A NEW DATA MANAGEMENT SYSTEM AT THE AUSTRIAN AVIATION MET-SERVICE

Alfred Hofstadler
Austro Control GmbH
Vienna, Austria

Summary: The data management system at the Austrian Aviation Met-Service based on asynchronous serial data transmission, one UNIX server machine per airport met-station and ASCII terminals is in the process of replacement by an object oriented client-server database system written in C++ and PERL, running on SUN servers connected by a TCP/IP-network, with graphical user interface applications written in JAVA.

1. INTRODUCTION

For the last 15 years the data management system at the Austrian Aviation Met-Service of Austro Control GmbH, was based on one UNIX server machine per airport met-station, connected to the main MOTNE/RTH Vienna database by asynchronous serial data transmission. The data, mainly standard WMO ASCII messages, were kept in a filesystem database. ASCII terminals (Televideo 9xx) were used as user interfaces to request and display weather information and sensor data.

Due to problems in acquiring new terminals and transferring / handling large binary data volumes (eg. weather radar images or GRIB data) the replacement of the whole data management system had to be planned.

2. PROJECT CHARACTERISTICS

The project was started in early 1996 with one computer analyst and one meteorologist working together. After an analysis phase of 8 man-months followed a design phase of another 4 man-months (using ORACLE CASE). The implementation phase took another 8 man-months. At an early stage the decision for an object oriented client-server data management system was taken. The main objectives:

- Handling of all available data
- TCP/IP communication
- Interfaces for an Internet (WWW) pilot briefing system
- Improvement of database performance (higher independence from MOTNE database)
- Logfiles for database transactions
- Multi-platform GUI (replacement of ASCII-terminals)

The main server characteristics:

- SUN/Sparc
- Solaris Version >= 2.5
- C/C++, PERL
- Object oriented software development
- Object oriented databases
- Software documentation in HTML

The main client characteristics:

Hardware independent

GUI with MS-Windows or OSF-MOTIF look-and-feel

JAVA, PERL

Object oriented software development

Documentation and online help in HTML

3. DATBASES AND CONTAINERS:

It was decided to use data containers for manipulating the data and databases just for storing and retrieving data containers. The following database classes show just a selection of possible databases:

NetworkDatabase	base class of all database classes; does the necessary networking
MedasDatabase	main database node; handling of all container types
MotneDatabase	network interface to MOTNE/RTH message switch
WarningDatabase	warning messages (eg. SIGMET, AIRMET) with expiration time
StationDatabase	observational station and airport information
FlightrequestDatabase	request macros for airliner briefing
FormsDatabase	forms for product generation
SensorDatabase	sensor data
SensorCacheDatabase	sensor data, kept in memory for better performance
Flightaccident	all available data for the time of a flight accident
DatabaseLogDatabase	logfiles for databases
WeatherLogDatabase	logfiles for weather report generation
BriefingLogDatabase	logfiles for airliner briefings

There are just five methods for every database class:

server	to start the database server
open	to open the database
read	to query from the database; a data container containing the query is passed as argument, and a data container containing the requested data is returned.
write	to insert into the database; a data container is passed as argument
close	to close the database

This design was modelled after the database design at ECMWF (*Hofstadler*, 1993 and *Raoult* 1996). The internals of the different databases should be kept completely hidden from the user. The user should not even be aware of the fact where the database resides. To keep the interface this simple should make it very easy to replace one database or use, in case of a computer problem, just another database at another server location. It allows also to write database interfaces in any computer language for any platform (as long as they allow TCP/IP connections) very quickly.

All the knowledge about the data is kept with the data itself in the so called container classes:

Container	base class; a very versatile "piece of memory"
RequestContainer	list of request statements for different databases
SensorContainer	sensor data
StationContainer	observational station and airport information
WmoContainer	all kind of WMO messages (text and binary)
ReportContainer	single report out of a WMO message
WarningContainer	warning messages (eg. SIGMET, AIRMET)
FlightrequestContainer	request macros for airliner briefing
DocumentContainer	plain files

This is again just a selection of possible containers. When transferring a data container, inserting or retrieving a data item (which can as well involve a network transfer), it is always just the basic container type which is used. This guarantees that any database and any arbitrary client application is able to deal with the data item.

4. INTERFACES

All user interfaces should run on any possible hardware platform and all our customers (mainly airliners) should be able to use their preferred hardware. That was the reason why it was decided to use JAVA. After implementing a database interface and a few simple data container classes in JAVA we were able to commission the implementation of most of the GUIs (eg. a text editor, a sensor data display, a tool for creating and editing charts and other graphical products).

A CGI database interface written in PERL enables us to provide pilots with weather information via the internet using HTML pages.

5. OUTLOOK

The new data management system is now in a testing phase and should go operational in early 1998. At the same time there will also be a pilot briefing system available on the internet. The object oriented approach using standard network protocols and standard computer languages should enable us to make any necessary changes to our new data management system with minimum efforts.

6. REFERENCES

Hofstadler A., R. Gibson, K. Rider and B. Raoult (1993):"Meteorological Applications at ECMWF Utilizing EMPRESS", in ECMWF Newsletter, number 64, p. 11-15, Reading, UK.

Raoult B. (1996): "Data Handling at ECMWF", in Proceedings of the Fifth Workshop on Meteorological Operational Systems, held at ECMWF, p. 207-219, Reading, UK.