SPIRIT: a Resource Database for Mobile Users

Noha Adly* Pete Steggles‡ Andy Harter‡

* Computer Laboratory
Cambridge University
Cambridge CB2 3QG, UK

‡ ORL
Olivetti & Oracle Research Lab
Cambridge CB2 1QA, UK

1 Abstract

Users of computer systems need continuous access to information and services, but, as they move around in a richly equipped, networked environment, the available hardware resources change; software systems must adapt to these changes, offering location-aware personalisation and control. The SPIRIT project (SPatially Indexed Resource Identification and Tracking) is building a resource database which integrates the configuration data for heterogeneous networked hardware and software with fine-grained location data for users and equipment, allowing software to dynamically configure and reconfigure itself as users move around the networked environment. The ultimate goal of the project is to make it seem to users as though an application knows as much about the physical environment as they do. This paper describes the architecture of the SPIRIT database and provides an overview of some of its client applications.

2 Introduction

The motivation for SPIRIT arises from three areas: our interest in producing mobile applications which need information about resource capabilities; our monitoring of the environment with sensor systems, which provide us with location information for resources; and our work with networking technologies and multimedia which makes ours a highly heterogeneous environment. Previous mobile applications developed at ORL have used a database to store a limited amount of resource and location information; as our requirements become more sophisticated, we need to store a wider range of resource data, reported by a large number of different devices and to integrate this with fine-grained location data. This paper describes the architecture of the system which we are designing to this end.

This paper is organised as follows. Section 3 gives an overview of the infrastructure at ORL, describing the hardware resources and the location techniques available as well as some of the mobile applications that are potential users of SPIRIT. Section 4 describes the general architecture of SPIRIT. Section 5 focuses on the design of the database server and the system integration issues. Section 6 briefly discusses the spatial monitoring and indexing techniques used. Finally, Section 7 concludes the paper and gives an overview of the current work in progress.

3 The Infrastructure at ORL

3.1 Hardware Resources

Some hardware resources are only useful when the user is colocated with them. These are of particular interest to mobile applications; the main hardware resources of this type at ORL include:

- **Computers** including laptop and desktop PCs, workstations, file servers and application servers.
- **Tiles**, which are ATM-based network computers incorporating a high-resolution display and a pen input system. Using ORL's radio ATM system, tiles can be carried around.
- **ATM multimedia peripherals** such as cameras, microphones, displays and loudspeakers.
- **Printers**, including monochrome and colour printers with various speeds and resolutions.
- **Telephones** which can be controlled using a CTI switch.

Detailed information about all these devices, including the location of each one, is stored in the database.

In addition, the status of the hardware infrastructure such as ATM switches, modems, routers, file servers and application servers is potentially of interest to all applications. Detailed information about these devices is stored in the database, though in these cases location information is less important than information about system availability and load levels.

3.2 Location systems

The environment in ORL is monitored by a range of sensors. The main system currently deployed is the infra-red based *Active Badge* system. This will be augmented in the next year by a finer-grained ultrasound-based system. These sensors provide the main source of location information for the database.

3.2.1 Active Badges

The Active Badge System[1] [HH 94] provides a means of locating individuals within a building by determining the location of their active badge. This is a small device which is worn by personnel; it transmits a unique infra-red signal every 10 seconds. A simpler version of the Active Badge has been designed for tagging equipment. Each office within a building is equipped with one or more networked sensors which detect these transmissions. The location of the badge (and hence its wearer) can thus be determined to a resolution of a few metres on the basis of information provided by these sensors.

3.2.2 Ultrasound sensors

ORL's ultrasound location system, which is currently under development, operates at finer granularity than the active badge system. It uses ultrasonic transmitters attached to people and equipment which can send signals several times a second. The signals are detected by an array of sensors in the walls and ceiling of a room, and the position of the transmitter is thus calculated. The current system locates the position of the transmitter to a granularity of less than 10 cm.

3.3 Applications

Work in the area of mobility and mobile systems has been an important aspect of ORL's research for the last five years. Several mobile applications currently implemented are relevant users of the SPIRIT database such as the Teleporting System and the Mobile Streams. These applications and systems use SPIRIT for personalisation and adaptation as they can share some of the users' perception of the environment.

3.3.1 The Teleporting System

The Teleporting System [Richardson 95, BRH 94] is a tool for experiencing an alternative approach to mobile computing in which application interfaces, rather than the computer on which the applications run, are able to move. It operates within the X Window System, and allows users to interact with their existing X applications at any X display. This *teleporting* is achieved by a *proxy X server* which is able to dynamically re-route the input and output of a set of X applications to any X display.

