

Enterprise Application Designs

In Relation to ERP and SOA



DESIGNING ENTERPRICE APPLICATIONS
HASITH D. YAGGAHAVITA
20th MAY 2009

Table of Content

1	Introduction.....	3
2	Patterns for Service Integration.....	3
2.1	Message Brokers	4
2.2	Web Service Remote Facade	5
3	SOA Platforms for ERP Implementations.....	7
3.1	Business Process Management Software (BPMS)	7
3.2	SOA extensions from existing ERP vendors.....	8
	3.2.1 <i>SAP NetWeaver Process Integration (PI)</i>	8
	3.2.2 <i>Microsoft Dynamics AX</i>	9
4	Conclusion.....	10
5	Bibliography.....	12

1 Introduction

Enterprise Resource Planning (ERP) is defined as a system that integrates (or attempt to integrate) all data and processes of an organization into a unified system. A typical ERP system will use multiple components of computer software and hardware to achieve the integration ^[1].

A successful ERP implementation can bring tremendous value to an organization. However if not properly managed ERP implementation can be a difficult and high cost process (both is financial and effort aspects). With the estimated success rate of 33% organizations are increasingly looking for more effective methods of implementing ERP systems ^[2].

A common criticism on single vendor ERP systems is inability to customize ERPs to suit the unique business processes of every business aspect of an organization. Traditional ERP implementation is seen as a procedure of molding the organization processes in to inflexible ERP processes. Although this can bring some industrial best practices in to the organization, there is a business risk of hindering the organization uniqueness. Due to this reason, organizations increasingly prefer assembling the ERP system from multiple vendors as suitable. This allows organizations to pick best suited modules from different vendors to obtain a unique ERP.

The prominent technology platform preferred for such ERP module integration is the 'Service Orientation'. According to Wikipedia, Service Oriented Architecture (SOA) provides methods for systems development and integration where systems package functionality as interoperable services ^[3]. SOA infrastructure allows different applications to exchange data with one another effectively.

This paper discusses the domain of enterprise application design concepts with related to SOA applicability on ERP implementations. Using SOA techniques for ERP module integration is the main focus of this paper. Following sections critically discuss some of the prominent design concepts with a comparative analysis.

2 Patterns for Service Integration

In practice service integration is not a trivial task but many different integration aspects need to be taken care of. Technically different application modules can be on different platforms but still should be able to technically understand each other. On a different note, data from different application modules should also be able to semantically understand each other in communication ^[5]. These two aspects are widely referred as 'application interoperability' and 'information interoperability'.

Diversity in technology platforms can make application interoperability considerably difficult between modules. For example, binary data services such as RMI, Remoting heavily restrict cross platform modules to be integrated. Most of the medium to large enterprises do have systems that are based on different technology platforms. SOA implementation technology for building the ERP should be able to manage such application interoperability issues.

Similarly it is not always common for all services to carry a uniform domain model representation. Data schemas, service contracts differ from each other making information interoperability also a considerable challenge. Integration between these heterogeneous domain models is another issue that needs to be solved by the SOA technology.

Following sub sections describe two of the prominent architectural styles used for building SOA based ERP products.

2.1 Message Brokers

A message broker is a component that act as an intermediately between the integrating modules. Modules do not directly invoke each other but communicate only with the message broker. With each communication, message broker is provided with the logical identification of the destination module where the message broker delivers the message to that designated receiver. In a typical implementation, all modules need to get them registered with the message broker prior to operation.

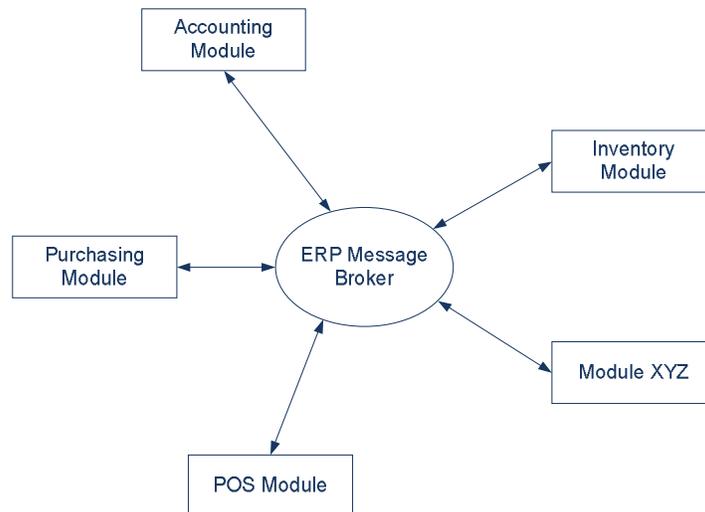


Figure 1: Message broker mediating integration between modules

In addition, advanced message broker implementations can route the messages conditionally based on the infrastructure conditions. This is called context-based message routing and useful for implementing infrastructure sensitive message delivery to achieve versioning, quality of service, etc.

