

# WHITE PAPER



**MAKE YOUR SYSTEMS  
TALK TO EACH OTHER  
WITH EVENT DRIVEN  
ARCHITECTURE (EDA)**

# MAKE YOUR SYSTEMS TALK TO EACH OTHER WITH EVENT DRIVEN ARCHITECTURE (EDA)

By Dan Matthews

Chief Technology Officer, IFS AB

Just when many of us have become familiar with the term service oriented architecture or SOA, the yet another new three-letter acronyms (or TLAs as we call them) is being touted by analysts and industry journalists. This latest TLA is event driven architecture, or EDA.

So what is EDA? EDA is a term that can be applied to any application that reacts in predetermined ways to changes in the condition of data. That change constitutes an event, and that event may have implications to other parts of a system or to external applications. When data in a particular database field changes, that event can cause an application to send an XML message to other interested applications – or even send an email message or other electronic communication to a specific person interested in that change. So for instance, if a customer is put on credit hold by someone in the accounting department your enterprise application, if it is designed to facilitate an event-driven architecture, can directly notify that customer's account representative. Or a message can be relayed to a third party customer relationship management system or to other interested applications. This ensures that the multiple parties interfacing with the affected customer are aware of their change of status, and that other systems can make appropriate changes to processes and activities related to that account.

IFS Applications has harnessed elements of EDA since 1999, at first using events within the application to trigger emails to specific individuals who could take action as necessary. But with the introduction of SOA, EDA found new utility in the industry. When IFS introduced SOA into its products, we naturally took EDA to the next level by adding the ability to send an outbound Web service call as the result of an event. With the rise of enterprise applications built on services, an EDA can be configured to send and receive these messages from one application acting as a service provider to a web service within another application that acts as a service consumer. Of course each event can still trigger a communication to an

MAKE YOUR SYSTEMS TALK TO EACH OTHER WITH EVENT DRIVEN  
ARCHITECTURE (EDA)

individual user as well, but it is becoming more and more common for events to trigger messages for other applications to act upon.

So when a customer gets credit blocked, the finance package can send an outbound XML message that can be routed to an integration middleware or sent directly to another system. The CRM system can get the message and make a notation in the customer record. The logistics functionality of an application can get the message and take note to avoid shipping further product to the customer. Manufacturing systems can put existing projects that might be in process for the customer on hold.

This high level of internal communication has always been desirable, but today, EDA is rising in prominence because it is an excellent way to construct SOA-facilitated integration across technology platforms, making EDA an excellent complement to SOA. In particular, EDA is an effective tool for managing change in SOA integration environments.

### The more things change ...

Change is happening more rapidly in business than at any previous time in history. Product lines are added and dropped, new regulations are imposed and geographic scope shifts so quickly that it can be difficult for a company's internal systems to keep pace. And the more things change within an enterprise that relies on several integrated IT systems, the more challenging it is to manage these complex integrations. For instance, imagine a company running IFS Applications for finance, manufacturing, and enterprise asset management with these IFS tools integrated with outside CRM and chemical labeling compliance packages. Let's say the company decides to replace its existing CRM system with a product from a different company. EDA makes swapping out the CRM system easier.

Sometimes the role of an application in an integration changes, and EDA helps manage this type of change as well. Let's say the CRM application in the above scenario was initially used to handle business development, sales and customer service functions, but was redeployed to also coordinate after-sale service and maintenance in conjunction with a maintenance repair and overhaul application. Given that the CRM and MRO applications and underlying enterprise applications involved in the integration are based on robust SOAs, EDA can manage the change in roles of applications in a system as well as changes in the systems themselves.

How will these changes affect a company's integration environment? Thanks to EDA, this change in CRM product need not affect the integration as a whole all that much.

How does EDA do this? One could compare EDA to the email distribution lists that you probably use to keep employees of your company informed. In your company, you likely have a number of these email lists that go to broad spectrums

MAKE YOUR SYSTEMS TALK TO EACH OTHER WITH EVENT DRIVEN  
ARCHITECTURE (EDA)

of people interested in various topics. For instance, there may be a list of people involved in planning the company holiday party or a list of people who need to be informed of new contracts. As your personnel change or as new people need to be added to these lists, they simply subscribe or unsubscribe to the email list. There is not a lot of change required to the distribution system as a whole.

Just as someone sending a message to an email list does not have to be directly concerned with communicating with each individual member of a list, an EDA allows an application to simply distribute event-driven messages to other computer applications that are interested or need to act on that data. Of course there is a much higher degree of complexity involved in integrating applications using EDA than there is in setting email distribution lists. But as you change out the CRM product in the above scenario, it is simpler to “subscribe” the new CRM system to messages that impact its operations than it is to rebuild a point-to-point integration each time you make changes to your integrated application environment.

