ꟼ periodica polytechnica

Transportation Engineering
37/1-2 (2009) 15–21
doi: 10.3311/pp.tr.2009-1-2.03
web: http://www.pp.bme.hu/tr
© *Periodica Polytechnica 2009*

RESEARCH ARTICLE

Airport information system integration by using message broker

Zsolt Kelemen

Received 2007-09-30

Abstract

Modern airports are being equipped with an ever growing number of modern electronic systems for specific tasks: Baggage Handling Systems (BHS), Flight Information Display Systems (FIDS), Departure Control Systems (DCS), Weight & Balance Systems; Common Used Terminal Equipment (CUTE) Systems; Building Management Systems (BMS), Access Control Systems (ACS), Airport Operation Databases (AODB), Resource Management Systems (RMS) ... there is virtually no end to the list.

Nevertheless, many airports are still difficult to navigate for passengers, still have to face departure delays and frequent changes in operations, and still miss substantial revenue opportunities or waste money because of unused resources of every kind.

Key to using all those powerful systems to their full capacity and reaping all their benefits is bringing them into tune. A task that is commonly known is the System Integration by using Message Broker solutions. This rather technical term subsumes all activities aimed at establishing the physical connection of disparate systems. The concept of system integration is one step beyond the simple installation of cables and the translation of a few protocols.

This paper provides details on the Integration Message Broker solutions in the airport information system industry.

Keywords

air transport · airport · message broker · integrated information system

Zsolt Kelemen
Department of Application Management, Budapest Airport Pte Ltd., H-1675 Budapest Ferihegy International Airport, Hungary
e-mail: zsolt.kelemen@bud.hu

1 Introduction

Airport Applications worldwide are undergoing reformation brought about by new demands communication and inter-modal transportation. Messaging middleware is software that provides an interface between applications, allowing them to send data back and forth to each other asynchronously [4].

Data sent by one program can be stored in a queue and then forwarded to the receiving program when it becomes available to process it. Without using a common message transport and queuing system such as this, each application must be responsible for ensuring that the data sent is received properly. Maintaining communications between different types of applications as they are revised and eventually replaced with newer architectures creates an enormous programming burden in large enterprises.

Message Broker is either a complete messaging system or software that works with existing messaging transports in order to add routing intelligence and data conversion capabilities. A rules engine analyzes the messages and determines which application should receive them, and a formatting engine converts the data into the structure required by the receiving application.

Business today is no longer defined by what it does centrally but includes activities across many locations and business units. Fast, reliable and accurate information are vital to many organisations operating in the global economy in order for them to gain competitive edge or simply to increase efficiency. Disparate systems need to work together easily, seamlessly and immediately to obtain vital information for planning and decision-making [5].

Message Broker allows disparate systems to communicate with one another and share critical business information, hence enabling organisations to produce quality and meaningful information fast where it is needed.

Message Broker also enables businesses to adapt to different systems that have multiple or different operating systems transferring real-time information between each other with ease without any costly reengineering processes. Also, the business will have the freedom to choose the best systems to suit its needs and implement it without restricting business growth.

2 Integration technologies

During the 1990s, companies bought packaged software solutions such as SAP, Oracle ERP, and so on. Although such packaged software solutions worked well individually, they created information islands. In most cases, each system produced redundant information (like customer information). As a result, when common data changed, employees manually updated the associated information in each system, a process that quickly becomes cumbersome. Eventually, some of the data across systems became inconsistent.

When people noticed the resulting double data entry, inconsistent data, and data isolation problems, they decided to find ways to integrate the systems. From that search, enterprise application integration was born.

This combines separate applications into a co-operating federation of applications. Two logical integration architectures for integrating applications exist: Direct point-to-point connections and middleware-based integration [5].

2.1 Direct point-to-point connections

Initially, developers chose point-to-point integration because it was easy to understand and quick to implement when they had just a few systems to integrate. An example of a point-to-point integration is when one application makes direct database calls to another application's database tables. Another example is when one application communicates with another using TCP/IP sockets [2].

