RODAIN: A Real-Time Object-Oriented Database System for Telecommunications

Juha Taina
University of Helsinki
juha.taina@cs.helsinki.fi

Kimmo Raatikainen
University of Helsinki
kimmo.raatikainen@cs.helsinki.fi

Abstract

Future telecommunication services will extensively exploit database technology. The persistent and temporal information needed in operations and management of the telecommunication networks and services will be in databases. The current Intelligent Network (IN) Recommendations of ITU-T imply that real-time transaction processing capabilities should be provided. Telecommunications Management Network (TMN) and Telecommunications Information Networking Architecture (TINA) are object oriented. Current and future mobile telephone systems have their own requirements for database services. An ideal database system supporting various telecommunications applications should be a fault-tolerant distributed real-time object-oriented database system.

In the research project Rodain the ultimate objective is to design and to specify a real-time object-oriented database architecture for telecommunications applications and to implement a prototype based on that architecture. The design and specification take place in 1996 while the prototype will be implemented in 1997.

1 Introduction

Databases will already in the near future have a central role in telecommunications networks. The information needed in operations and management of the nets will be collected into a logically uniform database. The world-wide nature of telecommunications prescribes that the only possibility to obtain the logical uniformity is the co-operation of autonomous databases.

During the next ten years great changes are expected in the ways of doing business on the telecommunications markets. The relative importance of traditional transmission and switching services will decrease. This is due both to liberalization of telecommunications and to developments in the technology. We anticipate that in the future transmission and switching are done using generic technology based on international standards.

The ability of a telecom operator to survive in the international competition will base either on huge volume of the few bulk services or on enhanced services. The ability of introducing enhanced services will primarily be based on the competence of responding in time to the needs on the markets. Future telecommunication architectures like IN and TINA are developed to meet the goal of rapid and efficient creation, deployment, and management of new services. An efficient and reliable database system is one of the key factors in meeting those goals.

Most of the recommendations issued in the 90's by ITU-T and other standardization bodies are object oriented. Therefore, object orientation is a natural starting point for the telecommunications database system, too. However, current object-oriented databases are primarily designed for situations where objects have complex structures and transactions are long-living. In telecommunications the situations are different. Most objects have quite simple structures and transactions are short. In addition, we anticipate that real-time properties, guaranteed response time in particular, are soon needed.

2 Key Requirements for Databases in Telecommunications

In the research projects Darfin[1] and RODAIN[1] we have examined database needs in telecommunications. Based on our results and experience the most important features will be the *real-time* and *object-orientation*. Real-time transactions are really needed in telecommunications. In telecommunications we will need both *soft transactions* that may continue their execution after deadline but with a reduced lower priority and *firm transactions* that are terminated when their deadline expires. We do not believe that *hard transactions* will be used in near future because systems supporting hard transactions are too expensive for open telecommunication markets.

Below we summarize the key requirements while detailed discussion can be found in [RT95].

[1] Darfin and RODAIN are acronyms of the project names *Database ARchitecture For Intelligent Networks* and *Real-time Object-Based Database Architecture for Intelligent Networks*, respectively.

Data distribution: A set of database nodes may cooperate to accomplish database tasks. A node may request information from other nodes in case information needed is not locally available. It is possible to implement the underlying database system without data distribution. In such an implementation all applications use a single database server. This differentiates logical data distribution from physical data distribution among database nodes. The former is necessary but the latter is a decision internal to the design of the database architecture. Our current belief is that only a few requests need access more than one database node. We also believe that the request distribution between different database nodes can be arranged to be almost uniform. Thus using a distributed architecture allows several transactions to be run in parallel on several database nodes and give better throughput.

Data replication: It is stated on [ITU94a] that a database node contains customer and network data for real time access. This implies that the underlying database architecture must be able to answer requests in real time. We believe that currently most distributed operations are reads. Therefore, replication is an effective way to speed up distributed operations because it gives better throughput.

Object orientation: It is a common belief that the best long term telecommunications architectures are object oriented. The implication is that a database system must support object access. In other words, the database system must either be object oriented or have an object interface.

