

# SOA magazine

Service Orientation, the Cloud and Beyond...

Issue L May 2011

## Feature Article



### The Integration Between EAI and SOA - Part II

by Jose Luiz Berg

Published: May, 2011 (SOA Magazine Issue L: May 2011)

[Download PDF](#)[Digg This](#) • [De.licio.us](#) • [Slashdot](#) • [Technorati](#) • [StumbleUpon](#) • [Google Bookmark](#)

This is second part of a two-part article series. The first part is published at [EAI and SOA - Part I](#).

#### Data Model

As previously stated, the data model in EAI deserves a separate chapter. A common mistake in systems integration is to transfer their job requirements as the systems communicate. It is common to hear from integration teams, "they are only the pipe, and don't care about what's going on inside it". In fact, the integration concerns the exchange of information (represented by your data) between systems. This is your goal and that is what adds value to the business. This is only possible through communication, but if the information exchanged is incorrect, or simply arrives at the wrong time, they will be worth nothing to your business. Therefore, an old diskette sent by regular mail with the correct information recorded and delivered at the right time is worth more than the most complex Web service, running at best SOA suite on the market, but that has the wrong information.

Defining the format and content of the data bus is important because handling this data is the objective of any integration. However, defining a data model for the bus is a difficult task because they should be attached to the definition that each entity has to business processes. To create this model, we must first understand and document the business process, which is a time-consuming and complex task involving interactions with the various areas of the company. This also implies that the areas need to be familiar with their business processes, which unfortunately is not usual. Moving forward, this bus data model will be named "corporate data model".



dynamics of change end up causing the updates not to be replicated to the integration.

There are advantages in using the BPM systems and standards. An advantage to this system is that in addition to implement state diagrams, it also allows the construction of business diagrams using a language that enables the business analyst to define their process. These are mapped directly to the systems without the need for a programming layer. The evolution of these systems will soon be how business areas retain their processes, reducing the cost and time of change and minimizing the flaws. This development combined with the new tools incorporated to BPM systems, such as the control of human workflows, rule engines and business monitoring, will give companies an agility and quality never before achieved.

There is another point in favor of using a corporate data model; this model will be the common language that will be used by business processes to exchange information. Without this common language, it is impossible to construct business processes that consider interactions between the various areas of the company.

## EAI & SOA

Conceptualize the main components and objectives of EAI; this will better your understanding of the interaction between EAI and SOA. However, we still need to introduce some concepts about SOA and clarify some common confusion in the market.

Web services and SOA are two completely different concepts. SOA is a set of concepts used to create an architecture and Web service is a standard for remote communication between systems. Typically, the SOA architecture is implemented through Web services, but the concept of service does not specify the technology, and there are numerous other technologies that can be used for implementation.

### NOTE

"Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains." "Note that although SOA is commonly implemented using Web services, services can be made visible, support interaction, and generate effects through other implementation strategies. Web service-based architectures and technologies are specific and concrete. While the concepts in the Reference Model apply to such systems, Web services are too solution specific to be part of a general reference model."

Oasis - Reference Model for Service Oriented Architecture 1.0

Web service is a communication standard that uses XML messages validated by XSD schema definitions, it usually exchange information using the HTTP(S) protocol. It is supported by most existing application servers on the market today by being across all operating platforms, and turning to be the principal pattern for communication, especially between heterogeneous platforms.

A service is a real world action provided through a specification and governed by restrictions and policies established through a contract. The technology used to build a service doesn't matter, as there are other additional concepts. The most important is isolation: the consumers of the service do not need to know what will happen internally, but only the information in the service description.

### NOTE

"A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description"

Oasis - Reference Model for Service Oriented Architecture 1.0

With these concepts in mind, we can now try to understand how SOA and EAI interact, and what the relationship between them is. Moving forward we will be using Web services to exemplify services, but always keep in mind that could be replaced by any other similar technology.

### Communication Services

Within our EAI infrastructure, Web services can be found in two ways:

1. When systems provide Web services as part of your infrastructure
2. When the integration bus provides the Web services. Let's see these two case in more detail:

- System Services

System Web services are found when they have in-house application servers as part of their infrastructure, and offer services that can be used for remote execution of their functions by other systems. Naturally, the features provided by the system and the data model used in services follow the system standards. The main advantage of these services enable the system to handle a job that was supposed to be done by EAI and is that other systems can make services calls directly. The disadvantage of this option is that we go back to the espagueti problem, as all systems will start to run calls directly to each other, increasing the complexity.

