

A white paper by Alistair McLeod
December 2007

SERVICE ORIENTED ARCHITECTURE

Contents

Exploring the place of SOA in a modern integration strategy	3
Strategies for using IT to deliver business integration	3
Start with the business requirements	4
Traditional Integration Strategies	4
Service Oriented Architecture	6
The Hybrid Approach	8
Summary	9

Exploring the place of SOA in a modern integration strategy

As the trend for employing a Service Oriented Architecture (SOA) gathers pace, it is important to remember that the more traditional approaches to business integration can still deliver results. Waterstons' executive consultant, Alistair McLeod will cut through the hype and get down to the fundamentals of integration, discussing the pros and cons of SOA, traditional and hybrid approaches so that the reader can make an informed decision as to which approach they should use and when.

Strategies for using IT to deliver business integration

Integration is a proven enabler for many organisations as they strive to connect customers, locations, people, business applications and information. Modern integration technologies such as SOA are now providing a business focused approach. The whole organisation can be involved in defining and creating business process automation through easy to use workflow applications and simple service requests for data from the core business systems. This is providing a greater scope for aligning IT systems to the demands of the customer and the requirements of the internal business processes.

The real challenge is to ensure the business is integrated as a complete entity, which requires co-operation and an accurate definition of the requirements driven out of the strategic goals and objectives. It is also important to understand that new technologies such as SOA are not necessarily the answer to all your integration problems. Traditional integration strategies are still relevant today and should be used appropriately - it is critical to understand which approach to use and when to use it.

Start with the business requirements

Any integration project should start with the business requirements; the project team should engage the whole business when defining specific details. Mapping the business processes and measuring the current performance will help to identify issues and highlight where improvements can be made. Change management will be an issue, and building the case for changing the status quo will be critical. The user community will only buy into the project if they can envisage the benefits, and they should be consulted at an early stage.

Traditional Integration Strategies

The traditional approach aims to select off the shelf, best of breed business applications that are tightly integrated and meet most of the business requirements. 'Tightly integrated' (or 'tightly coupled' as it is often known) describes a system where all of the modules are fully integrated with each other, share a common database and use a standard user interface (the presentation layer). Fig 1 illustrates this concept.

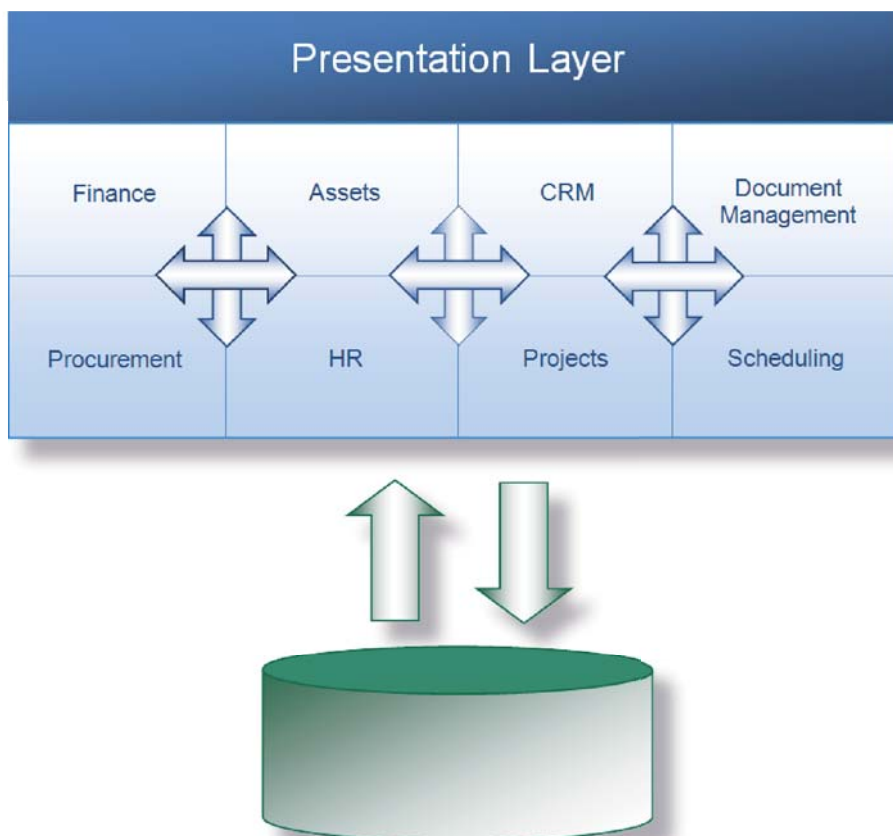


Fig 1. A tightly integrated business system

Ideally tightly integrated systems should meet all the business requirements, although a degree of pragmatism is required, and integration with additional third party applications may be required (with bespoke interfaces developed to tightly integrate the applications together). The main advantages of tightly integrated systems are:

- They are highly scalable and very responsive.
- They are functionally rich, containing many value adding features and a pleasant user experience.
- They are low risk as they usually have a proven track record, particular in specific industry verticals.
- They ensure best practise is followed as the systems are built around proven business processes.
- Upgrading, support and maintenance is less risky and usually more cost effective.

