



# Enterprise Application Integration (EAI)

White paper - September 1999  
by [Marc Buyens](#), Xpragma

Recently, Sterling Software presented its new strategy during a seminar in Brussels titled "Enterprise Application Integration (EAI)". For people like me who are following a bit what happens in the EAI market, Sterling Software is not really seen as a "typical" EAI vendor. They are of course well known for their development tools, previously known as Composer, but now marketed under the label of COOL:Gen, etc. Sterling's new strategy is an illustration of the reality that the EAI market that we knew from vendors such as New Era Of Networks and TSIsoft is getting more and more mainstream and increasingly e-business focused.

---

## Everybody seems to have it

We know of at least a dozen companies that claim to be some kind of "leading EAI vendor". We also know for sure that all these companies are offering quite different solutions. Don't they know what they are talking about? Are they lying? Where is the truth?

Well, as always, all these statements are partly right and partly wrong. They are right in the sense that these companies are indeed offering solutions that can be used for EAI. Also, it is possible that they can rightfully claim to be "leading" in some specific aspect of EAI. However, they are completely wrong in the sense that today no single vendor can offer a complete EAI solution. Most vendors are merely addressing a small subset of the requirements and you will need at least 5 vendors if you really want to address most aspects of EAI.

The reality is that EAI has so many facets that it even is a challenging task to give an all-encompassing definition. For the purpose of this article we will use the following definition:

*EAI is the ongoing **process** of putting an **infrastructure** in place, so that a **logical environment** is created that allows **business people** to easily deploy **new or changing business processes** that rely on IT.*

Well, it is certainly not a perfect definition, but it gives us a start for discussion. As said, most EAI vendors will only address a small piece of it and today's reality is that you will have to combine several products and technologies to really get to an EAI implementation that is close to this definition. Moreover, to be very honest, even the combination of all available

technology will not yet give you this.

Therefore a first warning: do have realistic expectations. As with all technologies or paradigms that are surrounded by a lot of hype, things tend to look fuzzy. Often this results in either unrealistic expectations or decisions to do nothing. Neither of these is a good approach. EAI is something with an enormous potential, so do evaluate it. But do this, based upon an understanding of the reality.

Let us have a closer look at the underlined key parts of our definition.

## Process

First of all, to eliminate (or create?) already some of the confusion: EAI is not something you can buy. EAI is not a product. It is not some kind of development tool. It is even not a category of middleware. EAI is something you do. It is a way of increasing the business value of your IT environment. However, it is also true that it is an IT approach that typically will make use of various middleware products and technologies.

## Ongoing

EAI is not a one-time exercise. In the initial phase, where the company evolves from its current IT application environment to an EAI-enabled infrastructure, many one-time changes will occur. But even when arriving at the "final" stage of integration, it will be mandatory that, whatever further changes in the IT environment occur (new systems, new technology, changes to applications...), these are deployed in accordance to the EAI principles.

## Infrastructure

The EAI process will result in the deployment of an infrastructure that implements (imposes) the principles of good application integration. Many organisations have implemented EAI-style solutions as a tactical approach to solve a specific problem. Such implementations will often be IT-driven and there is no harm to that. However, if your ambition is to work towards an EAI implementation such as the one discussed here, then "infrastructure" is a keyword. Consequently, such project must be business driven.

Some pieces of this infrastructure will be very low-level technical, far away from the abstraction layer used by the business people. However, this abstraction layer will not be able to provide its functionality and business services if the underlying groundwork is not done correctly. Message queuing is one of these parts of the underlying infrastructure that we see as being an essential element.

## Logical environment

The final deliverable of an EAI exercise is a high degree of abstraction, automation and flexibility. If feasible, this environment should only reflect the image of business processes. At such abstraction level, the notion of an individual system, application, etc. does no longer

exist. The technical infrastructure is not visible. Changes in this underlying infrastructure are completely transparent, without having a visible effect on the business logic layer. On the other hand, the business logic layer must have a clear view on the components that it can manipulate and the way that they interact.

This is another difficult point. An absolute requirement is a complete understanding of the behaviour of the underlying components. Especially while integrating packaged applications, this really requires specialist knowledge of the internals of the package. Also, since you have no control of the package, it introduces the risk that changes occur in some of the packaged components, without you noticing.

## Business people

Yes, IT people will probably not like this, but the business people are the ones who understand the business and they must be the ones that "build" business processes. No longer the complex exercises of translating business requirements into functional specifications, prototyping, RAD, etc. No, business people will simply do the "programming", be it that this programming will be quite a bit different from the one that is being used by the IT people who build the infrastructure.

Again, this underscores our requirement that EAI must be a business driven exercise. Business people have to be involved from the beginning. This also implies that top-management must drive the exercise.

## New or changing

EAI is not the thing you want to do if you have a very stable business environment (unless somewhere as a tactical solution). EAI is difficult, complex, and expensive. Therefore, you only want to start such an exercise if the potential gains are in balance with the guaranteed pains. However, in today's business climate only very few companies will have the luxury of stability. In most industries, "change" is the name of the game. EAI brings the promise of being able to handle "change".

Again, be realistic and know your business objectives. Not all companies will have the need to work towards an all-encompassing EAI implementation. For many companies, a more limited or even tactical approach will be the perfect fit. Therefore, it is necessary that you can clearly differentiate the different offerings in order to make the right decisions.

## Business processes

Business process is the final keyword of our EAI definition: the things you want to do so that your company can grow; the translation of business objectives into action. This is the world of business services. These services will be built as an assembly of business components. A component being the piece of business logic that can be freely (re-)used anywhere to change, complement or enhance a specific business process.

