The Mainstream
An article from the IBM @server zSeries software newsletter

Find the current version of The Mainstream here **ibm.com**/software/zseries/mainstream

SOA: Designing tomorrow's business today

from The Mainstream, Issue 15 - 2005
The IBM @server zSeries and S/390 software newsletter

You've seen them: College-educated employees flying paper airplanes across the room because an e-mail server has gone down and no one can work. Or in retail, when point of service becomes point of failure. Business comes to a standstill. That's because today's IT systems are equivalent to business processes.

This "equivalence" is between the concept of the process from the business person's perspective, and its implementation through automation. It's expressed in an IT model of the process that in turn is realized by applications that mirror the model. To maximize the value offered by the IT system – and eliminate those paper airplanes – the IT systems must be able to change as quickly as the lines of business generate new ideas. Why? Because lack of flexibility is costly.

The link, or equivalence, between business process and IT is evident not only during downtime when measuring the cost of even a few minutes of lost productivity can be painful. It's apparent when line-of-business managers must react quickly to changing business dynamics, or longer term changes, such as the enactment of new laws that raise compliance issues.

Service oriented architecture (SOA) prepares IT to be flexible, relevant and supremely responsive to anything thrown its way so you can continue with the job of powering the business.

ibm.com/software/zseries/mainstream

Natural evolution toward standardization

SOA is an evolution of what IT management and companies like IBM have been driving for a long time: a better and more standardized way to achieve business flexibility from the perspective of IT systems through easier and more flexible integration of applications.

IT shops have long tried tying their IT systems directly to business requirements, applications and initiatives – with limited and mixed results. You could say it began with procedural programming and subroutines. Some shops had a few brilliant systems and application engineers who took the subroutine idea and extended it across the enterprise. These were components and a sort of component-based architecture.

That evolved into more of a client-server model, and eventually into the Java™, J2EE and object-oriented programming models. SOA is a logical extension of the long-term vision of IT. It gives businesses a great opportunity to integrate applications under a single standard that enables reuse of existing assets.

The difference now is that the cost-effective IT shops of today reuse elements of their systems for different purposes at different times, as new projects are launched and old ones are retooled. Service orientation enables them to do that.

For hardware, this is often referred to as a shift toward composable hubs. From a software perspective, it's about composable assets. SOA gives you the architecture by which to decide how to make assets composable to an "atomic" level of work. This wholly contained unit of work can then be deployed as a Web service, which can be called by any program that needs to execute that unit of work. It might be a customer look-up unit of work, or an inventory update or any other number of processes.

SOA treats these Web services as common currency units that can be freely traded between applications – so that the same function can be used in any applications, no matter what their platform. There are protocols under Web services like JMS and HTTP and XML, message formats and SOAP message headers and SOAP message formats. When we call them Web services, we're generalizing them; we generalize them even further as SOA.

SOA doesn't standardize how individual processes work, or how different machines work, or how different programming languages in different machines work. It's a standardization of their interfaces. It lets each business code their applications in the way that makes sense for the system, for example COBOL or PL/1 for an IBM System z9®. Because it simplifies the movement of information between applications based on business rules, it reduces the time, cost and risk of integration projects.

It has become the preferred model among major IT vendors, including Microsoft®, which is supporting Web services and defining common repository formats for storing locations and definitions of Web services. In fact, IBM is working with Microsoft and industry consortiums on defining XML and WSDL standards, which define what a Web service does.

The development environment underneath the Web services may be different, but the overall architecture and the definition of a Web service – and how an application calls one – is standard across IT.

Why SOA is taking off: Rich rewards

When measuring the value of SOA, businesses are finding the most dramatic improvements in lowered costs of maintaining software systems.

When hardware is fully depreciated, businesses bring in the next generation. But software is the gift that keeps on giving for decades. There are plenty of examples: CICS, 36 years old and still critical; WebSphere® MQ, 12 years old; IMS™, a 37-year-old application.

ibm.com/software/zseries/mainstream

The cost of maintaining enterprise software takes a serious chunk out of IT budgets, more than 60 percent is common. This leaves little flexibility for investing in new applications, or making changes that come out of new business requirements, or changes required by new regulations.

Here's how it works: Businesses continue to develop systems, applications, hardware and networking that provide specific function to their enterprise. With specialized applications and specialized hardware supporting those systems, new requirements continue to pop up. So developers write yet another application that may or may not be based on, or reuse anything from, a previous application. Starting from scratch or rewriting existing code is a costly proposition.

SOA allows companies to reuse assets by building them into atomic units of work. When business requirements and related processes change, they can reuse entire pieces of existing business process, in the form of code, and replace or recode only small parts.

