DATA MANAGEMENT OF TELECOMMUNICATION NETWORKS

Kiyoshi Ono, Mikio Aoyama, and Hiroshi Fujimoto*

Telecommunications Software Division

FUJITSU LIMITED

629 Shimo-kodanaka, Nakahara-ku, Kawasaki 211, JAPAN

1. INTRODUCTION

As information diversifies and greatly increases in quantity, global Telecommunication Network Systems (TNSs) are becoming increasingly important. Thus, the information managed in TNS is rapidly increased in scale and becoming more and more decentralized. In a conventional TNS, the data management mechanism was relatively simple since the system emphasized massive traffic processing capability. However, as telecommunication service features become more sophisticated, data management of TNS does more important and costly. Hence, the modeling and analysis of data management of TNS are mandatory for better management, maintenance, and operation of TNS as well as for improving the development productivity and quality. This article proposes a new field of data management which is inter-disciplinarily incorporating the telecommunication and database (DB) technologies.

2. TELECOMMUNICATIONS DATA MANAGEMENT

2.1 Two Models of Data Management

The following two models can be used for data management in TNS.
1) *Centralized management* at a particular node, management center.
2) *Distributed management* at the respective nodes, i.e. switches.

We focus on the distributed data management since only a small amount of data is centrally managed in actual networks, and information resources tend to become more and more decentralized.

2.2 Data Management Architecture

The data in TNS can be considered as hierarchical distributed data in two layers: network and node. Thus, the data management of TNS can be modeled by a two-layer *Distributed DB Management System (DDBMS)*.
1) *Network layer*: Regarding the respective switches in TNS as a DBMS, the TNS itself can be seen as a loosely-coupled large-scale DDBMS.
2) *Node layer*: We propose a new architecture. It consists of a *Call Switching Processor (CSP)* and a *Maintenance and Administration Processor (MAP)*, centered around a DB. The CSP processes real-time switching features and needs high-speed data access. The MAP provides a high-level interface with maintenance personnel.

2.3 Switching System Architecture

Modern switching systems adopt multi-processor distributed architecture. We have developed a switching system of two-layer function and load distributed architecture [1]. It consists of a number of *Call PRocessors (CPRs)* made up of multiple load distributed processors and a *Management PRocessor (MPR)* providing the total system administration. Functions of the CSP and MAP correspond to the CPR and MPR, respectively. The CPR and MPR incorporate a DDBMS in a switch.

3. DATA MODEL AND DATABASE ARCHITECTURE

3.1 Abstract Model of Data

For the modeling of a DB structure in TNS, we have adopted the three-layer schema proposed by *ANSI/SPARC*. The logical structure corresponding to the respective layers is defined as follows:
1) *External schema* corresponds to the DB of respective switches.
2) *Conceptual schema* defines the generic DB structure for all the DBs in a family of switching systems.
3) *Internal schema* corresponds to physical data structure.

Due to the complexity of data structure and real-time operability, access through external schema is not always practical in switching systems.

* Aoyama is currently visiting University of Illinois at Chicago.

Therefore, the system is designed to accommodate DB access in either layer by opening limited portion of the three-layer to some specific application through standardized interface.

In order to guarantee the integrity of data during the access to network layer DB, an interface to the network layer DB is implemented in the DDBMS to synchronize the accesses among the nodes.

3.2 Data Classification by Access Method

Data can be classified into two types from the viewpoint of dynamics:
a) *Static data*: Data usually does not change except for the system configuration change, such as terminal attributes and hardware configuration, and
b) *Dynamic data*: Temporal data such as status of terminals and system.

Considering access speed, the dynamic data can be further classified into:
1) Dynamic data of real-time read/write, and
2) Dynamic data of real-time read and non real-time write,

For the data requiring real-time access from the CSP, high-speed access is achieved by allocating the data replication in both the MPR and CPRs.

3.3 Distributed Database Management

Concurrency control and *failure recovery* must be considered.
1) Concurrency control: A locking technique has been implemented.
2) Failure recovery: Hardware and software techniques are incarnated.

4. DATABASE GENERATOR: SYDGN

4.1 Requirements

The requirements of the generation method are: 1) A high-level *DB Description Language (DDL)*, 2) Supporting the three-layer model description, 3) Handling a family of DBs from small-scale to large-scale, 4) Supporting the multi-processor distributed architecture, and 5) Implementing the method into an integrated environment.

4.2 Data Definition and Generation Method

We have been using a DB generator by the following procedures.
1) Generic DB definition by using DDL. The data allocation to the respective processors is also specified.
2) Instantiation: DB instances are generated by *SYDGN (SYstem for DB GEneration)* from the generic DB definition and instantiation conditions for the respective DBs. SYDGN also generates the replication. Designers can reuse the data definitions among different DBs.
3) Together with application software, the DB generated in (2) is integrated into a complete system for a node.

4.3 Logical Design of Databases

From the semantic viewpoint, we classified the data into three categories:
1) *System data* is defined from the system configuration. It is independent of respective customer requirements.
2) *Office data* is based on requirements to respective switches (offices),
3) *Terminal data* specifies the individual terminal attributes.

The DB generation by SYDGN covers from the skeleton design of DB to the generation of the system data and skeleton of the office and terminal data. Therefore, designers can specify two types of instantiation parameters: one for the definition of the DB skeleton and the other for detail definition of the office and terminal data. The contents of the office and terminal data are defined at the user site.

4.4 Coordination with Data Access Methods

In order to support multi-processor architecture, we have implemented automatic generation of access methods corresponding to the three-layer schema at the time of DB generation. The idea is that, if the application access method can be automatically generated at the time of compilation of DDL by incorporating the interface descriptions of the three-layer access methods, we can reduce the efforts for application development, and data independence would be improved.

5. FUTURE WORK

Data management is the focal point of the management of TNSs, and an appropriate data model reflecting TNSs is necessary. The exploration of such model and management techniques is an urgent issue for the sound evolution of TNSs and requires further research.

REFERENCES

[1] T. Ogawa et al., "A New Generation PBX for Integrated Office Services," *Proc. of IEEE Int'l Switching Symp.*, Mar. 1987, Phoenix, No. B.7.2.