DISTRIBUTED DATABASE SYSTEM TECHNOLOGY FOR
MOBILE COMPUTING AND COMMUNICATION SYSTEMS

Bhavani Thuraisingham
The MITRE Corporation
Burlington Road
Bedford, MA 01730

ABSTRACT

Mobile computing and communication systems are expected to support mobile users with personal computers with access to information which would facilitate their travel and other related activities. Before such systems become common practice, several technologies need to be integrated. One such technology is distributed database systems technology. A distributed database management system manages a distributed database which is stored at different sites. It must ensure that users access the distributed database in a transparent manner. Such systems are becoming vital for the efficient processing of military as well as commercial applications. This paper describes distributed database system functions to support mobile computing and communicating systems.

1. INTRODUCTION

The rapid growth of the networking and information processing industries have led to the development of Distributed Database Management System (DDBMS) prototypes and commercial products (see for example CERI84). In such a system, the database is stored in several computers which are interconnected by some communication media. The aim of a DDBMS is to process and communicate data in an efficient and cost-effective manner.

While DDBMSs have continued to advance, the developments in computer communication technology and cellular radio technology have resulted in the emergence of Mobile Computer and Communication Systems (MCCS). The intent of such systems is to support mobile users with access to information which would facilitate their travel and other related activities. It is expected that such systems will be necessary not only for commercial and personal applications, but also for military applications, especially in the areas of command, control and battle management. As stated in a recent issue of the SIGNAL journal [SIGN93], “electronically interactive simulations linked to Army battle laboratories; the National Training Center, Fort Irwin, California; and to commanders are critical to providing flexible battle field information systems.”

Although significant advances have been made in MCCS technology, several technologies need to be integrated with MCCSs before such systems become common practice. One such technology is Distributed Database Systems Technology. This is because the efficient storage and management of the data are critical to support the needs of the mobile users. For example, information about the personnel at different echelons in a Army command hierarchy may be required to carry out the battle management functions. In various applications, users may want answers to queries such as “find the most efficient route to travel from city X to city Y”, “locate John Smith who is on his way from New York to Boston as his spouse has been taken ill”, and “locate the car with license plate number PPP as it has a kidnapped child.” In order to answer such queries in the least amount of time, the system must know which databases and/or caches access the locations of these data repositories. It must also return the response to the mobile user who could have traveled some distance since the time he issued the query.

This paper addresses data management issues for MCCS. In particular, object location, query processing, update management, and integrity maintenance functions are discussed. Architecture and data modeling issues will also be described. In Section 2, we provide an overview of mobile information systems. A more detailed discussion of such systems is given [WALK90]. In Section 3, we described data management for MCCS. The paper is concluded in Section 4. The developments and challenges in Distributed Database Systems Technology, which have influenced this paper, can be obtained in [THUR93]. It should be noted that in addition to database technology, communication technology also plays a key role in MCCS. However, the communication issues are beyond the scope of this paper.

2. OVERVIEW OF MOBILE COMPUTING AND COMMUNICATION SYSTEMS

Mobile information systems cover a wide range of systems. These include paging and message systems, cellular radio and telephone systems, satellite mobile communication systems, and more recently those systems which include wireless personal computers. Paging and message systems have been around since the mid-50s and are still used in hospitals and government organizations. The advent of cellular radio systems over a decade ago resulted in the car telephone. As stated in [WALK90], the
key feature of cellular radio is to divide an area into cells, each served by a low-power transmitter, with frequencies being used nearby. This would allow a higher number of mobile subscribers. To minimize interference between the users, the cells which use the same frequency are physically separated into clusters. That is, groups of cells form clusters which between them can use all available radio channels. When the number of subscribers that can be handled by a cell increases to the maximum, then the cell is split into smaller cells with each one having the same number of channels as the original one (see the discussion in [WALK90]).

