Administering a Distributed Data Base Management System

by

Henry M. Walker[1]
Bell Laboratories, Piscataway, N. J.

Abstract

The early design of a Data Base Management System (DBMS) must include a specification of the functions associated with administering the system, so that subsequent activities, such as documentation, training, and staffing, can occur. Therefore, this paper undertakes a careful definition of administrative tasks for a Data Base Management System (DBMS) in a distributed computer environment. The paper first discusses a general framework for administering a distributed DBMS, and then the paper presents a specific listing of administrative functions. While the above work is based on a specific system under development at Bell Laboratories, the paper concludes with several comments extending this work to other types of distributed systems.

1. Introduction

The early design of a data base management system (DBMS) must include a specification of the tasks involved with administering the system, so that subsequent activities, such as documentation, training, and staff planning, can occur. Typically, these responsibilities involve several areas, including

- communication with user, management, operations, and maintenance groups
- monitoring hardware and software
- testing
- security
- backup and recovery
- performance

---

1. The author is on leave from Grinnell College, Grinnell, Iowa, where he is an Associate Professor of Mathematics.

- training

- documentation

- scheduling

Each of these topics includes several specific functions. In a simple DBMS, these tasks are rather well known and well understood. However, when a DBMS is deployed on a distributed system rather than on a single processor system, then these functions become more complex, and some additional coordination functions must be considered.

This paper undertakes a careful definition of administrative tasks for a DBMS in a distributed environment. The majority of this work is based on an actual distributed computer system now under development at Bell Laboratories, and some features presented reflect the specific nature of this system. Section 2 discusses some basic characteristics of this distributed DBMS, and Section 3 describes the basic approach to administration resulting from these characteristics. Then, Section 4 presents a specific listing of functions and responsibilities for administering this type of DBMS. Finally, Section 5 contains several remarks allowing much of this work to be extended to a wider variety of distributed DBMSs.

This work has been greatly aided by many discussions and supporting materials involving numerous people at Bell Laboratories and at Illinois Bell and Michigan Bell. The author wants to acknowledge the considerable help received from E. V. Cracovia, D. Seefelt, N. Sharko, and P. J. Toal. The initial seed for this paper was planted by R. L. Martin and D. S. Watson. Finally, L. Bernstein and D. L. Casulli have given considerable support and encouragement for work on administration.

2. System Characteristics

Distributed computer systems may exhibit many different architectures and characteristics. The next three sections focus on a particular type of distributed system, so this paper can focus on specific administrative tasks in some real systems.

The significant characteristics of this distributed system, from the standpoint of administration, are described here.

- Several distinct component subsystems are tied together by a communications network and connected to numerous terminals through the same network.

- Each component has its own unique features, such as distinct
    - hardware
    - operating system
    - application programs
    - data base structure
    - DBMS utilities and procedures
- Data is distributed into component data bases.
    - Component data bases have very different sizes and complexities.
    - Some data items are duplicated in the data bases of more than one component.
    - Some data items are specific to each individual component.
- Processing a user request may require work in one or in several components.
    - Each component can process only certain specified transactions, and each limited transaction can run only on one component.
    - Flows between components are completely specified in system requirements for when a user request requires multiple transactions to be satisfied.
    - When a user initiates a request in one component, that component tracks any required processing flows in other components.
- The communications network is used for other computer systems as well as this distributed system.
    - Terminals in the network may log onto any of several large systems.

Thus, this distributed system involves several different processors and application components tied together into a unified package.

3. Basic Approach to Administration

This approach to the administration of this distributed system depends on two observations and related conclusions.

- The distinction between system level responsibilities and component responsibilities.

- The relationship between the administration of the communications network and the administration of this distributed system.

This section discusses each of these areas in turn.