- Integration Bus Services

Integration Bus Web services are found when we start to build services that enable systems that do not have the ability to provide Web services, to provide their functionality to other systems through the bus, which will forward requests to the systems using old technologies already used by the integration. In this case, the functionality and data model also follows the system standards, because integration is only providing the technical resources (acting like an application server) to build the Web services.

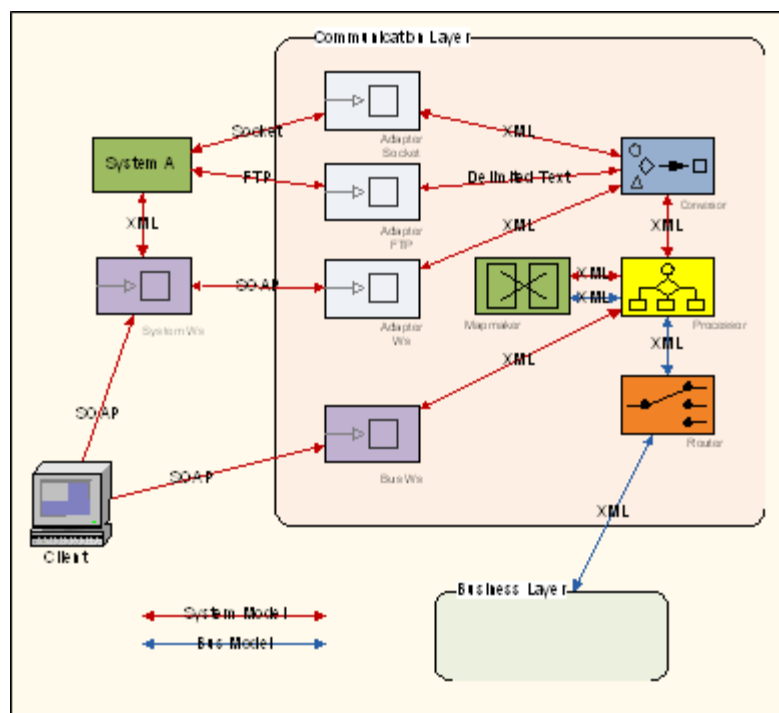


Figure 2 - Communication Services

One important point is that these services are in communications layers, so they are called "communication services". If we want to build a service in the business, the service starts to become more interesting, and similarly to the other layer, we will call then "business services".

### Business Services

We saw that in this layer, the tasks performed are the company's business processes, and the data model is the corporate model. Then we have a service that performs the business process, using common definitions adopted for the company. More importantly, this process is independent of the systems that are connected to the bus, because whatever they need, the business process will forward the request to the correct platform, and the communication layer will be responsible for making the requests to the correct system.

Existing services in this layer are quite powerful, because they correspond to the official process of the company.

They are not changed by infrastructure problems and changes, and are independent of the systems that will execute the request. Then, once the clients are using one of these services, they will know that the task will be performed correctly and homogeneously, and that if a business process is changed, the new version will be automatically used by all clients simultaneously.

The goal of a SOA infrastructure: build a set of services that execute business rules, and that can be used by client applications, enabling a homogeneous and centralized vision of business process by abstracting the technical details, and ensuring that corporate objectives are achieved.

Let's see the almost final model of our integration architecture:

