CICS Transaction Server and WebSphere Process Server
White paper

WebSphere software

Extend and enrich the value of your most-critical transactions and data.

By Klaus Deinhart and Steve Wood, IBM Software Group

April 2007

Contents

2 *Introduction*
3 *Delivering on the SOA promise*
4 *WebSphere Process Server for BPM enabled by SOA—choreographing service interactions with systems and people*
7 *CICS Transaction Server: Transaction-processing enabled for SOA*
8 *Combining WebSphere Process Server value-add with CICS transactions*
12 *How to integrate CICS applications with WebSphere Process Server*
15 *Synchronous and asynchronous communications*
18 *Transactionality*
20 *Local compared to remote deployment*
21 *Qualities of service*
22 *Teamwork for integration across technologies and SOA enablement*
23 *Summary*
23 *For more information*

Introduction

Many organizations are under constant pressure to deliver more business value while containing IT costs. In addressing this balance companies are exploring how they can reuse selected parts of their existing IT assets to achieve their business goals rather than constantly expanding heterogeneous, widely distributed and already complex infrastructures. A successful approach to reuse involves transforming your IT resources to a composite application-development framework that is built on an open-standards-based service oriented architecture (SOA). Instead of your having to reinvent the development process with every new application you build, this approach takes advantage of the flexibility of an SOA to help your business adapt quickly as your needs change.

With this flexible architecture in place, companies can gain the momentum they need to achieve competitive advantage in the marketplace. To keep pace, you must ensure that your business processes integrate across your company and with partners, suppliers and customers by building an ecosystem that can respond to any market opportunity or competitive threat.

IBM CICS® Transaction Server is the high-volume transaction-processing platform of choice for most large enterprises in a wide variety of industry sectors. IBM WebSphere® Process Server is designed to enable you to transform your business processes based on an SOA approach. This comprehensive process-automation software uses open-standards-based technology to integrate business processes with a unified programming model that spans people, workflows, applications, systems, platforms and architectures. Today, you can combine the robust capabilities of WebSphere Process Server with the latest CICS Transaction Server Web services capabilities to deliver the flexible platform you need to move your organization toward implementing strategic SOA initiatives.

This white paper discusses how you can take advantage of the power of WebSphere Process Server with CICS Transaction Server to build new processes, take advantage of new channels and drive new business. This solution enables you to enhance business flexibility and IT responsiveness by directly integrating existing business transactions with new business processes. You can build composite applications on the IBM System z™ platform to help optimize responsiveness and total cost of ownership (TCO), and gain improved incremental economic benefits from continued use of enterprise application services.

Combining WebSphere Process Server and CICS Transaction Server in an SOA helps you maintain full transactional control and IT governance over business processes integrated end to end, with advanced fault-tolerance and error-detection capabilities.

Delivering on the SOA promise

The demands for a more-flexible approach for aligning IT capabilities with business strategy are well known. Simple economics and competition put tremendous pressure on IT departments to extract maximum value from their organizations' existing applications. Business managers have often had to ruefully delay their plans because their IT systems had to be laboriously reworked to meet the latest business initiative. SOA and composite-application development enable you to solve this problem by using existing IT resources as services that you can rapidly assemble to create business-process flows.

The strength of an SOA depends on the extent to which it can help you use your existing IT infrastructure to meet your business objectives. An effective SOA should:

- *Take into account the complete life cycle of business processes to help ensure that IT aligns with the business.*
- *Render existing IT artifacts as consumable services.*
- *Manage services to enable efficiency and reuse.*

WebSphere Process Server for BPM enabled by SOA — choreographing service interactions with systems and people

WebSphere Process Server for z/OS (including IBM WebSphere Enterprise Service Bus [WebSphere ESB] for z/OS) enables your business to benefit from SOA by helping you to identify and enable your existing assets as Web services. It helps you ensure that data is always delivered on time and in the right formats, as well as helping you measure, manage and adapt your processes. As illustrated in Figure 1, the functional components of WebSphere Process Server are all delivered as modular service components that can be used stand alone or in combination with each other, as well as with external services. These elements are shown in combination with CICS Transaction Server in Figure 3.