Object directories: The conceptual model of IN Service Control Function [ITU94b] defines a Service Data Object Directory that is used to access data stored in the SDFs. This implies that object directories must be supported. The invocation mechanism of CORBA is a good alternative to implement the functionality.

Multiple application interfaces: Different telecommunications architectures define different access interfaces. The IN Capability Set 1 defines IN Application Protocol (INAP) [ITU94c]. TMN has its own access methods based on the OSI (X.700) management. Agents and managers exchange information using the services of *Common Management Information Service Element* (CMISE). CMISE uses the *Common Management Information Protocol* (CMIP) and utilizes the services of *Association Control Service Element* (ACSE) and *Remote Operations Service Element* (ROSE). The current definition in TINA is based on TINA ODL which is quite similar to the OMG IDL interface.

Fault tolerance: Real time access implies that data must be continuously available. The result of the implications is that the database system must be fault tolerant. In the current definitions the maximum allowed down time is a few seconds in a year. However, the actual allowed down time may be minutes instead of seconds. Nevertheless, a fault tolerant database system is a necessity.

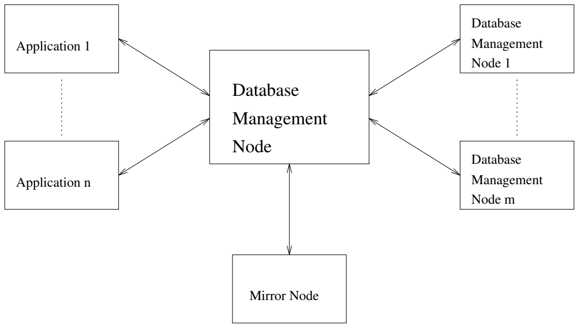

Figure 1: Overview of the RODAIN Database Architecture

Real-time transactions: A real time transaction has an explicit deadline that the transaction must meet. Although the real time data access as stated in [ITU94a] does not directly imply that the underlying database system must support real time transactions, we believe that the most convenient way to support real time access to data is to use a real-time database system.

3 RODAIN Database Architecture

The RODAIN database architecture is designed to fulfill database needs of future telecommunications applications. The architecture It is a *real-time object-oriented database management system architecture*. It has three primary levels: global architecture, node architecture, and object-oriented database management system. Below we briefly summarize each of the levels. Detailed description can be found in [TR96].

RODAIN DBS consists of a set of autonomous database nodes that interact with each other. Each node may communicate with one or more applications, and an application may communicate with one or more database nodes. Each node is replicated into a mirror node (Figure 1).

A RODAIN DBMS node consists of a set of subsystems that communicate with each other. The subsystems are:

1. *User Request Interpreter Subsystem*,
2. *Distributed Database Subsystem*,
3. *Fault-Tolerance and Recovery Subsystem*,
4. *Watchdog Subsystem*, and
5. *Object-Oriented Database Management System*;

see Figure 2.

The *Object-Oriented Database Management System* (OO-DBMS) consists of a set of database processes that use