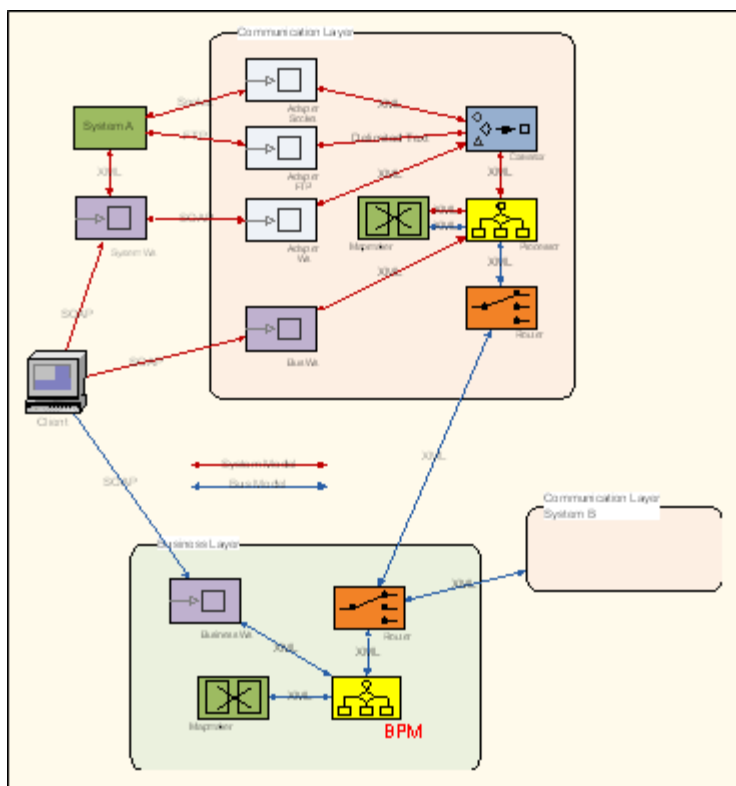


Figure 3

The final goal of SOA is to use only business services; it no longer makes sense to have communication services with the same functionality published to clients. Then the communication services exist only for the integration bus to use in communication layer. However, we know that the Web services are not very efficient compared to other technologies. Its main advantage is compatibility, but if we have only the integration bus performing calls, this advantage is no longer relevant, and we can return to using other technologies more efficient to communicate with systems. With this change, we now have the final model of the integration architecture:

With this change, we now have the final model of the integration architecture:

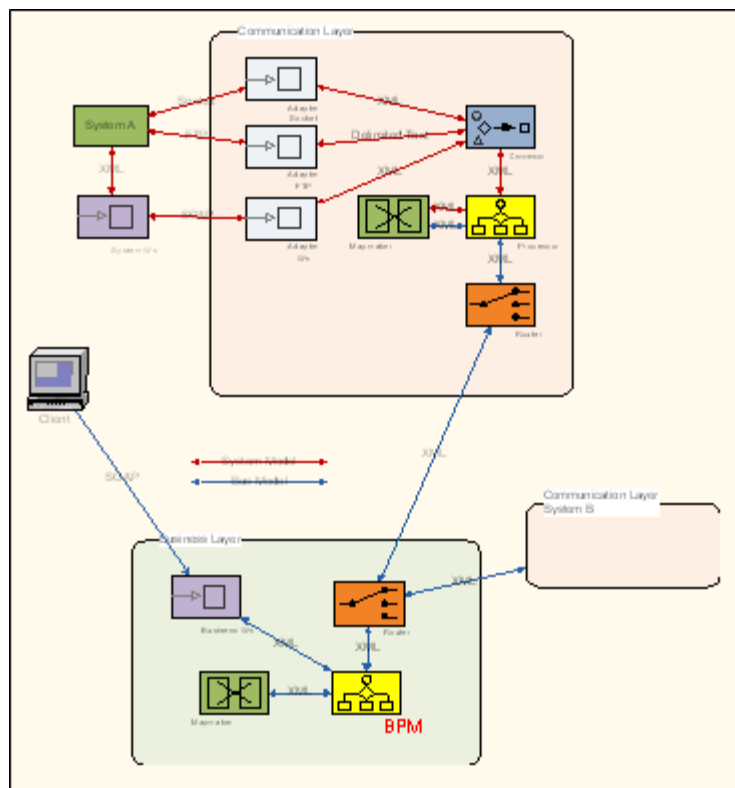


Figure 4

With the implementation of business services, a problem occurs for the applications to call business services using the corporate data model, they will have to make mapping from its data model to the corporate model, and vice versa. This adds to each application a level of complexity that only exists because of SOA architecture, which was previously provided by EAI. Each system has its degree of complexity and flexibility to deal with this problem, and there is no definite rule for its resolution, but some guidelines can be followed:

- when a system participates in a business process it usually uses the communication layer, and forwards its data in the native model for integration
- when a system makes calls to external processes without participation in it, it usually uses the business services, and need to manage the mapping of data

In this case, a possible solution is to build a mapping service that could receive data in the system model and return in the corporate model and vice versa; being treated as another service provided by integration. It may seem interesting at first glance, but this approach has performance and availability constraints that need to be addressed very carefully before implementation.

Always remember the basic concepts of integration: there are no simple solutions to complex problems.

## Conclusion

In this document, we review the history of EAI, and describe the main blocks that make up its structure, showing where and how SOA infrastructure is connected to. With this vision, we can realize that rather than replace the integration, service oriented architecture fits and complements the integration architecture and the modern "SOA Tools", despite having many conceptual and technical enhancements, have all components before presented in traditional integration tools, and in reality exists as an evolution of then.

The conclusion therefore is that the two visions were complementary and fundamental for the construction of a system architecture that meets the needs of a modern company, using the new features and functionality without forgetting the old concepts and standards, by taking the best advantage of each one and fulfilling its role of bringing agility and quality for business.

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL



[Home](#) [Past Issues](#) [Contributors](#) [What is SOA?](#) [SOA Glossary](#) [SOA School](#) [SOA Books](#) [About/RSS](#)  
[Legal](#)

Copyright © 2006-2010  
SOA Systems Inc.