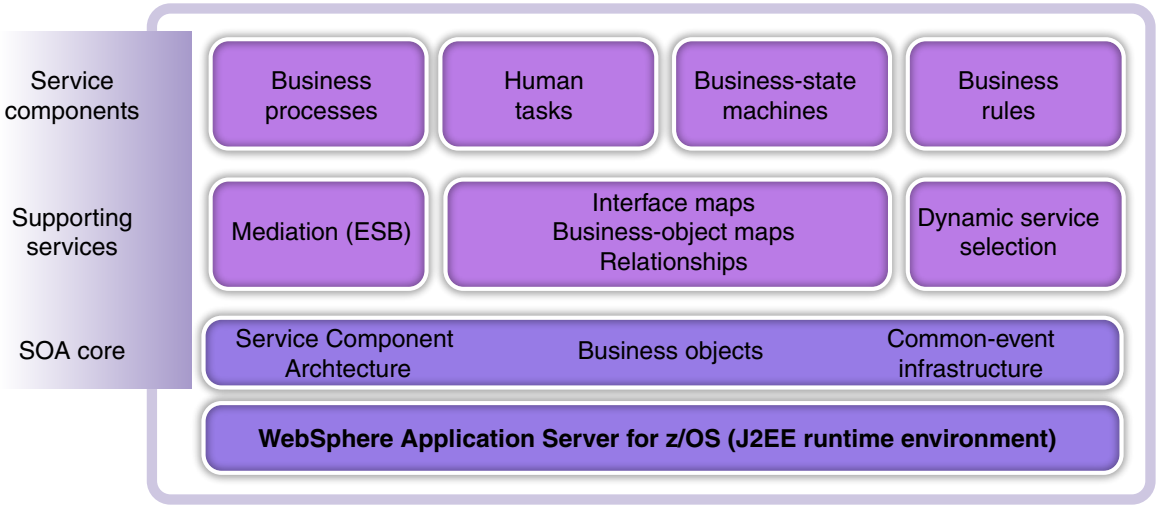

Figure 1. WebSphere Process Server for z/OS combines the power of open standards with z/OS qualities of service and business-centric SOA to support composite applications.

The quickest, most-efficient route to responding to the demands of the business while taking advantage of existing value is through reuse. This enables business processes directed by WebSphere Process Server to choreograph critical CICS line-of-business (LOB) applications, composed into business services, to form a business infrastructure that can be used again and again.

In many ways, CICS users are in an enviable position. CICS applications represent a rich profusion of proven applications that you can depend on to run your business reliably and securely. This inventory of highly valuable, critical business applications are a major business asset. However, it is likely that your CICS applications are locked into processes that prevent you from redeploying them in a flexible fashion because of common industry challenges that inhibit the flexibility that promotes growth.

WebSphere Process Server for z/OS is the primary hosting environment for often cross-functional business processes that can involve CICS transactions and other resources. It uses a common programming model for all types of processes to connect and use CICS resources as Web services and to define new business processes that deliver and consume Web services in an SOA. It also helps you wrap and reuse your existing applications as services, interconnect services across a network, and rapidly assemble and change new business solutions simply by dynamically reconfiguring or rewiring services. WebSphere Process Server includes constructs for defining dynamic business processes, business-state machines, business rules, selectors, role-based human tasks, relationships, events, business objects and mappings to interact with external service providers. And IBM WebSphere Integration Developer tools enable you to define implementable business processes and supporting services, while streamlining hand-offs to related application life-cycle tools, such as IBM WebSphere Business Modeler (targeted at business-process analysts), IBM WebSphere Developer for System z (supporting COBOL and PL/I applications, for example) and IBM Rational® tools.

The exciting new WebSphere products, including WebSphere Process Server, WebSphere ESB, IBM WebSphere Adapters, IBM WebSphere Service Registry and Repository and IBM WebSphere Business Services Fabric (adding industry-specific content packages) are built to interoperate seamlessly with each other and can help reinvigorate and revitalize your business, enabling it to become less technology-centric, more business-focused and more on demand.

Figure 2 shows a typical architecture and benchmark scenario for using WebSphere Process Server in an SOA. The scenario is order processing. Service requests come in from one or more front-end access channels, which could be a portal Web site, Web services applications, or your customer relationship management (CRM) or enterprise resource planning (ERP) system. WebSphere Process Server automates service-delivery tasks, including acknowledging receipt to the service requester, checking credit worthiness of the issuer with an external agency, approving the request (automatically or manually through the organization chain), waiting for the payment to be confirmed (through a CICS transaction), setting up delivery (using an IBM DB2® database transaction), updating the stock level (using an IBM IMS™ transaction), logging progress for traceability and audit purposes, and completing the process with a confirmation to the front-end application or user.

