



Fiorano
Enabling change at the speed of thought

www.fiorano.com

Message Driven SOA -- Enterprise Service Oriented Architecture

SOA to meet stringent market realities

Evolution of SOA platforms

--- Atul Saini

AMERICA'S

Fiorano Software, Inc.
718 University Avenue Suite
212, Los Gatos,
CA 95032 USA
Tel: +1 408 354 3210
Fax: +1 408 354 0846
Toll-Free: +1 800 663 3621
Email: info@fiorano.com

EMEA

Fiorano Software Ltd.
3000 Hillswood Drive Hillswood
Business Park Chertsey Surrey
KT16 0RS UK
Tel: +44 (0) 1932 895005
Fax: +44 (0) 1932 325413
Email: info_uk@fiorano.com

APAC

Fiorano Software Pte. Ltd.
Level 42, Suntec Tower Three 8
Temasek Boulevard 038988
Singapore
Tel: +65 68292234
Fax: +65 68292235
Email: info_asiapac@fiorano.com

Entire contents © Fiorano Software and Affiliates. All rights reserved. Reproduction of this document in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Fiorano disclaims all warranties as to the accuracy, completeness or adequacy of such information. Fiorano shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The opinions expressed herein are subject to change without notice.

SOA TO MEET STRINGENT MARKET REALITIES

Evolution of SOA platforms

Executive Summary

Applications and Data play a vital role in corporate survival today. Businesses have invested significantly in upgrading their IT hardware into massively scalable infrastructures. *Application Integration Platforms* (AIPs) leverage the underlying IT infrastructure, and support enterprise-wide deployments of integrated applications.

While the underlying hardware IT infrastructure has progressed to a true peer-to-peer resource network [1] [2], AIPs have lagged with earlier generations of centralized hub-and-spoke or Integration bus oriented architectures that do not leverage all the compute and storage power available at the end-points of modern IT networks. Resulting operational inefficiencies between the AIP and IT infrastructures lead to greater deployment complexities and costs. Forrester has estimated that the G3500 companies will spend an average of US\$ 6.4-Million *each* in 2003 on integration projects, and further, that only 35% of the integration projects will be delivered on projected budgets and schedules.

Given the enormous opportunity, AIP vendors need to address customer requirements such as:

Simplicity “Can the majority of non-technical corporate users create, manage and easily modify an integrated application under secure and controlled access mechanisms?”

Risk mitigation “I need to leverage my multi-million dollar investments in current proprietary solutions. However, my competitors are deploying the next-generation alternatives. How do I migrate into an affordable, proven standards driven solution?”

Predictability “Code customization and unwieldy complexities are a recurring nightmare. Can the new AIP solutions help me manage my projects on time, at cost? Will they help me reuse all the work going into creating the new solutions?”

This paper provides a perspective on the evolution of AIPs from point products to second-generation solutions such as the **Enterprise Service Bus – ESB**.

First Generation AIP's

The evolution of AIPs can be traced back to Wall Street in the 80s. Stock traders using standalone terminals saw immediate benefits in an integrated solution that could bind their terminals, asset trading data and management systems into a single solution. The early AIPs were a collection of point-products that met the immediate customer requirements.

Several business drivers created a need for the first-generation AIP, key amongst them were:

- The move from a mainframe to client-server based IT computing model
- Increasing Mergers and Acquisitions resulting in heterogeneous application environments
- The internet-explosion leading to B2B and B2C opportunities across corporate firewalls

It became clear that the complexity between components needed to be reduced by reducing dependencies between components. An emergence of standards such as a stateless web, platform neutral Java, and data-agnostic XML began to provide the standards-based interfaces that allowed for a decoupling and hence scaling of individual application components.

