SUMMARY AND CONCLUSIONS

In this study, we have examined a data flow graph based approach for distributed systems, and presented two concurrency control algorithms based on data flow graphs. One is an approach for concurrency control for a distributed database system, which is an alternative to distributed two-phase (static) locking. The other is, an approach for concurrency control for a replicated database system, which is an alternative to voting based approaches. Here, an attempt is made to summarize, what we have presented in this dissertation, and report some of the positive remarks of the two algorithms. At the end, we have suggested directions for further research.

6.1 Summary of the study

In first chapter, after a brief introduction, important aspects pertaining to the problem of deadlocks in a distributed database system has been discussed. Various kinds of failures in a distributed database system have been analyzed later. Further, the motivation for data replication, its advantages, problems and issues involved in replicating data are shown here. The objectives of study presented at the end of the chapter.

In chapter two, after explaining the transaction model, we considered the correctness criteria for a distributed and replicated database environment. Subsequent to that, the concept of time-stamps and transaction numbers is explained. At the end of the chapter, the
concurrency control algorithm, for centralized database systems based on data flow graphs is presented.

In chapter three, the local access graph has been defined and explained with example. The algorithm for construction of a local access graphs has been presented. We explained the process of synchronization through example. Next, we have analyzed the communication requirement of the algorithm. Lastly, the simulation results have been presented on use of the graphs. The implementation issues have also been described at the end.

In chapter four, the concept of data flow graphs and transaction data flow graphs is presented with suitable example. Next, the algorithm for construction of transaction data flow graph for a transaction along with its application is presented. The behavior of the algorithm in the event of site and media failures is explained. Subsequent to that, a comparison with the voting based approaches have been presented through use of a simulation study. At the end, the implementation issues have also been considered.

In the algorithms proposed in chapter three and chapter four, the two-phase commitment protocol is assumed for committing the transactions. The two-phase commit protocol is described in the chapter five. We have also discussed the possibility of integration of the stages of commit protocols with the synchronization stages of proposed algorithms.

In this section, we have summarized some of the positive features of the algorithms presented in this dissertation. In addition to this, we also contrast the proposed algorithms with the existing algorithms with their counter parts. Here, we list the positive points of concurrency control algorithm presented for distributed databases.
1. In this algorithm deadlocks do not occur. The transactions are not rejected in the event of formation of cycles.

2. It requires no additional algorithms to be run, for detection and removal of deadlocks.

3. It is a distributed approach.

4. The approach requires no extra lock table storage as compared to distributed two-phase static locking. In the two-phase locking approach, if more than one transaction requests the same data item, these requests wait in queues. Instead of queues, in this algorithm, a local access graphs is constructed.

5. In this algorithm, the local access graph contains the number of transactions for which it must wait for execution to get the required data items. We can estimate the time duration for getting the required data items, based on the preceding transaction information. In this way, it is possible to get deterministic estimate of the time-duration to get an access to required data items. This provision, can be used in the real-time database systems.

6. In this technique, the odd edge removal guarantees a deadlock free environment. The odd edge can be removed instantly when it forms; hence, no transaction blocks forever. As a contrast, in the distributed two-phase locking, if a cycle forms, a transaction is blocked until the next deadlock detection cycle is invoked.
A similar list of advantages is also associated with the TDFG approach used for replicated data environment. We list some of the positive points of the concurrency control algorithm for replicated databases.

1. The algorithm does not reject transactions in the case of conflicts. On the contrary, in voting based approaches, if a group of transactions, requests a common data item, then only one among the group do gets a consensus vote, and the remaining transactions are rejected. The proposed algorithm eliminates the problem of repeated rollback of submitted transactions.

2. It is a distributed approach.

3. It allows parallel transmission of transaction requests, to sites, which is not possible in voting based approaches. As proposed by Thomas [THO79], in the majority consensus approach, each transaction request visits the sites in a circular fashion. Each transaction request, in the proposed approach, strictly visits a majority of sites to get consensus. In the voting based approaches the transaction request may visit all sites, to get consensus.

4. It is as reliable to site failures and partition failures as the voting based approaches.

5. In this algorithm, after visiting the majority of sites, the transaction data flow graph contains all the preceding transactions. By knowing the transactions, an approximate time estimate of the delay in waiting to get access can be made. In this way, in a real-time database environment, if an estimated time duration
is with in a specific deadline of a transaction, then the transaction can wait. Otherwise, an alternative procedure can be followed.

6.2. Directions for further research

Here, we briefly indicate some of the problems that require more research. Also, we examine critically some of the limitations of the work presented in this dissertation.

1. The performance model presented in chapter three, and chapter four is not complex. Our aim has been to communicate the main advantage of the algorithms. It would be interesting to do a performance of proposed algorithms by designing more complex models. Also, by testing in the real application environments the performance parameters can be better understood. The efficiency of the algorithms can be tested and compared with their counterparts in the literature in a much more diversified manner. Methods from queuing theory can be employed to analyze the sensitivity of time-stamps (transaction numbers), probability of a transaction conflict, probability of formation of odd edges and even edges and the probability of deadlock cycle. Different types of clocks, such as processor clocks, and program counters, can be simulated for assigning transaction numbers to transactions.

2. As mentioned earlier, the algorithms can be applied to real-time environments. Modifications are needed in the algorithms presented, in order to apply them
for real-time environment. That can become an interesting topic to further research.

3. The algorithms presented are based on transaction-to-transaction conflicts. These can be extended to study read-write conflicts, as these exist in dynamic locking in a distributed databases. For replicated databases, the weight of each site can be changed as per requirements (availability level). Also, number of votes to be collected can be varied as per the requirements of operation (read or write) to be performed. Research in this direction would also be an interesting piece of work.

4. For the concurrency control algorithm proposed in chapter three, we assumed no replication. And for the algorithm proposed in chapter four, we assumed full replication. But, in the real life applications, partial replication is many times preferable as per the requirements. So, in future, by mixing these two algorithms, we believe that algorithms can be designed for by considering the partial replication. It is worth to design such algorithm and evaluate their performance.

On the whole, with the proposed improvements as per the data flow graph technique, the transaction processing environments with a high volume of conflicts would tend to benefit. With these approaches, the domain of application of the concurrency control technique would be broader, as compared to the earlier approaches.