3.1 System-Level versus Component Responsibilities

In administering this distributed system, most functions and procedures common to any DBMS, such as performance monitoring and integrity checking of a data base, must be performed. Further, since the components have widely differing characteristics, the administration of each component requires mastering a specialized set of procedures and utilities. However, in addition, administering this distributed system includes some further tasks, such as data base synchronization and component coordination, which are due to the distributed nature of the system. Therefore, this specification of administrative functions and responsibilities considers both System-Level Responsibilities and Component Responsibilities.

System-Level Responsibilities include those tasks requiring coordination among components as well as those tasks which do not lie solely within one of the components. Thus, System-Level Functions set system-wide policy and coordinate inter-component activity. However, these responsibilities do not include managing those tasks which lie totally within one of the components.

Each component system, on the other hand, has a set of Component Responsibilities which involve the tasks of managing the operations of that component. These functions expand the tasks performed by a traditional Data Base Administrator or Manager in small, non-distributed systems. Thus, Component Responsibilities include many customary data base management activities. However, in addition, these Component Functions include implementing the system-wide policies set on the System-Level as well as supervising the tasks local to the particular component.

3.2 The Communication Network and This Distributed System

In Section 2, it has already been noted that the communications network used with this distributed system is also used for other systems. In this situation, no one system can have control of the network as the network must serve each system adequately.

Therefore, this paper assumes that the communications network will be administered separately from this distributed system itself. Of course, since the network and this distributed system interact, there will be necessary interactions between the administration of each group. Thus, administration of the distributed system is not responsible for any of the specific internals of the communications network, beyond any interfacing requirements.

4. Specific Functions and Responsibilities

The System-Level Responsibilities for administering this distributed system parallel the Component Responsibilities in many respects. Both include responsibilities in the areas of communications, network interfacing, hardware and software, testing, security, backup and recovery, performance, training, documentation, and scheduling. However, Systems-Level Responsibilities involve those areas that cross boundaries of the component systems, while Component Responsibilities include those tasks localized in a specific component. A detailed description of the responsibilities in each of the above areas is presented in parallel below.

System-Level Responsibilities	Component Responsibilities

4.1 Communications, including trouble reporting

a. Maintain channel of communication

Maintain a channel of communication with each component system and with outside parties, including the communications network, for the collection of reporting data, the transmission of recommendations, the distribution of system policies and information, and the reception of trouble reports, and in support of the application within the various disciplines. In this communication function, provide a single contact point for each user and user group. This function includes the single user contact for interfacing with the Communications Network.	Maintain a channel of communication with component users, operations personnel, and support personnel for the transmission of recommendations and policies and the reception of trouble reports.

b. Establish formal trouble reporting system

Establish a formal trouble reporting system whereby all errors crossing component system boundaries are reported to the administrators at the System Level. This reporting system must ensure that each user will have precisely one contact point for any difficulties encountered.	Expand the formal trouble reporting system, developed at the System Level for the overall system, to include all personnel involved with this component.

c. Analyze and route trouble reports

Analyze trouble reports to determine the probable cause (component subsystem, communications network, user group). Route trouble reports to the appropriate Component Administrator or user group for resolution.	Within the component, analyze trouble reports to determine the probable specific cause (hardware, software, data base). Route trouble reports to the appropriate component support personnel for resolution.

The trouble reporting system and the individual trouble reports must be consistent with hardware and software vendor policies.

System-Level Responsibilities	Component Responsibilities

Each trouble report must contain complete log and descriptive information, and each report requires a formal response to the report's initiator specifying the resolution of the problem.

4.2 Monitoring Hardware and Software

a. Monitor Status and Report failures

Monitor the status of all hardware and software in the distributed system, and report failures and resumption of operations to administrative personnel in all affected component systems.	Monitor the status of all component hardware and software, and assign personnel and/or vendors to make any needed repairs. Report all failures to all affected personnel, and report major failures to the administrators at the System Level.
Monitor the flow of transactions between components. (Whenever one component sends a request to another component, the response back should be matched with the original request to confirm successful inter-component flow).	Generate any transaction data required in the inter-component monitoring process.
Resolve problems by identifying their cause and referring them to the appropriate component or to the communications network.	Resolve problems referred from the System Level.

b. Plan and coordinate replacement and upgrading

Plan, coordinate, and control all hardware and software replacements and upgrades, including the addition or removal of any terminals using the distributed system.	Determine requirements for hardware and software replacement and upgrading within the component, including needs for additional data base storage media.
Final authority and responsibility for the configuration of all parts of the system lies on the System Level.	Report all configuration changes to the administrators at the System Level.
	Coordinate replacement and upgrading activities within the component.