As a result, several separations came into play leading to the first generation of EAI solutions, typically in the form of hub-and-spoke architectures [3] or Integration-bus oriented architectures [4]. These first generation

AIPs provided logical separations as follows:

- Data was separated from Transport
- Publisher from Subscriber, and Requests from Replies, and
- Connectivity from Delivery/Guaranteed delivery of messages

For the first time, loosely coupled communication models based on standards began to form the core of AIPs. At the same time, COM and CORBA-based AIPs began to offer a proprietary alternative [4][5].

As businesses expanded along several fronts and the internet brought in global connectivity benefits (and security concerns), this generation of AIPs with their centralized processes and computational engines began to become a critical performance and scalability bottleneck.

In an age of irrational exuberance, customers continued spending lavishly for customizing code and a large number of Integration Servers and software licenses. However, with a reversal of the economy, the first generation of centralized solutions – including support for clustered servers – is running out of performance due to key architectural limitations such as:

Control and Data pertaining to information that flows between integrated applications still flows as a bundle over the Integration bus:

- The publish/subscribe model coupled with multicast mechanisms used render the overall system inefficient with message floods to several subscribers that do not need the messages. The implementations that have weak queuing mechanisms quickly degrade in performance when configured to support guaranteed delivery of messages.
- Security at the access and transport layers is often an afterthought as opposed to being ingrained.
- Most AIPs are looking to extend their functionality by adding a dizzying number of disparate products (often as acquisitions) into a patchwork of solutions. This approach ends up demanding significant management resources.
- Availability is becoming a key factor in most deployments. Deploying dual-redundant messaging busses can increase availability. However, replication increases costs.
- An interoperability nightmare with a plethora of multi-vendor, multi-standards assets.

As a result of these architectural limitations, current AIP solutions are struggling to address deployment requirements such as:

- Application Component level creation and re-usability
- Assembly and integration of the components into prototype workflows
- Ensuring a smooth and zero-downtime migration from prototype to production deployments, monitoring the ongoing usage and debugging flaws efficiently, ensuring that the deployment is bullet-proof secure, enabling non-technical end-users the ability to modify business processes dynamically and under authorized control
- Ensure that extending the AIP is as easy as altering a spreadsheet with macros. Adding a new component in real-time, for example, should be a task that need not require IT intervention each time

EAI software has evolved from its origins in the early 1990s when message oriented middleware emerged as a platform to integrate applications. Since then EAI has expanded to include message broking, business process flow management and workflow. In addition the solution focus has broadened from simply looking at application integration within a company to the needs of e-Commerce (B2C or Business to Consumer) and B2B requirements.

The Enterprise Service Bus (ESB) is a pioneer in second-generation AIP solutions. The defining characteristics of second-generation ESB-based solutions are simply summarized as: A set of built-in features that address modern-day business requirements efficiently, reliably and most importantly affordably.

The evolution of AIPs is illustrated in Figure 1 below, followed by a more detailed discussion of the modern ESB architecture as a representative example of second-generation AIPs.

