1988

THE ISSUE OF SITE AUTONOMY IN DISTRIBUTED DATABASE ADMINISTRATION

Hemant K. Jain
University of Wisconsin-Milwaukee

Hwa-suk Ryu

THE ISSUE OF SITE AUTONOMY IN DISTRIBUTED DATABASE ADMINISTRATION

Hemant K. Jain
Hwa-suk Ryu
School of Business Administration
University of Wisconsin-Milwaukee

ABSTRACT

With the increasing trend towards distributed systems, the administration of distributed databases has become very important. A key issue in this environment is the degree of autonomy for each participating site. This paper classifies the degree of site autonomy into three different levels and provides some guidelines for determining the appropriate degree of site autonomy from both organizational and technological perspectives. The alternative organizational structure for database administration and the specific tasks assigned to a Global DBA and Local DBAs under different site autonomy policies is discussed. Finally, the three site autonomy policies are compared.

1. INTRODUCTION

Over the last decades, database technology has established itself as a principal means of satisfying the information requirements of large organizations. As more and more databases are implemented, the need for inter-database communication becomes important. A distributed database management system (DDBMS) seeks to provide this. A distributed database is a collection of data located at different sites connected by a data communication network. Each site of the network has autonomous processing capability and can process local applications, which may require data from several sites. In a distributed database environment, there are two types of users, the global user requiring data from various locations and local users requiring data from only a local site. Ideally the user should not be concerned with the location of data. The DBMS is responsible for locating the data and providing it to the user.

In a distributed database environment, it is possible to identify a hierarchical control structure based on a global database administrator, who has the responsibility of the whole database, and local database administrators, who have the responsibility of their respective local databases. Even though the functions of distributed database administration (DDBA) are similar to those of conventional database administration, there are some distinct functions which need to be performed in a distributed database environment (e.g., database fragmentation, fragment allocation, data communication, catalog management, and coordination of inter-site relations). The nature of control at the global level and the relationship between global and local levels can affect the performance and site autonomy. The degree of site autonomy may vary from full site autonomy, where inter-site coordination is performed by the local database administrators without any centralized global database administrator, to almost completely centralized database administration and control.

The primary purpose of this paper is to analyze the issue of site autonomy and its impact on an organization in the distributed database environment. The following section describes the functions of database administration and the evolution paths of a distributed database system. Next, the degree of site autonomy is classified into three different levels and some guidelines for determining the appropriate degree of site autonomy from the organizational perspective are presented. Finally, specific distributed DBA functions under different site autonomy policies are described and compared.

2. FUNCTIONS OF DBA AND EVOLUTION OF DISTRIBUTED SYSTEMS

In recent years, attempts have been made to differentiate the role of data administrator (DA) and database administrator (DBA). Kahn (1983) described the distinction between DA and DBA. Data administration is responsible for the establishment and enforcement of policies and procedures for managing the data as a corporate resource. It involves the collection, storage, and dissemination of data as a globally administered and standardized resource. Database administration (DBA), on the other hand, is a technical function which performs database design and development, provides education on database technology, provides support to users in operational data management-related activities, and may provide technical support for data administration. For the purpose of this paper, DBA tasks are assumed to have functions of both DA and DBA because both functions have integral relationships and need be considered in the study of site autonomy.

2.1 Functions of DBA

Weldon (1981) classified the DBA functions into four categories related to the stages of database development: planning, design, operation and control, and usage. Specific DBA tasks in each of these categories are described here. It should be noted that all of these tasks are not done solely by DBA. Some tasks are performed in cooperation with others, such as system developers, operation personnel and end-users.

a) **Planning.** The DBA has an important role in both data processing and organizational planning. The following planning tasks can be identified:

- Define database goals to support organizational objectives.
- Develop and revise long-range plans to achieve database goals.
- Evaluate, select, and allocate hardware and software.
- Assess impact of changes in technology and information requirements on database.

b) **Design.** The DBA provides technical support to application system developers for both logical and physical database design. The following design functions are performed:

- Analyze information requirements (both global user view and local user view).
- Design logical and physical schema.
- Fragmentation and allocation of database fragments to local sites.
- Develop global and local Data Dictionary/Directory.

c) **Operation and Control.** The DBA role includes database implementation and maintenance. The tasks performed are:

- Maintain consistency and compatibility between sites.
- Develop and enforce data and operation standards.
- Set policies and procedures for database backup and recovery.
- Formulate and apply a database security policy.
- Develop and monitor database performance measures.
- Maintain DBMS and related software.
- Resolve database operational problems through reorganization or redesign.

d) **Usage.** The DBA performs the following functions to encourage and facilitate use of data:

- Resolve conflicts which might be created between end-users, operators, system developers, and sites.
- Education and training.
- Establish and maintain channel of communication with each component system, users, and operational personnel.
- Establish formal trouble reporting system.
- End-user support.