At the beginning, when integrating two applications, the point-to-point integration solution seems to be the right choice. But as additional applications are integrated, the situation becomes unwieldy. Each application is tightly coupled with the other applications through their point-to-point links. Changes in one application may break the applications integrated with it.

Another disadvantage with this approach is the number of integration points needing support.

- If we have five applications integrated with one another, we need ten different integration points.
- If we have ten applications integrated with one another, we need forty-five different integration points.

As a result, each additional application becomes exponentially harder to integrate and maintain. To avoid this problem, we need an intermediate layer to isolate changes in one application from the others.

2.2 Middleware-based integration

This mediation point between applications is called middleware. Middleware provides generic interfaces with which all integrated applications pass messages to one another. Each interface defines a business process provided by an application. This integration architecture allows us to add and replace applications without affecting the other applications. Compared to the point-to-point solution, middleware-based solutions easily support numerous integrated applications and require less maintenance.

In addition, middleware can perform complex operations-transforming, aggregating, routing, separating, and converting messages based on the data passed from applications to applications.

The only downside: the added initial complexity of setting up the middleware and converting existing applications to use the middleware application programming interfaces.

A specific middleware technology is the message-oriented middleware, the so-called Message Broker. Applications communicate with one another by passing messages and these messages are queued if the receiver is unavailable thus guaranteeing that the messages will eventually be delivered.

2.3 Java 2 Platform Enterprise Edition

Further development of the Message Broker is the Java 2 Platform Enterprise Edition (J2EE). The purpose of J2EE is to provide platform-independent, portable, multi-user, secure and standard enterprise-class platform for server-side deployments written in the Java programming language. It is a robust suite of middleware services that makes life easy for enterprise application developers. Java is a portable language that is vendor and platform independent.
Some of the major technologies provided by J2EE are:

- Java Database Connectivity (JDBC)
- Java Messaging Service (JMS)
- J2EE Connector Architecture (JCA)
- Java for XML Messaging (JAXM).

J2EE technology covers all aspects of enterprise application integration. If we want to stick with point-to-point integration by direct database access, we can use JDBC. For middleware-based integration, JMS provides integration with existing systems. For accessing existing legacy or ERP systems, JCA provides the common connector architecture that makes life easier for enterprise developers and system vendors as well. Applications can communicate directly with one another using JAXM technology.

System integrators have incorporated many new features of Message Broker to enhance the performance such as synchronous messaging, priority-based routing and also support different protocols.

Technologies employed for this purpose often rely on an AODB, which concentrates up-to-date information for various systems at one place, accessible to all connected systems, and on a Message Broker solution, which distributes crucial information and any updates in a platform-independent, open manner.

Via Message Broker any AODB can collect, store and distribute information across an airport. From its original design criteria, through its development and deployment, AODB can be designed to be of benefit to the whole airport community. This means that although the airport operator owns and manages the system, much of the detailed participation in the running of the system is done by other members of the airport community, including airlines, ground handling agents, government agencies and specialist operators on the airfield. So the whole airport community has an interest in making sure the system operates effectively, and in return, the system is available for them to use in their own operations, subject to inbuilt access rules.

The end result is a system that the airport community can trust to deliver accurate, consistent and timely information. This is vitally important in a complex operational scenario, where many parties have to work in a co-ordinated way to ensure operational efficiency.

3 The goals of establishing a message broker

Airport management has to understand that the key question is to get the latest and best information to everyone who needs. In a typical systems integration project, experts analyze the data available at the airport, data types and quantities, the frequency of updates, the requirements for fresh data in all systems, and the optimum ways how to improve availability of crucial information. But system integrators do not even stop there: Included in the analysis are also the individual available systems involved, and the processes that the systems are meant to support.

