Exercise session 2: The design phase of Distributed Object-Oriented Application

06 April 2004

1. The Problem:
You are given the responsibility of conceiving a plane ticket reservation system and the associated check-in and boarding systems of an airport. This system comprises access points at travel agencies, and at check-in counters in the airport. The data comprises for each ticket the departure and arrival airports, the departure and arrival date and time, the date of purchase of the ticket, the flight number, the type of the plane, and similar information for the return ticket in case of a round-trip. To this are added the name of the company, and name of the passenger. For each passenger we have personal information like the address, phone number, means of payment, frequent flyer card, food and seat preferences, etc. For each flight we have its departure gate, its number of seats (both total and already booked). Someone should be able to look up for tickets, to book a ticket, and to consult the departure gate 1 hour before departure. Meanwhile, operators on the system should be able to check-in people, to interrogate it about the bookings of people, about the flights (booking rate, for example), and about the companies. There are timing constraints on response time to be respected when executing the functions.

2. Specification of the application:
Draw a class-relationship style diagram for the above-mentioned data. Write an interface specification for the functions.

3. Centralized solution:
The system is bound to be running on one machine that will be accessed through by the different users through a local-area network. What are the technical constraints you are facing? What will be the architecture of your application? What are the advantages and drawbacks of this solution?

4. Replicated solution:
To improve reliability, your boss says you should set up a second, replicated server. It would run in parallel with the first one, executing the same requests. What would be the possible architectures for this solution? What transparencies should you provide to the user? What evolutions can you see for the system?
5. Distributed solution:
The system is working well for a time (several days, weeks, months, or even years?)
until the incoming load makes it crumble several times, and just inaccessible on a
daily basis.
Just to try something new, you chose to build several distributed solutions. The travel
agencies have PCs, while the airport recently bought at an auction some Sun Sparcs
that belonged to an IT company.

There is no central database anymore. The distribution of the data is done according
to the desks in the airport, that keep the information about the flights of certain
companies and the tickets and check-ins of people flying with those companies.

How can you process such a request: give all the people that fly two times this week
from the same airport?