Intelligent Conceptual Message Routing in Enterprise Service Bus (ESB)

Amir Massoud Bidgoli[1], Payam Nabhani[2]

[1]Islamic Azad University Tehran North Branch. drbidgoli@gmail.com
[2]Islamic Azad University Science And Research Khozestan Branch. Ahwaz,P.Nabhani@gmail.com

Abstract A lot of researches have been done in Message Routing in Service Oriented Architecture (SOA). The main part of SOA is called Enterprise Service Bus (ESB) and services should be routed in this part. Message Routing is the primary functionality of ESB and message routing algorithms are very crucial. Many algorithms and method are announced till now but most of them are tight coupled to message using static routing and working in flat or liner structures. Intelligent Conceptual Message Routing (ICMR) is working based on ordered pairs of conceptual service metadata which is issued by service provider and they are stored in a tree. The request of service consumer is translate into ordered pairs and then searched by intelligent agent will be able to find appropriate service. In our work we concentrate on the efficient routing method that can be find appropriate services by defining the parameters by requester on the fly.

Keywords SOA, ESB, Message Routing, Middleware, Service, Intelligent Routing.

1. INTRODUCTION

Enterprise applications are increasingly being architected in a service-oriented architecture (SOA) style, in which modular components are composed to implement the business logic. The properties of such applications, such as the loose coupling among the modules, is promoted as a way for an agile business to quickly adapt its processes to an ever changing landscape of opportunities, priorities, partners, and competitors. The proliferation of Web services standards in this area reflects the industry interest and demand for distributed enterprise applications that communicate with software services provided by vendors, clients, and partners. [1]

The growing complexity of enterprise applications has favored the adoption of the architectural principles behind the Service-Oriented Architecture (SOA). The wide range of functionalities available through-out the enterprise network, often from legacy applications or systems, is encapsulated in services at the points where the functionalities are implemented. Once some functionalities (whatever its underlying technology, implementation or other details) are available as services, they can be reused in other enterprise applications. An SOA favors the construction of applications around independent building blocks (services), which can be arranged in different ways to create new applications. To enable maximum flexibility in the combination of services, these should be stateless, such that successive invocations of a service are independent from each other. At the core of such an architecture, there is an ESB (Enterprise Service Bus) enabling the operation of the applications by allowing the services to be invoked and to communicate with each other in a location-independent fashion. The ESB may also interconnect services that use different protocols and data formats by leveraging mediation functions. At the core of an ESB implementation there is at least one form of middleware that establishes a lingua franca used by services in the exchange of information, regardless of the actual location or invocation requirements of other services. This middleware may use a number of mechanisms, such as message queues, publish/subscribe messaging or Web Services (WS). WS provide well-defined standards that make them ideal for interoperability.[2]

Part of SOA is the infrastructure that allows you to use services in a productive system landscape. This is usually called the enterprise service bus (ESB). There are different opinions about the exact role and responsibilities of an ESB. Part of the reason for the different understandings of ESBs is that there are very different technical approaches to realizing an ESB. To run SOA in practice, you need a way of calling services. This infrastructure is the technical back-bone of the SOA landscape (sometimes also called the "backplane"). [3]

It is the responsibility of the ESB to enable consumers to call the services providers supply. This sounds simpler than it is. Depending on the technical and organizational approaches taken to implementing the ESB, this responsibility may involve (but is not limited to) the following tasks:

- Providing connectivity
- Data transformation
- (Intelligent) routing
- Dealing with security
- Dealing with reliability
- Service management
- Monitoring and logging

These tasks may need to be carried out for different hardware and software platforms, and even for different middleware and protocols. The ESB's main role is to provide interoperability. Because it integrates different platforms and programming languages, a fundamental part of this role is data transformation. Another fundamental ESB task is routing. There must be some way of sending a service call from a consumer to a provider, and then sending an answer back from the provider to the consumer Depending on the technology used, and the level of intelligence provided, this task may be trivial, or may require very complicated processing. [3]

The ESB should be equipped with routing mechanisms to facilitate not only topic-based routing but also, more sophisticated, content-based routing. Topic-based routing assumes that messages can be grouped into fixed, topical classes, so that subscribers can explicate interest in a topic and as a consequence receive messages associated with that topic. Content based routing on the other hand allows subscriptions on constraints of actual properties (attributes) of business events. The content of the message thus determines their routing to different endpoints in the ESB infrastructure. [4]

Our work is about to define a new mechanism to routing the message through ESB. The mentioned method is a Conceptual Routing Message that could find the appropriate service intelligently.

The rest of this paper is organized as follow. Section 2 introduces related work of message routing .The proposed Conceptual Message Routing method is discussed in Section 3. Section 4 gives a sample built by using ICMR. Conclusion is made in Section 5.

2. Related Work

A. Content-Based Routing [5] [6] [7]

Content-based routing (CBR) is a message driven dynamic routing mechanism and considered as a necessary feature on ESB. In CBR, the routing path is determined by analyzing the message and applying a set of predictions to its content and it usually depends on a number of criteria, such as the message type, specific field value, existence of fields, and so on.

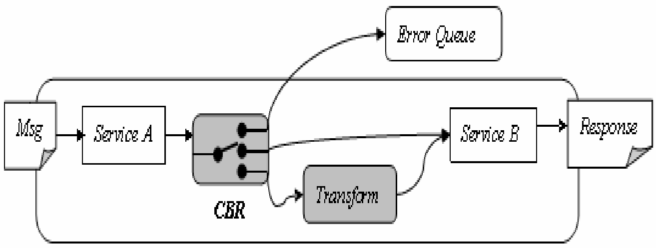

Fig. 1 CONTENT-based Routing on ESB

B. Pattern-Based Dynamic Routing [5] [8]

In conventional routing models, a service is usually maintained by single service container or unknown external server and their composition capacity usually depend on Web Services or BPEL, which are based on XML technology that has well interoperability but well-known performance drawbacks in high volume or intricate applications. Pattern-based dynamic routing (PBDR) [5] focuses on this issue and introduces application pattern (AP) concept to improve current solution. AP is the key to support the multi-service container responsible for service orchestration in PBDR since it defines a set of interoperable services and the relationship among them, so it can helps to obtain a routing path for message delivery. Generally, an AP instance maintains a steady composition application for PBDR module. Then by analyzing a received message, PBDR will choose an appropriate one and invoke the services involved in to process the message.

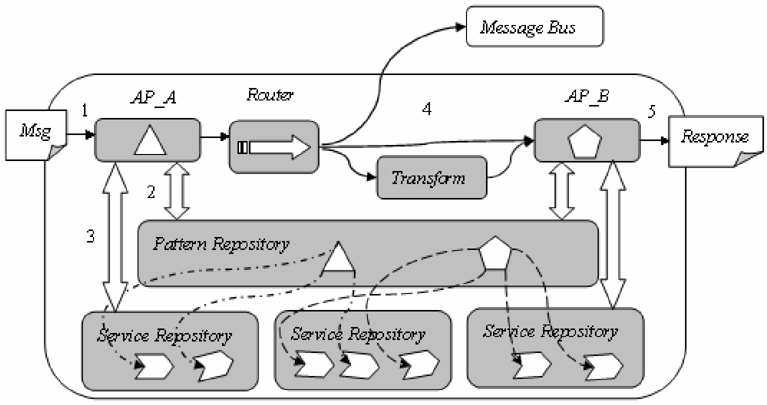

Fig. 2 Pattern-based Routing on ESB

Compared to CBR, PBDR (shown in Fig.2.) replaces the service node with AP and it has a pattern repository and service repository for AP and service registration. PBDR process a message in five steps:

1. When receiving a message, PBDR validate the message header and deliver it to an appropriate AP container (here is AP_A).
2. AP_A will select an AP instance form Pattern Repository by applying a matching predication to the message. The Pattern Repository maintains all the AP instances defined on an ESB.
3. After selection, a service executor module instantiates the related services through Service Repository, which is responsible for service abstraction, storage and reconfiguration, and invokes them to process the request.
4. A message may cross several AP containers that are linked by some router and transformation components on ESB. This step is similar with the conventional message process procedures.
5. Finally, a response is sent out to reply the request message. To provide relatively independent services for workflow layer, the ESB should manage some composition services with mediation flows. Moreover, ESB provides service containers that can host multiple services in a black box. PBDR follows the above two principles and provide such a more controllable and flexible routing solution.

C. Dynamic routing in ESB [5] [9] [10]

ESB provides the services for messaging transformation, asynchronous message passing, and message routing. The generic ESB architecture is made up of four parts:

- Message mechanism provides transparent communication services to the applications.
- Message transformation translates the messages of various formats and contents. It hides the heterogeneous in the messages and protocols.
- Routing mechanism establishes the service connections

according to the workflow and navigates the messages from one to the next.

• Container hosts services and provides support for selective deployment, service invocation and lifecycle management.

The key for DRESR to support dynamic routing is to separate service specification from service implementation in the routing path definition. DRESR supports service routing from three perspectives: routing path abstraction with ARP (Abstract Routing Path), instantiation with IRP (Instantiated Routing Path) and reconfiguration. An ARP is a description of the routing path in terms of abstract service names. An IRP maps the abstract service name into the URI of service provider. Reconfiguration reflects the changes in business process and service process. Fig. 3 gives an overall view of the DRESR approach.

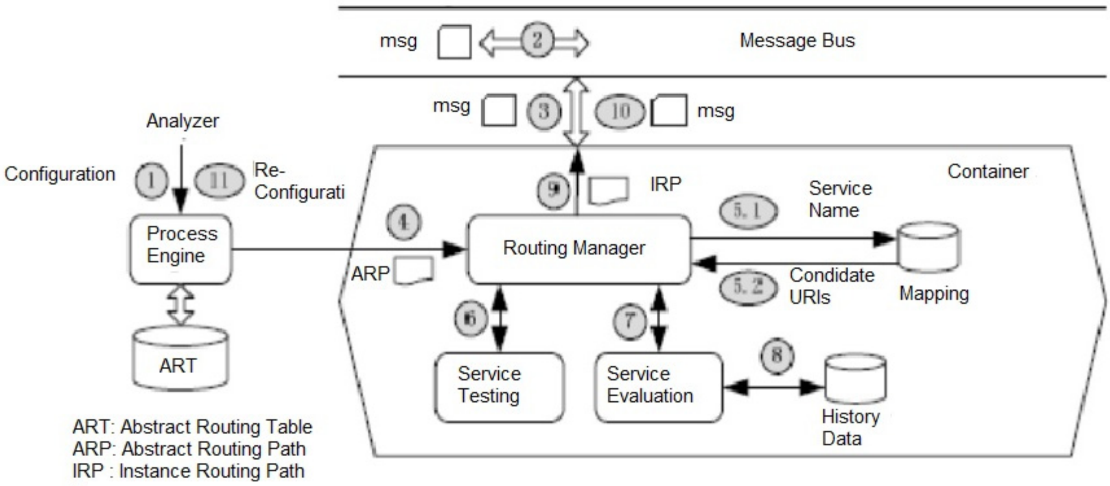

Fig. 3 Dynamic routing in ESB

D. Multifactor-Driven Hierarchical Routing

Multifactor-Driven Hierarchical Routing is a hierarchical routing model to synthesize the PBDR routing solution and introduce a domain specific language (DSL) on top of PBDR layer to support enterprise integration patterns (EIPs). As shown in Fig.3, a Message Filter is configured in MDHR to do a message sorting for the routing modules that have three routing levels: basic level; application level and business level.

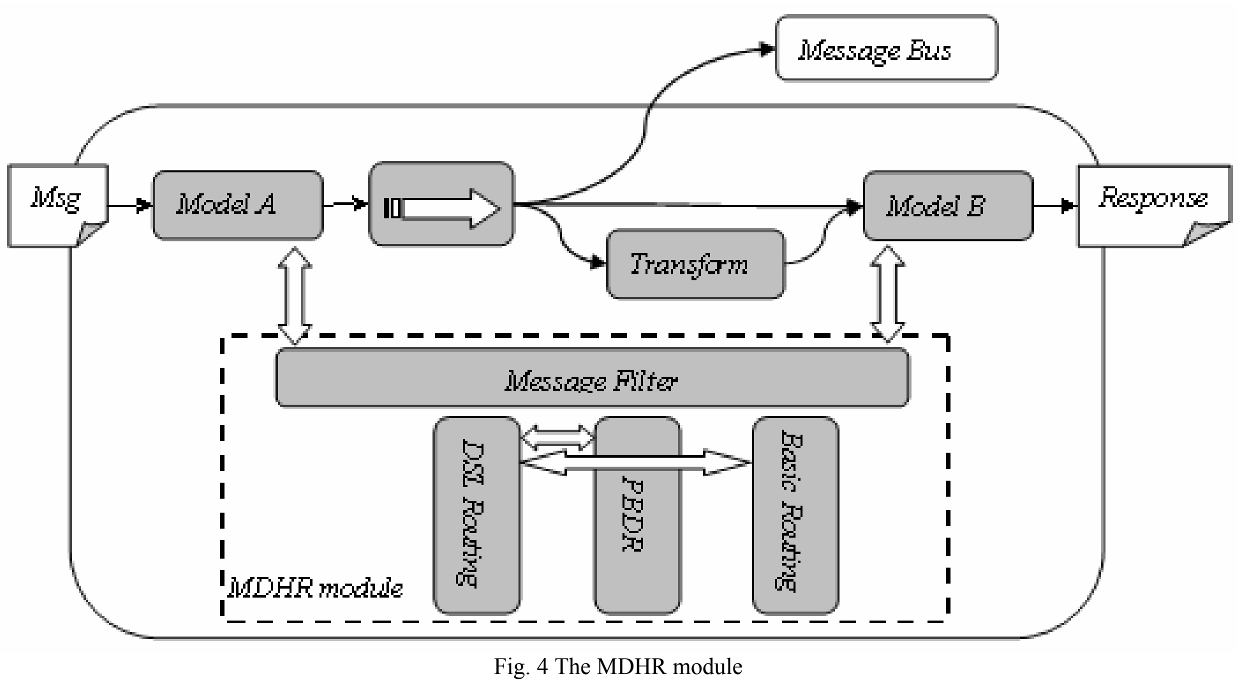

Fig. 4 The MDHR module

3. Problems of current routing solutions in ESB

Message routing is a very important service communication mechanism. The service requester should not be care which service provider can receive his messages and respond to him .In such condition the requester don't care about service provider anymore and the loose coupling about SOA substantially implemented. So, a kind of tool software to make this function work for dynamic service evocation in the SOA is needed. But the challenge is that how to decide which service provider can be used; is the key problem for it. With the development of the SOA, there are many middlewares designed to fulfill it. By analyzing current message routing in ESB, it could be considered that most of the existing methods are not support dynamic message routing in efficient way. In the real world we need a mechanism to find the service without know from where it comes dynamically. It means that, to complete a business, it is necessary to handle all locations of the valid service providers and access manner at first

4. Intelligent Conceptual Message Routing

Intelligent Conceptual Message Routing (ICMR) is based on tree structure and it illuminated in Fig. 5

E. Service Registration Unit (SRU)

This component is used for registering published services. When a service provider publishes new service, this component is responsible for getting all information about the service and the functionality of it.

F. Service Location Zone UNIT (SLZU)

This component is responsible for analyzing the content and the functionality of the resisted services and the assigning an ID to it. The ID is a unique number that shows the zone of the service in the tree. The second application of this unit is to check whether the service is available or not.

G.Service Delivery Unit (SDU)

This unit is responsible for getting the request from the service consumer and sends it into Service Explorer Unit (SEU) in order to find the appropriate service and to deliver it into consumer.

H.Service Explorer Unit (SEU)

This unit is playing great role in this model. The main functionality of this unit is to get the properties of the required service and find the suitable service according to the requested properties. In order to carry out this task, the SEU using intelligent algorithms to find the service and deliver it. SDU has a Message Agent Module (MAM) which is responsible for ranking the service base on the request and the quality. When one service is requested frequently, the MAM increase its ranking and it will be candidate in next service discovery at the same condition. MAM is using AI algorithms such as Genetic Algorithms to do the ranking and service delivering. In this way the service delivery will be delivered at the shortest time for the frequent requesting service.

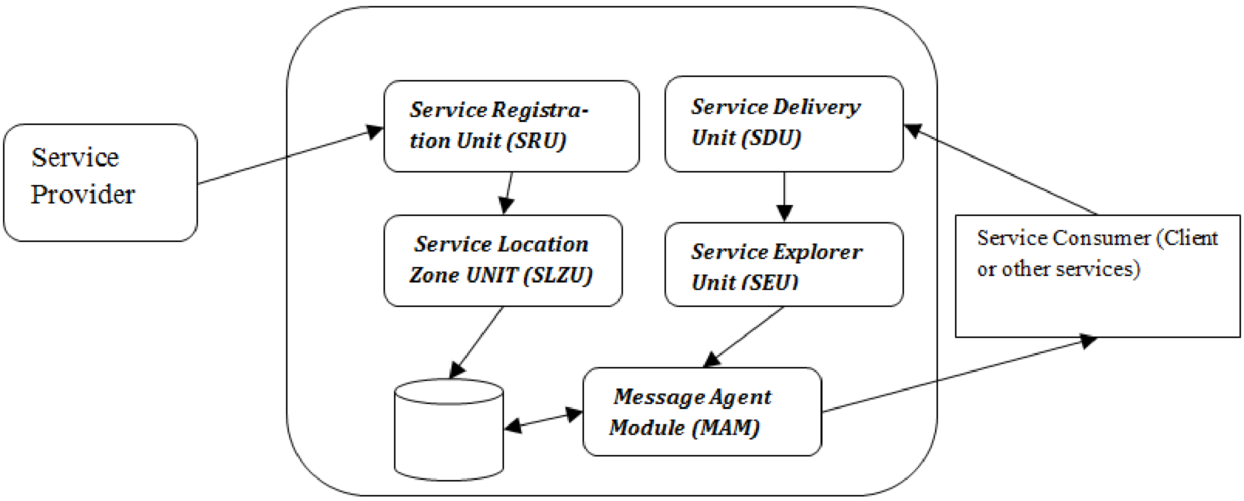

Fig. 5 Intelligent conceptual message routing

5. ICMR Vs. other routing methods

TABLE I shows the differences between our methods and other Message routing methods

TABLE I
ICMR VS. OTHER ROUTING METHODS

Routing Method	Advantage	Disadvantage
Content-Based Routing	• Dynamic routing in run time • Routing based on content of the message [5] • parallel processing of messages at multiple nodes at the same time	Routing is based on prediction Depends on the number of criteria such as the message type, specific field value, existence of field and so on. [5] It is not yet possible (i) to define SOAP message routes with alternative or parallel processing of messages at multiple nodes at the same time. Additionally, (ii) the sequence of message processing by additional services at intermediaries cannot be predefined, even though it is required for the correct execution of applications in multiple scenarios. (iii) The fact that message routes are predefined and allow no parallelism and branching results also in fault-intolerance, since reaction to failures at nodes on the message path is not supported. [11]
Dynamic routing in ESB	• Dynamic reconfiguration at run time by user and systems [5]	• Based on patterns • The invoked services are not guarantee to be alive [10]
Multifactor-Driven Hierarchical Routing	• Multifactor-driven architecture separates the routing configuration from the application arrangement and enables a more adaptable solution when building the service mediation on ESB. [5] • An important functionality of PBDR is the wide-ranging technologies support. [5]	If developers or deployment managers want to add some model on routing layer, they must turn to hard-code or static configuration that will make the ESB too complex to maintain. [5]
Intelligent Conceptual Message Routing	• Users and applications can communicate with the server provider and invoke the desired services by describing their needs. • Reliable service delivery(the dead services will not be delivered) • Search service in efficient way(i.e. searching by Intelligent algorithms) • Search the requested service based on semantics parameters	• Hard to implement some parts of the components(e.g. MAM and service exploring)

6. Conclusion and future works

This paper proposes a hierarchical routing mechanism with several layers to enhance the routing functionality for ESB. Service Registration Unit (SRU) provides several metadata fields for registering the service to mature routing implementations. Service Location Identification Unit (SLIU) is using a Conceptual data structure to arrange service in the hierarchical structure by assigning a unique ID to each service. Service Delivery Unit (SDU) introduces the unit for getting the message from application and delivers the requested service. Through this Intelligent Conceptual model, Applications can be request Service just by defining their needs ESB can be deliver the appropriate service without low performance solutions, like BPEL related system. The Future work includes 1) service mapping and selection algorithm design; 2) the integration with service testing tools based on Symantec service; 3) ESB performance analysis of the newly introduced mechanism.

7. References

[1] G. LI, V. MUTHUSAMY, H. JACOBSEN, *A Distributed Service-Oriented Architecture for Business Process Execution*. ACM Transactions on The Web, Vol. 4, No. 1, Article 2, Publication date: January, 2010.

[2] L. Garc´es-Erice, *Building an Enterprise Service Bus for Real-Time SOA: A Messaging Middleware Stack*. 33rd Annual IEEE International Computer Software and Applications Conference, 2009.

[3] N. M. Josuttis, *SOA in Practice*. O'Reilly Media, Inc, August 2007: First Edition. 978-0-596-52955-0.

[4] M. P. Papazoglou, Heuvel, W. J. Van den, *Service oriented architectures approaches, technologies and research issues*. Springer-Verlag, 2007.

[5] M. Xueqiang, T.Xinhuai, Y.Xiaozhhou, C. Delai, L.Xiangfeng, *Multifactor-Driven Hierarchical Routing on Enterprise Service Bus*. Springer, 2009.

[6] Z.Gulnoza, C. Eunmmi, M. Dugki, *Content-Based Intelligent Routing and Message Processing in Enterprise Service Bus*. International Conference on Convergence and Hybrid Information Technology, 2008.

[7] G. Cugola, E. Di Nitto, *On adopting Content-Based Routing in service-oriented architectures*. Elsevier B.V., 2007.

[8] *Enterprise Integration Patterns*. [Online] http://www.enterpriseintegrationpatterns.com/.

[9] B. Xiaoying, X. Jihui, C. Bin, X. Sinan, *DRESR:Dynamic Routing in Enterprise Service Bus*. IEEE International Conference on e-Business Engineering, 2007.

[10] B. Wu, S. Liu,L. Wu, *Dynamic Reliable Service Routing in Enterprise Service Bus*. IEEE Asia-Pacific Services Computing Conference, 2008.

[11] T. Scheibler, D. Karastoyanova, F. Leymann, *Dynamic Message Routing Using Processes*. [book auth.] Klaus David and Kurt Geihs. Kommunikation in Verteilten Systemen (KiVS). Springer Berlin Heidelberg, 2010, pp. 117-128.

[12] Red Hat. http://www.redhat.com/docs/manuals/jboss/jboss-soa-4.2/html-single/ [Online]
[13] A. Khalili, S. Mohammadi, *Using Logically Hierarchical Meta Web Services to Support Accountability in Mashup Services*. IEEE Asia-Pacific Services Computing Conference, 2008.
[14] W.Hui, F. Peipei, Research of Strategic Route In Heterogeneous System Integration Based On ESB-SOA. IEEE, 2009.
[15] D.Kun,D. Bo,Z. Xiaoyi,L. Ge, *Optimization of Service Selection Algorithm for Complex Event Processing in Enterprise Service Bus Platform.* Proceedings of 2009 4th International Conference on Computer Science & Education, 2009.