Tightly integrated systems also bring structure and control to organisations and are ideally suited to operationally demanding environments where information is required real-time such as manufacturing and engineering companies. Enterprise Resource Planning (ERP) applications are a good example of this. The disadvantages of using this approach are:

- The organisation is highly dependent on one specific vendor.
- Inflexibility - changes to the business processes could require expensive redevelopment, or the business may have to wait until the next upgrade is available.
- They are not always a perfect fit to the business requirements. Sometimes a business may have to change its processes to fit the system, or purchase additional third party systems to compensate.
- There is a high initial capital expenditure and a replacement system is usually required every 5-10 years.

Service Oriented Architecture

Service Oriented Architecture (SOA) is the practice of segmenting software into independent modules or applications that are loosely integrated (or loosely coupled). This means they are not integrated at all, but instead expose data as a series of interfaces. The integration is performed by a service layer (commonly known as the Enterprise Service Bus) that uses a workflow application (such as Microsoft Windows Workflow foundation or BizTalk) to automate the business processes, route the data and present the information to the users through simple and dynamic forms (delivered through a product like Microsoft SharePoint or Microsoft Office InfoPath). Fig 2 illustrates the concept.

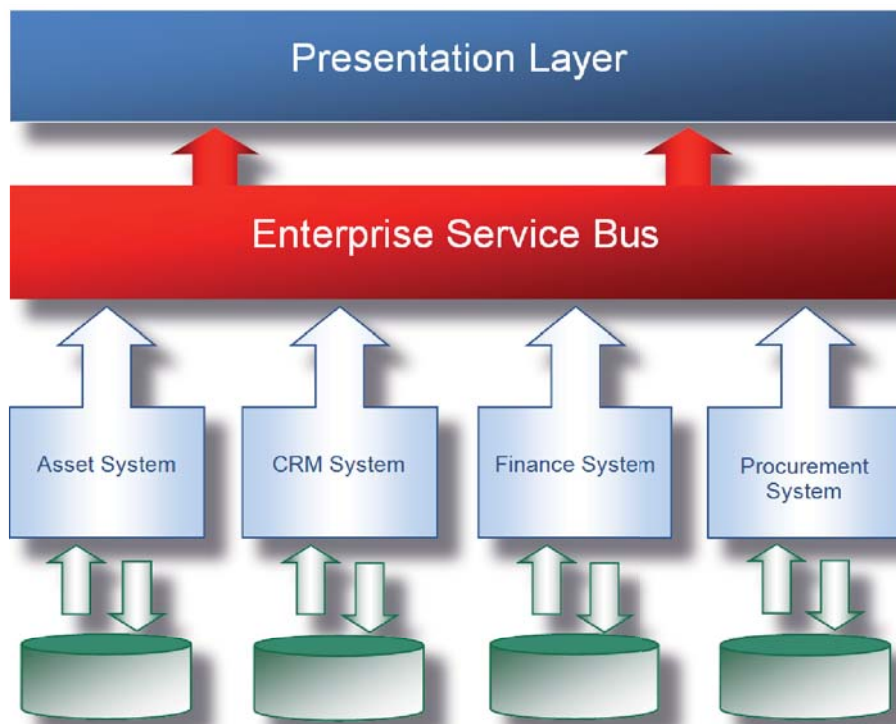


Fig 2. Service Oriented Architecture

Each module or application is completely independent and can maintain its own data and database. The main advantage of using this approach is that it creates a dynamic architecture to allow the delivery of business requirements. Business users or departments can be empowered to create and maintain the automation of business processes by using the graphical interface of the workflow application. Other advantages include:

- It provides an integration strategy for disparate business systems which is particularly useful if you have a lots of disparate legacy systems.
- It can minimise the impact of changes to software code on other software components through “loose coupling”. If a particular part of the system needs upgrading it does not impact on the other areas and can help to spread the cost of upgrades.
- It eases the future interoperability internally and to external third parties, making business to business automation easier.

It can also help lower your liability because you are not relying on one vendor and can also make the process of replacing core elements of the business systems less risky and easier to implement.

