not update item. It transaction has exclusive lock on item, can both read and update item, Reads cannot conflict, so more than one transaction can hold shared locks simultaneously on same item, Exclusive lock gives transaction exclusive access to that item.

1.7 TRANSACTION REQUIREMENTS IN CAD DATABASE

Design and development of a product is the first and foremost step in a manufacturing industry. This process is recurrent and repetitive until it reaches a final approved design and development stage. Design and development activity involves defining and describing the product, drawing the product in the computer using computer aided drafting(CAD) software making modifications in the drawing, proving suitable material combinations for the product, defining various sizes for the product, providing safety factor provision based on the end application, satisfying customer requirements. The entire process will be generally interactive between a designer and the customer with one to one direct contact, or interactive discussions between
designers at various locations, or independent design decisions by various designers who are located at different places and are accessing the same database which is centralized and sometimes distributed. When many designers are involved in designing an object in the database a major problem of concurrency as well as version of the product developed occurs.

   Majority of transaction will be done with long time gap. In the existing commercial database, all the equations and procedures are already coded with all the rules required which may change from time to time, Alexandtos et al. (1989), Bancilhon et al. (1985).Whenever a user is accessing the data in the form of read / write, the transaction takes place quickly.

**Transaction with longer time period:** In case of computer design process even though design formulae have been encoded at the time of software development, the users will have their choice of choosing their design requirements. They will do design based on their previous experiences and on the requirements of customers, based on the availability of machine capabilities in the workshop. The activity of deciding the optimum design will take long time for completing the transaction. Suppose, a similar design is done by another designer very quickly, which may be based on some criteria, the question is whether the second person can be permitted to update the database or should he have to wait for the first person to the design process. What type of transaction concept that has to be adopted is based on many factors which may not be readily fixed.

**Controlling transaction with users choice:** Most of the CAD transactions are based on interaction among many users. At least two users would transact the knowledge, discuss and come to a conclusion whether such design can be finalized. In such case, the final commit in the database should be possible. This should not become impossible because of basic transaction rules.
Cooperating the views of the designers in synchronization Many users working on shared objects cannot be serialized. The shared objects shall pass among them when modifying two parts of the same object parallelly to create new designs of the object. In this case, complete serializability is not possible.

1.8 EARLIER APPROACHES TO GROUP TRANSACTIONS

Developers of large project often work in small teams. A policy is to define the kinds of interactions allowed among members of the same team as opposed to interactions between teams. A group paradigm to deal with consistency of replicated data in an unreliable distributed system. One can hierarchically divide the problem of achieving serializability into two simpler ones: (1) a local policy that ensures a total ordering of all transactions within a group; and (2) a global policy that ensures correct serialization of all groups. Groups, like nested transactions, are an aggregation of a set of transactions, Fernandez et al. (1989). There are differences between groups and nested transactions. A nested transaction is designed 'a priori’ in a structured manner as a single entity that may invoke sub transactions, which may themselves invoke other sub transactions. Groups do not have any a priori ‘assigned structure’ and no predetermined precedence ordering imposed on the execution of transactions within a group. Another difference is that the same concurrency control policy is used to ensure synchronization among nested transactions at the root level and within each nested transaction. The group paradigm was introduced to model inter-site consistency in a distributed database system. It can be used to model teams of developers, where each team is modeled as a group with a local concurrency control policy that supports synergistic cooperation.
### 1.8.1 CAD Transactions in Groups

The group-oriented model does not use long-lived locks on objects in the public database. The conversational transactions model sets long-lived locks on objects that checked out from the public database until they are checked back into the public database. The group-oriented model categorizes transactions into Group Transactions (GT) and User Transactions (UT). Any UT is a sub transaction of a GT. The model also provides primitives to define groups of users with the intention of assigning each GT a user group. Each user group develops a part of the project in a group database. A GT reserves objects from the public database into the group database of the user group it was assigned. Within a group database, individual designers create their own user database, and they invoke UTs to reserve objects from the group database to their user database.

In the group-oriented model, user groups are isolated from each other. One user group cannot see the work of another user group until the work is deposited in the public database. Group transactions are thus serializable. Within a group transaction, several user transactions can run concurrently. In case of non-serializable schedule, each user in a group should have a common understanding of the way in which they are going to work with objects. The basic mechanism provided for relaxing serializability is a version concept that allows parallel development (branching) and notification. Versions are derived, deleted, and modified explicitly by a designer only after being locked in any one of a range of lock modes.