A good message broker implementation can cater for wide variety of interfaces having considerable differences. Typically a commercial message broker product contains a rich set of connectors such as for connecting files, databases, http services, etc. In addition they provide formatting and data transformation tools as value added services to increase integration ability^[4].

Integration through a message broker brings several advantages to the ERP implementation. One of the most significant characteristics of a SOA based ERP

system is its ability to change when required (loosely coupled). For example, POS module should be loosely coupled to its dependencies where switching to a new payment module minimally impacts the POS module. Many of the message broker implementations provide greater level of module decoupling which facilitate frequency service switching even dynamically. This is mainly because with the mediator approach it is possible to minimize particular module's knowledge on its dependencies.

One major disadvantage of message broker is the introduction of a single point of failure. Since all the integration is through a central broker product, if broker fails there is a risk of whole ERP system failing. However, a failover implementation can minimize this risk but with lot of additional complexity.

One main advantage of the centralize nature of message broker is ease of maintenance. Message broker integrations are mostly appropriate when most of the ERP modules to be integrated are internal to the organization with limited geographical distribution.

There are several commercial message broker implementations exist from vendors such as IBM, BEA, Oracle where some open source implementations such as from Apache are also available in the market.

2.2 Web Service Remote Facade

Web Services provide a platform neutral approach for integration using standard XML protocols. Use of text/XML makes messages makes integrations to be platform independent and preferred for most of SOA implementations. Web services are based on open standards such as SOAP, WSDL and HTTP and therefore widely adapted unlike proprietary standards.

As opposed to centralized message broker pattern, web service facades provide much distributed peer-to-peer integration between ERP modules. Use of web services as wrappers provides an interoperable interface to the heterogeneous ERP modules.

There are two main components exist in a sophisticated web service wrapper.

- **Adaptor:** Often direct service mapping are difficult due to the diversity of integrating modules. Placing an adaptor module prior to the web service can perform required adaptations (such as content adaptation, invocation model adaptation).
- **Web Service:** This is the remote façade component that makes functionality remotely available. Most popular is use of SOAP over HTTP protocol. But this is not mandatory and even simple XML over HTTP (e.g. REST services) can also be used.

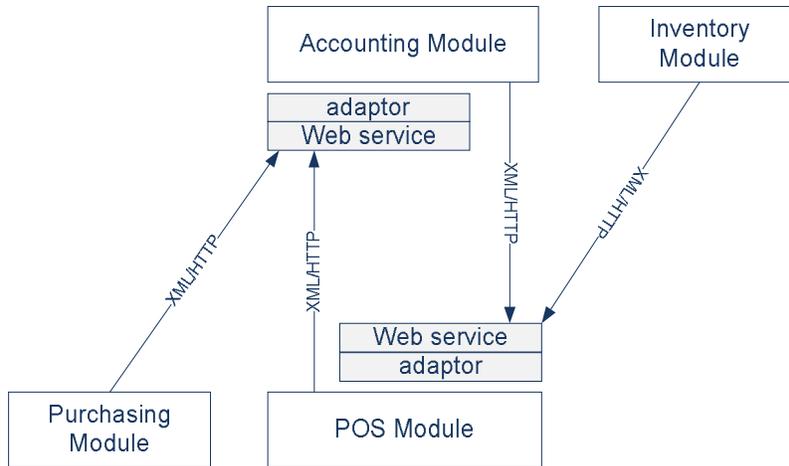


Figure 2: P2P architecture for web service wrapper integration

Web service wrappers are widely used for integrating legacy applications in ERP implementations. Wide tool adaptation (even in legacy languages) on web services has enabled organizations to develop wrappers for their legacy components. But it should be noted that real life implementations can be fairly complex especially when changes are required in the existing legacy code before service exposure. For example, if existing module classes are stateful, they might need to be modified to support stateless invocation which is the preferred method for service orientation. Although adaptors can help handling some of these complexities to good extent, legacy code changes are also often required reducing the attractiveness of web service facades.