Based on the nature of decisions made, the above DBA functions can be mapped to Anthony's (1965) model of three levels of managerial activities (see Figure 1).

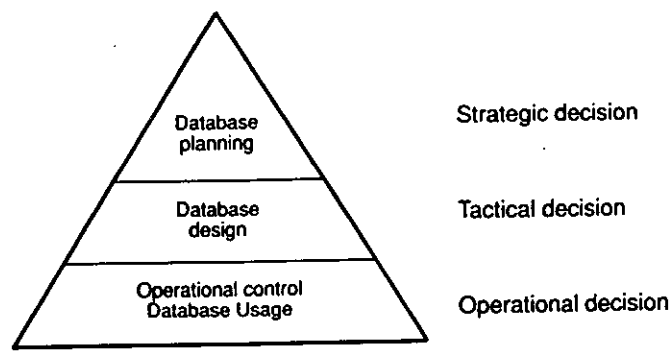

Figure 1. Classification of DBA Functions

2.2 Evolution of Distributed System and the Role of DBA

In traditional centralized database systems, there is a single DBA (single person or single group). A distributed database system, however, has many sites and the database administration functions tend to be more complicated. These functions may be performed by a Global Database Administrator (GDBA), responsible for organization-wide database administration functions, or a num-

ber of Local Database Administrators (LDBA), responsible for their own local site. A combination of GDBA and LDBA might also be used. The complexity of the DBA functions and the organization of the DBA group may depend on the way the distributed system evolved.

Bray (1982) illustrates three ways in which a distributed system may evolve in an organization:

- Design and develop a new application system as a distributed system.
- Take a system that is currently centralized and begin to distribute some of its functions to local sites.
- Add a communication system to link several currently independent sites and try to integrate their systems into a distributed system.

In the first two cases, the DBA functions are relatively easy.

When the DDBMS comes about as a result of merging several previously independent systems at different sites, the DBA functions result in many more organizational problems. Each of the independent sites may already have a database administrator performing the DBA functions for the site. In a distributed environment, the actions of these independent DBAs need to be coordinated. This coordination is complicated by the geographic dispersion of both the users of each site and the DBAs who represent them. Attempts to implement some coordination mechanism, such as a common data definition and uniform programming standards, can create serious organizational problems. Furthermore, the traditional database administration functions become more complicated in the distributed environment. If part of the data is to be replicated, there must be an agreement on how often the copies must be updated and who has the primary responsibility for the data. Also, if the database is partitioned, how is it partitioned and where is each part stored? Since these decisions can have a serious impact on a user's performance, it may be very difficult for several previously independent database administrators to arrive at a consensus.

Apart from these organizational coordination issues, several other technical functions needs to be performed in a distributed DBMS environment. Figure 2 represents a reference architecture for distributed databases (Ceri and Pelagatti 1984). The global schema represents the overall logical relationships between the records. The fragmentation schema represents the schemes for fragmenting the global database into a number of fragments. Allocation schema represents allocation of fragments to various sites in a redundant or an irredundant manner. Mapping to the local DBMS schema is provided via local mapping schemata.