The cellular radio technology combined with the increasing number of wireless personal computers and terminals that are in use today is being integrated to provide MCCSs. As stated in [IMIE92], it is expected that in the near future several million people will be carrying lightweight and inexpensive personal computers of which they could access a worldwide information network called Personal Communication Network (PCN). It is also expected that these users will be moving from place to place continuously to carry out their job-related activities. While traveling, a mobile user could cross several cell boundaries within a single day.

Several configurations are possible for a cellular network which groups cells into clusters. One such configuration is illustrated in Figure 1. Each cell is served by a base station which is responsible for transmitting and receiving data to and from the subscribers that are controlled by the cell. These base stations are connected to each other. The connection could even be point-to-point. The base stations in different clusters could communicate with each other possibly through other clusters as shown in Figure 1. For example, the base stations in cluster I communicate with the base stations in cluster III through the base stations in cluster II. The network could also be hierarchically organized as shown in Figure 2 (see [IMIE92] for a discussion of such a configuration). Each mobile unit is attached to a base station. The base stations are connected through location servers which are hierarchically organized. The location servers are responsible for locating other base stations and possibly the databases.

Figure 2. Cellular Network - Configuration II

3. DISTRIBUTED DATABASE SYSTEMS FOR CELLULAR COMPUTING

DDBMSs are being used for a variety of applications including those for command and control systems, process control systems, and banking systems. However, it is only recently that investigation on the application of such systems for mobile computing environments has begun [IMIE92]. The functions of DDBMSs include managing the data distribution, querying, and transaction management. Furthermore, in a multi-user update environment, concurrency control techniques must ensure that the distributed database is kept in a consistent state. Recovery techniques should ensure that the database should be brought back to a consistent state in the case of site link failures. The functions of a DDBMS have to be adapted to suit the application environment. In this section, we will discuss the functions of a DDBMS as well as architectural and data modeling issues for MCCS. We call a DDBMS for MCCS to be MCCS-DM (MCCS Data Manager).

3.1 ARCHITECTURE

One could expect the data needed to support the users of an MCCS to be stored in large databases at specific sites such as travel bureaus, traffic control stations, police stations, and hospitals, as well as caches and/or smaller databases at the base stations. The larger databases would typically store information such as maps, routes, and details of hospitals. The databases at the base stations could replicate some of the data in the larger databases as well as store data about its subscribers.

Due to the size of the databases as well as the distributed nature of the system, one would need the support of a DDBMS to manage the distributed databases. One possible architecture for an MCCS is shown in Figure 3. As can be seen, each base station has some data store which is managed by the data manager at the base station. The data managers at different base stations communicate with each other. Some data managers have increased responsibilities as they would manage cluster-related information also. Some others have access to the data.
managers which may be full-fledged database management systems (DBMSs) at sites such as travel bureaus and hospitals.

Due to the diverse nature of the databases and the data managers and the possible need to maintain autonomy of the data managers, one could expect the use of a federated architecture to be used in such an environment. Sheth and Larson have defined a federated database system (FDS) to be a collection of cooperating database systems which are possibly autonomous and heterogeneous [SHET90]. The intent is for a database system to continue its local operation and at the same time participate in a federation if it wants to. While heterogeneity brings about complexities not present in a homogeneous environment, autonomy, which enables a database system to join or leave a federation whenever it wishes to, makes the task of developing an FDS even more difficult. In an MCCS, there will be a greater need for sharing information between the various base stations. For example, if a user issues a query while under the control of cell A, and by the time the response is generated, the user has moved within the control of cell B, then the base station for cell A must communicate with the base station for cell B so that the user is informed of the responses. This means that the data managers at the two base stations need to have close communication with one another. However, the data managers at the sites may want some degree of autonomy and therefore, may not share all of the information with the base stations.

3.2 DATA MODEL

A data model represents the data in the databases. The data managed by an MCCS-DM could be quite complex and include multimedia data such as voice, text, images, and video, as well as the individual data types. Therefore, the data model should be powerful enough to support the representation of the diverse and multimedia data types. For example, photographs of certain suspects may have to be transmitted to the base station data managers and subsequently, to the police officers on the road by the data manager at the police station. In addition, video and audio scripts may also have to be transmitted.