### 1.8.2 Version Control

A number of issues often are thrown together under the heading “version control” or “configuration management”. This function is defined as: “The systematic approach to identifying, controlling, and accounting for the
status of the parts and assemblies required in a product and/or design from the point of its initial definition throughout its entire life.

There are three main components of version control, each of which imposes a set of requirements on a CAD/CAM DBMS:

1. Schema definition: recording status and descriptive information about products, numerically controlled tool programs, schedules, etc.;
2. Access control restricting access of various users to data at various stages in the life of a product; and
3. Change control monitoring updates, recording change requests and approvals, and changing data availability as a product moves from one design or manufacturing stage to another.

An effective CAD/CAM access control system should support four functions:

1. value based access privileges,
2. definition of access privileges at the attribute (function) level,
3. linking of some access privileges to particular predefined transactions, and
4. Flexibility granting and revoking of privileges.

Value based access privileges work together with the schema definition since database values are used to determine user access to data at a particular time. Attribute level access privileges are essential to provide the degree of control required in the CAD/CAM environment. Supporting attribute level access privileges becomes complicated when a schema allows entity valued attributes needed to capture complicated relationships in CAD/CAM. The CCDBMS can control the kinds of updates allowed by requiring the use of predefined transactions to update sensitive data.
1.8.3 Access Control and Concurrency Control

To maintain design data consistency and prevent unauthorized access, data retrieval, modification, and delete operations must be controlled. Without preventing out accidental or malicious update operations, one may have corrupted data as a result. The Design Data Management System (DDMS) provides two levels of control to enforce design data’s consistency and security. The first level, the access control level, is to provide a static protection scheme for design objects by allowing the user to wait access privileges. The second level, the concurrency control level, is to monitor design processes to synchronize concurrent accesses to design objects, and make sure that no concurrent editing is done to design objects.

DDMS regulates accessing requests for design data automatically according to the user specified access privilege information and runtime conditions. For each design object, four levels of access privileges can be set up based on the user classification. The working group is provided to allow object creator to further distinguish object developers from general users, and assign them higher working privilege. A user group defines all the possible users that may refer to the design object in their design. The public group consists of all employees of the company who can reach the design data.

For each access level, DDMS supports three types of access privileges, the read access, the write access, and the Delete access. The default setting is to allow all three privileges for the owner of an object, read and write access for the working group, and read access for the user group and public. These three privileges have different meaning when they are applied to different object types.
General system architecture, for performance reasons, of an engineering design system is the one based on the client / server computing environment. In such an environment, components or design modules reside in a central server (the shared database), which is connected to powerful workstations (private databases) through a high bandwidth local area network Katz et al. (1984) and Lorie et al. (1983). In many recent proposals, this model has been extended so that between the server and private databases an arbitrary number of databases may be established. Databases are associated with nested transactions and also in group transactions, Kim et al. (1988)

The first requirement is handled by introducing two kinds of references: static and dynamic (or generic) references. A static reference is to a specific version of an object, while in dynamic reference the exact version is left unspecified. Such dynamic references refer to a generic object that somehow represents the set of versions.

Concurrency control is provided to prevent corruption of design data from by parallel update to the design data. This is enforced in through two levels. The first level goes through checkout / check in mechanism. The checkout / check in mechanism enforces a resident lock on design objects. This prevents two different users to modify the same copy of design data. The second level is done at runtime by inter process locking. DDMS refers it as conversational lock. This precludes the possibility that one user modifies the same design data in different processes. The persistent lock may or may not be transparent to the user, since check out can be issued by user explicitly or done implicitly by DDMS when user issues command to edit a non-checkout cell view file. On the other hand, the conversation lock is
totally transparent to the user except that the user may receive error messages due to failing to pass DDMS runtime locking.

Conversational lock is a transaction lock issued by DDMS to make sure that only one process writes to the shared data file at one time. The lock is automatically released when the transaction is completed. A transaction is a series of DDMS operations which must be treated as a whole. A transaction is completed only if every atomic operation succeeds. If any intermediate operation fails, the design content returns to the state before the transaction are started.