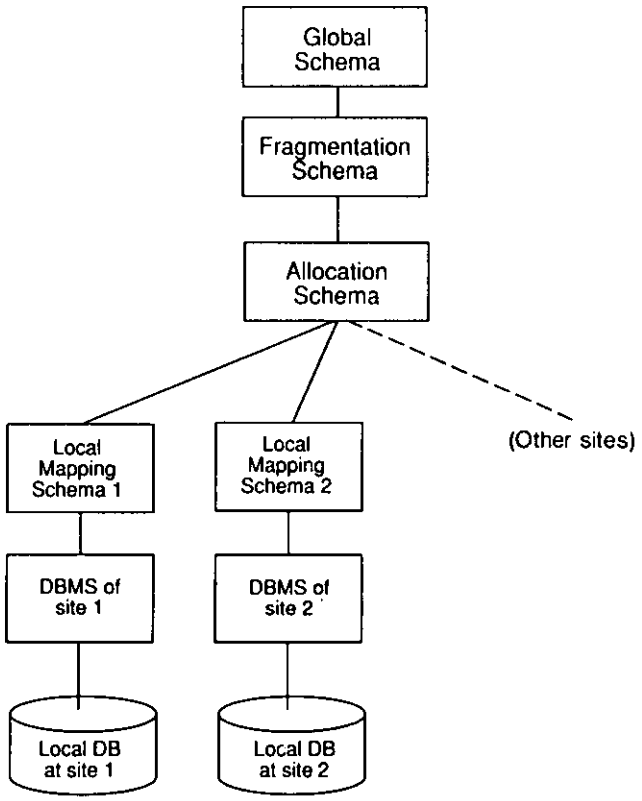

Figure 2. A Reference Architecture for Distributed Databases (Ceri and Pelagatti, 1984, p. 38)

3. THE ISSUE OF SITE AUTONOMY IN DISTRIBUTED DATABASE ADMINISTRATION

The word "autonomy" has been commonly used in the public administration field. In a distributed database administration environment, the term "site autonomy" describes the situation where a global database administrator is completely missing and the inter-site coordination is performed by the local database administrators (Ceri and Pelagatti 1984). The principal technique used to achieve the autonomy is decentralization of functions and data. Deadlock detection, recovery, locking, and catalog management are all performed in a decentralized manner.

Haas et al. (1982) used the definition of site autonomy in a very limited scope: that is, sites should perform all operations on their own data. It includes the ability of each site to control who may access its data as well as the ability to access and manipulate local data independent of other sites. This definition of site autonomy describes only a subset of the DDBA functions described earlier. Since DBA functions are assumed to include both data administration and database administration, the issues of site autonomy should consider all four categories of DBA functions. An organization may choose to implement different degrees of site autonomy in performing DBA functions.

3.1 Degree of autonomy

To analyze the impact of site autonomy on the design of a distributed system and to select a proper site autonomy policy, we have classified the degree of site autonomy into three different levels: *full autonomy, semi-autonomy, and no autonomy.*

Full autonomy. When the local sites have full autonomy, the global DBA function does not exist. All DBA functions, from planning to usage, are performed by inter-site coordination between local DBAs. A coordinating body armed with standards and procedures (but without any real authority or power of control) may exist.

Full autonomy is most easily achieved in a system where databases are loosely coupled and separately managed. Furthermore, site autonomy may be used to achieve resilience to failure of sites and/or communication lines. When the distributed system has been developed as a result of merging several previously independent nodes, full autonomy may be most acceptable. However, in this case difficulties can arise when the actions of individual DBAs must be coordinated.

Semi-autonomy. In the case of semi-autonomy, there exists a global database administrator in the central organization and local database administrators at each local site. Most routine DBA functions which are independent of other sites are performed autonomously by local DBAs. Some DBA functions which affect the efficiency and effectiveness of the global system are performed and controlled by the global DBA. The degree of delegation may differ between various distributed database systems.

The basic operational philosophy here is:

- Local DBAs have authority to propose and work out changes in detail and to cater to local information needs (inquiries, subschemata, etc.).
- The global DBA has the authority to sanction and to reject any change on the grounds of incompatibility with the requirements of another site and/or overall efficiency and effectiveness of the system. The global DBA also establishes the strategy for global system changes, which are then implemented by the local DBAs.

No autonomy. This environment has no local DBAs and all the DBA functions are centralized. The global database administrator has sole authority for any of the DBA tasks. The local DBA organizations, if they exist, are simply the "local representatives" of the central organization. This type of organization allows tight control and facilitates enforcement of organization-wide standards.

Thus the role of the GDBA and LDBAs depends on the degree of site autonomy adopted by the organization. Figure 3 shows the DBA functions performed by LDBA and GDBA under different degrees of site autonomy.

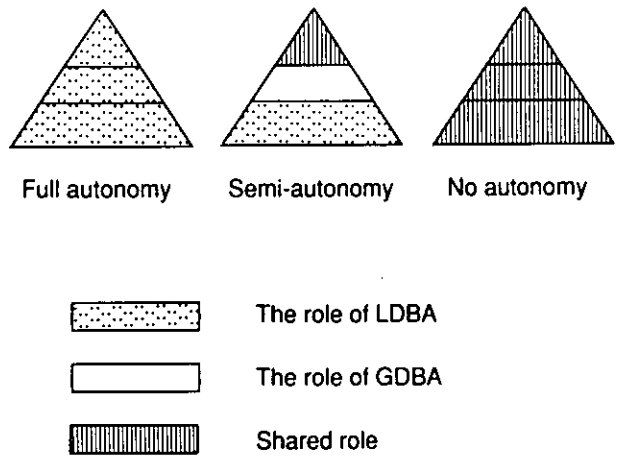

Figure 3. The Role of GDBA and LDBA Under Different Degree of Site Autonomy

Under full autonomy, three levels of DBA functions are performed by LDBAs through inter-site coordination. On the other hand, under no autonomy policy, all DBA functions are performed and controlled by GDBA.

Under the semi-autonomy policy, however, the role of the GDBA and the LDBA needs to be clarified. The GDBA is responsible for making strategic decisions, such as setting overall system objectives and long-range planning. The LDBAs are responsible for day-to-day operational control of their respective local sites. The role of both GDBA and LDBA for middle level managerial activity (management control) is flexible; this activity can be performed by either the GDBA or LDBAs. The border line between the GDBA's role and the LDBA's role may move up or down within the middle level of managerial activities. It will depend on the degree of delegation.

3.2 Guidelines for Determining the Degree of Site Autonomy

The decision on the degree of site autonomy should be based on the following considerations:

- Current structure of the organization (centralized versus decentralized).
- Organizational cultures.
- Information flow in the organization.
- Local data needs and global data needs.
- Type of evolution of distributed system (development of new distributed system or connecting existing independent sites, etc.).

- Data security requirements.
- Whether sites are loosely connected or tightly connected.
- Vulnerability of data communication system to failure.
- The availability of qualified DBA personnel with a background in data communication.

It is important to keep in mind that a major consideration in the selection of the degree of site autonomy is a fit between the functions performed by the GDBA and LDBA under different degrees of site autonomy and the overall organizational environment such as current organizational structure, organizational culture, information flow, and end user departmental needs. For example, when the overall organization is highly decentralized, the full autonomy approach fits more naturally with the structure of the organization. When the organization has a culture that emphasizes efficiency and control, the no autonomy approach is adequate because centralized control, uniform operations and economies of scale ensure efficiency (King 1983). On the other hand, the full autonomy approach is more appropriate when the organization's culture emphasizes effectiveness because users' needs are well reflected during database design and operations.

Table 1. Selecting the Degree of Site Autonomy

FACTORS	DEGREE OF SITE AUTONOMY		
	NO AUTONOMY	SEMI-AUTONOMY	FULL AUTONOMY
Overall organizational structure	Centralized ←		→ Decentralized
Organizational culture	Efficiency ←		→ Effectiveness
Information flow within organization	Mainly between sites ←		→ Mainly within site
Data need	Mainly global data ←		→ Mainly local data
DDBMS evolution path	From centralized to distributed ←		→ Merging existing DB
Data security required	High ←		→ Low
Connection between sites	Tight ←		→ Loose
Vulnerability of data communication	Hard to fail ←		→ Easy to fail
Availability of DBA personnel	Difficult ←		→ Easy

In some cases, however, technical and operational factors can dominate organizational considerations. For example, consider a situation where a distributed system is to be developed by merging existing independent data processing nodes in a highly centralized organization. The excessive redesign efforts required (such as changes in data definition, programming standards, and hardware replacements) in order to achieve uniform standards and control may not be justified in a cost-benefit analysis. In this case, the organization is confronted with a difficult decision. The trade-off is between high development costs and a high degree of site autonomy with less global control. Table 1 provides guidelines in selecting the degree of site autonomy with major environmental factors which should be considered in making the site autonomy decision.

4. ORGANIZATION OF DBA FUNCTIONS

Based on the site autonomy policy selected, the DBA functions can be structured in different ways. The alternative organizational structures for database administration and the specific tasks to be assigned to the GDBA and LDBAs under different site autonomy policies will be described here.

4.1 Organizational Structure of DBA Group

The degree of site autonomy is a critical issue when organizations adopt a distributed database system. If each local site has a DBA to provide the needed expertise, it would be difficult for the DBA not to be biased towards the local view of data instead of the overall corporate interests. On the other hand, if the functions of the DBA are completely centralized, then response to local data requirements may be slow and unsatisfactory. The alternative organizational structures of database administration functions with different degrees of site autonomy are described below.

4.1.1 Centralized DBA Organization

This type of organization implies strong centralized control. DBA functions are centralized; therefore, local sites have no autonomy. Figure 4 represents a centralized organization.