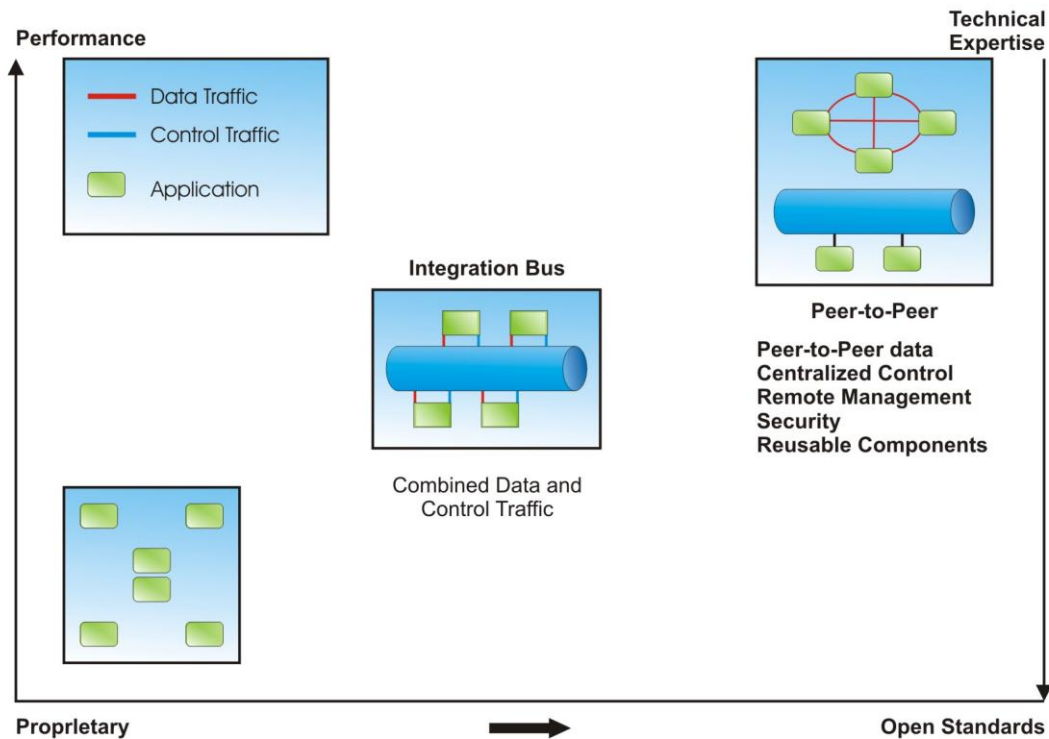


Figure 1: Application Integration Platform (AIP) Evolution Landscape

The ESB: Second Generation AIP's

ESBs enable customers to solve the most significant limitations of first generation AIP solutions as discussed in this section. The largest cost-driver in most integration projects is the fine-grained programming requirements of application components, including adapters. Most current AIP architectures necessitate embedding message routing, data translation and load-balancing engines inside each component. Message routing in particular, tied to the underlying topics/queue configurations also entails painful coding and testing costs that scale exponentially with larger deployments.

ESBs provide for coarse-grain process-level components that can be visually integrated to create enterprise application workflows. By segregating the data-translation from the message routing, An ESBs coarse-grained Enterprise Services are re-usable. Further, Peer-to-Peer Peer ESB infrastructure enables a distributed load-balancing ability that does not have to be embedded into each component.

ESBs take the component-level composition and re-usability to the next level of productivity with *Business Service Composition* tools. By using intuitive and powerful visual drag-and-drop composition tools, ESB Services can be easily and securely integrated into production-worthy workflows. EAI architects can experiment with a subset of their master workflows by tapping real-life data into prototype workflows composed of reusable Enterprise Services, and dynamically modify qualified and tested sub-flows.

Once the workflows are deployed, the application management staff can also use Business Service Composition tools to monitor and debug the workflows. Training costs are significantly lower since multiple stages of an EAI project consisting of prototype experimentation, production deployments and

ongoing management and extensibility can all be controlled via a single tool – the Business Service Composer (BSC) as shown below.