Many large multiuser software systems, such as Software Development Environments (SDE), generate and manipulate large amounts of data. Traditionally, users of such systems manage the data they generate either manually or by the use of special-purpose tools. Programmers working on a large-scale software project use system configuration management tools to manage the configurations and versions of the programs they are developing, Feldman (1979) and Tichy (1985).

Design environments need to store the objects they manipulate in a database and have it managed by a DBMS for several reasons, Bernstein (1987), Dittrich et al. (1987), and Nestor (1986). Since there are numerous commercial DBMSs available, several projects have tried to use them in advanced applications. Researchers discovered quite rapidly that even the most sophisticated DBMSs are inadequate for advanced applications, Korth et al. (1986). One of the shortcomings of traditional general-purpose DBMSs is their inability to provide flexible concurrency control mechanisms. There have been a few comprehensive discussions and surveys of traditional concurrency control mechanisms, Kohler (1981).
1.9 LOCKING MECHANISM IN CONCURRENCY CONTROL

In the absence of information about how and when the data items are accessed, Two Phase Locking (2PL) is both necessary and sufficient to ensure serializability by locking, Yannakakis (1982). In advanced applications, it is often the case that the DBMS has prior knowledge about the order of access of data items. The DBMS can use this information to ensure serializability by using locking protocols that are not 2PL.

One of the more common variations on locking is the addition of intention lock modes to support hierarchical locking. Hierarchical locking means that an explicit lock on an object implicitly locks the descendants of that object in the hierarchy. This substantially reduces the number of locks that must be held. Intention modes permit higher degrees of concurrency for transactions that only access a small number of items in a hierarchy. For correctness, there are specific protocols that must be followed to explicitly acquire or release a lock. Locks must be acquired from the root of the hierarchy down to the requested object. The methods of any data server that uses hierarchical locking must follow the locking protocols. As a convenience, some implementations of hierarchical locking choose to implement the locking protocols directly in the lock manager, rather than the methods. This is convenient and perhaps safer since the lock manager need not depend on the method to follow the protocol. The lock manager needs to know the details of the lock hierarchy.

Lock manager would proceed to acquire locks on ancestors of object. The particular set of ancestors is determined by the object and the lock mode. Lock manager should be independent of the concurrency control algorithm and of the semantics of objects. Observer introduces the notion of notify locks as a way of permitting groups of transactions to share
intermediate states of an object. In Observer, there are various forms of notification. A transaction may be notified that an object has been written, a lock request denied, or a lock acquired.

1.9.1 Lock Modes

The model supports five lock modes on a version of an object:
(1) read-only, which makes a version available only for reading.
(2) read / derive, which allows multiple users to either read the same version or derive a new version from it;
(3) shared derivation, which allows the owner to both read the version and derive a new version from it, while allowing parallel reads of the same version and derivation of different new versions by other users;
(4) exclusive derivation, which allows the owner of the lock to read a version of an object and derive a new version, and allows only parallel reads of the original version; and
(5) exclusive lock, which allows the owner to read, modify and derive a version, and allows no parallel operations on the locked version.

Using these lock modes, several designers can cooperate on developing the same design object. The exclusive lock modes allow for isolation of development efforts (as in traditional transactions), if that is what is needed. To guarantee consistency of the database, designers are only allowed to access objects as part of a transaction. Each transaction in the group-oriented model is two-phase, consisting of an acquire phase and a release phase. Locks can only be strengthened in the acquire phase, and weakened in the release phase. If a user requests a lock on a particular object and the object is already locked with an incompatible lock, the request is rejected and the initiator of the requesting transaction is informed of the rejection. This avoids the problem of deadlock, which is caused by blocking transactions that request unavailable resources. The initiator of the
transaction is notified later when the object he requested becomes available for locking. In addition to this flexible locking mechanism, the model provides a read operation that breaks any lock by allowing a user to read any version, knowing that it might be about to be changed. This operation provides the designer (more often a manager of a design effort) the ability to observe the progress of development of a design object, without affecting the designers doing the development. Designers in the same team should be able to 1) Synchronize their designs by being aware of the new versions as they are being produced, and 2) prevent premature disclosure of a design, produced by one of them and shared among them, to other designers.