Integration through web service wrappers placed between the modules make ERP system distributed and peer-to-peer (P2P). A common criticism is that it can make total system architecture more complex and hard to maintain. But one of the important advantages of P2P style is that, unlike message broker, web service wrapper does not introduce a single point of failure.

It should also be noted that the web service wrappers are mostly custom built unlike off-the-shelf integration techniques like message broker frameworks. Developers will require handling data transformation and formatting by themselves unless used specific framework for those purposes.

Web service wrappers are mostly recommended when discrete ERP modules work independently within their respective domains. In addition, web service wrappers are able to handle largely distributed scenarios such as B2B integrations. Another advantage of web service wrappers is that the web service interfaces built for inter module integrations can be used for further module extensions outside ERP scope due to the use of cross platform protocols.

As amount of traffic between two ERP modules increases, overhead of web services can cause issues such as poor performance. Therefore for very high volume communications, other integration mechanisms that can support more native communications can be effective over web service wrappers (e.g. ESB products).

3 SOA Platforms for ERP Implementations

Above section discussed some prominent patterns for integrating different business modules for developing a service based ERP systems. In addition there are numerous platforms and tools that enable SOA based ERP software implementation. These platforms operate at a higher level above the technical integration patterns discussed in previous sections (with much business layer integrations).

Following sections elaborate some of the tools and platforms used for service based ERP implementations.

3.1 Business Process Management Software (BPMS)

Business Process Management Software is a set of tools designed to aid development, monitoring, analyzing, optimization and automation of business processes of an enterprise.

BPMS empowers user to control business workflows and have a better flexibility over traditional integration approaches. Process optimizations provided by BPM suites allow analysis of business process information creating an opening for performance improvements based on the real-time business metrics ^[6].

BPM software generally provides powerful business workflow management system. Workflow integrations are soft wired allowing them to be adjusted for necessary optimizations.

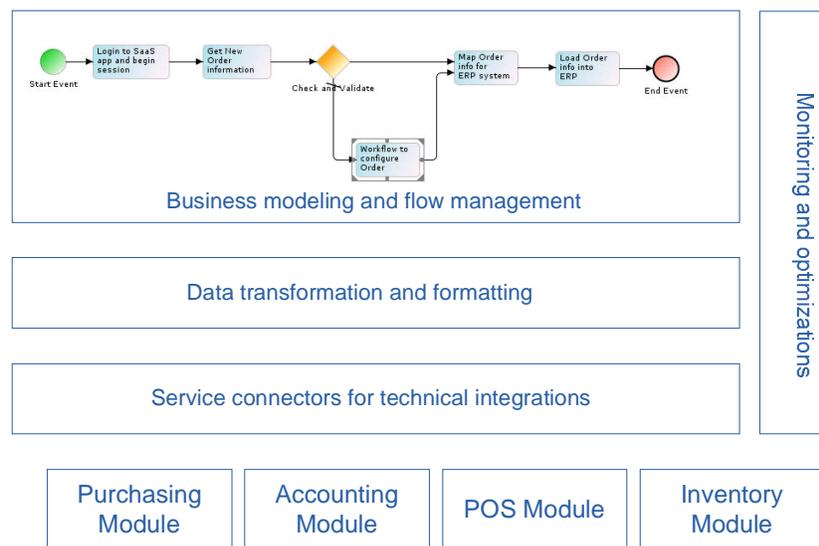


Figure 3: Employing BPMS as the user interaction layer of ERP

Another most notable difference of BPMS based service integration is the visual modeling of business processes. This allows business users to get better involved in the ERP system design and also to actively get involved in modifying workflows

when processes change. BPMS can bring total enterprise process integrations in to a central visual model providing more visibility and control.

BPM approach is much different to both above described integration patterns as BPM software act as a wrapper for all modules where users often interact with the workflows provided by BPMS. It is generally recommended that BPMS is more appropriate when ERP modules provide rich integration capabilities and when the functionally is more workflow oriented rather than data centric.

Business process automation being the top layer, most of the commercial BPM software provides toolsets required for technical integrations as well. They are commonly equipped with data manipulation tools and technical integration connectors.

Once services are integrated, BPM software can also emulate the production environments to identify possible bottlenecks and inefficiencies. These features are offered by the monitoring and optimization tools available in most of the commercial BPM software.

It has been commonly seen that almost all the leading ERP vendors building BPM capabilities in to their existing ERP infrastructure. But the operational concepts of traditional ERP systems are much different compared to the principles of BPM. This make most of the ERP adapted BPM systems to offer much less flexibility than BPM solutions built from ground up.