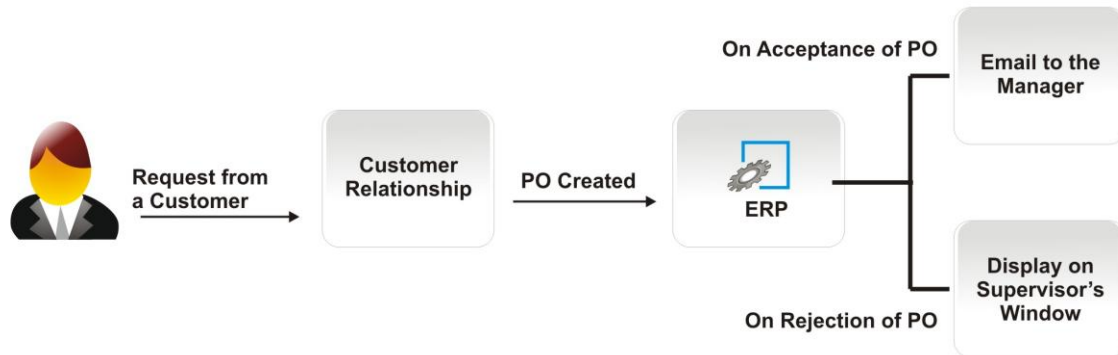


Figure 2: Application Workflow

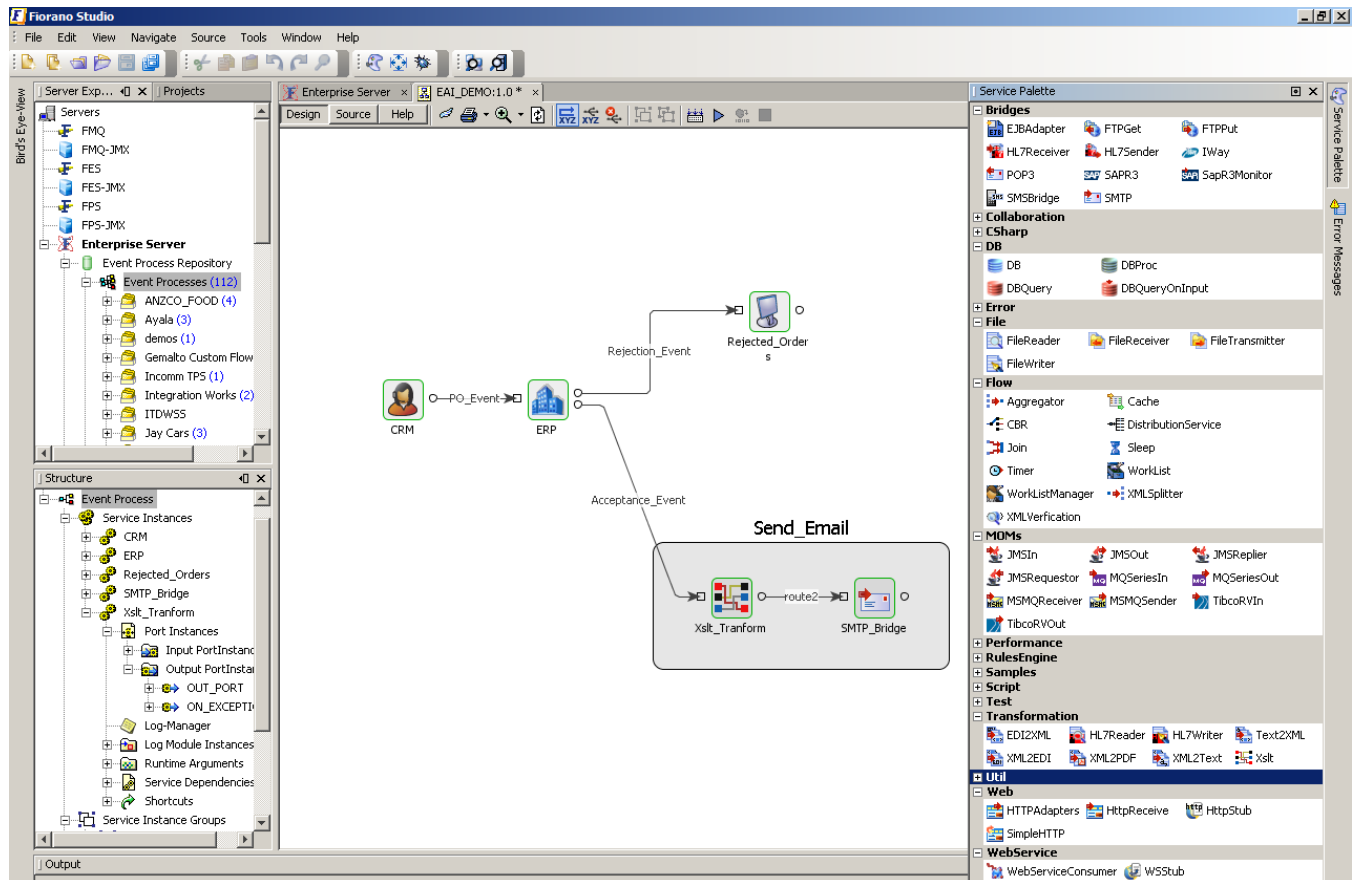


Figure 3: Business Service composition, deployment, monitoring and debugging

ESBs offer innovation at the component reusability level (by using coarse-grained Enterprise Services that segregate message routing from data transformations), application integration and monitoring using BSC tools, and business-process orchestration and extensibility in an easy to use, intuitive usage model for component developers, EAI architects and business managers alike. It is this integration of diverse

functionality into a seamless solution, distributed peer-to-peer scalability, self healing network-level reliability and ease of use that are the hallmarks of a second generation service oriented AIP solution.

Summary

Ever since mainframe computing moved to a client/server model, businesses began to add packaged applications to their legacy applications and serve an increasing number of employees, partners and customers. Application Integration Platforms have evolved to ease the management tasks and boost business productivity by offering a holistic view of all the applications within an enterprise. As client/server applications evolve towards distributed applications and Web Services standards, ESBs offer substantial savings in cost and the time to productivity with their architectural-level innovations

Table 1: Evolution of Application Integration Platforms (AIPs)

Key Business Requirements	First generation AIPs	ESB benefits
Performance scalability	Centralized hub-and-spoke architectures created performance bottlenecks; adding more instances of Integration servers leads to sub-optimal price/performance gains	ESB Peer Servers enable a true peer-to-peer architecture leading to a near-linear scalability in performance at very low fine-grained incremental costs
Risk mitigation via Interoperable solutions	Proprietary messaging-based architectures	Standards-based interoperability including at the messaging layer with other EAI vendors' products
Cost reduction	Component-level: Fine-grained components prevent component reusability leading to high costs per component	Component-level: Coarse-grained Enterprise Services with all peer-routing external to the component enables reusable components
