Information System for Operations at Medium-Sized Airports

D. Gillingwater, N. Ashford, and J. Shelton

A systems analysis of the information requirements of a medium-sized airport, which serves between 1 million and 5 million passengers per year, is presented. The proposed generic model of an airport information system (APIS) is based on this systems analysis and an evaluation of existing interactive software and hardware systems used at airports. Research was undertaken in collaboration with East Midlands International Airport (U.K.), together with Amsterdam-Schiphol Airport (The Netherlands), Frankfurt Airport (Germany), and Vienna Airport (Austria). The information requirements of an airport information system, developed using an approach based on structured systems analysis and design methods (SSADM), are described. Topics include an account of airport information system functions and system design objectives, definition and development of a generic model of APIS, and conclusions drawn from the research. The main conclusion is that current airport information systems can neither meet the information requirements associated with the operation of medium-sized airports nor approximate, to the specification of APIS, the proposed generic model.

Safety and economic reasons have accounted for the increasing use of computer systems within airports. Such systems contribute directly to flight efficiency by rapidly and economically coordinating flight preparations and by checking completed and ongoing operations. Airport authorities need to cater not only to their own interests but also, as partners with airlines and many other airport-reliant businesses, to the interests of their clients.

Airports thus may be viewed as concentrated networks of diverse but complex activities that link passengers and cargo to aircraft arrivals and departures. As such, they generate and require considerable amounts of information and are excellent examples of "information-rich" environments. Until recently, airports have relied on a variety of means to organize and manage this information and the flows that are generated. For medium-sized airports, which serve between 1 million and 5 million passengers a year, information transfer and information management have relied as much on manual paper-driven systems as on electronic data processing.

Subsystems have been created, typically on functional lines (e.g., to cope with financial accounts and passenger information), that essentially are separate entities. As a consequence, there is evidence of information duplication between subsystems and a reliance on personal communication and information transfer at the interfaces. The take-up of information technology (IT) has been comparatively slow and largely uncoordinated, at least in IT planning terms. The take-up has been driven mainly by the twin imperatives of internal financial management and external pressure from airlines. The overall view is that airport management have proceeded extremely cautiously with the implementation of IT.

The main portion of this paper is based on a 2-year grant-aided study, with the following objectives:

- To undertake a systems analysis of the information requirements of medium-sized airports,
- To develop a generic model of an information system, and
- To evaluate existing interactive IT systems in collaboration with East Midlands International Airport (U.K.), together with Amsterdam-Schiphol Airport (The Netherlands), Frankfurt Airport (Germany), and Vienna Airport (Austria).

Designing a complex IT application such as an airport information system has many methodological difficulties. Given the checked history of system design, it is hardly surprising, perhaps, to discover a degree of skepticism among airport managers about the ability of a single information system to meet the requirements of their diverse operations. That said, there still is strong anecdotal evidence that suggests considerable duplication of effort. In addition, many airport managers believe that airports are not unique undertakings and, as a result, that it is not necessary to search for problems and solutions specific to airports.

The research reported here leads to a different conclusion. The choice does not appear to be as stark as is often presented: either minimize initial outlay by purchasing proprietary business applications that may be capable of modification or get the system that is wanted by hiring a software consultant to design and deliver a system ab initio.

There is a third, preferable course to take: develop a generic solution that exploits the benefits of proprietary software and yields a design that airport managers will find familiar and want to use. For this to succeed, several preconditions must be met. First, the heavy development costs must be shared because they are beyond the capacity of an individual airport to fund from its own resources. Second, close involvement and collaboration are required on the part of the users (i.e., airport managers). Third, system analysts and software engineers are needed to produce the application.

A particular system design methodology has evolved, partly in response to such issues. Called the structured systems analysis and design method (SSADM), it is rapidly becoming a standard method for the analysis and design of information systems in the United Kingdom and elsewhere (1).

For these reasons, this research has tried to follow the discipline of SSADM an approach. As Downs et al. (2) make clear, SSADM is prescriptive, because it sets out the way in which a systems development effort should be conducted, and reductionist, because it breaks down a project into phases that are then divided into stages and subdivided into steps (with each step having a list of tasks, inputs, and outputs). Finally, Downs et al. (2) clarify that SSADM provides structural and procedural standards, covering everything from diagram notation and syntax to the conduct of interviews.
SSADM is a data-driven approach to system design and development that views problems from a data base management perspective. This approach follows the basic assumption that systems have an underlying, generic data structure that changes little over time, even though processing requirements may change. It also recognizes that there probably will be different views as to how a system's information requirements can be best met; for example, SSADM places great emphasis on the need to cross-check between different views for consistency and completeness. Finally, SSADM separates logical descriptions of a system from the physical aspects of development, converting a logical system to a physical design as late as possible, "... when the 'cost to fix' any errors is low but their potential impact very high" (2).