[1] Active Badge is a registered trademark of Ing. C. Olivetti & C., S.p.A

For teleporting suitable displays must be specified according to the current location of the user. Such information can be drawn from SPIRIT.

3.3.2 Mobile Multimedia Streams

The goal of the mobile multimedia streams project [MM 96] is to integrate continuous media, such as audio and video, into a mobile adaptive, distributed computing environment. The major sense of *mobile* relates to the mobility of the user. So for instance, a video stream would *follow* its user from room to room as s/he moves around the lab. The system is *adaptive* in the sense that the transport parameters of multimedia streams can be modified to take into account the processing, display or capture capabilities of the source, sink and intermediate objects in the data pipeline.

In order to select suitable end points for mobile streams, the system needs to know about the current location of the user and equipments as well as the capabilities of the resources surrounding him. This information is provided by SPIRIT.

3.3.3 Browsers

The location data provided by the badge system is useful for internal organisation purposes. It is possible to find where colleagues are and how to contact them by using a browser for active badge data such as *xab*.

3.3.4 Pangea

The Pangea project is producing a multi-processor application server, which is primarily intended to support the ORL tiles. With Pangea there is no one host machine for a session; when a request is made to launch an application an appropriate processor is chosen on the fly according to the current status of the processors and network. This information is collected from the processors and publicised by SPIRIT.

4 Architecture of SPIRIT

SPIRIT organises and makes available all kinds of information about the environment, including the location of people and hardware, and the capabilities and status of hardware and software accessible over the network. An overview of the architecture of SPIRIT is shown in Figure 1.

It is based around an *Oracle7* [Oracle 92] database which stores all the relevant information describing hardware and software resources as well as people. We rely *Ora-*

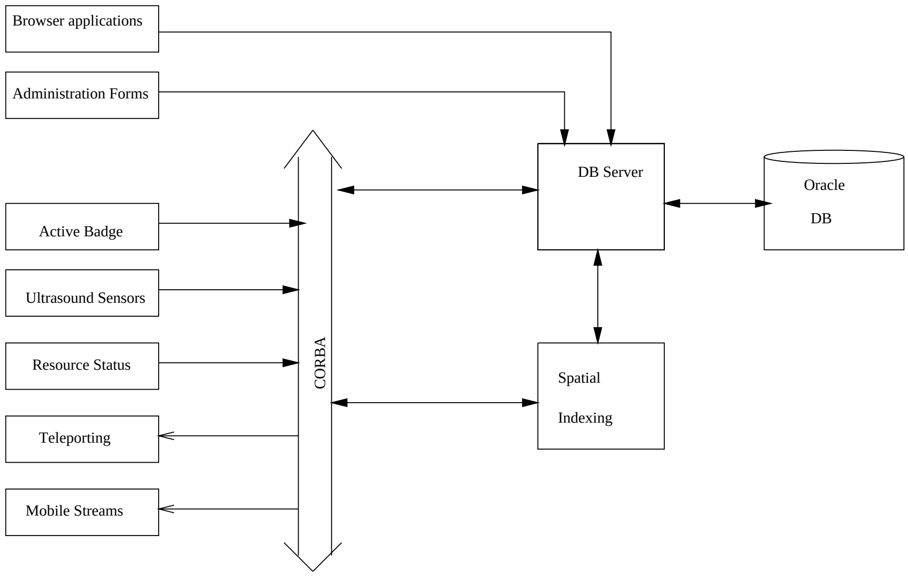

Figure 1: Overview of the architecture of SPIRIT

cle7 for managing concurrency control, locking, transaction processing, query optimisations, indexing and so on. The database maintenance packages are implemented using PL/SQL [Feuerstein 95] which is Oracle's procedural extension to SQL combining data manipulation and transaction processing capabilities of SQL with procedural programming constructs.

Spatial queries and monitoring are supported by a dedicated spatial indexing system which communicates via shared memory with the Oracle database.

The data stored into the database can be classified into two categories:

- **Dynamic data**, comprising location data which is obtained via the Active Badge System and the Ultrasound sensors and data associated with dynamic reconfiguration of the resources such as the load on a machine, the current running operating system on a machine, or the current topology of the network.
- **Static Data**, comprising all static attributes of the resources such as name of a computer model or a new employee. This is the data which cannot be reported by telemetry.

SPIRIT communicates with its sources of information as well as its application clients via CORBA interfaces. CORBA (The Common Object Request Broker Architecture) which is developed by the Object Management Group OMG [OMG 91] is a specification and an architecture that provides an implementation independent access to objects on heterogeneous systems. CORBA has been chosen as the distributed object infrastructure for its universal platform coverage; its "off the shelf" component standards; and its ability to interoperate with propriety technology, such as OLE and new languages, such as Java.