EDA is easy – and is getting easier – to use as an integration tool. When IFS first offered event-driven functionality, we created an Event Server that allowed users to specify actions taken based on 200 different application events within IFS Applications. In early 2007 however, a service pack to IFS Applications 7 included a Custom Events feature which allows our users to create an event when any of the objects in IFS Applications are created, updated, modified or change state. This effectively creates 20,000 potential events. Anything from an invoice being cancelled or modified to a change in a “valid to” date on a service agreement can route a signal to your integration middleware that will forward it to other systems and other applications that need to act upon that information. No one will likely use all 20,000 potential events, but there is certainly enough agility built into IFS Applications to allow users to harness EDA for their integrations in precisely the way that makes sense for their business.

### Not a panacea

EDA will become a more and more popular choice for enterprise application integration due in large part to the rise of SOA-based applications like IFS Applications. Using EDA to unite applications with easily exposed Web services that can serve as integration points, we can move from a model of point-to-point integration of our systems towards a publish-subscribe model of communication. However, EDA is not the right tool for every job. Furthermore, like any popular TLA or buzzword, the term EDA may be misapplied or used to obscure the nature of specific technologies.

EDA is not, for instance, a good tool for building an enterprise application, but excels rather at uniting applications. In the context of a large enterprise suite, it is possible to use EDA to unite various modules in the suite using EDA. But current

MAKE YOUR SYSTEMS TALK TO EACH OTHER WITH EVENT DRIVEN  
ARCHITECTURE (EDA)

interest in EDA has more to do with integrating different applications across an enterprise and integrating the applications of a company with those of its customers, suppliers and partners. Why is this? Because it is a lot more labor intensive and less efficient to run transactions within an application through a generic event mechanism than it is to use traditional function calls generally used to construct enterprise applications. EDA is an excellent way to achieve greater degrees of decoupling and independence between elements of an IT system. But components of an application that are designed to work together – including the thousands of granular components that make up IFS Applications – share design elements so they deliver a higher degree of dynamic interplay between components and richer functionality than could be achieved easily with EDA. While the many granular components that comprise IFS Applications are carefully engineered for the high degree of independence necessary for a true SOA, they are exchanging large amounts of complex data. The XML messages that make up an EDA tend to be much smaller and more rudimentary, and generic enough to be understood by a broad spectrum of other applications.

EDA is not, in and of itself SOA, nor is it a replacement for SOA. So when Virgin Islands-based Infor Global Solutions in March of 2007 announced that their “Infor Open SOA” would provide an “event-driven SOA,” both the terms EDA and SOA were likely being used imprecisely. In order to allow the large number of products that Infor has acquired over the last several years to be integrated with each other, it would appear that they will open integration points in each product and provide an event mechanism so that specific types of XML messages can pass from one Infor-owned application to another. However, employing this degree of EDA does not really deliver an SOA within each application, and Infor does not seem to be planning to re-architect its portfolio of applications to deliver a granular, agile SOA. Rather, they will be offering a middleware conduit to allow data to pass back and forth between their various legacy systems.

## Conclusion

Once again, EDA is not a replacement for an SOA, but rather a way to integrate applications using Web services. The more granular the application’s SOA design is, and the more different services your application can open up to you, the more flexibility and agility you will have in configuring and reconfiguring EDA-based integrations.

*Dan Matthews is the Chief Technology Officer of IFS R&D. He has been with IFS since 1996, working with software architecture and IFS technology platform. Dan holds a MSc. In computer science and software engineering from Linköping University, Sweden.*

## About IFS

IFS, the global enterprise applications company, provides solutions that enable organizations to respond quickly to market changes, allowing resources to be used in a more agile way to achieve better business performance and competitive advantage.

IFS was founded in 1983 and now has 2,600 employees worldwide. IFS has pioneered component-based enterprise resources planning (ERP) software with IFS Applications™, now in its seventh generation. IFS' component architecture provides solutions that are easier to implement, run, and upgrade. IFS Applications is available in 54 countries, in 20 languages.

IFS Applications provides extended ERP functionality, including supply chain management (SCM); enterprise asset management (EAM); maintenance, repair, and overhaul (MRO); product lifecycle management (PLM); customer relationship management (CRM); and corporate performance management (CPM) capabilities.

IFS has over 500,000 users across seven key vertical sectors: aerospace & defense, automotive, high-tech, industrial manufacturing, process industries, construction & facilities management, and utilities & telecom. IFS also provides a cross-industry solution for Retail & Wholesale Distribution.

More details can be found at [www.ifsworld.com](http://www.ifsworld.com). For further information e-mail [info@IFSWORLD.com](mailto:info@IFSWORLD.com)

**[www.IFSWORLD.com](http://www.IFSWORLD.com)**

This support offer has been made in order to respond to the requirements of IFS' customers. Since the customers' requirements may be different in some markets, variations of this offer may exist.

IFS and all IFS product names are trademarks of IFS. The names of actual companies and products mentioned herein may be the trademarks of their respective owners. The example companies, organizations, products, domain names, email addresses, logos, people and events depicted herein are fictitious.

No association with any real company, organization, product, domain name, email address, logo, person or event is intended or should be inferred.

This document may contain statements of possible future functionality for IFS' software products and technology.

Such statements of future functionality are for information purposes only and should not be interpreted as any commitment or representation.