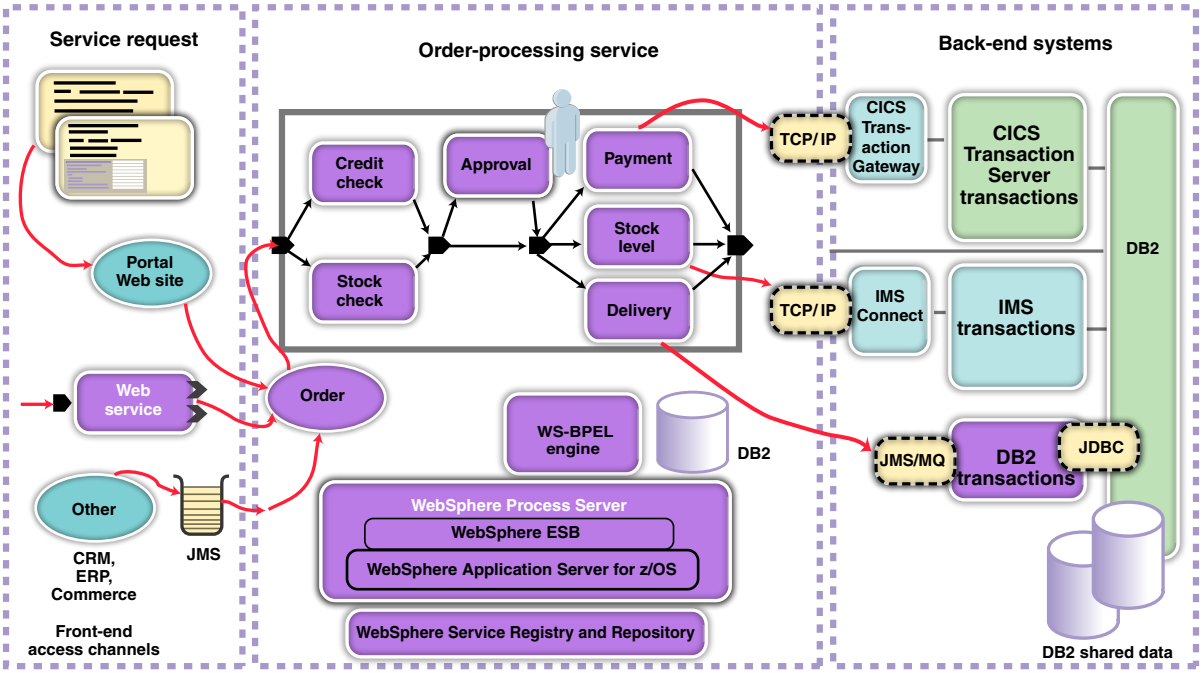

Figure 2. Typical architecture for using WebSphere Process Server in an SOA

This architecture enables you to:

- *Support a variety of front-end access channels, including Web browsers, client programs and a network of front-end application systems.*
- *Automate processes that control, track and coordinate service delivery across a set of activities, interacting with systems and people. Fully automated straight-through processes might well coordinate up to four CICS and DB2 transactions. Longer-running processes can interact with people when needed and coordinate the invocation of a chain of application transactions. Significant event-handling capabilities are built in to cope with expected and unsolicited error situations, including the ability to invoke compensating application transactions for process steps that need to be undone but have already been committed.*
- *Reuse, extend and enrich existing assets in back-end systems as services, while maintaining full transactional control over your business processes end to end, including your most-critical transactions and data.*

CICS Transaction Server: Transaction-processing enabled for SOA

CICS Transaction Server is a high-volume, mixed-language transaction processor for enterprises that need to process a large volume of transactional requests on critical, dynamic and frequently accessed shared information that must be consistent across the enterprise. On large IBM eServer™ zSeries® and IBM System z9™ servers, CICS Transaction Server easily supports thousands of user transactions per second, making it a mainstay of enterprise computing. It also offers a choice of programming languages for application implementation, the most popular ones being COBOL, PL/I, C, C++ and Java™. CICS applications implement a wide range of critical LOB processes in many industry sectors, including bank-teller applications, airline reservation systems, automatic teller machine (ATM) systems and so on.