For dynamic data, we use telemetry services that allow hardware and software resources to enter status information into the database. For instance, services are provided to update the operating system currently running on a certain machine when the machine restarts, to update the topology of the network describing the connections of switches, cameras, microphones and all other devices attached to the networks, to monitor the status and the load of the resources as well as tracking the location of people and resources and feeding the database accordingly.

For entry of static data we use *Oracle7*'s SQL*Forms which permits fast development of GUI-based database clients. To implement browsers for SPIRIT data we can use SQL*Reports.

5 Design Issues

5.1 The Database Model

SPIRIT maintains information about a highly heterogeneous environment of different kinds of sensors, software and hardware objects as well as people. Each individual resource/object is represented as an entity. Various attributes are associated with each entity describing the relevant characteristics of the object. An object can inherit from another object and consequently inherits all its attributes. The set of all entities and relationships between entities constitutes the SPIRIT data model. We have used standard Entity-Relationship (ER) diagrams to represent the different schemas in the data model and we have implemented a simple tool to translate ER diagrams into SQL DDL.

5.2 CORBA Interface

In order to provide CORBA interfaces to the resource database several alternatives have been investigated:

- **Providing functions with SQL strings as arguments:** In this approach, the user submits the query or update in SQL forms as an argument which is then sent to the database server to be executed. This approach has been adopted by the MIND project [Kilic 95] which is a system that uses CORBA implementation in developing a multidatabse system. Although this approach is easy to implement, it requires all end users to be expert in SQL and to be aware of the details of the data model and schemas which is not practical.

- **Using the CORBA Object Property Service** The Property Service [OMG 95] is a standard CORBA 2.0 service that allows users to associate a PropertySet to an object then add, update or delete properties to this PropertySet dynamically. Using OPS, one can provide functions that gather related attributes together in a PropertySet and returns a handle to the PropertySet. The user can then modify the PropertySet defining, updating and querying properties and then write it back to the database when s/he is done. However, this approach has some limitations. First, although properties within a PropertySet are typed, a PropertySet itself is untyped. So, one cannot guarantee any type checking at the CORBA front end. Further, the database server cannot perform a mapping between a certain property and the corresponding attributes in the database unless a client gets a handle on the PropertySet first and then uses it. However, there is no way of enforcing this. Also, writing back a PropertySet to the database might involve updating a whole set of attributes spanning a variety of tables while only few entries have changed which would result in poor performance. Finally, with OPS there is no easy way to handle sets of sequences which is what is usually returned by a database query.

- **Hand-written interfaces** That is, implementing functions to perform specific updates and queries and providing a corresponding IDL (Interface Definition Language) mapping for them. This approach seemed to be the most suitable and was the one adopted. An example of a portion of an IDL interface is given in Figure 2.

  As shown in Figure 3, the database server consists of a CORBA server written in Pro*C/C++ [Oracle 93] using embedded SQL to implement the specified functions. The CORBA server can access the database directly using SQL commands or call one or more of the stored PL/SQL procedures or functions. A set of CORBA interfaces are provided for the clients, one for each resource.

- **Sedona**. We are looking at other alternatives for providing the CORBA interfaces which would be more efficient. In particular, we are looking into using the Oracle's Sedona tool which runs on top of an ORB. Sedona allows users to directly define CORBA interfaces for Oracle 8 objects (Oracle 8 is an object-relational database using SQL3 and incorporating inheritance, dynamic binding, references etc). Also it allows us to publish interfaces to CORBA clients from or to the ORB's interface repository. This seems as a more efficient way of registering Oracle with CORBA as it would provide the mapping for all the complex data types of Oracle.

```
typedef string error_message;              // NULL if no errors
exception DataNotFound{};
interface hwr {

        // IDL operations

        // add an Operating System type
        error_message add_os_type(in string os_type);

        // update the current running operating system
        void update_running_os(in string comp_name, in string os_type, in string version)
            raises (DataNotFound);

        // get the current running operation system
        void get_running_os(in string comp_name, out string os_type, out string os_version)
            raises (DataNotFound);
};
```

Figure 2: An example of IDL interface

6 Spatial Monitoring and Indexing

In keeping with the ultimate aim of SPIRIT, that the applications appear to know as much about the physical environment as the users do, we want to be able to trigger actions directly by moving resources and people in relation to each other. For example, we may wish mobile multimedia streams or teleporting sessions to automatically move from one display to another as a user moves around the environment, or we may wish multimedia peripherals to be associated with each other in some way when they are moved close together. To achieve this, we monitor the positions of resources and people as they move around in the environment; applications can register for callbacks with the monitoring engine, so that when events occur (such as a person moving with respect to a screen, or two multimedia peripherals coming close to each other) the applications are notified by the monitoring engine. An indexing system is used by the monitoring engine to enable monitoring for hundreds of different events simultaneously.

