

The Project Factory
IT Project Services

The Project Factory Ltd
46 Lynx House
Caxton Close
Andover
Hampshire
SP10 3FG
UK

www.Project-Factory.co.uk

Tel (+44) 0870 145 4390

***An Introduction to
Enterprise Application Integration
and
Service Oriented Architectures***

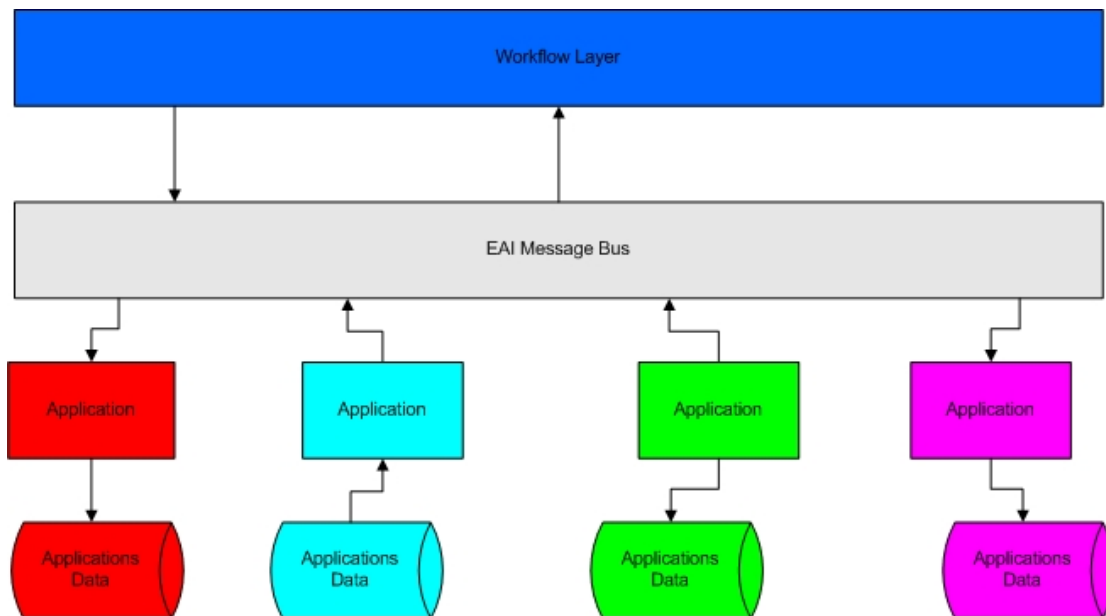
By Cliff Leach

The main issues with this approach are:

- **Sustainability**- as the number of applications grow so does the number of inter-application connections
- **Performance**- Multiple connections utilise resources and modelling the performance implications becomes increasing difficult as the number of applications and point-to-point connections grow
- **Security**- if applications and their data sources reside in differing systems or security domains, then tracking the flow of data and access rights for example, becomes increasingly problematic
- **Scalability**- on either an active/passive, active/active or other topology scaling for transaction rates and geographic scale becomes more complex for the reasons above
- **Control and management**- less really is more when it comes to managing applications, data and connections

An alternative approach is to use more formal Enterprise Application Integration (EAI) and to focus this around a Services Oriented Architecture (SOA). The principles of such an approach are straightforward in outline:

- **Consistent data strategy**
 - Definition of the key business processes to be supported by the enterprise and that will be within scope for the architecture
 - Determination of key elements of enterprise data and definition of an enterprise logical data model for that data related to the business processes concerned
 - Define the master sources of each of the key elements of enterprise logical data
 - Determine the precedence and update logic (now) of the key elements of enterprise logical data
 - Develop a mapping of enterprise logical data elements to their physical data representations for each application
- **Application interface strategy**
 - Mapping of applications and their key data to the business processes concerned
 - Clear definition of all existing interfaces to these applications and their associated data
 - Definition of required interfaces to these applications and their associated data based on the mapping of the business processes to these
 - Shortfall analysis defining those interfaces to be modified and those requiring development
- **Workflow strategy**
 - Definition of the workflows and their detailed logic mapped to application functionality
 - Determination of where workflow logic will sit- in the workflow layer or in the application
 - Analysis of how workflow layer will interface to the applications



In principle (in fact this is a lot more subtle and complex) the workflow layer utilises a messaging backbone to talk to standard APIs (called connectors in EAI speak) and to access both applications functionality and data. This allows re-use of application functions and simplifies the management of the associated data. Some of these functions can be collapsed (and often are) with amalgamation of (for instance) some components of workflow into the adaptor interface for the application.

Back in the real world

Let us leave this very high level summary and think now about the implications of the EAI / SOA strategy in the real world of commercial systems.