If the architecture chosen is a federated one, then one makes the assumption that different databases with their different representation schemas already exist. The issue then is to efficiently integrate the different schemas. Should there be a single data model at the global level or would several data models co-exist? For example, could the users have a global view of the entire distributed database or could they have their own individual views? In either case, it would be desirable for the users to access the distributed databases in a transparent manner. If a global view is enforced, the query processor could transform the queries on the global view to the views of the individual databases. If each user has his own view, then the query processor could transform the user's view into the views of the individual databases. Other issues for a federated architecture include the representation of the individual schemas (which describe the data in the databases), determining which schemas to be exported to the federation, filtering appropriate information from the schemas at different echelons, integrating the schemas to provide a global view, and generating the external schemas for the users. In integrating the different schemas, the semantic and syntactic inconsistencies between the different representations need to be resolved.

Figure 3. An Architecture for MCCS-DM

Each data manager could be based on an extensible architecture. In such architectures, database systems are extended to provide the support for complex objects and deductions. Such systems can help the users with problem solving and decision support.

Figure 4. Schema Architecture

Figure 4 illustrates a schema architecture for an MCCS-DM. This architecture has been adapted from the five-schema architecture described in [SHET90]. Each component (or data manager) schema consists of the schema of each data manager. Cluster schema consists of the schema of each cluster. This schema is formed by integrating the data manager schemas. Note that each data manager may not export all of its schema to the cluster. That is, a subset of the data manager schema could be represented in the cluster schema. The cluster schemas
are integrated to form the MCCS-DM schema. Again, each cluster may export only a subset of its complete schema. Different views of the MCCS-DM schema may be created for the users. These schemas are called the external schemas. Note that the schema architecture described here does not include the schemas for permanent databases such as hospital databases and policy station databases.

3.3 FUNCTIONS

3.3.1 Overview

The functions of a DDBMS include query processing, update processing, transaction management, handling data distribution, maintaining integrity and security. Each of these functions have to be adapted for an MCCS-DM. Several issues are yet to be investigated. For example, will there be a need for the users to execute transactions? What sort of metadata management capabilities should be provided? Are special index methods and access strategies needed for such systems? What are the security mechanisms that must be enforced by an MCCS-DM? What are the security issues for telecommunications systems? What are the techniques to locate the users and other resources (which we will call objects) in a cellular network environment? Are there special backup and recovery procedures? Are there any real-time constraints imposed on the queries and updates? Are special concurrency control mechanisms needed? Will the query processing algorithms use any knowledge discovery techniques? What is the impact on the functions of an MCCS-DM to support multimedia data and different data types? What types of data management standards are needed for a mobile computing environment? These are difficult questions and require much research before appropriate answers are obtained.

Below, we discuss only some of the functions of an MCCS-DM. These are locating objects, query processing, update management, metadata management, storage management, integrity maintenance, and realtime data processing.

3.3.2 Locating Objects

The objects in this case would be users and resources such as databases, metadata, and directory servers. The questions is, how are the objects such as various police stations, hospitals, and users located? The problems are more difficult when the objects are mobile. For example, when a user has moved from one cell to another, how could he be located? In the case of stationary objects, each base station data store could have information on certain objects and also information on which base station or site to query if the location of the objects cannot be determined within the station. Another approach would be for a base station to broadcast the information to all stations and sites. A third approach would be to send a message only to certain designated base stations.

In the case of mobile users, should it be up to them to inform appropriate base stations that they are moving from cell to cell? This would make things very difficult for the user. The preferred approach is for the base station to keep track of its users and when it detects that a user has gone outside of its control to inform some designated server. Similarly, if a new user comes under the control of a base station, it would also inform a designated server.

3.3.3 Query Processing

One would expect queries to be rather ad hoc. One issue is whether a special query language is needed for the users. In general, it is not desirable for the mobile users to have to learn about the different data models and therefore, the query languages for the associated data models. On the other hand, if each user issues queries with his own language (either natural language or a query language) a greater burden would be placed on the MCCS-DM. A third alternative, which seems the more attractive one, is for users to learn a common language. This could cause some problems with international users.