The benefits are evident in industries across the board since most major businesses face regulatory compliance of some sort. That's one reason why industries that are heavily regulated are finding SOA so appealing: They never know when the next law is going to be passed.

The concern about changing laws is a legitimate one, says Rick Thomas, program director for IBM's Enterprise Software Platform Marketing. "The question is, how much does it take from scarce resources to make your processes and applications comply with new and far-reaching amendments to regulations, such as Sarbanes-Oxley and HIPAA? You can't possibly anticipate what's coming next from lawmakers, for example. Or new trade regulations you heard a United States legislator debating last month. Will it result in a changed business process? Is it going to be a major issue for IT to remediate? With SOA, companies can be ready for whatever hits them."

In addition to the fast action that components enable, SOA makes business processes more easily tracked, managed and audited by making it simpler to follow the transaction paths, for example, that Sarbanes-Oxley requires companies to report on. The granularity of reporting will be enhanced with some of the upcoming process management products and technologies that IBM is developing.

How to ensure success with SOA

There are potential pitfalls to SOA, but following a methodical, best practices approach will help you avoid them. Let's look at the highest level possibilities.

If you tackle too small a project, you won't see demonstrable results in timely manner. This can ruin your momentum, turning an exciting challenge into lackluster performance. More importantly, it can impact subsequent budget allocation when you're unable to show results.

Or, you can have the opposite problem.

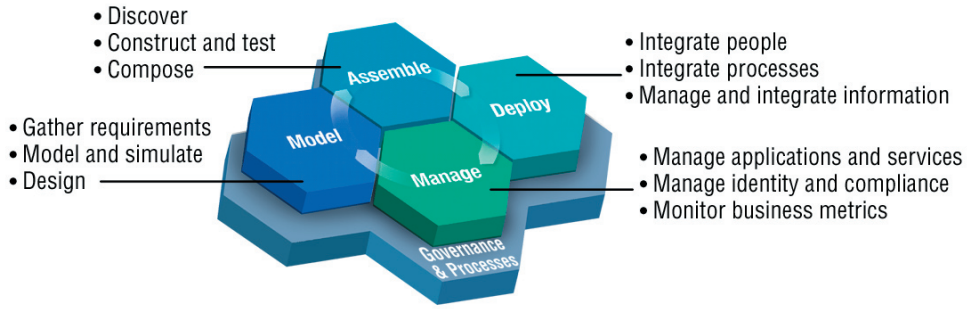
The power of building Web services for an SOA is that you are taking that subroutine way of thinking and extending it to your entire business. Subroutines can be big, small, or they can be subroutines within subroutines. When you take business processes, as represented by subroutines, pull them apart and turn them into atomic units of work, what do you have? If you've tackled too much without adequate planning and modeling, you can end up with a bag of nuts and bolts, and no instruction manual for putting them together.

The essence of a successful SOA implementation is having the appropriate tools to build a well-thought-out suite of Web services, tools to deploy them as composite applications and tools to manage them to the highest service levels. The approach is just as important as the scope and scale of the undertaking.

ibm.com/software/zseries/mainstream

The life cycle approach

There is a middle ground between too slow and too overzealous, and that's why IBM recommends that you manage the process using a proven methodical approach: from modeling and assembling, to deploying and monitoring. This is what we call the "SOA life cycle."



Model

The Model phase of the process starts with discovering which program assets you already have that you can reuse in new applications. Discovering these hidden assets, and determining which programs are good candidates for reuse in Web applications, is made easier with the WebSphere Studio Asset Analyzer.

Once you understand what business operations you have already coded and available, you need to create end-to-end process models that represent key business processes throughout the enterprise. IBM WebSphere Business Modeller offers business modeling, simulation, analysis, and collaboration capabilities. The models it helps you build can direct your Assemble and Deploy activities for creating composite applications. Other tools help you reuse your existing applications by identifying the business rules coded within your core applications so you can restructure them into more manageable segments.

At the end of the modeling phase, you should have a clear idea of your assets, and where they can be used in the new business processes that you have modeled.

ibm.com/software/zseries/mainstream

Assemble

The Assemble phase is where programs are wrapped as services, creating composite applications from core assets which are them assembled across multiple platforms. IBM tools can help make composite application development faster and more efficient. If you use CICS, IMS or WebSphere transactional environments, the tools simplify the development of new Web user interfaces, traditional terminal interfaces, and back-end business logic.

IBM offers tools for rapidly assembling business solutions with features for integrated development and testing of the models. When you're happy with your composite application model, then you are ready to deploy it.

