

# SOA's apprentice

*You probably think you've heard it all before, but Brian Tingham discovers that a services orientated architecture (SOA) is the start of something big, new and surprisingly powerful*

**I**f there's one point practically all the analysts agree on today, it is that SOA (services orientated architecture) finally enables IT to deliver precisely what business needs today, tomorrow and the next day, without costing the earth. If there's another, it's that any move to an SOA should not be IT-driven, nor for that matter wholesale; it absolutely should be business-driven and piecemeal. Further, they say that pilot work should be undertaken as soon as possible, but also with IT involved upfront like never before – because that's the new deal.

Why now? Because, at last, the IT industry has created a standards-based, platform-agnostic framework that really can make your IT (both legacy and new) agile, yet robust enough to react quickly and predictably to that most inevitable aspect of modern business – change. And the plain truth is, no-one can afford to miss out.

Why the insistence on a much closer business and IT link? Because, quite unlike its less than successful (and ultimately only IT-focused) CORBA (common object

**“Services are platform- and vendor-neutral, so systems can co-operate”**

*Matt Deacon, Microsoft*

request broker architecture) and COM (common object model) predecessors, SOA – or rather, the tools, environments and methodologies that make it work – moves 'programming' largely up to the business process modelling (BPM) level, while still requiring attention to the detail of resources and service provision.

In short, SOA is not just another terminally expensive and ultimately flawed IT bandwagon. Those doing it are making enthusiastic noises about progress and fulfilment of the business flexibility dream – albeit with the usual caveats. They're also making the point that you can do it without having to rip the heart out of your existing IT and start again. Indeed, they're even suggesting that, whatever your next project, you'd seriously have to question why not to go the SOA route.

All of which might come as a surprise to the sceptics who, understandably, see SOA (with its less than illuminating acronym) as leading to yet another hyped up, highly risky mega-project – perhaps even worse than



Illustration by Magictorch

ERP. It may also raise eyebrows among those who consider themselves more SOA-savvy, but have been too stuck doing the day job to see the gathering storm.

Either way, since there are going to be a lot of SOA apprentices out there, it's important to get some explanations, advice and guidance under your belt. Which is why I asked a bunch of SOA aficionados to answer some fundamental questions. Such as, what actually is SOA and what makes it so special? How is it able to make such a valuable difference at both the business and IT levels? And how might you sensibly get started?

First, let's have a description of SOA that makes some practical sense. Daniel Ball, a director with e-procurement software and services company Wax Digital, puts it thus: "Current IT architecture is fairly simple, with technology resources organised in order to perform tasks rather like businesses are organised, with structured roles and relationships in a matrix. SOA delivers a significant shift in that view, with 'services'



being designed without predefined task lists, and simply 'exposed' for use by other 'like-minded' systems and accessed whenever needed, from wherever they reside."

Putting that slightly more technically, Matt Deacon, Microsoft UK's development and platform group chief architectural advisor, says: "It's a way to integrate systems, but with the key difference that it uses a common set of protocols and languages that boil down to the technologies around web services, such as SOAP [Simple Object Access Protocol] and XML. So the point is, services are platform- and vendor-neutral, meaning different vendors' technologies can exchange messages in a secure and cooperative manner."

Hence the services approach takes your view up to functional building blocks that can sit anywhere and, most importantly, can be rendered to mimic the way business people describe their processes. Which is how SOA allows IT to deliver new solutions again and again from a combination of the best of existing IT silos,

mixed and matched with whatever IT innovations are required – and relatively quickly.

Ball again: "Because of the way IT assets are exposed, in the form of re-usable services, rather than hard-wired or point-to-point connections, changes in the way they collaborate can be made quickly and easily, without coding. Equally, additional computing assets can be introduced to quickly extend the network and deliver new functionality." Hence again, SOA responds to today's business prerequisite that systems be 'built to change', rather than the old way of 'built to last', putting business people virtually in the driving seat.

What's more, SOA encourages better design of processes and better innovation, with legacy applications and new. As Tony Dobson, CTO of Wax Digital, observes: "Being able to paint a veneer on existing services, and add to those with new ones, means you can secure your investment in ERP and production systems; you can cut the cost of ownership and maintenance; and you can enable ongoing flexibility that's relatively quick and cheap to develop."

### Accelerate change

How? Martin Percival, marketing director at web services tools specialist BEA, says: "Wrapping technologies are sufficiently advanced that we can pull pretty much any systems of record into an SOA. But the other good thing is we do so with a high degree of abstraction. So you can extend the life of that system and take the decision to replace or upgrade it, whenever it makes business sense."

So yes, SOA does make a serious difference both at the IT and directly at the business levels. Ian Broughton, also with BEA, puts it another way: "If you've got a warehouse management system and ERP etc, linked using traditional point-to-point integration, that's a fairly brittle environment. The likelihood is that large packages will have been customised to death – meaning you're tied to a particular version, it's harder and more costly to upgrade and there's very little agility. But if instead you start migrating to an SOA, you can break down the artificial barriers by re-using services from those existing applications and others that might be required – and now it's sustainable for the future."

