Enterprise Service Bus - A Backbone for SOA

RĂZVAN DANIEL ZOTA, RADU STEFAN MOLEAVIN, LAURENŢIU CIOVICĂ
Department of Economic Informatics and Cybernetics
Academy of Economic Studies Bucharest
Calea Dorobanţilor nr. 13-15, sector 1, Bucharest, ROMANIA
zota@ase.ro, ramos_cisco2003@yahoo.com, laurentiu.ciovica@gmail.com

Abstract: - Nowadays, enterprises are using multiple specialized applications to solve different aspects of their business, from simple content management systems to complex financial or business intelligence applications. These different specialized applications were developed using different communication protocols and programming languages. During time, business requirements and goals are changing continuously and along with them the application complexity is also raising. Many applications have to be integrated together in order to communicate and to share resources and data. A big help in dealing with these challenges is the new architectural style called SOA. The SOA, which can be extended also at business level, ease the bridge between business requirements and software. In order to assure a neutral language communication and resource sharing, the Enterprise Service Bus (ESB) pattern is used. In this paper we describe the main advantages of an ESB model along with its importance for a SOA approach.

Key-Words: - ESB, SOA, Web Services, Business Processes

1 Introduction
Economical challenge but new business opportunities out there created by the technological evolution. Many companies nowadays have a great success and a good economical evolution due to a good adoption of latest techniques, technologies and architectural styles.
But also, many other companies are putted in their knees, and sometimes bankrupted, due to wrong approach using different technologies.
During past couple of years, a new architectural style was defined and compared to others, had a great success being adopted, with success, by many companies being also supported by them. This new style is called SOA – Service Oriented Architecture. The main concept behind is that, everything is seen as an atomic service, capable in executing a single particular thing.
These services have to be capable in communicating in a language and protocol neutral way. This communication channels have to be standardize and enhanced by introducing a specialized unit in protocol and message translating and conversion, routing, dispatching and delivering them; this unit is called Enterprise Service Bus (ESB).

2 Service Oriented Architecture
There are many definitions within the literature, but they have some common grounds: service oriented architecture is an architectural style where the entire system is built using atomic, independent and reusable services with the scope of increasing availability, performance, flexibility and scalability. This is the perfect bridge between business and IT because, like in the real world, where everything is a service, within the software system every task / job can be seen as a service. By creating this new cultural environment, the business people can easily communicate their needs, goals and requirements, which can be translated in software tasks and user stories by an architect.

The main SOA layers are:
- Operational Systems Layer
- Enterprise Components Layer
- Services Layer
- Business Process Layer
- Presentation Layer

The main SOA components are [1]:
- Service Consumer
- Service Provider
- Service Registry
- Service Contract

Strategic goals and benefits of SOA are [2]:
- Increased Intrinsic Interoperability
- Increased Federation
- Increased Vendor Diversification Options.
- Increased Business and Technology Alignment
3 Enterprise Service Bus (ESB)

Within an enterprise, different applications – different from point of view of usability, field, complexity – are used to fulfill its goals, requirements, cut costs and increase ROI. These applications are developed in-house or bought from third parties or even inherited as legacy systems. In order to make them communicate and use the same resources many companies adopts the SOA as the main strategy and if implemented right they succeed in their mission.

From the IT and business perspectives a good infrastructure must be designed and implemented in order to have a correct adoption and implementation of SOA.

This infrastructure has to be able to support resources distribution, message end-points discovery, message dispatching, routing and delivery. Many companies fail in a good SOA adoption due to the fact they omit to take in consideration the installation and configuration of an ESB.

The ESB is nothing else than a pattern which uses web services, intelligent routing and message transformation and services orchestration in order to make services more flexible and re-usable. Thus, the company is not just prepared for the short run but also for the long run, for the future technological changes and evolution.

According to [3] an ESB provides functionality for five functional areas as follows, as shown also in Figure 1.