The structure of the method can be described as follows:

Stage 1. The current system, in its current implementation, is studied first in order to gain an understanding of the environment of the new system.

Stage 2. This view of the current system is used to build the specification of the required system. However, the required system is not constrained by the way in which the current system is implemented.

Stage 3. The specification of requirements is detailed to the extent that detailed technical options can be formulated.

Stage 4. The detailed design is completed at the logical level before implementation issues are addressed.

Stage 5. The logical design is converted into physical design by the application of simple (first cut) rules. The resulting design is tuned using the technique of physical design control before implementation. (1)

Background information for the research was obtained from in-depth field studies. These studies were conducted by examining the existing and well-regarded information system installations at three airports: Vienna Airport's system, called MACH; Amsterdam-Schiphol Airport's system, called CISS; and Frankfurt Airport's system. Additionally, a more in-depth analysis of information flows was carried out at one airport—East Midlands International Airport—during a 6-month period. The field and related studies were carried out during an 18-month period.

INFORMATION SYSTEM FUNCTIONS AND DESIGN OBJECTIVES

An airport information system lies at the center of any airport's operations. The primary objective of such a system is to improve the overall efficiency of operations and the quality of service to passengers, airlines and service companies. In practice there are at least four core activities with information system functions of critical importance (3):

1. Airport Management
   - Airport management per se (e.g., building, engineering, maintenance, finance, and personnel),
   - Air traffic control,
   - Airport information desk,
   - Airside/ramp operations and apron management, and
   - Airport operations monitoring (including noise and pollution monitoring).

2. Airline and Airline Handling Agents
   - Intra-airline information (e.g., use of systems such as CUTE and STIA), and
   - Aircraft servicing (e.g., cleaning, refueling, engineering checks, and catering).

3. Public Information
   - Off-airport videotext information services (e.g., teletext and similar TV-based information), and
   - On-airport public display monitors (e.g., flight departures and arrivals).

4. Security and Immigration
   - Airport police and security, and
   - Customs and immigration authorities.

From these four core activities, it is possible to arrive at a list of nine external entities requiring access to an airport information system. In other words, these entities are the main producers and consumers of information generated and used within an airport environment. Taken together, they constitute the principal ingredients of an airport information system:

- Airport management system,
- Airline/handling agent systems,
- Air traffic control,
- Airport information desk,
- Ramp operations and apron management,
- Videotext systems,
- Public display monitors,
- Security, customs, and immigration, and
- Airport operations monitoring system.

Figure 1 shows the external entities graphically. To indicate their crucial roles, three functions are described more closely: airport management system, airline/handling agent systems, and ramp operations and apron management.

Airport Management System

The planned daily flight schedule, or "Mayfly," is of great use in manually operated systems. However, in information system terms, it is the seasonal flight schedule that provides the initial impetus. This schedule stores all scheduled and charter flights planned on a seasonal basis, usually during the summer and winter seasons. It contains flight details required for the daily flight schedule. The essential data are

- Planned arrivals and departures, shown by clock time (00.01 to 24.00 hr),
- Flight numbers (e.g., BY 482 A), and
- Departure and arrival gate to be used by each flight (e.g., Gate 6).

The preparation of the daily flight schedule, however, provides the operational pressures for airport managers. The seasonal flight schedule provides the base for this schedule, with any new but pre-planned flights normally being entered 7 working days before the departure date as a "rolling update."

Several functions therefore must be built into any flight schedule data base; they include the abilities to

- Change the seasonal and daily flight schedules,
- Flag any flight on the data base (i.e., to call it up "by exception"),
- Print any data in real time, and
- Automatically delete historical flight data.
In a multiterminal airport, dedicated terminals need to be differentiated between those giving users read-only access and those giving direct or partial access to the flight schedule database. Where air traffic control or airport managers—and to some extent airlines—are involved, they must be able to add, delete, or modify flight data rather than have read-only access. All changes must be flagged for operational personnel. For software requirements, this function must incorporate well-defined data fields to allow, for example, causes of delay to be entered.

Flights in the daily flight schedule database should be created automatically from the seasonal flight schedule listings where possible. Each day a “spooling-off” of the previous day’s schedule should begin to form the actual daily flight schedule. This process should allow for real-time hard-copying of records to permit manual checks and to provide greater security of historical data in case of subsequent loss.

When deleting flight records is required, the data are transferred to an archive database for flight schedules that should at least include the following:

- Airline, point of origin, and destination, including en route stops;
- Scheduled start and end dates of the flight;
- Whether the flight was a scheduled, charter, cargo, or private flight;
- Details of handling agencies involved and any problems experienced (e.g., indicated by code); and
- Any relevant information from the airport operations monitoring system (e.g., whether a flight breached current noise regulations).

In addition, the following details must be recorded to comply with current regulations:

- Scheduled arrival and departure times;
- Logged flight-plan route;
- Nature of flight (i.e., passenger, cargo, commercial civil, private civil, royal/presidential, or military); and
- Handling agencies used.

As a guide, any airport information system should be able to deal with the following methods of updating the data bases, manually via a terminal or automatically via a screen update program:

- The system must allow full-screen updates by authorized personnel only.
- Repetitive daily updates should not require user intervention (i.e., flights that repeat daily in the week should not need to be reentered daily).
- Flagging of any changes must be marked clearly for the user’s benefit.

When an individual flight record is incomplete (e.g., when no “on-stand” time has been entered or the baggage handling agent is missing), airport personnel will need to access the actual daily flight schedule database to perform remedial updating procedures. To avoid corrupting this database, it is best for such amendments to be entered on a different database at the flight performance monitoring stage, which will require various levels of system access to be incorporated at the design stage.

**Airline/Handling Agent Systems**

Given the complexity of the interfaces between airlines and their handling agents, consideration here is restricted to three particularly important functions: baggage belt handling, boarding gates, and fuel and catering.
Baggage Belt Handling

The status of baggage belt handling can be divided into four phases:

1. An arriving flight is allocated a designated belt.
2. The “first bag” is on the belt.
3. The “last bag” is on the belt.
4. The belt is cleared for reallocation to another flight.

This information must be shown on public display monitors and recorded on a data file for airport records. The relevant handling agency therefore has to have direct read-and-write access to this part of the daily flight schedule data base to allow for real-time updating. Indeed, this access is an essential requirement for the smooth and efficient running of any arrival or departure area to allow passengers to be directed correctly to their allocated baggage carousel.

Any system also must account for errors in the baggage handling system. Under normal circumstances the in-built requirement for belt displays to regulate themselves automatically would be sufficient. If no last bag is entered, for example, there needs to be a specified time lapse before the flight disappears from the allocated belt so that handling staff have time to correct any error.

Boarding Gates

There are four separate phases to the boarding gate procedure:

1. Boarding gate is declared “open.”
2. Passengers board the flight.
3. Final call is announced.
4. Boarding gate is “closed.”

Any status change to a flight normally is initiated by an airline employee making data entries directly into a dedicated terminal at or near the boarding gate. When any gate message is received, the exact time of the change should be both stored and highlighted so that it is readily recognizable as an alteration.

Fuel and Catering

Both an airport’s “fuel farm” and the catering organization need to be linked to the airport’s daily flight schedule data base, but on a read-only basis. This specification is to allow for estimated times of arrival and delays to be seen and considered when both organizations are planning or executing a daily rota. The timely arrival of refueling and catering vehicles is necessary to avoid turnaround delays when an aircraft is typically on-stand for only 45 min to 1 hr. At well-coordinated airports with comprehensive information systems, a medium-sized, 130-seat aircraft can be simultaneously refueled, restocked, and cleaned in half an hour from initial “on-blocks” time.

Ramp Operations and Apron Management

Aircraft parking and boarding-gate slots need to be allocated by the ramp marshal’s office. An information system must be able to cope with these operational requirements and provide facilities for the on-line updating of the daily flight schedule data base through the following methods:

- Direct command to update a specific flight with a new gate slot and to highlight this change on screen (e.g., via reverse color modes),
- Full-screen update that alters several flights in a specified order,
- “Flagged” automatic command that shows flights and satisfies certain criteria that are entered into the daily flight-schedule data base.

The main elements of the flight schedule data base, which must be updated daily, are as follows:

- Originating airport of flight,
- Aircraft registration,
- Aircraft type,
- Public display monitors,
- Airline users (new airlines),
- Baggage handler,
- Baggage belt allocation, and
- Parking stand allocation.

These elements need to be updated fully on a rolling basis (often every few minutes) to ensure an efficient cascading of data to the finance and accounts section of the airport management system, thus allowing timely and accurate invoicing of customers.

It is most important that an accurate record of both boarding gates and parking slots is kept because, in the event of a query from an airline, records from the flight schedule archive may have to be cross-checked. In practice, many such queries result from the inaccurate entering of on-block and off-block times, which leads to incorrect charging of users. Incorrect identification of an aircraft type also can lead to under- or overcharging because of varying passenger numbers carried and varying takeoff and landing weights.

A simple data entry procedure is all that is required for ramp and apron staff for real-time updating. It has been considered that ramp staff need to be given a printer so that hard copies of the daily flight schedule can be carried around on the apron. Hand-held terminals with read-only access are a great advantage because very recent updates may be seen in real time.

The three preceding information system functions—airport management system, airline/handling agent systems, and ramp operations and apron management—indicate the complexity of the information requirements of an airport and should be incorporated in the design of an airport information system. If these entities, together with the other six external entities, constitute the main information system functions of an airport, then the next step is to identify the critical system design objectives to be met. Five such objectives would form the backbone of any airport information system design.

The first objective would be to provide an efficient fault-tolerant information transaction system. Such a transaction system would contain all the application software for accessing an airport’s data bases, presenting terminal users with a menu-driven data base system. As such, the transaction system would interface with terminal users and the core of the data bases: the daily flight schedule. Commands to access on-line data would be given processor priority; requests for historical data would be dealt with in queued sequencing by a background processing system (which is the second objective). A further interface would be provided within the system for automatic signaling to the information control system for all updates to the data bases. This interface would enable the information control system to display accurate records from the daily flight schedule.
The second objective would be to provide an efficient background processing system. The processing of low-priority or batch-control jobs would be dealt with through the background processing system. This system would be used primarily for producing historical data printouts or jobs that do not require direct interaction with a terminal user. Jobs would be queued either after a request from a terminal user (providing the user has clearance) or automatically at a certain time interval (specified by the system user). Printouts from terminal users normally would be handled serially to simplify any design software required and to minimize costs.

Providing a fault-tolerant data communications system would be the third objective. The data communications system deals with the hard-line connections to other airport or external computer systems. Changes to an airport's data bases could be initiated via incoming messages. Outgoing messages from the data bases could be generated automatically or by request from a terminal user. Changes to the data bases and telex-type messages (e.g., via SITA) also would be generated in this manner.

The fourth objective would be to provide an effective information control system. To ensure that the total system functions optimally and according to predetermined access criteria and arrangements, the information control system would be required to generate, monitor, and maintain new and updated information.

Finally, the fifth objective would be to provide an effective information distribution system. To display information in appropriate formats for specific predetermined uses, the information distribution system would consist of both hardware and software. It could receive information from a wide variety of hosts and from the information control system. This information, stored in paginated form, could be viewed by terminal users via the data base menu. Typically, television monitors and LED and LCD boards are used as display devices for presenting page contents at airports. Because a display device shows the complete contents of one page, any updates would have to be shown in real time on public monitors showing that particular page.

Two methods are available to specify which page must be displayed on a generic console: (a) define a display device as fixed—that is, the console will always show a fixed daily timetable or other related data unless updated by the central data base (e.g., passenger departure monitors), or (b) provide a monitor with a keypad for self-selection of a page of data (e.g., staff keypad monitors at the airport information desk). Both methods can be used together, too.

**DEFINITION OF GENERIC MODEL**

The scene has been set for revealing the structure of the generic model being proposed, on the basis of the detailed findings of the preceding research. This section describes the structure of information processes and flows following SSADM principles generally and the data flow diagram (DFD) conventions specifically. Figure 2 presents standard DFD notation.

The structure of the proposed model for the generic airport information system (APIS) is shown in Figure 3. The model starts with the nine external entities; the hashed box delineates the system boundary between these entities and APIS functions. Within the system boundary are seven information processing functions: seasonal flight schedule, air traffic control communications, daily flight schedule, flight information display, actual daily flight schedule, flight schedule archive, and flight performance monitoring. Together these constitute this APIS. However, the arrows indicate the main data flows between the external entities and information processing functions and the data flows between information processing functions.

In summary, the main points about this model are (a) the nine external entities are linked to seven information processing functions by 25 separate data flows; (b) the core, or strategic, external entities, as evidenced by the number of separate data flow links, are airline/handling agent systems (4 links) and air traffic control (3 links); (c) the core, or strategic, information processing functions are the flight information display (9 links) and the daily flight schedule (7 links). Taken together, these four activities account for 23 of the 25 data flow links.

Before the model is explored further, it is necessary to explain the seven information processing functions. The role and significance of the seasonal flight schedule consists of the advance program of flights that airlines or their handling agents, or both, plan to operate in the upcoming months—typically these months are winter and...
summer schedules. Air traffic control communications represents the interface between the control of airspace movements and airport operations; in the context of APIS, it is associated primarily with aircraft takeoffs and landings as well as with apron and taxiway movements.