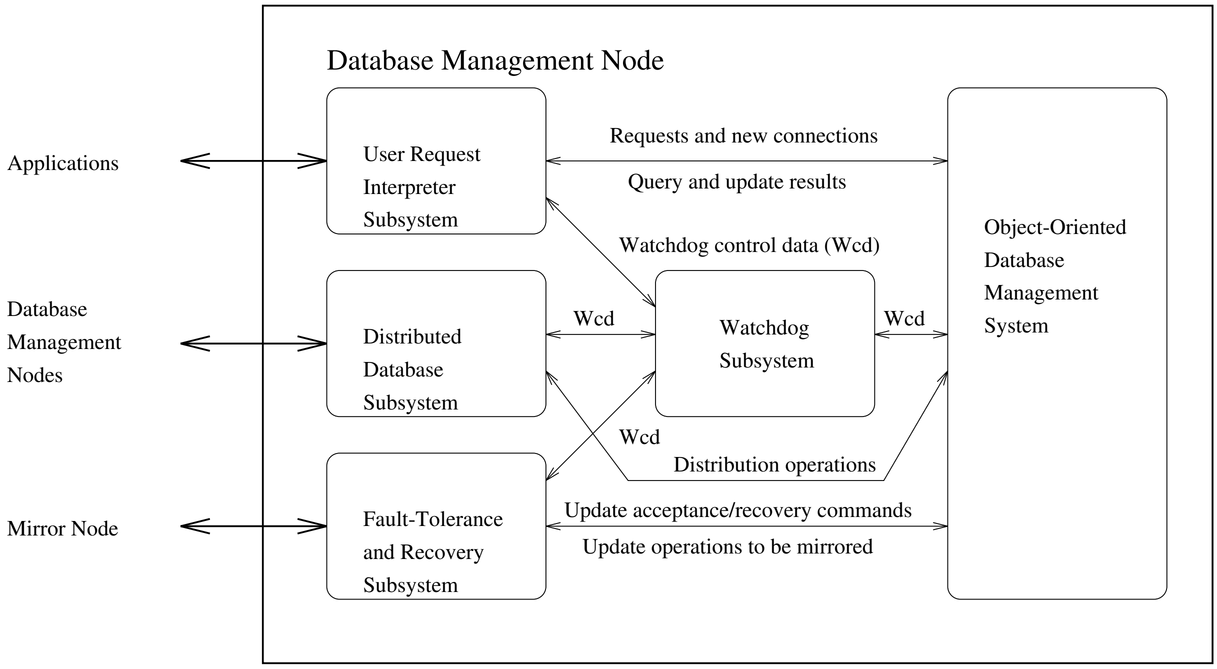

Figure 2: RODAIN Database Management Node

database services to resolve requests from other subsystems, and a set of manager services that implement database functionality. The functional structure of RODAIN OO-DBMS is shown in Figure 3.

The manager services are divided into four layers depending on their nature. A service in a layer uses services in the next lower layer. Some services in the lowest layer are used by all other layers. The layers are *Database Interface Layer*, *Global Entity Layer*, *Object Layer*, and *Real-Time Core Layer*.

The **Database Interface Layer** is the interface layer for the database processes to the database services. It offers Schema information services for metadata use and maintenance, Object information services for object oriented interface, and Directory services for directory maintenance.

The database architecture should be flexible due to different kinds of needs present in telecommunications. We believe that the required functionality is the easiest to implement with an object based approach. We think that it is better to use a reduced set of object features than to use a fully object-oriented database when we are developing a real-time database to be used in telecommunications.

The principal item in Darfin real-time object model is an object. It can be either a regular object or a real-time object. Objects have identity and they belong to a known type. A real-time object has predefined attributes for isolation level and access type. Different isolation levels allow transactions to reduce serializability when referenced objects allow it. Access type restrictions allow transactions to be optimized for certain types of objects.

Real-time objects are referenced by real-time transactions. We want to support two kinds of real-time transactions: 1) *soft transactions* that may continue their execution after deadline but with a reduced lower priority and 2) *firm transactions* that are terminated when their deadline expires.

A real-time transaction is also a real-time object, either persistent or transient. It has a deadline and transaction type. A transaction may suggest its isolation level to other transactions. In case of isolation conflict the higher isolation level is chosen. Isolation levels between objects are also supported. For instance, a transaction that runs at a low isolation level does not care if an object is accessed by another transaction. However, it will not get access to the object if a transaction demanding a higher isolation is accessing it. Similarly, if two low isolation level transactions want to access an object that has a high isolation level, the latter transaction must wait for the first transaction to release the object.

The **Object Layer** offers the actual object implementation services: Physical Object Manager service and Dynamic Linking Manager Service. The Database Interface Layer managers use the services to implement the visible object model. On this layer the chosen object model is no longer visible.

The Physical Object Manager Service is the core manager for objects and object operations. It offers services for object creation, object deletion, object access, object attribute access, and object operation access. It uses the global entity layer when a replicated operation occurs, or when it cannot resolve an object request on the local database. It

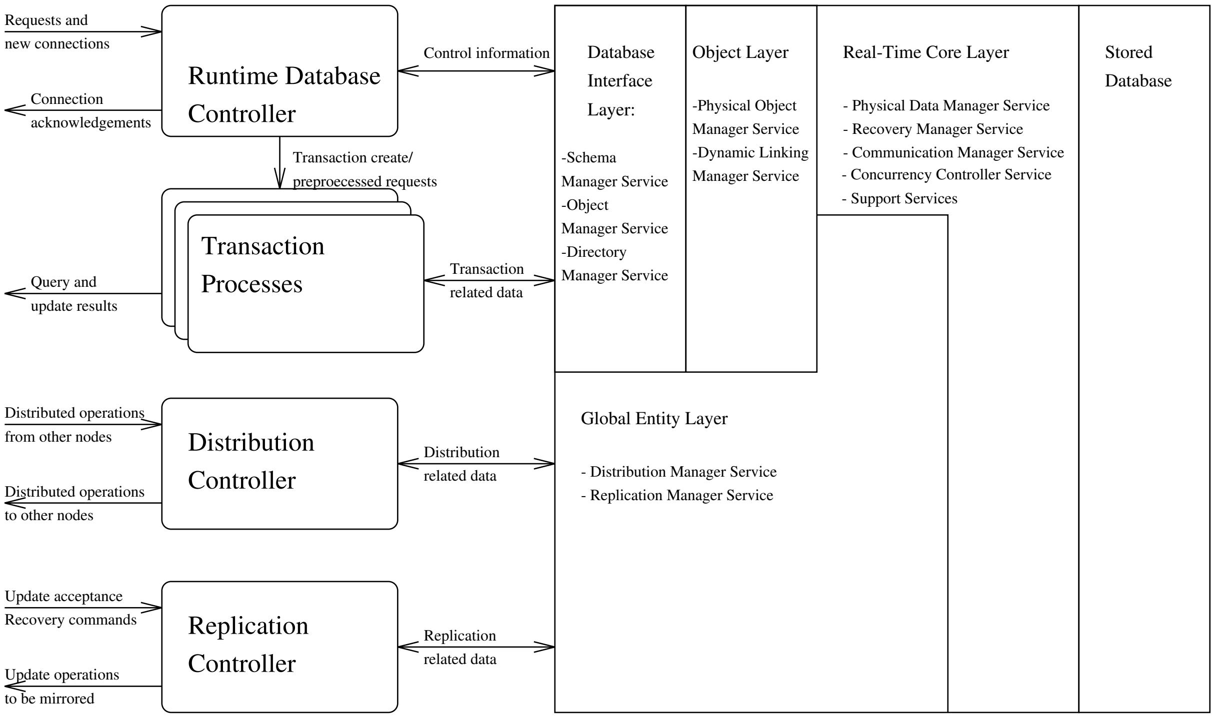

Figure 3: Processes and Database Services in RODAIN OO-DBMS Subsystem

uses the real-time core layer services to implement object functionality.

The Dynamic Linking Manager Service offers services to dynamically link new object operations into the object database. It uses the services of the real-time core layer to implement object operations.

The **Global Entity Layer** offers services for other layers and special controller processes to support distribution and replication in the OO-DBMS. It consists of Distribution Manager Service and Replication Manager Service.

The Distribution Manager Service is used by the Distribution Controller process when it needs database services. It is also used by other services when an unresolved database operation occurs. An unresolved database operation triggers the service which then informs the Distribution Controller to take control of the operations.

The Replication Manager Service is used by the Replication Controller process when it needs database services. It is also used by other database services when a database update operation that should be mirrored occurs.

The **Real-Time Core Layer** implements the core operations of real-time database management system. This lowest layer consists of Physical Data Manager Service, Recovery Manger Service, Communication Manager Service, Concurrency Controller Service, and Support Services. Of the services the Communication Manager Service and Support Services are used on all other layers.

The Physical Data Manager service is the only interface to the physical data in the database. It is the only manager that has knowledge of physical data locations in the database. It offers services for data storage, retrieval, and access estimates.