Although CICS Transaction Server has its highest profile among financial institutions, such as banks and insurance companies, more than 90 percent of Fortune 500 companies rely on CICS Transaction Server (running on the IBM z/OS® platform) for their core business functions. Most state and national governments do as well.* Most people mean *CICS Transaction Server* when they say *CICS*; however, the *CICS* family refers to a portfolio of transaction servers supporting various platforms, connectors (especially IBM CICS Transaction Gateway) and CICS tools.

CICS Transaction Server has a proven track record of successfully delivering new technology, and enabling you to gain advantage at a pace that makes sense for your business, by taking advantage of the investments you have made in CICS technology over the years, while helping to minimize the risks inherent in adopting new technologies. The most-recent CICS Transaction Server enhancement is the introduction of Web services technology for SOA enablement to help increase ease of integration and reuse.

Combining WebSphere Process Server value-add with CICS transactions

Figure 3 shows a typical organization's setup and configuration, involving several CICS regions and Virtual Storage Access Method (VSAM) files with a central DB2 for z/OS database, together with the rich functionality delivered by WebSphere Process Server to extend and enrich existing assets.

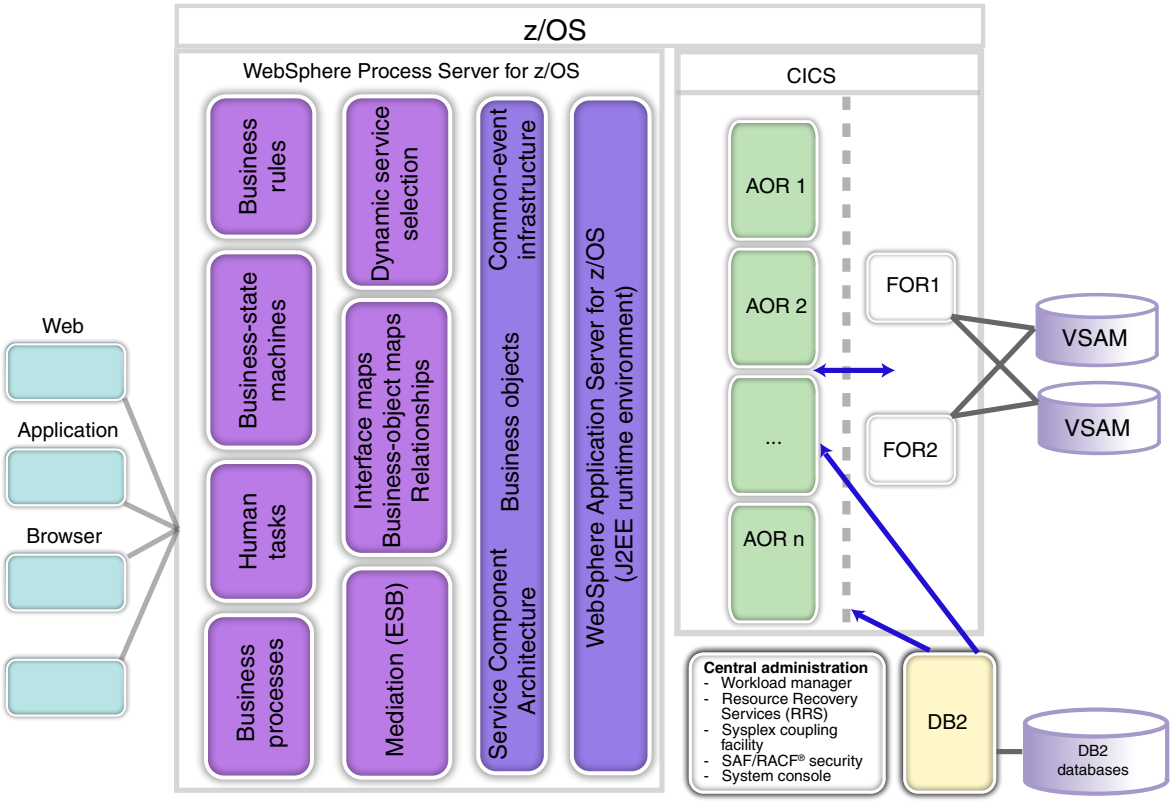

Figure 3. Processes running consistently across multiple front-end access channels and back-end information systems

Key WebSphere Process Server for z/OS, Version 6 capabilities include:

- *Service Component Architecture to facilitate composite-application development, enabling solution assembly from service components, together with integrated testing and deployment*
- *Rich human-task management and business-user client generation to combine human-based services with Web Services Business Process Execution Language (WS-BPEL)-based process automation*
- *Business-state machines to implement event-state action models for process automation*

- *Business-rules support for flexible decision making*
- *Relationship management for data-synchronization workloads*
- *Information service activity to integrate WS-BPEL processes with Structured Query Language (SQL), as well as information server for extract-transform-load (ETL) and enterprise content management (ECM) integration scenarios*
- *Built-in ESB to support message-format transformations*
- *Support for a common-event infrastructure that enables end-to-end monitoring*
- *Integration with WebSphere Service Registry and Repository for service governance*
- *Plug-in support for application adapters as service on-ramps*
- *A common administrative console*
- *Support for open standards, including WS-BPEL, Java 2 Platform, Enterprise Edition (J2EE), Eclipse and Common Base Events (CBEs)*

For details about any of the rich capabilities that are built into WebSphere Process Server and its companion WebSphere Integration Developer tool, visit the product WebSphere Process Server Web site listed in the "For more information" section of this white paper.

The extensive range of WebSphere Process Server capabilities offer a choice of flexible combinations that you can adopt as needed to support a specific scenario at hand, or you can choose a similar or different combination to optimally support the next scenario without having to choose a fundamentally different technology or tool set. You can also select from different entry points, depending on where you are today and where you want to go next. WebSphere Process Server can support the entire spectrum of business process management (BPM) scenarios with a common programming model that supports a wide range of typical workloads, and any combination thereof, including:

- *Human-centric integration processes for workflow automation, rules-based decision making, forms routing, ECM integration, human-based Web services and ad hoc collaboration*
- *Integration-centric processes for enterprise application integration (EAI), business-to-business (B2B), object synchronization processes, straight-through processing, state machines and WS-BPEL*
- *Single transaction microflow and macroflow chains of multiple transactions including support for invoking compensating transactions*
- *Choice of service-component bindings for back-end integration including Java application programming interfaces (APIs), SOAP, Java Message Service (JMS) and IBM WebSphere MQ, involving CICS Transaction Server and other resources, on the same system or anywhere in the network*

The business value of combined solutions typically includes:

- *The ability to serve new channels consistently*
- *Standards-based integration using SOA*
- *Making processes visible, flexible and changeable, end to end*
- *Transaction coordination from start to end, including compensation*
- *Reusing the same CICS transactions with multiple business processes*
- *Intermixing CICS transactions with human-task management without forcing people to learn traditional applications*
- *Improving customer care, regulatory compliance (traceability, auditability), flexibility of delivering new products and services, and reuse of services anywhere (infrastructure cost reduction)*

How to integrate CICS applications with WebSphere Process Server

A range of technical integration options are available with WebSphere Process Server and WebSphere Integration Developer and related complementary products to help unlock and extend critical application functions. These options include:

WebSphere Process Server and WebSphere Integration Developer CICS adapter

WebSphere Integration Developer provides built-in support to generate J2EE Connector Architecture (JCA) adapters to CICS transactions and deploy those adapters as an integral part of composite applications to be deployed on WebSphere Process Server. It provides wizards to quickly and simply expose CICS (or IMS) programs as enterprise services, including the ability to import definitions from COBOL, C structures, CICS basic mapping support (BMS) and IMS Message Format Service (MFS) definitions. You can look up existing functions using the enterprise metadata-discovery wizard.

This option uses CICS Transaction Gateway and supports locally optimized transport protocols when deployed entirely on the System z platform. CICS Transaction Gateway provides connectivity between WebSphere Application Server and CICS Transaction Server, enabling WebSphere Application Server to handle Web services and XML processing.

Adapters provide a service on-ramp to process automation, leaving existing application code untouched and reusing it as-is. The CICS adapter supports synchronous invocations (tight coupling) and supports two-phase-commit control coordination for the microflow process and the CICS application transaction.

JMS and WebSphere MQ

JMS and WebSphere MQ provide for transactional-messaging-based integration with CICS transactions. If you have a WebSphere MQ technology-based infrastructure, you can opt to use the Service Component Architecture (SCA) WebSphere MQ bindings available with WebSphere Process Server, often together with ESB mediation capabilities, to adapt message formats to the needs of an end point or a bridge used by the end point.