The daily flight schedule, its prominence described earlier, represents the immediate and current 24-hr period of flights planned and is at the heart of day-to-day airport operations. Links between the daily flight schedule and flight information display are sometimes difficult to disentangle; for example, in many airports both are virtually one and the same (4). The flight information display, however, is much more than the system that drives public display monitors in airport terminals: it is the core of real-time information processing at an airport, as the number of data flow links implies.

If the flight information display records and displays data dynamically, the actual daily flight schedule begins the process of information storage and retrieval before the data are archived in the flight schedule archive. As the title implies, the actual daily flight schedule consists of a 24-hr record of the flights that actually took place from 00.01 hr to 24.00 hr. It also provides an “after-the-event” opportunity for airlines or their handling agents, or both, to add data to the information input via the daily flight schedule. At the end of this period, these historic data are transferred to the flight schedule archive for subsequent retrieval and analysis by the external airport management system.

Because if Figure 3 presents the overall structure of APIS, it is now possible to proceed to the next level of system design by examining the data flows within each of the seven information processing functions. Using the same logic and notation, Figures 4 through 9 describe these flows.

Figure 4 begins to unpack the structure of the seasonal flight schedule. In data base management terms, it shows that this function contains two data bases: the planned seasonal flight schedule and the current seasonal flight schedule, the former being updated by a processor called update planned seasonal flight schedule. The output of this process is passed to the daily flight schedule, which is shown in Figure 5. As a consequence, this output becomes the input data to be processed and reconstituted as a third data base, the daily flight schedule, which is then passed to air traffic control via the air traffic control communications black box and the flight information display.

Figure 6 presents the crucial role of the flight information display function and demonstrates that it is much more than an interface between airport operating functions and the display of information on public monitors (4). Three data bases are generated following two information processing stages: two data bases, current flight records data and public display data, are generated via a processor called prepare flight records and public display, and the third, video-text data, is generated separately (largely for technical reasons) via the prepare video-text data processor. Two of the data flow outputs become inputs to two further internal information processing functions: the actual daily flight schedule (Figure 7) and flight performance monitoring (Figure 8). In data base management terms, both functions are essentially data merging, data verification, data cor-
Airline/Handling Agent Systems

Air Traffic Control

FIGURE 4 Seasonal flight schedule.

Seasonal Flight Schedule

Airline/Handling Agent Systems
Airport Information Desk
Ramp Operations and Apron Management
Air Traffic Control

Prepare and Change Daily Flight Schedule

Daily Flight Schedule

Air Traffic Control Communications Flight Information Display

FIGURE 5 Daily flight schedule.
roboration, and data updating activities before data storage and retrieval. The product of the actual daily flight schedule is the data base confirmed flight record data, and that of the flight schedule archive is the flight record archive.

The final data flow diagram for flight performance monitoring (Figure 9) consists of one data base, flight record monitoring data, created by combining data from the external airport operations monitoring system (e.g., an integrated noise monitoring and flight tracking system) with the data merged from the actual daily flight schedule and the flight information display.

Thus having established the structure of the relationships between the key external entities and the internal information

---

**FIGURE 6** Flight information display.

**FIGURE 7** Actual daily flight schedule.
processes as constituting the overall model for APIS (Figure 3), and having identified the data flow relationships within each of the internal information processes (Figures 4 through 9), the next stage would be to identify the internal requirements of and specify the structure for the nine data bases. This stage would come before any attempt to develop software or consider hardware requirements.

CONCLUSIONS

This research has attempted to meet the requirements of the first two of the five stages of an SSADM approach to system development, which was outlined earlier. What has been presented should be seen as the first steps toward developing software for an APIS configu-
ration. The proposed APIS is generic because, in principle, it should apply to any and all airports; it has been based on tracking flows of information between users and translating those flows into a systematic structure that does not necessarily conform to existing organizational boundaries or corporate functions. It demonstrates an approach to thinking about information management in data base management terms, which is independent of information “owners” and IT professionals.

ACKNOWLEDGMENTS

The research in this paper, and reported in full elsewhere (5), relied on the active collaboration of a large number of individuals and organizations. Needless to say, the paper reflects the views of the authors rather than those who collaborated in its production.

Due acknowledgment is given to the following organizations for their helpful collaboration and enthusiastic responses: Amsterdam-Schiphol Airport, East Midlands International Airport, Frankfurt Airport, Intersystems bv (Amsterdam, The Netherlands), Tecnost SpA (Turin, Italy), Vienna Airport, and Airport Information Systems (Nottingham, England).

The research was funded principally by the Science and Engineering Research Council of the United Kingdom.

REFERENCES


Publication of this paper sponsored by Committee on Airport Landside Operations.