Ebusiness technologies: foundations and practice
What SOA offers

v/o
What SOA offers is the means for communication between business and IT to take place at the higher level of business process and business service.

Man
Well first of all business services are very much a level of abstraction from an IT sense in terms of architecture built upon a kind of inheritance tree of, you know, the IT paradigm around component based architecture. So the IT guys are, kind of, already familiar with those kind of constructs.

v/o
Although a business service is a high level abstraction, the implementation is by means of low level components, the individual web services. Does the publish and re-use model hold any pitfalls for business?

Man
A web service, at the end of the day, that links two partners, two parties, especially if those are external parties, is a contract. Once a contract is established, you can't change it, you know, and if you change it there will be issues. So you need to get that contract right, you know, as much as possible and getting that contract right basically means that a lot of thoughts needs to be put in before publishing, before defining this kind of interaction.

v/o
The detailed level of planning required is manifest as WSDL descriptions and XML schemers for each business service. Beneath the public interface, individual web services are orchestrated into discrete business processes by means of business process management languages.

Man
BPEL has really been the enabler for that ability to turn real business process, you know, real business process artefacts codified in tools, which are not just Visio, which are not just PowerPoint, but is equally malleable and easy to use and to learn as those kinds of tools in a way that's actually much more meaningful. Actually codifies intellectual property that the business understands, that differentiates itself that then we can turn through and transition actually into an engineering and assembly and a deployment phase.

Man
We definitely see business process management as being a real enabler in terms of being quickly putting together solutions from a number of different services within the organisation. In terms of those languages, the key ability that we're looking for is a dialect that we can use to exchange information between the business and IT and that's where things like BPMN come into play as being a notation we can all agree on.

What orchestration gives you is a way of representing your lower level message flows up as a single business service, which can then be consumed by BPM. And I think that by doing that it kind of gives you a dial that you can adjust to tweak the granularity of your business services. Granularity is very important in implementations in terms of how you divide up your landscape into services and it's easy to get wrong and what choreography can do is it can

give you a knob to tweak to kind of bring some services, which may be too low level, together into a higher level service, which you can then represent and meaningfully re-use.

v/o
So what of the future? The software vendors face the challenge of creating tools for the developers of SOA solutions built around open standards.

Man
Very much the trend is around the industry flavour, around SOA. So making sure that as, certainly as a software vendor like IBM, we need to gear out what we're doing and what we're delivering, to be much more relevant from a sector perspective within the industry.

v/o
But others see the ultimate goal as placing the power of business process languages such as BPEL, within the hands of business managers.

Man
So I think this is where we're moving towards and business rules community whereby business users can use their business knowledge to define rules that define relationships between the islands of information that typically exist in an organisation and to help them execute their decisions easier.

v/o
But perhaps we should leave the final word to the web services developer.

Man
Whether services oriented architecture gives us the freedom to create better web services. I guess it's hard to say as a developer, depends what part of the development organisation you might be in. You might be buying into a bigger architecture in which case, depending on the web service you're creating, it may fit straight into the overall architecture. Or if there's a business requirement that requires a web service that is quite different to the rest of the architecture, then maybe architectural constraints could be a problem there, need to be worked around.