System-Level Responsibilities	Component Responsibilities
c. Maintain hardware	
	Maintain the hardware associated with the component including central processor and peripherals, notifying vendors of any required activities.
4.3 **Testing**	
a. Perform Acceptance Testing	
Coordinate and control the testing of all new procedures, software, and hardware throughout the distributed system.	Conduct tests of all new procedures, software, and hardware, in consultation with the administrators at the System Level.
b. Schedule data base structural and consistency checking	
Regularly run procedures to compare data duplicated in the data bases of different component systems for consistency, and specify procedures for resolving any discrepancies found. The synchronization of the data bases in the various component systems is the responsibility of the administrators at the System Level.	Regularly run procedures to test each data base for structural errors, and correct any errors. Further, within the component, run programs to test duplicate items in the data bases for consistency and specify the procedures to be followed in resolving any differences. The data base management and editing utilities exist for the exclusive use of the administrators of the component.
c. Coordinate Accuracy Checking	
Coordinate efforts by all parties to achieve a high level of data base accuracy.	Periodically run programs to sample the data base to determine its accuracy. Work with users and with administrators at the System Level to correct problems encountered in data base accuracy.
d. Maintain internal tables	
	Maintain all tables used by the component software.

System-Level Responsibilities	Component Responsibilities

4.4 Security

a. Consult Management about security considerations and monitor security

Consult with Company Management concerning what user and system groups are authorized to access what records in the data bases. If different groups are to be given different capabilities, determine appropriate security classifications for each group, including data items which are to be accessible by each group. Further, maintain documentation to demonstrate compliance with Company Management security policies.	Within the component, implement security classifications determined by Company Management, monitor compliance with these measures, and report compliance data to those administering at the System Level. Log all attempted security violations.

b. Review operations procedures in light of protection of data bases

	Periodically inspect computer operations for proper procedures to assure the protection of each data base against inadvertent or accidental damage by users, by support, operations or maintenance personnel, and by other component systems.

c. Oversee physical protection of data base disks, tapes, etc.

	Institute procedures to insure the physical protection of the component data base medium (disk packs, tapes). Physical protection is the responsibility of those administrating the component.

System-Level Responsibilities	Component Responsibilities
4.5 Backup and Recovery	
a. Schedule backup procedures	
Coordinate scheduling of processes, such as backup and preventive maintenance, to minimize the impact of these activities from one component system on the operations of other component systems.	Schedule procedures for backing up the data base and for verifying the contents of any software tables.
b. Test effectiveness of backup procedures	
	Test backup procedures periodically to assure their effectiveness to restore each data base.
c. Develop and test contingency plans	
Develop and test contingency plans for failures in the system. This includes disaster planning as well as consideration of procedures in case of failure of one or more parts of the system.	Develop and test contingency plans for failures concerning the component system. Further, develop and test plans to minimize loss due to fire or natural disaster.
Develop and test degraded mode operation plans, including plans for operations when part of the network or when one or more components are down.	Develop and test plans for the operation of the component system when one or more other components are unavailable.
Develop and test plans to recover and resynchronize components following degraded mode operation.	Participate in the development and testing of procedures required when this component is unavailable.
	Participate in the development and testing of plans to recover and resynchronize components following this degraded mode of operation.