Such a process and information map is fundamental to designing the right technology and the required degree of systems integration. It allows a customized approach combining periodical updates of systems (e.g. manually or via broadcast) and on-line connections for automatic mutual update of systems in real-time.

The objective is to enable relevant system connections to be done more efficiently than is possible by manual methods and at least as economically. Deploying a Message Broker will help airports to achieve the following business goals:

- Cost effective integration of existing and new IT systems based on reusable services;
- Loosely coupled integration avoids supplier lock-in;
- Uses industry standard technology e.g. XML and Web Services;
- Proven solution results in lower cost and risk;
- Central location for airport and integration business rules;
- Easy user configuration of additional system interfaces;
- High availability architecture with failure recovery.

Bottlenecks, dead-ends and unwanted information modifications are eliminated and, as a side effect, network traffic can be reduced, freeing up capacity on the entire system for a better response rate throughout the overall system.

The application work around the most advanced technologies can bring all the related benefits to the users, hiding the related complexities by means of proprietary middleware:

- Internet paradigm;
- Secure protocols;
- Pervasive computing (hand held terminals, mobile offices, wearable devices, mobile phones, PDA, etc.);
- Voice and data integration;
- Wireless networking (WIFI, GPRS, TETRA, etc.);
- Multilanguage Automatic Speech Recognition and Text to Speech.

All these technologies could be independent from:

- Telecommunication channel;
- Language;
- HW device,
- Local infrastructure.

Developers continuously work on new technologies to validate their usage within the products so that the software is always up-to-date and innovative.

Furthermore by implementing a Message Broker airports will no longer need to store the same data on a number of different systems, that is why it is a necessary step in the information system integration process [2].

Preserving existing investments and allowing airports to choose any new system without restrictions makes this a compelling system for an airport to own. Integration between systems becomes simple and possible. Finally, the need for better efficiency as well as security and safety will be met simultaneously as well as the homogenous data in the heterogeneous airport systems environment.

4 Integration Message Broker basic functionality

Message Broker is the only integration tool that an airport will ever need to resolve its problem in sharing, transferring and transmitting information to various systems at the same time between different vendors and to look forward for high efficiency operations.

Message Broker can provide a powerful platform to integrate, consolidate, manage and distribute the large and diverse amount of data affecting the multiple organisations and systems involved in the ground operation.

Departing from traditional point-to-point interfaces, it ensures that the information is entered only once, at the source, and that data and process ownerships are clearly established. Message Broker makes certain that data updates are shared amongst the affected parties from Air Traffic Control through all management and operational systems to FIDS.

Message Broker can distribute the data updates to all connected systems, including FIDS, Baggage Sorting and Air Traffic Control and to the Ground Handling applications (for example Air Cargo Community System) as well. The open architecture allows for the easy addition and integration of new users and systems [3].

If the Message Broker is developed on the J2EE Java platform, it will provide highly scalable and robust architecture. It can support systems running on different platform (hardware, operating systems, application servers, and databases).

In order to support all actors in the airport industry to reach best efficiency and quality targets, software developer companies have spent huge efforts in finding solutions and services able to bring them the power of technology innovation in a transparent way and working with a business approach. All results of this strong commitment of software developer companies have been integrated in a global framework.

The best philosophy is to manage the complete airport lifecycle for flights, airport resources, operators and passengers from seasonal schedule through planning and operation processes up to invoicing as it is shown in Fig. 1.

An extension of this concept concentrates on sending and receiving information between different systems. Message Broker allows incremental and modular changes to the systems, along with the addition of new systems, without disrupting ongoing operations.

New features of Message Broker include: [6]

- Support for Web Services;
- Support for Java Message Service producers and consumers;
- Contents-based routing;
- Support for more text message formats like ‘delimited’ and ‘name/value pairs’;
- Support for native TCP/IP sockets interface for senders and receivers.