CICS Service Flow Feature

IBM CICS Service Flow Feature is a business-service integration adapter for all CICS applications—both terminal-oriented and programmatically callable. Provided with CICS Transaction Server, Version 3, it delivers the tools and runtime capabilities to build CICS business services so that you can create new services more quickly and easily by composing a sequence of CICS application interactions to form a noninterruptible sequence of several fine-grained CICS transactions.

The high-level, business-service interfaces created with this approach enable you to unlock critical IT assets that can be easily integrated into long-running business processes handled by WebSphere Process Server that can wait for input and can be interrupted by events. These processes can be invoked by choosing from a range of transport-protocol bindings, including SOAP and WebSphere MQ, and JCA adapters, promoting reuse as part of your SOA—a far more-efficient process than creating new applications. It supports COBOL data structures, BMS maps and Web Services Descriptive Language (WSDL) documents. Applying this approach enables you to gain extra value from your existing assets by putting them to use in new business-process solutions without having to write new code.

WebSphere Developer for System z
WebSphere Developer for System z provides enterprise developer tools (COBOL, PL/I and so on), including new XML Services for the Enterprise (XSE) to build XML and Web services interfaces into CICS and IMS application functions.

IBM WebSphere Host Access Transformation Services (HATS)
HATS provides tools to transform 3270 screen interfaces for terminal-based application functions into a Web service, generating a WebSphere Application Server application as the client of a 3270 session.

IBM WebSphere Studio Asset Analyzer
WebSphere Studio Asset Analyzer provides tools to help understand dependencies within and across applications and associated data across the enterprise (supporting the mainframe as well as J2EE application programs).

IBM Asset Transformation Workbench
This tool enables you to prepare applications for reuse in an SOA. It can analyze application dependencies, find business rules and help create reusable components.

Which option you use depends on the specific scenario and related solution requirements that you need to meet. Key decision points include:

- *Synchronous versus asynchronous communication*
- *Transactionality and granularity of WS-BPEL processes, including microflow, macroflow, state machine, compensation and human workflow*
- *Local compared to remote deployment*
- *Security, availability (failover), performance and scalability attributes*
- *Development approach (direct integration as opposed to cooperative modernization)*

Synchronous and asynchronous communications
Synchronous communications typically imply tight coupling, where the operation of one system depends on the availability of the other. Asynchronous communications typically imply loosely coupled systems, and can also often support synchronous interactions with similar "good enough" performance. One of the premises of SOA is to help bring both communication patterns together with common Web services standards.

We live in an asynchronous world, with unsynchronized network architectures, platform diversity and wide-area geography. As a result, one system talking to another often cannot rely on the availability of the other.

However, application developers often want a synchronous view partly because users have a synchronous view. Processes share the same fate or outcome, and all must succeed for any to succeed. If one fails, all should fail.

In most scenarios, you will need to support a combination of both patterns:

- *Asynchronous interactions between loosely coupled systems are often used to encapsulate synchronous interactions among tightly coupled system components. For example, adapters often use synchronous interactions for function calls to back-end applications, providing a service on-ramp that interacts asynchronously with a long-running business process.*
- *Synchronous interactions in a WebSphere Process Server microflow can be committed as a single unit of work providing a transactional service (or activity implementation) to a long-running macroflow, which inherently operates asynchronously to coordinate a chain or sequence of distinct transactions.*
- *Information-service activities in WebSphere Process Server provide direct integration of SQL with WS-BPEL, automating synchronous or asynchronous service delivery.*

WebSphere Process Server often interacts with loosely coupled systems and databases while maintaining state information in a tightly coupled database (deployed on the same server or on a separate tier).

You shouldn't design a WebSphere platform application without understanding the nature of the back-end systems it's talking to. That's because it is hard to implement synchronous interactions between asynchronous systems. Synchronicity cannot be guaranteed if the caller cannot rely on the (immediate) availability of the system called. Better options are:

- *Synchronous communications for synchronous interactions between tightly coupled systems*
- *Asynchronous communications for synchronous interactions with back-end systems*
- *Asynchronous communications for asynchronous interactions between loosely coupled systems*

Useful examples of synchronous communications include JCA adapters, Java Database Connectivity (JDBC), HTTP Secure (HTTPS), call and return, SOAP/HTTP (request/reply message interchange), database lookup, information service activity and microflow transaction coordination. Useful examples of asynchronous communications include JMS, WebSphere MQ, SOAP/JMS (one-way interfaces), e-mail, business-to-business (B2B) interactions, event-state-action machines and long-running workflows.