EAI and SOA are very strong architectures. By this I mean that they give robust, flexible and open solutions to multi-application, multi-data-source situations. They are strongly founded in sound principles of computer science and well supported by the tools and methods needed to deliver them and their adoption is very slow despite the proven major benefits associated with them.

Why is this? How can it be that such a strong and well-supported set of architecture principles and tools so well aligned to the needs of real commercial problems is so slow in its adoption? The headlines below summarise the key reasons why:

Scope Definition

The key issue with an architecture is that its principles will apply to a complete systems domain. This implies:

- The scope of the systems is adequately defined
- The starting point has to be a clear definition of the workflows and business logic within the scope
- The priority and logical interrelationships between the workflow elements needs to be defined- what is needed when and how does it fit together

Data Definition

Too often I find that data are not even basically defined for an organisation. A tiny percentage have a complete enterprise logical data model (and few can justify the need or costs of such a model), a small percentage- perhaps 5-10% have an adequate logical data model for the applications and workflows key to their business and the rest, in excess of 80% simply do not have any form of accurate, appropriate and effective logical model of the data key to their organisation's workflow and business operations, This implies:

- Most business's data are out of control and not big exploited to the benefit of the business
- Before any move to an EAI can be made a model of the data needed by each workflow has to be made
- There will be multiple version of the same logical data on multiple systems and all of them will be different since in the majority of cases no single master data copy will have been defined within the context of the workflows key to the business

Applications Interfaces

For any EAI strategy to be implemented, interfaces between applications and the EAI bus have to be available. This implies:

- APIs must be published and properly documented by the application vendors
- APIs will need to be created or modified by the application vendors to make their applications open within the context of a best-of-breed approaches
- The costs of APIs has to be economic for the clients

So what goes wrong- why don't we have lots more EAI and SOA projects?

In summary there are several things that hinder the take up of EAI and SOA in the real commercial domain:

- Businesses don't define their business process or workflows at all or adequately and this is the core of their business and the key to their process, quality and cost/productivity improvements
- Linked to the above, Businesses don't define their business data at all or adequately and this is the core of their corporate information and the key to their performance analysis and process design
- Applications vendors are too paranoid about being open with their APIs, closing an API by not publishing it, refusing to develop it or pricing it too high may keep your client right now but they are unwilling hostages not partners, the first guys on the block that provide the same functionality and open up their APIs will have a major benefit over you so open up, talk to your clients and realise that cooperation will win over confrontation

The following table gives a summary:

Issue	Implication for Application Owners / Users	Implication for Application Vendors
Need to have open APIs	<ul style="list-style-type: none"> • Must identify existing and required APIs • Either you own the right to an API or not • Need contractual and relationship issues covered • No API will impede your EAI /SOA strategy 	<ul style="list-style-type: none"> • Need to publish your APIs and make them open technically and commercially • Must make clear the ownership and licensing issues surrounding such APIs • Make the cultural leap to support APIs and cooperate with client strategies
Must have a logical and physical data model for the enterprise and its applications	<ul style="list-style-type: none"> • Need to scope just the workflows you will want to include in your EAI/SOA and map this to applications • Must map logical to physical data structures 	<ul style="list-style-type: none"> • Must make physical data structures and database structures available to clients • Need to consider the impact of changes to your data structures on clients who have implemented and EAI/SOA
Move to Best Of Breed (BOB) for selection of products based on functionality	<ul style="list-style-type: none"> • Need to consider functional fit with the business needs more precisely than is often the case • Expect to see selections to throw up some platform issues and need to accommodate these 	<ul style="list-style-type: none"> • View this as a major benefit in the sales cycle and strengthen your sales approach, for instance with more sector-specific use cases • Be sensitive to the platform choices of your future client sector

This is a short introduction to the commercial problems of getting an EAI strategy into play and I apologise for the dearth of detail that inevitable comes with an article of this length. I would strongly suggest to all organisations that they take a look at an EAI/SOA approach to their long-term applications and data management strategy.

From the experiences I have had working with my clients, from the largest telecommunications companies to the smallest public authorities, EAI and SOA deliver immediate and long term benefits despite the problems and pitfalls I have outlined above.

When it comes to applications vendors, closing the doors may feel comfortable and charging the Earth for an API may look attractive in the short term but EAI is here to stay and the new entrants to every application market I have worked in are making their applications more open and at affordable prices so to borrow the words of Bob Dylan ‘... the times they are a changing...’.

The Project Factory *IT Project Services*

The Project Factory welcomes any questions on this article or any other aspects of EAI and SOA, our contact details are: sales@project-factory.co.uk

This paper is an abstract from our seminar 'An introduction to EAI and SOA' this is a 1 day overview aimed at senior managers and non-technical decision makers who need to get an appreciation of the subject.