And that needn't be hard. Yes, there's bound to be an upfront cost of training and rethinking, but everyone I speak to says the savings are so big, you can expect real returns from the second project. Julian Dobbins, director of legacy software tools firm Micro Focus, says: "One of the fantastic things about SOA is that it's at a higher level, so it doesn't involve you in a new programming model at all. Simple web services interfaces enable applications to interact with one another. So, on the one hand, organisations can take their legacy systems forward because technology now exists to let them expose the building blocks they need. On the other, new package developers are providing web services interfaces to their new applications anyway."

He likens SOA to an evolution of Cobol, but closer to object orientation and so having inheritance. "It's like

▶ having a shuffle board with business components,” he explains. “If you’re a supermarket, and someone says there’s an opportunity to become an insurance company, or a bank, then there are issues to deal with in terms of IT, and taking data and functionality from one area to another. If you can present all that as graphical business processes on a shuffle board, then you can see the changes you need to make, and make them very easily.”

Once again, that’s how business leaders get to take on a much more central role, very much hand-in-hand with IT, but now working at the pace the business wants and using recognisable business language. And it also works by dint of good old-fashioned business analysis and modelling.

### Flexible foundation

Kobi Korsah, senior product marketing manager for Computer Associates’ Wily (application performance monitoring) division, says: “Not only are business services exposed in such a way that they enable coherent interaction between different parts of an organisation and its partners and customers, but they explicitly tie business objectives to the technology and methods – as a result of business process modelling.”

Which is the answer to the question about where you need to start. Micro Focus’ Dobbins again: “You can’t change with confidence what you don’t understand. Any IT strategy involving an SOA needs to be rooted in three things – a high-level understanding of the corporate goals, what IT assets are already in place [including what they do and their interfaces] and what value they deliver. Once you have that foundation, you can extract the costs, complexities, languages, platforms etc [there are technologies that help], and then make decisions appropriate to the needs of business.”

“So, if the business says it needs that functionality from this application, but it’s costing too much to maintain – and that could be due to the platform or the complexity of the application – that might be your first SOA initiative. Or maybe some aspect of your IT keeps on having to change because it’s part of a dynamic business process. So that could be your pilot project.”

“Say it’s a mainframe application and your users are interfacing to it with a bunch of green screens. To make that more usable, you can wrap it up and present it as business components that can be assimilated into some composite application or rendered in HTML onto a portal – either way, a system pulling data or business information from different places. Another example could be building self-service applications (for customers) that work over the Internet – providing information over XML and plugging into partners’ local systems or presenting the information to local portals.”

Meanwhile, if your main IT provider is one of the big packaged software boys, such as SAP or Oracle, it might be a slightly different story. As Adrian Simpson, senior business consultant with SAP Netweaver, says: “For SAP, they need to upgrade to the current ERP 6.0 release – then it’s easier to use SOA because the web services are exposed and they get the Netweaver platform.” Also,

you’ll get what SAP terms ‘enterprise services’, meaning high-level services with rich content and business semantics, such as ‘cancel order’.

As for the ‘how’ to do it, there are several ways, but it makes a lot of sense to talk to application toolset firms such as Oracle, IBM and BEA. Fundamentally, you need an integration development framework, software tools that provide the wherewithal for everything from BPM to generating BPEL (business process execution language) or equivalent, and an application server. So that’s tools to enable object modelling, process modelling, data modelling, potentially writing Java code etc, supporting UML (unified modelling language) that allows business and IT people to generate ‘code’.

You’re also going to need an ESB (enterprise service bus), so your services can communicate and users can have single sign-on, security and other essentials. It’s not as complicated as perhaps it sounds. As Oracle’s vice president of product management for Fusion middleware, John Aisien, puts it: “SOA is no different from other applications – you need to define and enforce security, you need a service registry that governs services around the components, and you need BAM [busi-

**“It’s like having a shuffle board with business components”**

*Julian Dobbins, Micro Focus*

ness activity monitoring] software to give an insight into in-flight activity against KPIs.”

So finally, when should you start? Unsurprisingly, the universal answer is immediately – and not just because it will save money and improve flexibility. BEA’s Percival reminds us of the current unsustainable position: “Ideas and new business models are being hampered by the old way, because the vast majority of IT budget and resource is being consumed with just ticking over – and serious change can take 18 months.”

Just one note of caution: make sure you have performance monitoring and governance under control. Frank Hill, worldwide manufacturing director with non-stop computing firm Stratus, suggests that, as you deploy interdependent services, your reliance on continuous availability increases – and that’s across the network, database and all levels. “The entire ecosystem has to be available, so you have to take a more holistic approach to monitoring the architecture – and that has to include the people and processes too.”

The last word goes to Wax Digital’s Dobson: “Don’t be sceptical. It really is the first time that the IT function has had the tools to do the business job well, and it’s the first time that the business has really been able to get involved in what IT is doing. SOA will force together business and IT, and create fast, agile development methodologies. With SOA, you don’t need a 300-page spec any more; you can start just by mapping business processes with the business users involved.” ■

**Enter 301 at [www.mcsolutions.co.uk/enquiry](http://www.mcsolutions.co.uk/enquiry)**