This assembly of new or changing business processes should only have to deal with this

manipulation of well defined business components, the definition of the flow of interaction and the association of business rules to it. All underlying infrastructure, including complete ERP solutions, should be masked. Very few solutions today already provide support for such level of abstraction.

As business processes are key in EAI implementations, do have a serious look at your current business operations. Some integration exercises will be a lot easier if first some level of process re-engineering takes place.

## Architecture

As said above, EAI is about building an infrastructure that allows for business agility. And for an infrastructure we need some kind of conceptual view, an architecture. Giving the all-encompassing character of EAI it is difficult to give one that really addresses all aspects. Especially now that EAI is extending its reach to include the e-business paradigm, lots of technologies will have their place in an EAI exercise. Therefore we will limit ourselves here to an architecture description for messaging oriented integration approaches.

Definitions of architectures are always a fun piece. Nobody agrees on the classifications and definitions, and often they are defined based upon their ability of being graphically represented in an interesting way.

Do not expect something fancy from our side. We simply needed a way to classify a bit the various pieces that make the EAI puzzle and we have based this classification upon the definition that we gave earlier. We define the following four major layers:



This is of course a very generic model and you can argue upon the terminology used. However, we want to use a classification that is as simple as possible, but highlights the major steps that are needed to get to a full EAI implementation.

From top to bottom the layers are:

## The Business Process layer

This is the layer where the business modelling is done, where the flow of business processes and associated business rules are defined. According to our definition, business people should be able to handle and control this layer. Of course, installation of the required software pieces that support this layer remains an IT task. Today, reality is that this layer still is largely undefined. None of the vendors that favour a bottom-up approach (starting at the messaging middleware layer and gradually adding functionality on top of it) already address this layer in a suitable way.

On the other hand, there are a number of vendors that entered the application integration space directly at this level, even without underlying messaging based infrastructure. Their focus typically was on the integration of packaged applications. Gradually they are extending their space to the lower layers, seeking partnerships with some of the leading vendors of transformation engines and hooking up their product to "standard" messaging environments such as MQSeries. Examples of such approach are Oberon and CrossWorlds.

In our architecture, the business layer implements the "translation" of business flows and processes. The basic object that it can manipulate is the business component. The business layer defines the rules for the chaining and the interaction of the components.

The normal evolution will be that workflow-style solutions will play a major role here. But workflow alone will not be sufficient. A complete solution will have to incorporate process modelling facilities, simulation capabilities and support for complex logic.

## The Component layer

The component layer is the highest stack level that is under control of IT. This layer provides all necessary building blocks that the business people can manipulate in above layer. In our messaging based EAI approach, business components are logical objects that represent the execution of one or more business functions. Being a logical object, they are instantiated by one or more (physical) executables that perform the programming logic. These instances receive requests for processing and send replies via well defined but not necessarily uniform (messaging) interfaces.

Business components are fully logical elements, without an understanding of the underlying technical infrastructure. Components manipulate "logical data" (meta-data). Therefore, any business component is free to interact directly with any other business component.

Component logic will be executed by the means of an underlying application or program. These are of course physical elements that all have their own internal representation and capabilities. Therefore, these underlying instances have to resolve their technical incompatibilities. Typically, this is handled by techniques such as data transformation, bridging, connectors, etc.

## The Message Services layer

The message services layer provides generic services that can assist the functioning of above layers. This includes services such as: reformatting of messages, routing, load balancing, alternate path switching, message warehouse, etc. This layer together with the Transport Services layer discussed further on is often referred to as a "message broker".

Message broker is one of these often (mis-)used terms in the EAI literature. Since there is no clear definition of what a message broker must be, vendors will use this "label" for very different functionality. Therefore, look at the individual capabilities and not at the generic label.

## The Transport Services layer

The transport services layer is the world of the "basic" middleware products. Message oriented middleware is one of the key players there, but this layer will also incorporate other middleware products such as object request brokers and transaction managers. This layer also has to provide several of other supporting services such as audit, logging and security. From a development point of view, this also includes aspects of API's, message formats and templates.

## Conclusion

Even with such a simple architecture, it is obvious that the challenge is enormous. As said, no single vendor does address this today. Currently we are living a period that traditional EAI technology is merging with e-business oriented solutions such as application servers. Also ERP vendors are increasingly moving into this space. Given the anticipated importance of e-business the stakes are enormous but as yet no clear leaders do exist.

Does this mean that you have to wait? No. The potential business benefits of EAI approaches, even in today's context, are too important. Nevertheless, your first implementation will most likely be a tactical one: selecting the appropriate vendor for a specific business requirement or even a technical need. For a real all-encompassing EAI approach things are still a bit too early.

However, even when not yet going for the full 4-layer implementation model, you still can deploy your "tactical" solution with respect for the principles behind the model. Whatever vendor product will become the "leading" EAI solution of the future, the basic principles of EAI will remain the same: isolation of functionality, deployments that reflect the business process flow, clean interfaces, independence of technology, etc.

EAI does implement an infrastructure that can allow for the "seamless" replacement of business components or even complete applications. Make sure that the same principle is also respected for the EAI components themselves.

---

© 1999-2008, Xpragma bvba. All Rights Reserved.

Xpragma bvba - Mechelsesteenweg 254 - 2820 Bonheiden - Belgium  
Tel. +32-(0)15-340 845 - [info@xpragma.com](mailto:info@xpragma.com) - [www.xpragma.com](http://www.xpragma.com)  
RSS feed: <http://www.xpragma.com/english/rss/xpven.xml>