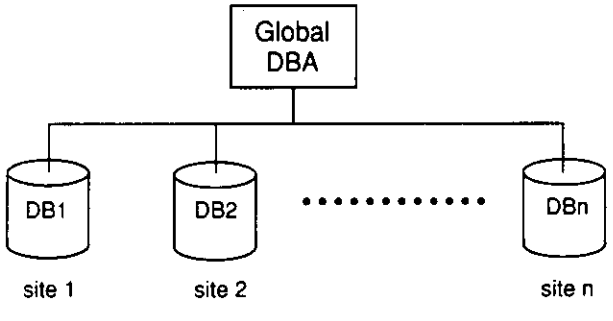

Figure 4. Centralized Organization

4.1.2 Hierarchical DBA Organization

The hierarchical organization implies a central control site (global DBA) and subordinate local sites (local DBA). This type of organization is suitable for a semi-autonomous environment. However, the degree of autonomy may vary based on the organizational policies. Figure 5 represents a hierarchical DBA organization. If the number of local sites exceeds the global DBA's span of control, intermediate DBAs who are responsible for controlling a group of local sites and reporting to a global DBA may be used. These intermediate DBAs can be characterized by region or by functional areas.

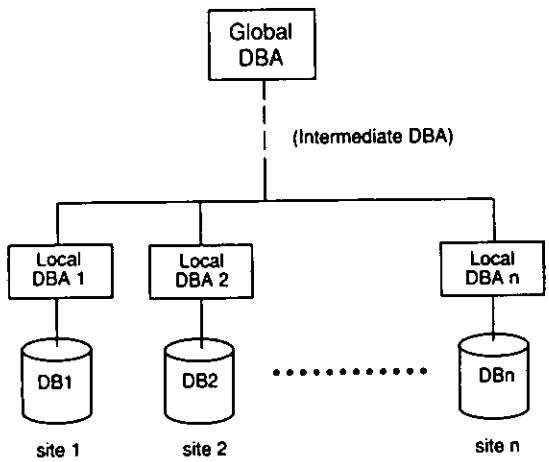

Figure 5. Hierarchical Organization

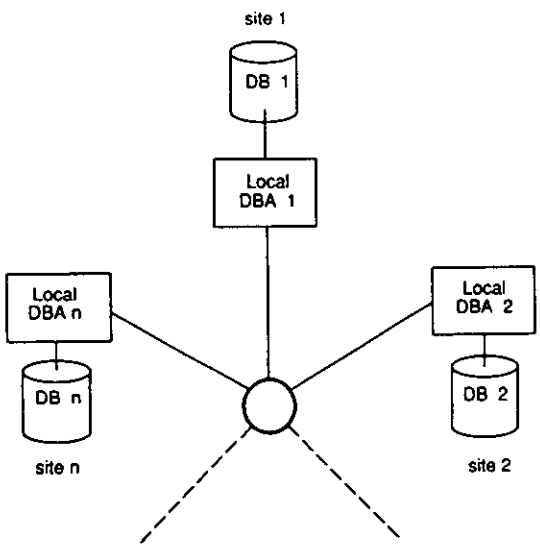

Figure 6. Autonomous Organization

4.1.3 Autonomous DBA Organization

In an autonomous organization, there is no GDBA because local sites have a high degree of autonomy. In order to function properly, this configuration must involve the joint effort of LDBAs, perhaps in the form of a committee. A central coordinator may be present to facilitate coordination among the sites. The coordinator may not have actual authority for imposing consistent database management practices at all local sites. Standards are negotiated and agreed upon by the committee. Figure 6 shows an autonomous organization. The circle in the center represents a LDBA committee or coordinator.

Based on the above three structures of DBA organization presented above, the relationship between the degree of site autonomy and the organizational structure for database administration function can be defined. Table 2 shows this relationship.

Table 2. DBA Organization and Degree of Site Autonomy

DBA ORGANIZATION	DEGREE OF SITE AUTONOMY		
	NO AUTONOMY	SEMI-AUTONOMY	FULL AUTONOMY
Centralized	1	2	3
Hierarchical	2	1	2
Autonomous	3	2	1

1 : Highly Recommended
2 : Better to Avoid
3 : Not Recommended

Obviously, the best combinations between the degree of site autonomy and organizational structure for DBA are the centralized DBA organization to be used with the no autonomy policy, the hierarchical DBA structure with the semi-autonomy policy, and the autonomous DBA structure with the full autonomy policy. Two extreme combinations which are the use of centralized DBA structure with the full autonomy policy and the autonomous DBA structure with the no autonomy policy should be avoided since these combinations may cause serious organizational problems.