Empowering the business community with the ability to automate and improve business processes obviously comes with a risk. SOA relies heavily on the management of the workflows, components and metadata which can result in an administration cost on the business. Other potential problems are:

- You need someone (or several people) who really understand the business processes in order to effectively model them.
- Performance may not be as good as with more tightly coupled interfaces. You need to understand the types of processes that can be automated using this approach, for example systems that require high degrees of transactional processing are not good candidates.
- Security is difficult and complex to implement, with many systems, workflows and users screens.
- The functionality of the user interface may be reduced providing a less intuitive user experience.

The Hybrid Approach

The integration strategies we have discussed so far are not mutually exclusive and a combination of the two can be used effectively to provide a robust and flexible integration strategy. Many ERP vendors have recognised this and have moved towards providing an SOA layer with their systems to provide a way of integrating with third party applications and customer systems. Additionally, a minority of businesses who are in specialised industry sectors and are not well served by off the shelf business systems could use the hybrid approach as a proposed solution to reduce the risks inherent in building large bespoke solutions (i.e. a series of best of breed packages and small bespoke projects coupled together within an SOA).

Figure 3 illustrates how a business may want to integrate a document management system and an MIS system with a tightly integrated business system.

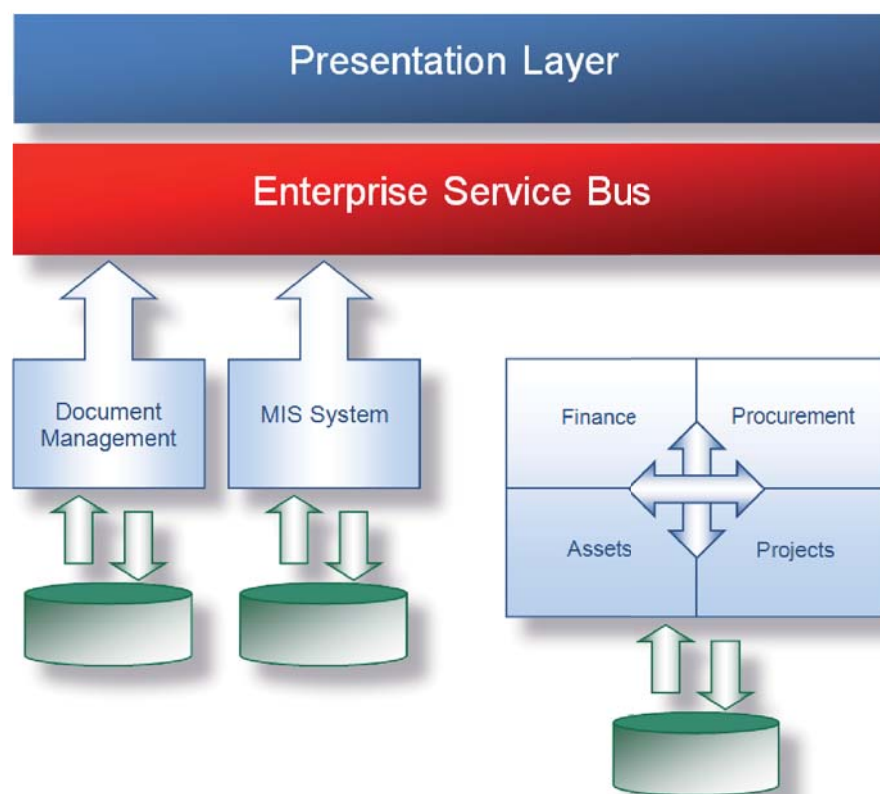


Fig 3. A hybrid system using SOA and Tightly Integrated Strategies

Summary

In the real world most businesses have an evolving IT architecture, with legacy issues and different levels of maturity. When deciding upon an integration strategy it is important to consider the current landscape and consider the following:

- Is there a business case or economic value in building or buying best of breed?
- Do you have multiple legacy systems that work well but are not integrated?
- Is there a cost justification for replacing disparate systems with one tightly integrated system?
- Do you want to decrease your dependency on vendor-specific software products and use multiple software service components?
- Do you want to maximize your ability to create flexible business processes?
- What are the scalability issues in the next 5 years and are your systems up to the job?

SOA certainly has a place in the future of integration but it needs to be part of a bigger strategy. Tightly integrated systems are undoubtedly the preferred starting point to business process automation. They bring structure, control and a great user experience. Tried and tested systems are low risk, if somewhat more expensive. SOA is great at adding value to further enhance business processes or resolve the nightmare of disparate legacy systems, but it can be difficult to maintain and can lead to a proliferation of bespoke components throughout the IT architecture. It is important to always remember that anything you do should be against a back drop of a well defined IT Strategy with cost and benefit justification.