A modular and flexible architecture can provide the following major infrastructural components:

- Airport Operation Database in the middle with its Foundation Modules;
- Airport Service Bus around AODB as a Message Broker using different protocols (Corba, Http. XML, JMS);
- Multi-channel Distribution Services
- Applications: airport owned systems and 3^{rd} party systems as well.

Fig. 2 illustrates schematically logical structure of the components and their connections as an example.

The core component of the framework includes all the basic modules constituting an integrated application environment. It includes an AODB as the central airport repository, and a set of specific services called Foundation Modules. Examples of Foundation Modules are Allocation Middleware, Diagnostic Management, DCS gateway, Airport Desktop, Airport Configuration Manager, and User access security, Peripheral Management, Flight Programming and applications used to configure the centralised airport resources.

In addition, the Airport Service Bus as a Message Broker presents hub and spoke notification channels and multi-standard interfacing services, which allow a strong integration between systems, whilst ensuring an actual functional and technological uncoupling among applications.

One of the most interesting features of the framework is the capability of performing a multichannel distribution of the information to the users. Particularly, a rich set of tools allows selecting the most suitable media in respect of the specific sort of information that should be transmitted to both airport staff and passengers.

Particularly, the Multichannel Distribution Services allow the use of the following channels:

- Display Unit Multicasting;
- TV Teletext;
- SMS;
- Web Portal;
- E-Mail;
- Fax;
- Wireless;
- Voice.

All these channels can be supported by advanced technologies and specialised application services providing high-level interfaces that hide the complex technologies employed.

As regards the applications dedicated to the different operational areas, the framework can make available a wide range of advanced solutions. It is also possible the integration of third parties solutions, without any limitations of their abilities to interact with other native applications.

In the future AODB will play a complementary role, as each is dependent on the level of computerisation and automation of supporting functions, as well as the integration level, to be able to receive the relevant data real-time. An interoperability mechanism like a Message Broker will be key component to provide the level of integration needed and the flexibility of data interface required.

5 Benefits of Implementing a Message Broker

Information system solutions can help the business cycle with tools and features that assist with planning, support real-time operations and enable full follow-up [1]. A real-time information and communications model underlies all aspects of airport