Other combinations, which are marked "2" in Table 2, are less efficient than the combinations marked "1". For example, under the hierarchical DBA structure with the full autonomy policy, the conflict between a GDBA who wants to control and LDBAs who assume full autonomy may result in many more organizational problems. Therefore, an organization with the combination marked "2" should pay much more attention to the possible conflicts and try to resolve them when they exist.

4.2 Specific Responsibilities of GDBA and LDBA under Different Autonomy Policies

Based on a specific system developed at Bell Laboratories, Walker (1982) discussed the administrative tasks in a distributed database environment as system level res-

ponsibilities and component responsibilities. The system level functions set system-wide policy and coordinate inter-component activity. The component functions include implementation of the system-wide policies and supervision of the tasks local to the particular component. From the viewpoint of site autonomy, this specific system can be considered as semi-autonomous. Walker (1982) also listed the specific responsibilities of GDBAs and LDBAs. However, he did not consider the importance of site autonomy in defining the functions. Haas et al. (1982) discussed several aspects of the distributed database design which affect the autonomy of the participating sites. The successful implementation of site autonomy policy requires a careful distribution of function and management responsibility among the participating sites.

Some important tasks that can be assigned to GDBA and LDBA under three different levels of site autonomy are listed here. This list is intended to provide guidelines for implementing site autonomy in distributed database administration.

FUNCTION: Define Goals and Develop Database Plans

Full Autonomy. A committee of LDBAs defines and develops overall goals and plans. However, conflicting local needs may make it difficult to arrive at a consensus plan.

Semi-Autonomy. The GDBA defines overall system goals and develops organization-wide plans. The LDBAs are responsible for defining and developing goals and plans for each site. An alternative bottom-up approach, where the LDBA develop local plans and the GDBA coordinates local needs with overall system plans, can be used.

No Autonomy. The GDBA defines and develops goals and plans for both the global and local levels. The GDBA needs to pay attention to local requirements and may frequently contact local representatives.

FUNCTION: Develop Standards and Policy

Full Autonomy. A committee of the LDBAs is responsible for developing system-wide standards and policies for efficient operations. In this case, it may be difficult to coordinate different view points. Consensus standards developed through coordination are implemented by the LDBAs.

Semi-Autonomy. The GDBA sets standards and policies and LDBAs implement them. Alternatively, the LDBAs can propose standards and policies and the GDBA coordinates and integrates them with system wide goals and policies.

No Autonomy. The GDBA sets overall standards and policies. In this case, the GDBA tends to emphasize overall efficiency and effectiveness, which can cause local user resilience.

FUNCTION: Evaluate and Select Hardware and Software

Full Autonomy. The LDBAs plan and control hardware and software in each site under the standards and policies which were developed by coordinating different LDBAs' requirements. The hardware and software needs to be compatible. This type of autonomy is adequate for heterogeneous configuration.

Semi-Autonomy. The LDBAs determine requirements for hardware and software within each local site, and report all requirements and configuration changes to the GDBA. The GDBA coordinates the local requirements to maximize overall system performance.

No Autonomy. The GDBA plans, coordinates, and controls all hardware and software installations, replacements, and upgrades. The GDBA has full authority and responsibility for the overall configuration. This type of autonomy is very effective in homogeneous DDBMS environment.

FUNCTION: Distributed Database Design

Full Autonomy. The local schemata are required to be integrated into a single, global schema. LDBAs participate in this integration process and a global schema is produced as a compromise between LDBAs. There are many potential conflicts between local requirements which need to be resolved. A common data model for describing the global schema may be used. LDBAs are also responsible for the translation of each local schema into the common data model.

Semi-Autonomy. The GDBA is responsible for designing the global schema based on local user's view. The GDBA coordinates and controls the conflicts between LDBAs to reach a compromised global schema. Both the GDBA and LDBAs participate in the design of the fragmentation schema and allocation schema. LDBAs are responsible for mapping the local schema.

No Autonomy. The GDBA starts by designing the global schema; the database is then fragmented. The fragments are then allocated to the sites, creating the physical images. This design method, known as a top-down approach, is attractive because it allows rational design. However, the data needs of local sites may not be met in a completely satisfactory manner.

FUNCTION: Backup and Recovery

Full Autonomy. The LDBAs are responsible for scheduling procedures for backing up the local databases and for testing backup procedures periodically to assure their effectiveness in restoring the database. A committee of LDBAs may coordinate the development and testing of contingency plans for handling failures of local sites to minimize losses due to natural disaster and schedule backup and recovery processes between local sites.