SOAP is a protocol for exchanging XML-based messages over computer networks, normally using HTTP. SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework that more-abstract layers can build on.

The SCA of WebSphere Process Server enables you to flexibly specify, for each of its service components, synchronous and asynchronous interactions using a range of data and transport protocol bindings to ease integration with other services.

Transactionality

In business systems, transaction processing is information processing that is divided into individual, indivisible operations, called *transactions*. Each transaction must succeed or fail as a complete unit; it cannot remain in an intermediate state. Transactions have been applied to user interactions (a classical CICS transaction), as well as to databases and distributed applications. The key properties of a transaction are often referred to as atomicity, consistency, isolation and durability (ACID). Together they guarantee that transactions are processed reliably. Without them, the integrity of an information system cannot be guaranteed.

Transaction processing in distributed, or composite, applications requires commit coordination across multiple participating nodes, which must succeed or fail as a complete unit. The two-phase-commit protocol (XA standard) is a standardized algorithm that lets all nodes in a system agree to commit a transaction. Popular implementations are available with J2EE application servers and IBM z/OS RRS, for example, supporting transaction coordination for tightly coupled distributed systems.

For loosely coupled systems, transactional-messaging infrastructures such as WebSphere MQ support assured message delivery, helping to ensure that messages are delivered once and once only. In combination with two-phase-commit coordination at the end points, this transactional messaging has enabled exciting new capabilities for transactional workflow and process management to choreograph a sequence of interactions with services, both local and distributed. These services can be provided by classical transactions programs (like those hosted on CICS Transaction Server or database-management transactions), by nontransactional applications programs (updating a spreadsheet or sending an e-mail, for example), by human task-management components or by B2B protocols.

WebSphere Process Server can support all the resulting combinations, providing important built-in capabilities to flexibly specify and control the transactional scope of WS-BPEL processes in relation to the transactional characteristics of the services that are integrated. Long-lived interruptible (macroflow) processes represent a chain, or series, of multiple, independent services or transactions. Each navigation step in a macroflow is performed as a transaction, helping to ensure that the navigation is performed once and only once. State information is updated persistently in the runtime database when the navigation step is committed. A navigation might actually span multiple activities, such as when branching into multiple parallel paths.

Short-running noninterruptible (microflow) processes are run as a single unit of work in a single thread, and limited to synchronous interactions with invoked services. State information is transient, updated only in memory until the microflow transaction gets completed and committed.

Invoked services might or might not participate in the transaction of the process they interact with. Services themselves can be transactional or not, and a process can itself be exposed as a service. Every process, microflows and macroflows, can be invoked synchronously or asynchronously, supported by the full range of service-component bindings.

Due to their interruptible nature, long-lived processes allow for:

- *Asynchronous interactions with other services*
- *Coordination of parallel paths*
- *Waiting for unsolicited events*
- *Involving human interactions*
- *State machines (event-state-action paradigm)*
- *Complex B2B protocol interactions with other stateful services*

In contrast, short-running microflows typically support fully automated straight-through transactions choreographing synchronous service interactions. Microflows can support two-phase-commit coordination with participating transaction-processing systems. Long-running processes are forward-recoverable. If there is a failure, the transaction that was in flight rolls back, and navigation continues from there.

Microflows implement the ACID properties. If a transaction rollback occurs, then the transaction manager rolls back all of the transactional services that were invoked. Others remain untouched during rollback, unless a compensating transaction has been defined that can be invoked by the process. As a result, WebSphere Process Server can coordinate transactional and nontransactional services using its transactional WS-BPEL process engine. Transaction scopes can be flexibly specified in WebSphere Integration Developer, for short-running microflow transactions and long-lived interruptible business processes. WebSphere Process Server makes sure that nothing gets lost or is run twice. Even in the case of catastrophic system failures, WebSphere Process Server knows exactly where to resume processing for the composite application.

Local compared to remote deployment

You can add integration software to an existing server, or deploy it stand alone on a separate server in the network. Distributed systems are often perceived as more cost-effective for new solutions than mainframes, typically ignoring the benefits of the mainframe when it comes to the ability to run thousands of applications concurrently on the same box or sysplex, using a central database and central administration. Deployment on a mainframe can dramatically reduce the labor cost needed to maintain the number of distributed servers required for the composite workload, while maximizing utilization of shared processing power and providing 99.99 percent availability for the composite applications.