Another issue with query processing is getting the responses to the users in a timely manner. If the user has not moved since the time he issued the query, then many of the optimization techniques could be applied to process the query. In addition, deductive database techniques could also be used to make some intelligent deductions. For example, if the user requests the most efficient path from city X to city Y, then the MCCS-DM would attempt to select the path which has been predetermined. However, if there are unusual situations, such as accidents, then the data managers will have to compute the path dynamically. There are additional concerns if the user has moved to a different cell since the time he posed the query. Then the response has to be routed to the base station which received the query, but to the base station which controls the user. Determining this base station would depend on how good the algorithm is to locate mobile objects.

Deductive database techniques are useful not only in making intelligent deductions, they can also help the user in problem solving and decision support. For example, consider the case where the police are chasing after some suspects and information about the case is being continuously updated from different sources. In such a situation, deductive database techniques will be useful for not only processing the queries and answering the questions of the police in an intelligent manner, but also guide the police in locating and identifying the suspects.

In addition to the above issues, issues for query processing in a federated environment have to be addressed for an MCCS-DM also. For example, different data managers may utilize different query processing and optimization strategies. One of the research areas here is to develop a global cost model for distributed query optimization.
3.3.4 Update/Transaction Management

One can expect for users to update the distributed database concurrently. For example, several users may enter a cell at the same time. Therefore, the database managed by the data manager at the corresponding base station will have simultaneous updates. Appropriate concurrency control techniques need to be enforced so that the consistency of the database is maintained. Databases have to be updated even when a user leaves a cell. The database associated with the cell that the user was attached to has to be updated so that the user is deleted from it. In addition, appropriate designated servers have to be notified about this movement so that the databases at these servers have to be updated also.

The databases (or the caches) at the base stations have to be updated when the information that it contains becomes obsolete. For example, the databases at the various sites such as the hospitals and police stations will probably be updated periodically. The frequency of these updates is determined by factors such as the environment and other events. When these databases are updated, the databases at the base stations may become obsolete. Therefore, the main question is, when should the base station database be updated? Should it be done immediately after a change has occurred at the sites, or should it be deferred until it is determined that the base station databases have obsolete data?

In the case of replicated databases, the copies have to be kept consistent. The issues that deal with replica control have to be handled by an MCCS-DM also. In addition, if the updates are issued as transactions, then appropriate scheduling algorithms have to be developed. Also, in a federated environment, different data managers may utilize different algorithms for transaction processing. Work is being directed toward integrating the various transaction processing mechanisms. For example, techniques which integrate locking, timestamping, and validation mechanisms are being developed. It has been suggested that the notion of strict serializability may have to be sacrificed for a heterogeneous environment. That is, one may have to contend with a weaker form of serializability.

3.3.5 Metadata Management

Metadata describes the data in the database. They include the schemas, integrity constraints, access control rules, information on index methods and access strategies, optimization techniques used, and other administrative policies. A discussion of the various types of schemas were described in Section 3.2. In addition to performing schema integration and transformations, metadata management functions include querying and updating the metadata.

In a mobile computing environment, a user may want to have answers to questions such as how many users are in the system? How many base stations are in the system? How many data managers are in base station A? How many clusters are there in the system? What types of information is maintained by the data manager in base station A?

Metadata manager should also provide the support for a dynamically changing environment. For example, a base station may support 20 users at time t1 and 30 users at time t2. Also the data in the database may be changing continuously. Schema evolution support seems essential for mobile applications.

3.3.6 Storage Management

One could expect an MCCS-DM to maintain a very large number of smaller databases. In addition, the MCCS-DM should access the larger permanent databases located at sites such as hospitals, police stations, and ground stations. Information from these larger databases would often be cached and stored in the smaller databases.