Semi-Autonomy. The GDBA is responsible for coordinating schedules of backup and maintenance processes at local sites to minimize their impact on the operations of other sites. Also, the GDBA develops and tests contingency plans for failures of the overall system. This includes disaster planning as well as plans to handle the failure of one or more local sites. Finally, the GDBA develops and tests degraded mode operation plans (plans for operations when one or more parts of the network are down), including plans to recover and synchronize components following degraded mode operation.

The LDBAs periodically test backup procedures which were developed by the GDBA to assure their effectiveness in restoring local databases. LDBAs also develop and test contingency plans to handle the failure of local sites. Plans for the operation of each local site when some other sites are unavailable are also developed and tested.

No Autonomy. The GDBA is responsible for periodically scheduling backup and recovery procedures to assure their effectiveness. The GDBA also develops and tests contingency plans for failures of the whole system. Some of the above functions can be performed by local representatives or local operation staffs. In this case, the GDBA mainly performs development, coordination, and control of global plans. Actual operations are done by local staffs.

FUNCTION: Security and Privacy

Full Autonomy. The LDBAs cooperate in developing database security policies concerning which users and system groups are authorized to access a particular part of the database. It is convenient to form a committee to coordinate different viewpoints in establishing these policies. After consensus policies are developed, each LDBA is responsible for local enforcement of these policies, establishment of control measures, and monitoring compliance with those measures. Periodic inspection of computer operations is performed to assure the protection of each local database.

Semi-Autonomy. The GDBA develops system-wide policies about what parts of the database each user and system group is authorized to access. Appropriate security classifications for each group are determined. Documentation to demonstrate compliance with developed organizational security policies is maintained.

The LDBAs are responsible for implementing security classifications determined by the GDBA, monitoring compliance with those measures, reporting compliance data to the GDBA, and periodically inspecting computer operations for proper procedures to assure the protection of each local database against inadvertent or accidental damage by users. LDBAs are also responsible for physical protection of the local databases.

No Autonomy. The GDBA is responsible for security and privacy functions for the entire system such as developing system-wide policies, monitoring compliance with those measures, and inspecting computer operations for proper procedures, etc. However, it is difficult to enforce system-wide policies because of the distributed environment. In this case, it is useful to perform these functions by support of local operations or maintenance personnel.

FUNCTION: Performance Monitoring, Database Reorganization and Redesign

Full Autonomy. The LDBAs assign priorities to transactions at the local site. Any inter-site transactions are coordinated by corresponding LDBAs. The performance of transactions, hardware, operating system, and application software are monitored by the LDBAs. Finally, the LDBAs monitor database size and growth, tune the system by adjusting appropriate system parameters, and perform database reorganization when it is necessary.

Semi-Autonomy. The GDBA assigns priorities to transactions that compete for time and resources within the distributed database system. The GDBA also monitors system performance to ensure consistency across local sites, collects statistics on transaction volume, response time, etc. The GDBA also monitors database size and growth and reorganizes the database when necessary.

The LDBAs implement the priorities determined by the GDBA, monitor transactions, hardware, and software performance at the local sites, and send a summary of the performance data to the GDBA. The LDBAs monitor database size and growth and report current database layout and capacity for growth to the GDBA. The GDBA coordinates and controls any needed reorganizational activities.

No Autonomy. The GDBA is responsible for all functions described above which were done by both the GDBA and LDBAs under the semi-autonomy policy. The GDBA assigns priorities, monitors system performance, monitors database size and growth, executes database reorganization not only at the global level but also at the local system levels. It may be more difficult to monitor and understand local site performance characteristics.