Fig. 1. Complete Airport Lifecycle for Flight

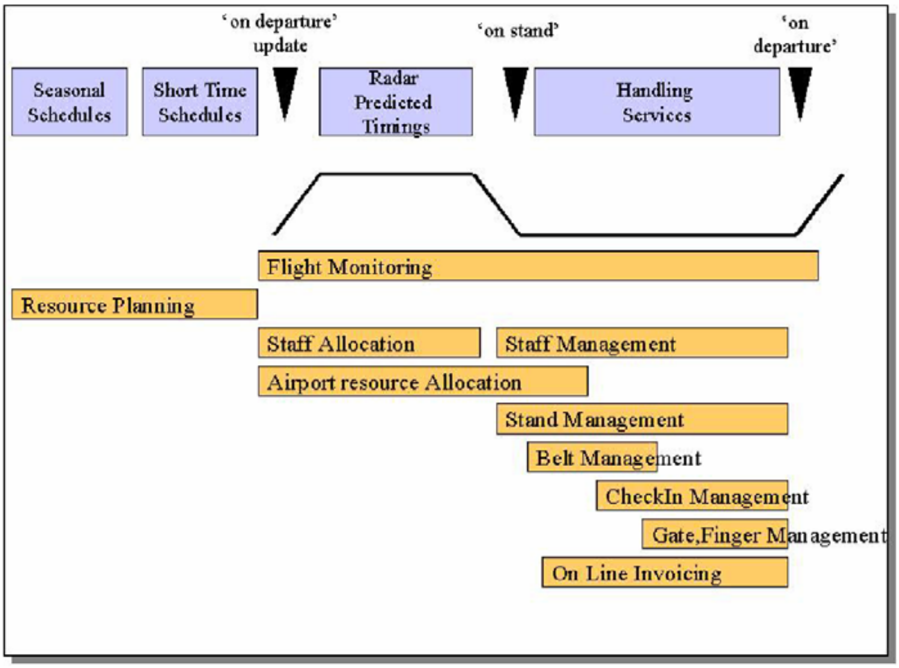
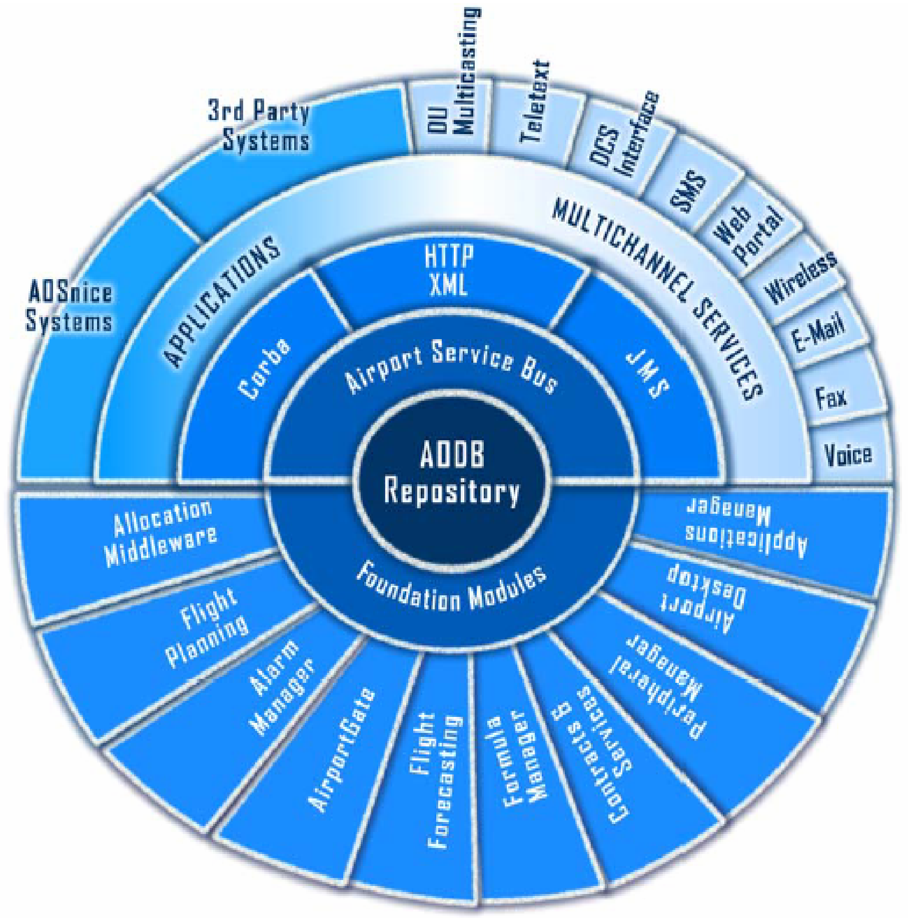


Fig. 2. Logical Structure of a Typical Airport System Components Architecture [6]



operational management to give the user the resources to make informed business decisions.

Powerful automation tools can collect data from all major airport systems. The system then processes and organizes this data into a consistent enterprise-wide view to help planning, decision support, forecasting, billing, scheduling and more. Most relevant benefits enjoyed in using such kind of Message Broker are:

- Cost saving
  - High benefits/costs ratio;
  - Personnel training reduction;
  - Facilities and human resources planning optimisation and management allowing operative costs reduction;
  - Safeguard of investments on legacy systems.
- Passenger satisfaction
  - Multi-channel access to the business and operational information of the airport (e.g. Web, SMS, PDA, Public Address, Teletext, Internet, etc) allowing a dramatic reduction of the staff dedicated to customer relationships (e.g. Customer Care, Lost and Found, etc.);
  - Continuous enrichment of services;
  - Improved passenger handling.
- Efficiency and optimization
  - A unique desktop to access any application and a standard user interface approach for all modules reduce training needs allowing easy introduction of new resources and the reallocation of existing resources.
  - A new easy and integrated way to contemporarily manage flight information, advertising and general message dramatically simplifies the work of terminal responsible and customer care offices.
  - Resource allocation solutions allow optimization of resource planning and management completely integrated with all other operational tools.
  - Improved terminal management.
- Safety
  - Centralization of user profiles;
  - Dynamic user access control and certification;
  - Secured communications.
- Reliability
  - Reliable central database and application servers;
  - Application Independence avoiding cascade faults;
  - Manual operation support for contingency plans.
- Maintainability
  - Usage of open systems design principles such as well-defined interfaces and object modelling;
  - Maintenance cost elimination for client stations by using internet design;
  - Integrated and controlled management of all airport systems and peripherals gaining high efficiency in the ability to discover and identify the anomalies and the related fault sources.
  - Fast quality improvement and average intervention time reduction of maintenance services through real-time notification tools (mail, SMS, etc.) which inform the staff in charge about the anomalies occurred.
- Extensibility
  - Modular and multi-tier architecture able to simplify business requirements and technology changes;
  - Loose coupling interfaces among applications by means of a communication bus allowing module changes without impacts on others;
  - Openness to third parties solutions and legacy systems (furthermore, in many cases, cheaper than building their own systems and in this way they can get more accurate and better quality information);
  - Openness to new airline's information systems without impacts on deployed applications.
- Scalability
  - Performance requirements depending on airport figures allowing also little airports to gain advantages of an integrated platform;
  - Use of techniques to exploit clustering and multiprocessor architecture;
  - Distributed and multi-tiered architectures able to be balanced on a server pool.
- Interoperability
  - Presence of an integration information model defining airport metadata and objects exchanged;
  - Presence of a Message Broker allowing multi-technology real-time hub and spoke notifications and point-to-point exchanges.

Furthermore airports have become competitive commercial enterprises that must operate efficiently and create shareholder value [1].

As airport systems become more complex and air travel increases, airport managers must find: [2]

- Better ways to coordinate information distribution across the airport community;
- Manage resources;
- Control expenses and increase profitability.

Use of an Integration Broker allows a structured approach to system interfacing, creating an environment here information exchange between systems becomes cost effective and less risky.

6 Conclusion

Low cost, low risk integration of airport systems is now possible using standardized application services. A new breed of airport Integration Message Broker provides interface services, which satisfy the specific needs of airport integration, including the distribution of real-time flight information. Based on open XML, Java and Web technologies and incorporating an underlying Integration Message Broker product, these services are flexible and bring plug-and-play control to the hands of airport staff [2].

Legacy adaptors integrate existing systems with new in a common manner, which minimizes cost and maximizes benefits in high-risk field of system integration. Integration with key airport and service partner systems helps to increase productivity by reducing the number of manual inputs and hand written reports.

Nevertheless Message Broker Solutions will be continually developed with new functions based on the continuously arising needs of the airport and of course the passengers.

References

1 **Kelemen Zs**, *Recommended Process for the Airline Informatics System Development*, Periodica Polytechnica ser. Transp. Eng. **29** (2001), no. 1-2, 137–145.

2 ______, *Latest Information Technology Development in the Airline Industry*, Periodica Polytechnica ser. Transp. Eng. **31** (2003), no. 1-2, 45–52.

3 *Message Broker – Air Cargo Edition – Supporting the change in cargo management applications*, Cap Gemini Ernst & Young Advanced Development Center, 2006. White paper.

4 available at `http://www.answers.com/topic/messaging-middle\discretionary{-}{}{}ware`. 30th September 2007.

5 available at `http://www.xybase.com/`. 30th September 2007.

6 available at `http://www.swdes.it/portal/page/portal/swdes`. 30th September 2007.