Key points you should consider for deployment options include:

- *The proximity advantage of deploying the composite workload on the System z platform, as opposed to splitting the workload between a distributed server at the front end of the mainframe*
- *The impact of two-phase-commit coordination between new workload and existing functionality using XA (distributed deployment, tight coupling) or RRS, and potentially in combination with WebSphere MQ for assured delivery*
- *Failover to a local or remote server*
- *Number of interactions between the tiers, and the volume of data passed*
- *Passing security contexts between components and systems*
- *The performance of cross-memory communication (such as hypersockets) as opposed to network latency, some of which can be mitigated if high-speed lines are available*
- *The cost of administering the operation of a single centralized system compared to multiple systems that work together for providing composite business solutions*

Qualities of service

Security, availability, performance and scalability are important nonfunctional requirements for any business solution. For composite applications and mixed workloads, you need to decide on the best deployment platform from a physical-infrastructure and operations-management perspective. Colocating SOA infrastructure with core z/OS assets enables you to use native interfaces for optimized connectivity, and also enables you to make the most of the ability of the z/OS platform to satisfy a wide range of nonfunctional requirements. Important considerations are:

- *Scale*
- *Availability (on the hardware and software level)*
- *Dynamic application of software upgrades*
- *Backup and recovery procedures*
- *Security management policies*
- *Use of system capacity for mixed application workloads and peaks*
- *Number of people to configure, monitor and adjust infrastructure*

Consolidating the most critical pieces onto a single system offers significant economies of scale, and helps ensure the SOA infrastructure becomes as reliable and as secure as all of the most important assets that you extend and integrate.

Teamwork for integration across technologies and SOA enablement

From a development perspective, you have the following options:

Direct integration of existing business transactions with new business processes

In this case, existing application code remains unchanged and is reused in place, typically using an adapter for message mediation as a service on-ramp to your processes. That is, the integration developer wraps existing application functions as a service to be deployed on WebSphere Process Server. This approach is useful when the CICS transaction already delivers the functionality required and if you need to complete the project without involvement of a CICS application developer. Note that both synchronous (tightly coupled) and asynchronous (loosely coupled) interactions can be supported, depending on the integration technology that you choose to apply.

Developer teamwork between the CICS application developer, the WebSphere integration developer and potentially a business process analyst

The beauty of this approach is that you enhance your CICS transaction system for services that are needed by new business processes. This approach is useful, especially when the existing CICS transactions do not really deliver the requested service functionality yet. It can be a major element to modernizing CICS Transaction Server for active participation in SOA and related standards that better support reuse and business flexibility.

In this case, existing CICS transactions are enhanced to expose application functionality as services that meet the needs of the process-automation project at hand, and for reuse with other SOA projects. That is, the CICS application programmer extends existing transactions to expose modern services, working hand in hand with integration developers who focus on automating business processes while hiding the implementation details of application services that are integrated into the process. Again, a range of integration technologies and interaction patterns are supported with this approach, giving you full implementation flexibility while modernizing your CICS applications for a service-oriented world.

Summary

WebSphere Process Server for z/OS introduces rich business-integration capabilities to run your SOA on the z/OS platform. Built on a common, unified, standards-based platform, WebSphere Process Server for z/OS is designed to extend the value of core applications and databases while retaining the availability, security and recoverability qualities of the z/OS platform. Powerful capabilities are available today to combine WebSphere Process Server with existing CICS line-of-business applications that represent a rich seam of untapped potential for reuse.

By employing this combination of capabilities you can transform critical, proven business assets straightforwardly into Web services, and compose them into business flows directed by WebSphere Process Server, that export all the qualities of service that you depend on into new environments. You can also extend all the nonfunctional qualities of service that you depend on today by colocating your SOA infrastructure with your core assets to support new composite business solutions. And the available tools can help you harvest the benefits without special skills, and with reduced risk.

System z applications and data are among the most valuable assets a company owns. Incrementally reusing these time-tested and proven assets through extension and integration is a cost-effective, reduced-risk, fast path to achieving your strategic business objectives.

For more information

To learn more about IBM WebSphere Process Server, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/integration/wps/
ibm.com/abstracts/sg247378.html

To learn more about IBM CICS Transaction Server, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/cics

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
04-07

WSW11314-USEN-00