System-Level Responsibilities	Component Responsibilities
4.6 Performance	
a. Assign priorities	
Assign priorities to any activities, such as running transactions, generating reports, and performing special tasks, that compete for time and resources within the distributed system.	Within the component system, implement the activity priorities determined at the System Level.
b. Set installation and generation parameters	
Coordinate and control the setting of all installation and generation parameters in each component system to assure consistency across component systems.	Set all component installation and generation parameters, subject to consultation and review by the administrators at the System Level.
c. Monitor system performance, tuning the system as required	
Monitor system performance, collecting statistics on transaction volume, response time, and error rates and on hardware utilization. Further, seek input from system users to assess user satisfaction with the system's performance.	Monitor transaction, hardware, operating system, and application software performance. Send a summary of the performance data to the administrators at the System Level.
Coordinate and control any needed tuning.	Tune the system, adjusting appropriate system parameters, when necessary.
d. Monitor data base size and growth, executing data base reorganizations, as appropriate.	
Review data base performance. Coordinate and control any needed reorganization activities.	Monitor data base size and growth. Execute file expansion programs and perform data base reorganizations when necessary. Report current data base layout and capacity for growth to the administrators at the System Level.

System-Level Responsibilities	Component Responsibilities
c. Review activity and abnormal termination logs	
Review summaries of system activity logs and abnormal termination logs for each part of the system.	Review all component activity logs and all component termination logs. Prepare summaries of these logs for the administrators at the System Level.
4.7 Training	
Coordinate and schedule all training activities for personnel within the distributed system.	Coordinate and schedule the training of all operations, support, and administrative personnel for the component, after consultation with the administration at the System Level.
4.8 Documentation	
Compile a list of the release numbers of each software item running on each component system, and maintain a complete set of all documentation.	Compile a list of all component releases, and maintain a complete set of documentation related to the component, including vendor documentation.
4.9 Scheduling	
a. Regular activities	
Coordinate schedules for each of the following activities to be run on a regular basis. - data base integrity checking, including checking consistency of data across component systems - data base checking, including consistency of data stored within the system and that stored in external systems	Develop schedules for each of the following activities to be run on a regular basis. - data base integrity checking, including both structural checking and consistency checking within component data bases - procedures to verify contents of component tables - backup procedures - preventive maintenance - utility runs

System-Level Responsibilities	Component Responsibilities
b. Special Tasks	
Coordinate and schedule each of the following activities to be run as required. - cutovers to new software releases - installations of new hardware or modifications to existing hardware - special user tasks	Schedule required machine time for each of the following activities. - cutovers to new software releases - runs for special user tasks

5. **Conclusions and Extensions**

The previous sections of this paper have discussed the administration of a particular type of distributed data base system. This type of system had three basic characteristics.

- Several component subsystems with differing architectures.
- Data items split over the data bases of these components, with some data duplicated in several components and some unique to each component.
- User requests requiring transactions in one or more components.

Responsibilities for the administration of this distributed environment were divided into two general categories:

- System level administration focusing upon the coordination of the component subsystems.
- Component administration specializing on an individual component.

Then, each area of administration had parallel responsibilities on the system level and for each component.

This basic list of administrative tasks may now be adapted to a variety of other types of distributed systems. Two illustrations of possible modifications are presented below.

1. Some distributed systems may involve identical hardware and software running parallel transactions. In such an environment, separate component administration may be unnecessary. Instead, responsibilities may be organized by functional area, such as performance monitoring or data base checking, and one responsibility may extend over all of the parallel subsystems.

2. In other systems, components may have a common hardware and operating system; application programs may differ from one subsystem to another, but these applications operate in the same support environment. In this setting, system-level coordination and responsibilities may remain as in Section 4. However, here, subsystem responsibilities may be divided into those related to the common execution environment and those specific to each application. This yields three parallel sets of functions, including system-level, environment, and applications. With this division, many coordination activities are simplified as responsibility for each subsystem's hardware and operating system is brought together.

In any of these types of distributed systems, many of the same functions must be performed. Thus, the basic list of functions and responsibilities presented in this paper can be useful in determining the administrative tasks required in many types of distributed systems.