3.2 SOA extensions from existing ERP vendors

Traditional ERP systems often offer monolithic architecture and centralized data store. But with the strong tendency of enterprises to move in to service integrations, most of the traditional vendors now providing rich SOA support as well. With these tools, organizations who already own commercial ERP software can leverage on these tools for extending their ERP solutions to consume services from other vendors as well.

It is commonly accepted that the service orientation is not easily achieved as a one-off solution but need to be implemented incrementally. Organizations with an existing ERP implementation can increase their value by integrating the current ERP with service available outside. Planned and incremental process of repetition can gradually move the organization to own a fully-fledged SOA based ERP solution that replaces traditional monolithic ERP solution.

Following sections discuss SOA capabilities offered by two of the leading ERP product vendors.

3.2.1 SAP NetWeaver Process Integration (PI)

SAP is one of the main ERP product vendors in the market. NetWeaver PI product from SAP provides the development and runtime environment for SAP applications

and can be used for custom development and integration with other applications and systems ^[8].

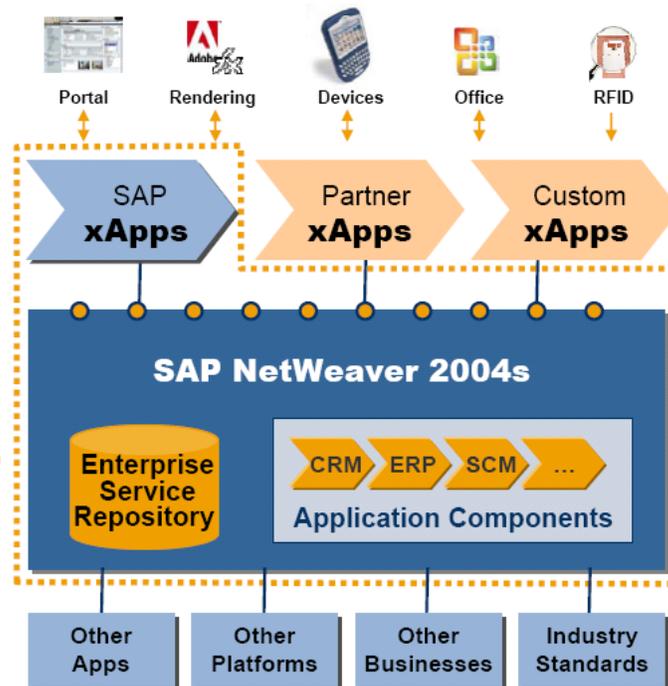


Figure 4: Creating composite applications with SAP NetWeaver ^[7]

SOA middleware platform named ‘SAP NetWeaver PI’ facilitates communication between disparate services and the SAP ERP. These tools can be effectively used for integrating non SAP services in to the organization’s ERP platform.

SAP NetWeaver PI is built on message broker architecture variant, named ‘hub-and-spoke’ model. Through the spokes, the central message processing hub is connected to all relevant applications. The hub system then hosts all necessary information such as document formats, transportation rules, and security requirements to handle integration activities.

SAP also provides wide selection of adapters for the most common business systems such as IBM CICS Transaction Server, Oracle E Business Suite, PeopleSoft Enterprise, and Siebel Systems ^[8].

Organizations owning SAP can leverage on the tools provided with ‘SAP NetWeaver PI’ platform for integrating services from other systems. This allows the organizations to integrate legacy applications and external services to the existing SAP infrastructure.

3.2.2 Microsoft Dynamics AX

Dynamics AX is the ERP solution provided by Microsoft Corporation. Dynamics AX provides various ERP modules such as for financial, human resource, supply chain,

customer relationship management, etc. For SOA based ERP extensions, Dynamic AX provides a powerful service integration tool set.



Figure 4: Dynamic AX platform for SOA ^[9]

Dynamic AX provides several levels of service integration capabilities such as at ‘application integration framework (AIF) level’ and ‘BizTalk Server level’.

Much powerful and higher level integrations are generally done through BizTalk integrations. Microsoft BizTalk Server, a business process management server that helps connecting services for building SOA enabled applications. The combination of Microsoft BizTalk Server and Microsoft Dynamics AX helps building sophisticated integrations between heterogeneous services enabling the ERP systems to extend their scope.

Application Integration Framework (AIF) is much lower level framework that enables integration through the exchange of data through formatted XML. This formatted XML is referred to as a document, and each document contains defined data and business logic. In AIF, data is exchanged with external systems through electronic documents. Microsoft Dynamics AX also ships with over 70 standard documents that support common business processes ^[10].