- Increased ROI
- Increased Organizational Agility

- Architecture
- Connection
- Mediation
- Orchestration
- Change and Control

Some of the most important services and features of an ESB, according to [4], are:

- Transformation, Augmentation and Mapping
- Conversion
- Routing
- Invocation

In [5] are described the main service platform features which are supporting by an ESB:

- Service hosting
- Service component model
- Orchestration
- Security
- Service Registry
- Auditing
- Monitoring
- Resource adapters
- Composition

An ESB provides a loosely coupled, highly distributed approach to integration [6]. The most important feature of an ESB is that that it has the ability to be highly distributed and thus providing highly availability in such environments. This feature is defined, in fact, by three main componentes: abstract remote services’ endpoints Service Container, Message-Oriented Middleware (MOM) and a Management Facility.
4 Message Oriented Middleware
The Message Oriented Middleware (MOM) is the most important component of an ESB [7], representing a highly distributed network of message servers; this leads to its name, the backbone of an ESB. The relations between MOM, service containers and management facility are shown in Figure 2.

MOM permits to establish reliable, secure and manageable virtual channels and to transmit messages over them.

A messaging server manages various queues and topics and is able to store messages sent to those. They are either used to realize a point-to-point (queues) or publish-subscribe (topics) messaging model.

In the first case the sender sends messages to a virtual queue which is managed by a messaging server. The messaging server stores temporarily the received messages. To receive messages, a receiver can connect to a queue and fetch the oldest message. The oldest message in the queue will only be delivered to first receiver which fetches it.

In the latter case, the publisher writes messages to a topic. Message subscribers can subscribe to that topic in order to receive the published messages. In this case, the messaging server manages the virtual topic and for each subscriber a private queue in which the messages are stored.

ESB is composed from multiple interconnected messaging servers. The MOM routes the messages through this network of messaging servers via a store and forward mechanism. Each messaging server, from one route, stores the message, tries to send it to the next messaging server and deletes it only if the target server has acknowledged the reception, guaranteeing the message delivery.

Each message client is connected to a messaging server and runs inside a service container which is able to send and receive messages to/from this messaging server. The service container actually manages the client and takes care of transforming the received messages into service invocations.

In an ESB, a message is the base unit of transaction. A message consists of a header, properties and a body. The header contains identification and routing information. The properties are structured in a key-value pairs, allowing passing application specific values. The body contains the actual payload of the message.

The ESB uses a standardized message exchange protocol, meaning that the messages are sent in a normalized format. Therefore, on a business service invocation, they might have to be transformed from the normalized format to the format required by the business service and vice versa.

So far, in the paper was presented a quick overview of the basic functionality of an ESB, but the goal of an ESB is to integrate all kinds of isolated
applications into a decentralized infrastructure to provide the business functionality as reusable business services, to create, automate and integrate business processes using them, and to manage and monitor the created business processes.

An ESB is used mainly for the integration of the message exchanges and in providing mechanisms to route messages through the bus, by invoking multiple business services. In other words, it is used for itinerary-based routing and service orchestration using BPEL.

In order to route a message through the bus, each message contains an itinerary, consisting of a list of ESB endpoints that have to be visited. A BPEL process definition consists of a number of logical steps that are connected to each other by conditional or unconditional links and can be executed in sequence or in parallel. As in the itinerary based routing, each logical step refers to an ESB endpoint.

A service orchestration or BPEL engine is used to enact BPEL processes based on the process definitions. The BPEL engine simply sends asynchronous messages to and receives asynchronous message from the MOM. Depending on the kind of logical step, it can thereby invoke a business service or interact with a business process managed by another BPEL engine. The mechanism of invoking the business service is adapted by the find-bind-involve mechanism, as shown in [7].

5 Conclusions

SOA is commonly an architecture style that builds on loosely coupled, interoperable and composable components or software agents called services. The goal of SOA is to increase the alignment between business and IT and achieve business agility – the ability to respond to changes quickly and efficiently. Since a single true SOA requires the business services to be independent of each other, it is imperative to have a mechanism in place to enable these components to be assembled together.

The Enterprise Service Bus (ESB) architecture is a message based, distributed integration solution which provides integration services. Some of the basic integration services that we may take into consideration are: routing, invocation, mediation, support for transactions, logging, auditing, security and management services. Technically speaking, ESB represents a messaging backbone which does protocol conversion, message format transformation, routing, orchestration, accept and deliver messages from various services and applications, which are linked to ESB. Besides these functionalities it also provides a consistent approach to integrating applications. It is standards based, supporting web service specifications, and is a pluggable framework where any new service can easily be plugged into the bus. This helps in agile integration [8] and today the Enterprise Service Bus is the most popular and successful infrastructure for implementing Service Oriented Architectures.

References: