Implementation of A Multimedia Resource Mangement System Based on Distributed Objects

5T-1

Qianshan He and Hidehiko Tanaka
Department of Electrical Engineering, University of Tokyo

1 Introduction

Developments in workstations resently make it possible for an user to manipulate or display data of different media types (image media, text media or graphic media) by using a single workstation. Nowadays, sharing of the multimedia data in a distributed environment is available and practical. In a distributed environment, because the numbers and types of these resources available to users are being expanded, users may have troubles to access and manipulate these resources. It is necessary to use an uniform method to integrate different media types and provide some convenient facilities for users to use resources. In [1] and [2], to solve the problems above, we have proposed a distributed system based on object-oriented service base model. This paper details the structure of object-oriented service bases and discusses the implementation issues of the system.

2 Structure of The Object-Oriented Service Bases

In our system, resources (multimedia data, programs, functions provided by computers) are abstracted as objects. A service base in each node is constructed to manage these objects and co-operate with other service bases. In a service base, we use three layered views [1] to manage distributed objects, and a directory to classify objects. The structure of a service base in one node is shown in Fig. 1.

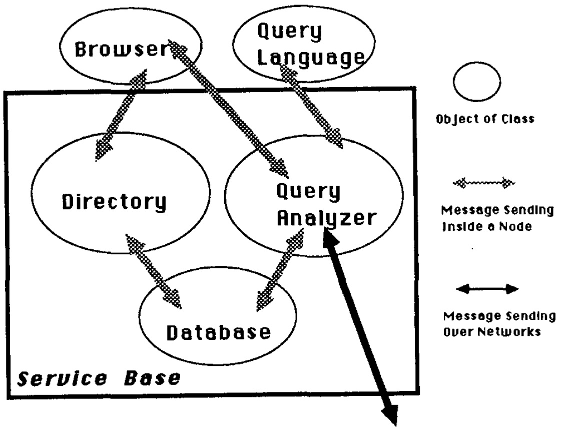

Figure 1: Structure of a service base

Directories A directory is a collection of classes which contain attributes information (for example, name, location, media and formate) of the distributed objects. In the directory, object classes are defined to classify the objects which have the same characteristics and attributes. For each object class, the operations which can be applied to this object class are also defined. The attributes of an object are expressed by instance variables of the object. Fig. 2 shows an example of classes in the directory. In this example, C++ language is used to define the class.

Directories follow a tree structure in which classes are hierarchical. We suppose directories in different nodes may have their different structures according to their environment. This enriches the extensibility of each node, and make our system applicable to different environments. On the other hand, we define some minimum standard to ensure the the whole system can cooperate with each others. This means that we must define some classes and attributes which are common to variable objects. In an actual implementation of the system, classes and attributes which are not defined in the standard can be added to a directory to make the system suitable to certain applications. For example, if a new type of media is to be added to the system, then a new class for that specific type should be defined.

A directory class is only a description of one kind of resources. To register a resource in the service bases and make it available to network users, a class instance which contains the specific attribute values of the resource is produced and stored in a database.

Query Analyzer When an user or an application program makes a request to the system, query analyzer searches the database to get the

Implementation of A Multimedia Resource
Mangement System Based on Distributed Objects
Qianshan He and Hidehiko Tanaka
University of Tokyo

attribute information of the requested objects. Basing on the attributes of the objects, query analyer in a service base works as follows: (1) *Location checking:* If some of the requested objects are remote objects, a message is sent to the node where the objects are located, then the objects are fetched. (2) *Semantics checking:* Before the requested objects are combined, the semantics of each object is checked. For example, to display an image data object, a window object which can display image rather than text should be used. (3) *Combination:* The combination of objects means that query analyzer combines some existing services to meet the needs of the user's request. (4) *Execution:* If the operations on objects is local, query analyzer forks processes to execute the operations.

3 Implementation Issues

We are implementing an experimental system. Our experimental system contains a number of SUN3, SUN4 and VAX workstations. The experimental system is written in C++ language.

Database An object-oriented database is expected to be used to store the registered objects. But in our experimental environment, object-oriented database in each node is not available. Therefore, we decide to use a relational database called INGRES. To use a relational database, the main problem is focused on how to map the directory schema to the schema of the database. To represent the directory schema which is a tree structure in a normalized relational form will need a lot of relations. To solve this problem, we use unnormalized form to represent the directory. One relation is used for one directory class and several relations are used to keep the relationship of the directory tree.

Interface A standard query language C++ is provided as a standard interface for application programs to access objects managed by the service bases. Moreover, we provide a browser for users to manipulate or display objects.

The browser has text windows, image windows and graphic windows etc.. Each window of the browser is used to display or edit different media objects respectively. The selection of using which window to display or edit a multimedia objects will be made by the system automatically according to the attributes of the objects. To display a document which contains elements of different media types, the text window will be first opened and the text parts of the content are displaied. Each element which is not text is represented by an icon and a text caption. Selecting the icon will cause the element to be displaied in a window which can display media type of the element. The browser in our system is implemented by using the X-window system in SUN workstations.

```
class dirODAdocument : public dirRoot
{
    Attribute name;
    Attribute location;
    Attribute formate;
    Attribute mediaType;
    Attribute creatTime;
    Attribute owner;
public:
    void editingProcess();
    void layoutProcess();
    void presentationProcess();
    void multimediaMail();
}
```

Figure 2: Class example in the directory

For applications written in common languages (Fortran, Mudula-2, etc.), a library is provided. The library provides a way for common languages to access the service bases through standard query language C++.

4 Conclusion

Based on the model proposed in [1] and [2], we are implementing a multimedia resource mangement system. This paper details the structure of the system and discusses the implementation issues of the system. As an application example, we are considering to construct a message system which contains multimedia documents (in ODA standard of OSI) on top of the experimental system.

References

[1] Q. He and H. Tanaka, "A Distributed Environment Based on Object Oriented Service Base System", Multimedia Communication and Distributed Processing Confrence of IPSJ, 41-6, March 13, 1989.

[2] Q. He and H. Tanaka, "Object Oriented Distributed Program Environments", 38th National Conference of IPSJ, 7H-4, March, 1989.