4 Conclusion

In general services are more self contained modular elements that expose a well defined contract for external and remote interactions. Self contained nature of services can encapsulate data and internal implementation details from the business reducing coupling. Services are generally reused by multiple applications and minimize duplication of logic considerably. Due to these reasons, it is evident that the enterprise applications built based on SOA are more flexible and open for business changes.

Despite many criticisms, traditional ERP systems have succeeded in obtaining a broader market penetration over last decades. However it is mainly the inflexibility of traditional ERP products attracted enterprises towards service oriented enterprise systems.

As described, existing ERP vendors increasingly provide tooling for service oriented integrations. Web service platforms have become the most prominent protocol for these integration tools allowing disparate systems to communicate with each other. These tools help integrating legacy applications as well as extended external services to the basic applications provided by the existing ERP infrastructure.

Building service oriented integrations around existing ERP solution can be seen as an incremental approach for service enabling the organization. Another approach for service enabling an organization is to integrate different services/ERP modules from different vendors to custom built an ERP solution for the organization. This is more aggressive and complex approach but brings more flexibility to the organization in selecting best matching ERP modules for the organization. This approach is more suitable if the organization already owns discrete IT systems that are well functional within its departments. By employing a suitable SOA enabler pattern such as 'service broker' or 'web service facades' the organization can build its unique ERP solution that leverage on most of the existing investments. Use of a good BPMS is another interesting alternative for service enabling the existing IT infrastructure.

One of the biggest hurdles in service enabling the organizations is the question of 'trust'. Some of the original promises brought by web service concept such as dynamic discovery of services (based on UDDI) have not really picked up due to the question of trust. For example, it is very unlikely an organization to dynamically search in a repository and integrate some vendors 'interest calculation service' to its ERP infrastructure.

Another major challenge for SOA enabling enterprise applications is the dealing with complexity. Distribution and diversity can make system maintenance a considerable challenge and high availability need to be ensured. Heterogeneous operational and data models can effect smooth integrations between the systems causing SOA projects to overrun budgets and schedules.

Despite lot of challenges, SOA has gained lot of attention of the business community. Especially the concepts like 'pay as you use' brought by the service vendors lower the cost of entry to the service use. Also the optimization flexibilities offered by SOA can easily facilitate competitive advantages for innovative organizations.

5 Bibliography

1. Academic ERP Terminology. *Threerivers-cams*. [Online] [Cited: May 17, 2009.] <http://www.threerivers-cams.com/Resources/Terminology/tabid/96/Default.aspx>.
2. Fryling, Meg. *The Dynamics of ERP Success*. State University of New York : s.n.
3. Service-oriented architecture. *Wikipedia*. [Online] May 14, 2009. [Cited: May 17, 2009.] http://en.wikipedia.org/wiki/Service-oriented_architecture.
4. WebSphere Message Broker Basics. *IBM Redbooks*. [Online] [Cited: May 18, 2009.] <http://www.redbooks.ibm.com/redbooks/pdfs/sg247137.pdf>
5. Enterprise application integration. *Wikipedia*. [Online] [Cited: May 17, 2009.] http://en.wikipedia.org/wiki/Enterprise_application_integration.
6. Ayres, Phil. BPM or ERP - Stand out from the crowd. *Improving It (Blog)*. [Online] Aug 01, 2006. [Cited: May 19, 2009.] <http://improving-nao.blogspot.com/2007/10/bpm-or-erp-stand-out-from-crowd.html>
7. What's New with SAP NetWeaver 2004. *SAP Community Network*. [Online] 2004. [Cited: May 19, 2009.] <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/4487dd91-0b01-0010-eba1-bcd64196ec6b>
8. End to End Process Integration with SAp NetWeaver. *SAP*. [Online] 2005. [Cited: May 19, 2009.] http://www.sap.com/platform/netweaver/pdf/BWP_SB_End-to-End_Process_Integration.pdf
9. Microsoft Dynamics and "real-world" service-oriented architecture . *Microsoft* . [Online] [Cited: May 20, 2009.] <http://www.microsoft.com/dynamics/SOA.msp>
10. Integrating Other Applications with Microsoft Dynamics AX . *MSDN*. [Online] [Cited: May 19, 2009.] [http://msdn.microsoft.com/en-us/library/aa877498\(AX.10\).aspx](http://msdn.microsoft.com/en-us/library/aa877498(AX.10).aspx)