Table 3. Comparison of Three Degrees of Site Autonomy

FUNCTIONS	FULL AUTONOMY	SEMI AUTONOMY	NO AUTONOMY
Database planning	Conflicting local site needs make planning difficult to arrive at global level.	LDBA coordinate local needs with global plan which was developed by GDBA.	Local needs may not be well understood and represented in global plan.
Database design	Local needs well understood. Difficult to develop global schema.	Local needs well understood. Easy to integrate local views into enterprise model.	Difficult to get local participation in design. GDBA may have difficulty in requirement analysis.
Develop and enforce standards	Difficult to coordinate. Tendency for conflicting standards.	Standards can be developed rationally (LDBAs propose, GDBA set standards). Local enforcement is relatively easy.	Difficult to enforce in local sites. Possible local sites' resiliences.
End-user support	Local user needs are directly supported by LDBA. Difficult to coordinate intersite access need.	GDBA develops plan for supporting end-user computing. LDBA directly supports user needs.	Difficult for GDBA to support local users' needs.
Conflict resolution	Difficult to resolve conflicts between local sites.	GDBA can easily resolve conflicts between LDBAs.	Easy to resolve but difficult to appreciate local grievance because of the absence of LDBAs.
Performance monitoring	LDBAs monitor local DB performance. No permanent group to monitor global performance.	GDBA monitors global level performance. LDBA monitor local DB performance.	Difficult to closely monitor local performance.
Technical support	Local support provided by LDBA.	Local support provided by LDBA. GDBA provide support to LDBAs.	Difficult to provide technical support to local users.
Training & Education	LDBA control programs. Could have varying levels of education & training.	GDBA develop consistent curriculum. LDBAs support them.	Difficult to do follow-up and continual education.
Data Dictionary /Directory	Very difficult to develop and maintain global DP/D.	GDBA develop global DD/D in cooperation with LDBA.	Easy to develop and maintain.
Documentation	Very difficult to achieve consistent & cohesive documentation.	LDBAs provide information. GDBA coordinate and control. Easy to achieve relatively consistent documentation.	Easy to achieve consistent documentation. Difficult to obtain local information.
Security & Privacy	Local enforcement of security policies developed by group consensus.	GDBA develop system-wide security policies. Local enforcement by LDBAs.	Difficult to enforce system-wide security policy.
Backup & Recovery	LDBA ensures proper backup and recovery. Tendency of inconsistent procedure between sites.	GDBA develops system-wide procedures. LDBA ensures proper local backup and recovery.	Difficult to ensure that backup and recovery is done properly at local site.

4.3 Comparison of Three Degrees of Site Autonomy

The advantages and disadvantages of each of the three levels of site autonomy in distributed database administration are summarized in Table 3. This table can be used along with Table 1 (Selecting the Degree of Site Autonomy) in providing guidelines for selecting a proper site autonomy policy.

5. CONCLUSION

Distributed databases may differ very much in the degree of site autonomy: from full autonomy without any centralized global database administrator to no autonomy with completely centralized control. In this paper, the degree of site autonomy was classified into three different levels: full autonomy, semi-autonomy, and no autonomy.

When local sites have full autonomy, all DBA functions are performed by inter-site coordination between LDBAs. On the other hand, under the no site autonomy policy, all DBA functions are centralized and the GDBA has sole authority and responsibility. Between these two extreme cases, local sites can have semi-autonomy. In this case, most of the routine DBA functions which are independent of other sites are performed autonomously by LDBAs Some DBA functions, which affect the global system's efficiency and effectiveness, are performed and controlled by the GDBA.

The selection of the appropriate degree of site autonomy in a distributed database environment is critical. It depends on the structure and culture of the organization along with information flow and needs. The advantages and disadvantages of different site autonomy policies were compared. The semi-autonomy policy is relatively superior to the two other more extreme cases, especially when designing a new distributed system. However, when existing independent databases are to be integrated into one distributed database, the autonomy policy may work better. An attempt has been made to provide guidelines for selecting an appropriate degree of site autonomy. The organization of the DBA function and the assignment of tasks to both global and local DBAs under different autonomy policies has also been discussed.

6. REFERENCES

Anthony, R. *Planning and Control Systems: A Framework for Analysis*. Cambridge: Harvard University Press, 1965.

Bray, O. H. *Distributed Database Management Systems*. Toronto: Lexington Books, 1982.

Ceri, S., and Pelagatti, G. *Distributed Databases, Principles and Systems*. New York: McGraw Hill, 1984.

Haas, M.; Selinger, P. G.; Bertino, E.; Daniels, D.; Lindsay, B.; Lohman, G.; Masunaga, Y.; Mohan, C.; Ng, P.; Wilms, P.; and Yost, R. "R*: A Research Project on Distributed Relational DBMS." *IEEE Database Engineering*, Vol. 5, December 1982.

Kahn, B. K. "Some Realities of Data Administration." *Communications of the ACM*, Vol. 26, No. 10, October 1983, pp. 794-799.

King, J. L. "Centralized versus Decentralized Computing: Organizational Considerations and Management Options." *Computing Surveys*, Vol. 15, No. 4, December 1983, pp. 319-349.

Walker, H. M. "Administering a Distributed Data Base Management System." *ACM-SIGMOD Record*, Vol. 12, No. 3, 1982.

Weldon, J. *Data Base Administration*. New York: Plenum Press, 1981.