The Recovery Manager Service is the first service to gain control when the Fault-Tolerance and Recovery Subsystem has started OO-DBMS recovery operations. It restarts database processes and offers services to the Replication Manager Service to to return data integrity and lost data. The Communication Manager Service offers services to the other managers to maintain communication channels to the other subsystems.

The Concurrency Controller Service allows transactions to run in parallel. It offers control services to other database managers and to the Real-Time Scheduler. It may also awake the Real-Time Scheduler when a conflict occurs. Concurrent transactions do not necessarily need serializability to produce a correct result. In many situations the result of a transaction can be regarded as correct as long as the result corresponds to the real-world situation presented in the database even if the transaction does not serialize with other transactions. The concurrency controller supports different serialization and isolation levels, as described in the real-time object model. An interesting correctness criteria, which we call τ-*serializability* [RKMT95], can be used to reduce the number of serialization conflicts. In τ-serializability a transaction is allowed to read old data as long as the update is not older than a predefined time.

4 Key Issues

The database system should be flexible due to different kinds of needs present in telecommunications. We believe that the required functionality is the easiest to implement with an object based approach. An open question is if a fully object-oriented architecture—something similar to ODMG-93 [Cat94]—is needed. Currently we believe that it is better to use a reduced set of object features than to use a fully object-oriented database when we are developing a real-time database to be used in telecommunications.

Another interesting area is correctness. In real-time applications concurrent transactions do not necessarily need serializability to produce a correct result. In many situations the result of a transaction can be regarded as correct as long as the result corresponds to the real-world situation presented in the database even if the transaction does not serialize with other transactions. A new interesting correctness criteria, which we call τ-*serializability* [RKMT95], can be used to reduce the number of serialization conflicts. In τ-serializability a transaction is allowed to read old data as long as the update is not older than a predefined time.

We can gain several advantages when we combine real-time and object orientation. Data objects can have attributes that specify the correctness criteria. Operations can have attributes that tell the resource consumptions of the operations, what kind of locks they needed (read, update, or replace). Transactions are also objects; either transient or persistent. They can have attributes that specify the priority, deadline, criticality, correctness criterion, among other things.

Acknowledgements

This work has been carried out in the research projects Darfin (1994–5) and RODAIN (1996–). The former was funded by Telecom Finland and the latter by the Finnish Technology Development Center (TEKES) together with Nokia Telecommunications, Solid Information Technology, and Telecom Finland.

References

[Cat94] R. G. G. Cattell, editor. *The Object Database Standard: ODMG-93*. Morgan Kauffmann, San Francisco, Calif., revision 1.1 edition, 1994.

[ITU94a] ITU-T Recommendation Q.1204. *Intelligent Network Distributed Functional Plan Architecture*. International Telecommunications Union, Geneva, Switzerland, 1994.

[ITU94b] ITU-T Recommendation Q.1214. *Distributed Functional Plane for Intelligent Network CS-1*. International Telecommunications Union, Geneva, Switzerland, 1994.

[ITU94c] ITU-T Recommendation Q.1218. *Interface Recommendation for Intelligent Network CS-1*. International Telecommunications Union, Geneva, Switzerland, 1994.

[RKMT95] K. Raatikainen, T. Karttunen, O. Martikainen, and J. Taina. Evaluation of Database Architectures for Intelligent Networks. In *Proc. TeleCom95 – Technology Summit*, Vol 2, pages 549–553, International Telecommunications Union, Geneva, Switzerland, 1995.

[RT95] K. Raatikainen and J. Taina. Design Issues in Database Systems for Telecommunication Services. In *Proc. of IFIP-TC6 Working Conference on Intelligent Networks*, pages 71–81, TeleDanmark Research, Copenhagen, Denmark, 1995.

[TR96] J. Taina and K. Raatikainen. An Experimental Real-Time Object-Oriented Database Architecture for Intelligent Networks. *Int. J. of Engineering Intelligent Systems for Electrical Engineering and Communications*, 4(3):57-62, September 1996.