BROADPEAK PARTNERS, INC.

EAI is Dead….Long Live Point to Point Integrations

For years we have worked with energy trading firms to streamline their operations. To that end, we often get into working sessions with the IT and Business groups to prioritize application development efforts and set integration strategy. One of my favorite questions is, "What EAI tool do you use?" The reaction is consistent; the CIO or IT Manager does that quick look up and to the left and says "Um we have [insert Tibco, Oracle EAI, MQ Series, Solace or other EAI tool]. *But we only use it in a very limited way as we still have a web of point-to-point integrations.*" We hear this again and again. So after 15 years of trying, countless millions of dollars spent with limited results, we declare:

EAI in Energy is Dead

For the uninitiated, EAI stands for Enterprise Application Integration. EAI generally refers to a set of software tools that facilitate communication between applications. At their core, EAI tools contain functionality that makes just about any application able to "talk" to another application. The value logic for this type of application is nothing short of impregnable. The second any energy trading group installs a new application, the demand for it to pull or push data to other applications is nearly instantaneous. But as IT groups know, this is easier said than done because applications "speak" in different languages, over different protocols. To use an example, think of a German speaker who only speaks over the phone sending communication. The recipient is an English speaker who only speaks over the Internet. Somehow the IT group has to get these two together to communicate and frankly, that only scratches the surface. What happens when the recipient is out for a cup of coffee? How do we know that the message was successfully delivered? This is a small sample of the problems IT groups face when trying to connect one system to another.

So along came EAI with a brilliant solution to this perpetual problem. EAI tools generally have four components that ease the burden of application integration.

1. Translation components: These are internal components that do the translation. Using the example above these would convert German to English or in the case of application speak, XML to Java JAXB.
2. Communication Components: These are components that do the communication conversion. For example converting the phone message to an Internet message. In the case of an application it might be a conversion from HTTP to JMS.
3. Integration Patterns: These components add intelligence to the integration and solve the problems that business users face. Just like the English speaker going to get a cup of coffee, an application might be offline, or running end-of-day calculations. Integration patterns enable things like retrying until the application is back up, or other features like multiplexing a message to multiple recipients. The pre-packaged integration patterns in EAIs relieve IT developers from continually having to re-invent the wheel. Every point-to-point integration gets coded up with these routines to meet the demands of the business. EAI offers an alternative to simply call the integration patterns and voila! No additional coding is necessary.
4. Message Queues: This is a big offering of EAI tools. A lot of people think of the message queue as the "big pipe" moving information between the applications. Once a piece of data is available in a connected application it is instantly available to every other connected system.

When you put all these components together you get the infallible logic of EAI. Any connected system can talk to any other connected system regardless of architecture differences and no more point-to-point integrations.

So where did we in the energy business go wrong? Why have so many energy EAI projects ended up far, far short of their original architectural plans? In a nutshell, the business demanded it be so. Here are some reasons why:

1. IT Architecture First, Business Second: This is a project killer. A lot of EAI tools were bought with a grand architecture in mind, but the projects to implement them don't deliver immediate business results. The project may start off with a lot of enthusiasm, but it is only a matter of time before the vigorous demands of the business kill it off. In an energy trading enterprise there are just too many needs and not enough people, and keeping up with the dynamic nature of the business takes priority over architecture projects. So a grand EAI effort gets about a week before it is killed down to a point-to point integration project. Thereafter, there are few IT managers with the fortitude to pick it back up again.
2. Substantial Underestimation of Effort: In combination with #1 this is the knife that does the slashing. Leveraging EAI tools requires learning technology specific to the selected EAI vendor. It is doubtful that the energy enterprise has people with these skills in house, thus special consultants are brought in. With an EAI, every application requires an "adapter" in order to "plug-in" to the big pipe. Consultants spend countless hours "plugging-in" a single application so it can be a participant on the pipe. With the high cost of this it is only a matter of time before the project peters out. Even if the organization has the budget to spend on initial integrations, the steep learning curve to EAI technology means a marriage to the vendor and will require ongoing special consultants to have any extensibility with the product.
3. Energy applications are nasty…really nasty. Adding fuel to the underestimation of effort is the very nature of the applications we use in energy. Not only are the applications we use far from cutting edge technology, they include a veritable spaghetti-works of complicated data. The "keys to the application kingdom" is an API (Application Programming Interface). Think of this as the gateway to getting data in and out of an application. In the energy business, when we walk into any given integration project, there is a 50% chance that a particular application will have no API at all, and for the other 50% there is a 90% chance that the API is either very limited to a narrow part of the application, entirely undocumented or both. The result of this is that the "plugging" part of integration into an EAI takes far longer and requires double the resources as originally thought. Double? With the challenging applications it takes a very strong subject matter expert on the application, coupled with another consultant that understands the EAI.

Put these items together and you have a failed EAI project within a short period of time. The only integrations that survive are those where the trading desk, absolutely positively needs the data in real time (read market prices, bids etc). But these are few and usually do not justify the massive cost of EAI tools.

So are we back to point-to-point?

Yes, in fact, we in the energy business never left it. We feel pretty comfortable saying that the energy industry never really got out of the EAI gate at all. But we can learn a lot from what EAI has to offer to make our point-to-point integrations more sensible, manageable and faster to deploy.

We were asked by a client to build an interface from ICE and NYMEX into their system of record (ETRM System). A typical application, the ETRM system had an API, but the API was unpublished. The thing is that

trades were not only needed by the ETRM system, but also several other downstream systems, including a risk system and margin systems; one with a limited API and one with none at all. Exchanges "typically "speak" in a language called FIX but none of the downstream systems could accept FIX messages. We could have used an EAI tool here. It would nicely translate between systems, but the client was really not up for a 7 month endeavor to connect all these systems to Tibco.

So that got us thinking. What if we built something in plain java with all the components and integration patterns but without all the baggage of EAI? Using a common language minimizes the learning curve making implementations simple, empowers clients to extend the product as their infrastructure grows and eliminates the dependence on vendor technology. We could accomplish all the promises of EAI without tethering clients to vendors and expensive consultants.

So that's what we did...we packaged up 140 communication and translation components together with 65 standard integration patterns into a framework called F1. Then we wrapped that framework into a simple user-friendly application called K3. K3 can take a message in any established language and automatically convert it to another. Just simply write a route through K3's F1 framework calling the required components for translation and patterns. Out the other end the user gets a message formatted properly and sent to desired destinations, which could be a system, file, database, messaging queue or others. Above all what we had to ensure was that absolutely no proprietary language/methodologies were used at all, just simple POJO (Plain Old Java Objects). Take any off the shelf IT Java developer and they will know exactly what to do.

This is the underlying technology framework that drives K3. K3 is designed to provide total business transparency to data flowing through it. For trades and prices it automatically reconciles to show whether a trade or price has been correctly booked in the downstream system.

The key is that F1 lives as a set of universal translation components for the energy enterprise. Part of the challenge with EAI tools is that the investment is massive. And once you go down this road with a vendor there is no going back without throwing out the whole thing. By connecting to K3's simple Java objects, the enterprise can use any integration approach they like; point-to-point or EAI if they should still choose and any EAI vendor at that. By running K3 with the F1 framework the energy enterprise can integrate far more easily and with far more flexibility than before.

If you are looking to efficiently manage point-to-point interfaces, wait no longer…give us a call.

Phone: +1.646.370.6804

Email: info@broadpeakpartners.com