7 Conclusions

In this paper we presented an overview of the project SPIRIT which goal is to support mobile users in mobile environments by making all kinds of information about the environment

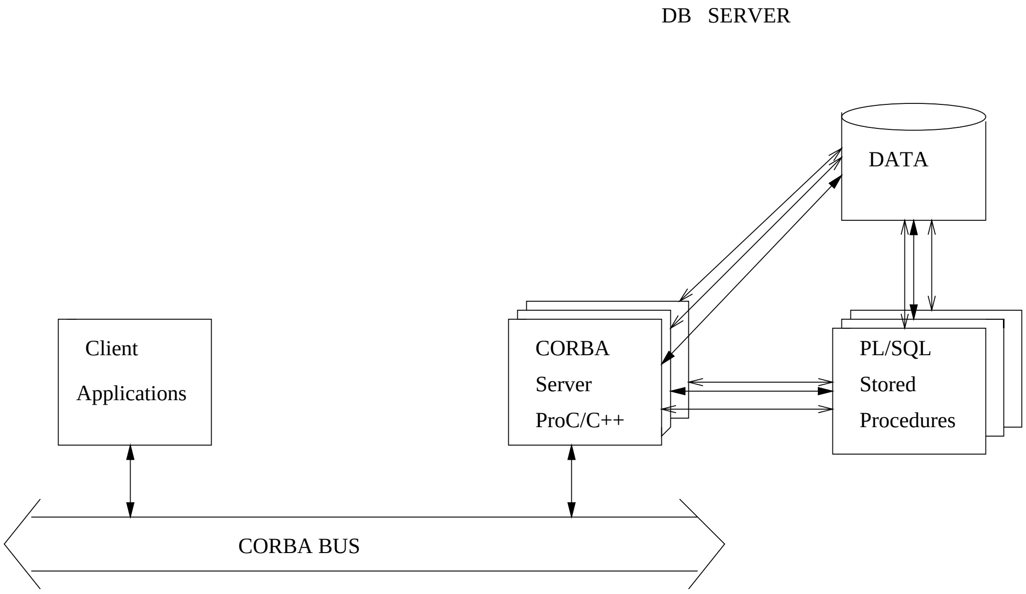

Figure 3: Database Server

including locations of people, hardware capabilities and the status of hardware and software resources accessible over the network. We have described some of the mobile applications available at ORL and showed how SPIRIT can be beneficial to them. Also, we have reported on our experience in providing CORBA interfaces to the resources database implemented in *Oracle7*.

Currently, the database for the hardware resources has been modelled, implemented and populated. This prototype has enabled us to experiment with the various aspects of the project.

Our future work includes extending the design and the implementation of the data model to incorporate all the software resources as well as information related to the IP network at ORL. This will be followed by the design and the implementation of the various CORBA services required by different mobile applications.

References

[BRH 94] F. Bennett, T. Richardson and A. Harter. Teleporting - Making Applications Mobile. In *Proceedings of 1994 Workshop on Mobile Computing Systems and Applications*, Santa Cruz, December 1994.

[Feuerstein 95] S. Feuerstein, *Oracle PL/SQL Programming*, O'Reilly & Associates, Inc., 1995.

[HH 94] A. Harter and A. Hopper. A Distributed Location System for the Active Office. *IEEE Network*, Vol. 8, No 1, January 1994.

[Kilic 95] E. Kilic, G. Ozhan, C. Dengi, N. Kesim, P. Koksal and A. Dogac. Experiences in Using CORBA for a Mulidatabse Implementation. In *Proceedings of the 6th International Conference on Database and Expert Systems Application DEXA*, London, September 1995.

[MM 96] B. Murphy and G. Mapp. Integrating Multimedia Streams into a Distributed Computing System. In *Proceedings of Multimedia Computing and Networking*, San Jose, CA, USA, January 1996.

[OMG 91] Object Management Group, *The Common Object Request Broker: Architecture and Specification*, OMG document Number 91.12.1, December, 1991.

[OMG 95] Object Management Group, *Object Property Service*, OMG document Number 95.6.1, June, 1995.

[Oracle 92] *Oracle7 Server Concepts Manual*, Oracle Corporation, December, 1992.

[Oracle 93] *Progammer's guide to the Oracle Pro*C Precompiler*, Oracle Corporation, December, 1993.

[Richardson 95] T. Richardson. Teleporting - Mobile X Sessions. In *Proceedings Ninth Annual X Technical Conference*, Boston MA, January 1995.