A MAGAZINE FOR AIRLINE EXECUTIVES 2006 Issue No. 2

ascend

taking your airline to new heights

the global advocate

A Conversation With...

Giovanni Bisignani
director general and CEO
International Air Transport Association

page 38

INSIDE

6	g overnment regulations affect globalization
42	l atin American carriers grow regionally
50	AirAsia overcomes challenges to its t hai-based subsidiary

wearelistening@sabre.com

At Your Service

Photo by Jan Stadelmyer/Shutterstock.com

As the airline industry becomes more global, carriers will increasingly need interoperable technology, which can be realized through the use of services, such as Web services and open application programming interfaces.

By Tony Brice | *Ascend* Contributor

By now, the positive effects of air transport liberalization have been well documented and, for the most part, appear to be generally accepted. The findings of recent studies cite traffic growth resulting in job growth and subsequent increases in gross domestic product for the country pairs involved.

The economic benefits of alliances, cross-border ownership and open-skies agreements are often obvious but so, too, are the resulting technology-related challenges. Today's world is, and will remain for the foreseeable future, technically heterogeneous. For all but a few of the newest air transport companies, it's likely that the current technical environment is a combination of old and new as well as proprietary and open source. In many instances, production system environments are simply reflections of how a company and its technology vendors have evolved through generations of hardware and software platforms, applications enhancements and industry-mandated changes.

The disadvantages of technical complexity are all too familiar to anyone who has ever been charged with leading a large multi-company integration effort. The high-level objective is clear. The sooner participants in any new venture can shift their focus away from technical challenges and onto the real business opportunity — profitably transporting passengers or cargo — the greater the likelihood of success. The technology imperative in most cases is to quickly and efficiently complete the work required to make two or more dissimilar systems work with one another in a prescribed manner.

Technology Integration: the Long Wait

Increasingly, these business needs result in placing a premium on interoperability. Even if participants desire some other level of integration as an end state, interoperability provides interim advantages since it allows a new venture to begin operating sooner rather than later. In fact, interoperability might provide significant temporary advantages even in situations where one company's technology will eventually be phased out in favor of another's technology.

For not only the air transport industry but also virtually any industry where business-to-business integration is a necessity, improved methods of achieving interoperability are welcome. New options for achieving interoperability are now available thanks to the emergence of "services." In a relatively short period of time, the range of services has grown quickly from those designed for "internal-use only" to controlled, "private" Web services to open application programming interfaces, or APIs. Each new type of service has evolved, to some degree, based on the knowledge and experience gained from the earlier phase. Perhaps more importantly, however, each also evolved to address new categories of users and business opportunities.

How do new approaches to interoperability help the air transport industry chart its way through liberalization? To answer that, it may be useful to look briefly at how services have evolved as a means of predicting where they are going.

Services — Past and Present

Much of the momentum toward services is the result of companies having embraced the potential of service-oriented architecture. The idea of SOA was first proposed in 1996 by Gartner analysts Roy W. Schulte and Yefim V. Natis. Prior to the currently accepted definition of SOA as a software design principle, the move toward it as a concept had actually been underway for several years. In its early forms, however, it was implemented through object, procedure or message orientation in a way that tied it too closely to specific execution environments, such as CORBA, COM/DCOM and J2EE. These early efforts facilitated interoperability within a given execution environment, a step forward for many companies at the time. Unfortunately, this approach fell short in that it made interoperability with components outside the environment a comparatively daunting task.

Photo by Marina/Shutterstock.com

By connecting to Web services, airlines can access a range of technology, opening new opportunities to facilitate interoperability.

In the last three to four years, developments in the software industry have driven the first true wave of service orientation. Within air transport and many other industries where legacy systems are prevalent, there are two major categories of drivers — the need to replace older systems and the need to enable new innovation.

Photo by Yuri Arcurs/Shutterstock.com

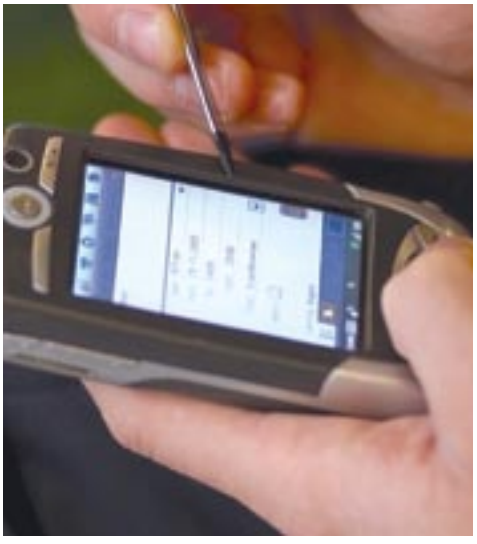

Photo by Yuri Arcurs/Shutterstock.com

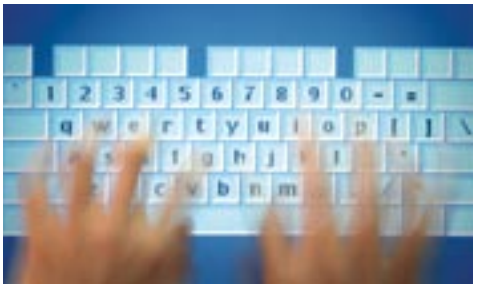

Photo by iStockphoto.com

Photo by JDaniela Andreea Spyropoulos/Stockphoto.com

The recent trend toward self-service continues to grow, particularly through personal user interfaces, which are increasingly available on the Web. Evolving devises such as the next generation of mobile phones provide access to context-related business functions that can be accessed anytime, anywhere.