Deploy

During the Deploy phase, assets are launched into a secure and integrated environment, and they should be taking advantage of specialized services that support integration of people, processes and information. Ensuring that the process runs efficiently, that applications are running well, and that systems are receiving the information they should receive to complete their actions – are essential requirements at this stage of the SOA life cycle.

To help you do this, IBM offers the WebSphere Process Server (WPS) that controls the execution of all the Web services created in the previous steps. It deploys the composite applications, and controls the execution of the process, choreographing the individual programs in the process into an automated flow.

Also available is an advanced ESB from IBM, WebSphere Message Broker (WMB), which handles connectivity to any application, whether standards-based or not. This means that you can choreograph any application through WPS, making it especially useful if you have a large base of trusted existing applications. WMB can mediate and transform messages in flight to meet the needs of the receiving applications, and offload the mediation processing to a zAAP processor, with an immediate benefit in performance and throughput.

Manage

Once deployed, you should manage and monitor the composite applications and underlying resources from both an IT and a business perspective. Information gathered during the Manage phase on key SOA indicators should provide real-time insight into business processes, enabling better business decisions and feeding information back into the life cycle for continuous process improvement.

The IBM WebSphere Business Monitor gives you this information and because it's dynamic, anything of interest – such as a group of users having very slow response times – can be investigated on the spot by clicking and drilling down the data. The answers are immediately available and presented in graphical form, as well as in columns and text. The Business Monitor also loads the real-time metrics back into the Business Modeler for analysis, completing the life cycle loop of continual improvement and reuse of existing assets.

What is the role of the mainframe in SOA?

An SOA can be designed for most computing systems, distributed or centralized. But there are some unique advantages to hosting it on the mainframe, says Thomas. "Scalability is probably the most important consideration for companies building on-demand applications with unpredictable capacity requirements. The mainframe has no true competition in the commercial world. With processors working together in a Parallel Sysplex, it's not unusual for an enterprise to run thousands of MIPs and process billions of transactions per day."

There are certain industries clearly dependent on the reliability of the mainframe platform, including banking and finance, with stock transactions that operate where no mistakes are acceptable. Other companies, such as online auction houses, may run their business on Intel servers and have thousands of them. If a customer performs a look-up and submits a bid, nothing critical happens if the bid doesn't arrive. Yet in both cases, the mainframe has an advantage: While most distributed platforms are reliable and available most of the time, the mainframe's availability is unassailable.

ibm.com/software/zseries/mainstream

The IBM mainframe Workload Management capabilities, unique among all platforms, can easily deploy any size unit of work, up to the most complex composite applications, and execute on it flawlessly. It was designed to manage mixed workloads in the most efficient manner possible and the technology embedded in the Workload Manager ensures that you can meet complex performance objectives for each composite application. Because of this, the IBM mainframe delivers exceptionally high rates of utilization, up to 100% during peak loads. While many companies turn to distributed computing for added hardware flexibility, the functionality required for a successful SOA is already built into the System z9 and zSeries hardware.

The same benefits apply when building your new applications. More than 80 percent of customer data, on average, is stored in mainframe systems, including DB2 and Oracle. Access and proximity to that data by any application – new or legacy – is critical to optimal performance. Applications hosted on the mainframe clearly will perform better, faster and more reliably when they aren't subject to network bottlenecks. In addition, IBM's recently announced zSeries Application Assist Processor, or zAAP, has cut the cost of running Java-based code on the mainframe, while keeping it tightly integrated with core mainframe and business systems.

Perhaps most important to note from a cost-savings perspective is the "home team advantage." When you build your new applications on the mainframe, you can use existing in-house knowledge and expertise not only for developing the applications, but managing them effectively, as well.

Can you afford not to?

On Demand business requires that companies shape their business to become integrated end-to-end, and to be flexible and responsive to the demands of the market and their customers. The cost of continuing to support that paradigm by developing specialized applications with specialized hardware is prohibitive, and change is inevitable – one way or another.

ibm.com/software/zseries/mainstream

Service oriented architecture brings lines of business and IT into a trusted partnership that can deliver an on demand experience in a cost-effective manner. Sounds like an old story you've heard before? In some ways, it is that same old IT dream of a seamless blend of heterogeneous platforms, applications and functionality. The difference today is that we now have the technology to support it, including an entire suite of tools from IBM that can help you reuse your existing assets.

In upcoming issues of the Mainstream, we'll explore SOA more deeply, including each phase of the SOA life cycle and how companies are measuring key SOA indicators for continuous process improvements.

For more information, visit ibm.com/soa

ON DEMAND BUSINESS™