There are several, storage management issues that need to be investigated. First of all, how often should the information be cached? What is the size of each cache? What sort of storage structures should be used. Traditional databases use structure such as B-Tress. Are these structures suitable for MCCS-DM? What sort of index strategies and access methods should be supported? Are special strategies ended to support multimedia data? What is the relationship the storage manager has to the other modules such as query processor, update/transaction manager, and metadata manager? What is the impact of data distribution on storage management techniques. Much research is needed on storage management issues.

3.3.7 Integrity Maintenance

Integrity maintenance techniques include concurrency control, backup and recovery, and enforcing integrity constraints. Concurrency control issues were discussed under update management. Recovery techniques must ensure that the databases are recovered to a consistent state in the case of site or network failures. In a cellular network environment, the base stations, the sites, and the links which connect the users to the base station as well as the different stations and the sites may fail. Therefore, the recovery algorithms must ensure that such failures can be handled appropriately. For example, when a user moves from one cell to another, he might lose contact with the network due to the link failure that connects him with the new cell. Then, when the link comes back up again, the user must be registered with this base station. Otherwise, it will be difficult to locate the user so that he can receive appropriate responses. Backup of the databases is an additional concern which is related to maintaining integrity. On-line backup procedures are being used for many databases. This is because off-line procedures will consume too much time especially for large databases. The issue here is to develop improved techniques for backup so that it will not impact functions such as querying and updating.
Integrity constraints may be application-independent or application-specific. Application-independent constraint depends on the data models utilized. Application-dependent constraints are often referred to as semantic integrity constraints. These constraints may be enforced across databases or within a database. In a cellular network, there could be constraints such as the number of subscribers for a cell cannot exceed a certain amount, or the total number of subscribers for a network cannot exceed a certain number. Lately, enforcing semantic integrity constraints in a federated environment is receiving a lot of attention. Some of the techniques designed would be useful for an MCCS-DM.

3.3.8 Realtime Data Processing

In a realtime database management system, the transactions (including the query and update requests) have timing constraints (or deadlines) that must be met. Therefore, traditional query processing and transaction management techniques have been adapted to meet the timing constraints.

One can expect certain queries, update requests, and transactions to meet timing constraints for an MCCS-DM. For example, in the case of a kidnapping case, it is crucial that the information be dispatched to the police officers on the road within a certain time. In such a situation, the message that is transmitted from the police station to the mobile users may have a timing constraint. The system will then examine the timing constraints assigned to the various message and assign priorities to them. Another example would be for a traveler to be taken ill while on the road. Then a time-constrained query could be issued to find the shortest path from the location to the nearest hospital. Such a query could be given a high priority.

Many of the issues in realtime data processing apply for an MCCS-DM also. For example, if the time constraint on a query, update request, or message cannot be met, then what is the value of processing the request. Also, are criticality values assigned to the data? Does the data become useless after a certain time? The impact of realtime processing on query, update, metadata management, and storage management functions of an MCCS-DM need to be examined.

4. CONCLUSION

Mobile computing and communication systems will require effective management, retrieval, and integration of databases which are possibly heterogeneous in nature. Developing a database management system for mobile computing and communication systems will require new technologies and novel approaches for data management. While hardware is rapidly advancing to provide data storage, processing, and transmission, the software necessary for the retrieval, integration, and management of data remains a challenge.

This paper has identified some architectural, data modeling, and functional issues for managing the data in mobile computing and communication systems. First, an overview of mobile computing and communication systems were discussed. Then some architectural and data modeling issues were given. Finally, a discussion of the issues for the various functions of a data management system for mobile computing and communication applications were given. The set of issues identified is by no means considered a complete list. As the progression of research and prototyping continue, additional issues will arise.

The next step is to conduct a detailed investigation of the data management requirements for mobile computing and communication systems and subsequently identify a more complete set of issues. Once this is completed, the current state of the technology should be assessed and its utility to address the needs of the such applications should be determined. Finally, areas that need further work should be identified and solutions to the issues need to be developed.

REFERENCES