The initial driver for much of the air transport industry was the need to reengineer older applications to reduce costs, deliver new functionality or, as it turned out in many projects, both. In some instances, entire applications have been rewritten as composite rather than monolithic applications. This afforded the opportunity to also deliver each discreet component as a service. In other cases, selected aspects of an application were rewritten, typically those supporting either user-facing transactions or external applications requiring new functionality via a service interface. The partial rewrites often involved "wrapping" remaining legacy functionality in a way that left it service enabled. An example would be a legacy connectivity component, such as EDIFACT, where there was little or no justification for change.

Another driver, no less important for the air transport industry over the long term, is the trend toward self service. This is especially true on the Web where personal user interfaces, both portal and non-portal style, continue to emerge. A growing number of projects benefit tremendously from the reuse of business logic across multiple user types and methods of access. User types include leisure and business customers, agents, vendors, and other new business partners. All user types are as likely to be at home, in a hotel, in an airport or on an airplane as they are to be in their office. They are also increasingly not sitting in front of a desktop computer. Instead, they are using their personal digital assistant, phone or laptop computer. The need to access a context-related business function can happen anytime, anywhere.

The ultimate benefit of any SOA effort is in how well its implementation promotes the design of applications that rely on loosely coupled services, each of which is designed to serve a discrete business function. This makes SOA, properly implemented, the perfect fit for multiple clients and multiple channels.

HIGHlight

> Airlines and their service providers are, in some cases, a few years into building and/or reengineering systems to be SOA compliant. The benefits of these efforts, executed properly, will be ongoing.

From Good to Great: Extreme Interoperability

It is important not to confuse the benefit of SOA with the value of SOA. The value is derived from the interoperability of services independent of the underlying platform and programming language. In a truly SOA-compliant environment, the definition of the service interface (the part with which an application interacts) hides the language-specific aspects of the service. This is what enables the use of a service independent of development technologies and platforms, such as Java, .NET, etc.

Even in situations where services have been implemented for internal use only, this has provided significant advantages. Because almost all technology environments that have evolved over a number of years are heterogeneous, the deployment of services has provided the opportunity to develop a new generation of applications — those that leverage advances in hardware, software and development methodologies.

Each new type of service that has emerged in recent years has led to new opportunities to ease interoperability efforts. The first — internal use services — has garnered the bulk of the industry's attention to date and has led to significant streamlining of integration processes within each enterprise including its service provider network. Indirectly, the benefits also extend to users of applications that rely on these services.

The second type, primarily Web services, has led to new opportunities to extend each enterprise. Through Web services, trusted relationships with valuable business partners manifest themselves in the sharing and reuse of selected service-based capabilities.

The third type, the open application programming interfaces preferred by an emerging global community of developer/entrepreneurs, is the newest and, thus far, most underused. In addition to creating the greatest levels of interoperability, such APIs hold the most promise for delivering innovative new applications. Opportunities range from mobile self-service capabilities to creative ways of leveraging social interaction in Web 2.0 (see sidebar).

Airlines and their service providers are, in some cases, a few years into building and/or reengineering systems to be SOA compliant. The benefits of these efforts, executed properly, will be ongoing regardless of whether the work is the result of new system development, simplification initiatives or to lay a foundation for future innovation that is fast and affordable.

The real key to maximizing the benefits of service orientation well into the future, however, is to envision how services will support the social and business environment the industry is speeding toward … not just the world airlines are living in today.

Tony Brice is director of emerging technologies in the labs group for the Sabre Holdings® *business. He can be contacted at tony.brice@sabre.com.*

Photo by Jeffrey Smith/iStockphoto.com

With more mobile, wireless access to the Internet through laptop computers and other devices, Web services can be accessed remotely rather than strictly from the office.

© markus.angermeier@aperto.de

Web 2.0, the second generation of services on the Web, is defined by several key characteristics and components — such as wikis, blogs, really simple syndication, mobility, microformats — that differentiate it from earlier uses of the Internet.

Web 2.0

"I can't tell you what it is … but I know it when I see it!"

This is a typical answer to the question, "What is Web 2.0?" In the absence of any "official" definition, it seems only fitting that the characteristics of Web 2.0 be quoted straight from Wikipedia, itself a Web 2.0 phenomenon:

"The term Web 2.0 refers to a second generation of services available on the World Wide Web that lets people collaborate and share information online. In contrast to the first generation, Web 2.0 gives users an experience closer to desktop applications than the traditional static Web pages."

While the definition of a Web 2.0 application is still hotly debated, it is generally accepted that a Web 2.0 Web site would exhibit some basic characteristics, including:

- The site should not act as a "walled garden" — It should be easy to get data in and out of the system.
- Users usually own their data on the site and can modify it at their convenience.
- Mainly Web-based — Most successful Web 2.0 applications can be used almost entirely through a Web browser: this is commonly referred to by the phrase "network as platform."
- Data returns should be dynamic rather than static, changing depending on variables associated with the user's query (keywords, location).
- An "architecture of participation" that allows users to add value to the application as they use it.
- Some social networking aspects.

— From Wikipedia, "the free encyclopedia"