One of the problems of locking mechanisms is the potential for deadlock. Deadlock occurs when two or more transactions are mutually waiting for each other’s resources. This problem can be solved by assigning each transaction a unique number, called a timestamp, which increases monotonically. The sequence is a function of the time of the day. Using time stamping mechanism, a concurrency control mechanism can totally order requests from transactions according to the transactions’ timestamps, Rosenkrantz et al. (1978).

The time stamp ordering mechanism assumes that only one version of a data item exists. Consequently, only one transaction can access a data item at a time. This restriction can be relaxed by allowing multiple transactions to read and write different versions of the same data item as long as each transaction sees a consistent set of versions for all the data items it accesses. This is the basic idea of the first multi version timestamp ordering scheme introduced by Reed (1978). In many applications, locking has been found to constrain concurrency and to add an unnecessary overhead. The locking approach has many disadvantages, Kung et al.
To avoid these disadvantages, the concept of optimistic concurrency control is presented.

It is sometimes desirable, to be able to access a set of data items as a single unit. A multiple granularity concurrency control protocol that aims to minimize the number of locks has been discussed, Herrmann et al. (1990). To determine whether to grant a lock on a node to a transaction, the transaction manager would have to follow the path from the root to the node to find out if any other transaction has explicitly locked any of the ancestors of the node. This is clearly inefficient. To solve this problem, a third kind of lock mode called an intention lock was introduced. The Orion object-oriented database system provides a concurrency control mechanism based on the multi granularity mechanism.

The idea of nested spheres of control, which is the origin of the nested transactions concept, was first introduced by Davies (1973) and expanded by Bjork (1973). A comprehensive solution is presented to the problem of composing transactions by formulating the concept of nested transactions. A nested transaction is a composition of a set of sub transactions; each sub transaction can itself be a nested transaction.

1.10 SCOPE OF THE PRESENT WORK

This thesis has focused in implementing Artificial Neural Network (ANN) for concurrency control. Different supervised neural network have been used for implementing concurrency control in CAD database.

A systematic approach has been developed to train the network; with different architectures the major algorithms proposed for locking in CC are as follows:
This thesis focuses in implementing intelligence into the concurrency control aspects for CAD database operation. A neural network approach is attempted and implemented into concurrency control. This thesis implements

1. Locking and unlocking of objects using Functional Back Propagation Algorithm (FUBPA)
2. Locally Weighted Projection Regression (LWPR) used for locking and unlocking of objects
3. Locking and unlocking of objects by Fuzzy Logic.

The locking type, ‘L’, locking status of the objects ('O_i') are implemented using ANN. By using ANN memory conservation is achieved when compared to existing table look up locking and unlocking method.

1.11 ORGANIZATION OF THESIS

This thesis has been organized in the following direction. Chapter 2 presents the sequence of entity formation in the drawing Fork used in a two wheeler front structure, Bolted connection and Bearing with approximate schema have been presented. The sequence in which locking and unlocking of objects ('O_i') that would take place in each drawing with main transaction and sub transactions are presented. Chapter 3 presents the different types of locks used for a transaction. The sequence in which FUBPA modules used for learning, object, transaction, lock type and the method of finding out if lock has been assigned for a transaction through testing of FUBPA are given. Performance metrics are presented. In actual practice that is during online implementation, if the time gap between transactions are longer, then the time taken for training will not affect locking and unlocking. If the time between transaction is less, then FUBPA is not suitable.
In chapter 4, the implementations of Locally Weighted Projection Regression (LWPR) for locking of transactions of objects in Fork used in a two wheeler front structure, Bolted connection, Bearing. LWPR consumes comparatively less amount of time in obtaining the final weights when compared to the time taken by FUBPA. This algorithm is very much suitable when time between transactions is very less.

Chapter 5 presents implementation of Fuzzy logic and unlocking of transactions.

Chapter 6 This chapter presents the comparison of the performances of FUBPA, LWPR and Fuzzy Logic for locking and unlocking of transactions of drawings objects in Fork, Bolted connection, and Bearing used in a two wheeler front structure.

Chapter 7 presents the conclusions and scope of the future work.

1.12 SUMMARY

This chapter has provided detailed information of concurrency control with locking and unlocking of long time transactions in CAD/CAM database. A detailed review of literature has supplemented with long time transactions and different locking methods used. Chapter 2 explains the experimental data generated using CAD software. The concurrency control with locking sequences used for transactions of objects in fork, bolted connection and bearing have been discussed in detail.