Real-time adaptability to business drivers	Creating new business processes to conduct scenarios that adapt to new business changes could take weeks or months	Using Service Composition tools, business managers could conduct real-time prototype workflows leading to informed decisions; BSC tools enable production engineers to rapidly translate the new prototypes into production level workflows
Business processes need to be secure	Security for most first generation solutions is weak at the AIP levels	Standards-based security is designed into the architecture at the transport (HTTPS, SSL), messaging, component and workflow log levels.
Easy monitoring, debugging and management tools	Most consist of a patchwork of five to ten different products with weak support for visual tools to achieve these tasks	Integrated visual tools that can be used by business managers, and provides easy and centralized management tools for developers, architects and workflow managers.

About Fiorano Software

Fiorano Software (www.fiorano.com) is a leading provider of enterprise class business process integration and messaging infrastructure technology. Fiorano's network-centric solutions set a new paradigm in ROI, performance, interoperability and scalability. Global leaders including Fortune 500 companies such as Boeing, British Telecom, Credit Agricole Titres, Lockheed Martin, NASA, POSCO, Qwest Communications, Schlumberger and Vodafone among others have used Fiorano technology to deploy their enterprise nervous systems.

References

- Freenet website: <http://freenet.sourceforge.net>
- Gnutella website: <http://gnutella.wego.com>

- TIBCO's EAI evolution presentation
<http://www.search.org/conferences/1999symposium/Presentations/TIB%20Evolution%20EAI%20012599.ppt>
- M. CONDIT, Hitachi Systems: *CORBA-based Enterprise Applications Integration*
http://www.omg.org/news/meetings/workshops/presentations/eai_presentations2/conduct.pdf
- N. Byrne, IONA Technologies: Tying it all together EAI and I-Portal
http://www.omg.org/news/meetings/workshops/presentations/eai_presentations2/OMG%20EAI%20Feb%202000.pdf