

WebSphere Business Integration: An architectural overview

by K. Bhaskaran
M.-T. Schmidt

Business integration is a set of capabilities that characterize service-oriented architectures for e-business solutions. These capabilities include the ability to integrate and manage business operation systems, enterprise information assets, business partners, and a collaborative network of decision makers to address specific business problems. WebSphere® Business Integration is an IBM software platform that embodies these capabilities to offer rapid time to value for delivering integrated e-business solutions. In this paper, we provide an overview of the WebSphere Business Integration platform—the architecture, programming model, tools, and runtime.

To become flexible, efficient and responsive, businesses are trying to address their end-to-end integration problems. For example, a solution that integrates customer relationship, financial, logistics, and procurement systems can enable a business to become more responsive to customers' demands. Typically, such integration solutions feature an ad hoc assembly of a variety of technologies and technology components. This, however, need not be the case.

WebSphere* Business Integration provides a systematic and replicable way of building business integration solutions. In this paper, we highlight fundamental business integration challenges and introduce the WebSphere Business Integration solution capabilities that address them.¹ WebSphere Business Integration provides business modeling tools to capture solution requirements, as well as direct-to-middleware tools to build and assemble service-

oriented software components to meet the needs of the solution. The solution is then deployed to an integrated runtime that can be monitored and managed.

The next section of this paper defines the business integration problem and discusses the WebSphere Business Integration capabilities. Then, the third section introduces the WebSphere Business Integration Platform—the architecture, programming model, and runtime. The fourth section describes the WebSphere Business Integration tools. The fifth section provides a WebSphere Business Integration solution scenario and is followed by a summary.

Business integration

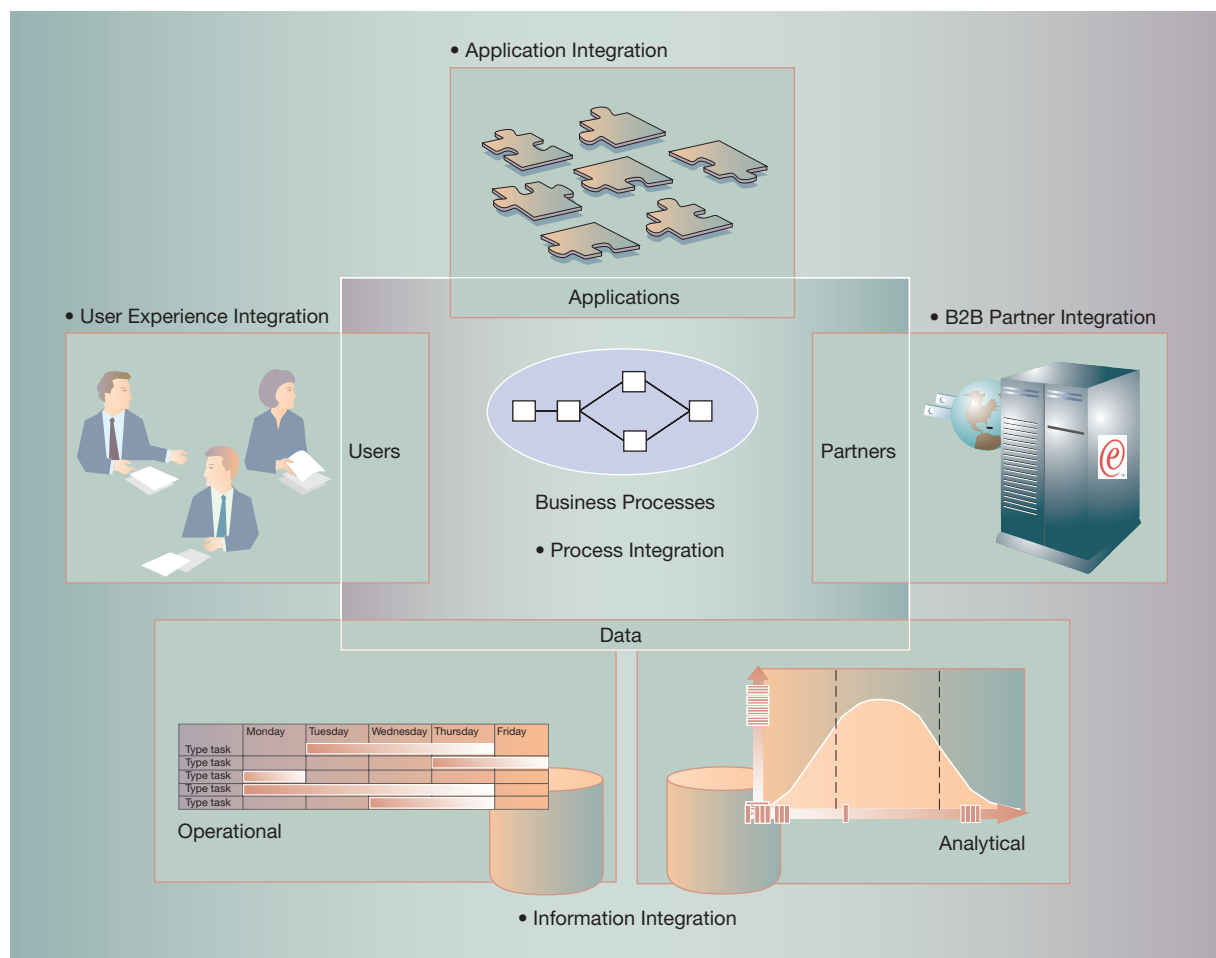
In this section, the challenges and capabilities of business integration are presented.

Business integration challenges. Business integration addresses the integration of information (sourced from applications and data sources) with people and processes; the processes are within an enterprise and between enterprises (see Figure 1). The following integration points characterize business integration challenges:

1. User experience integration—Provide information visibility and access to business functions

©Copyright 2004 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

Figure 1 Business integration challenges



(processes, applications, and information) based on users' roles.

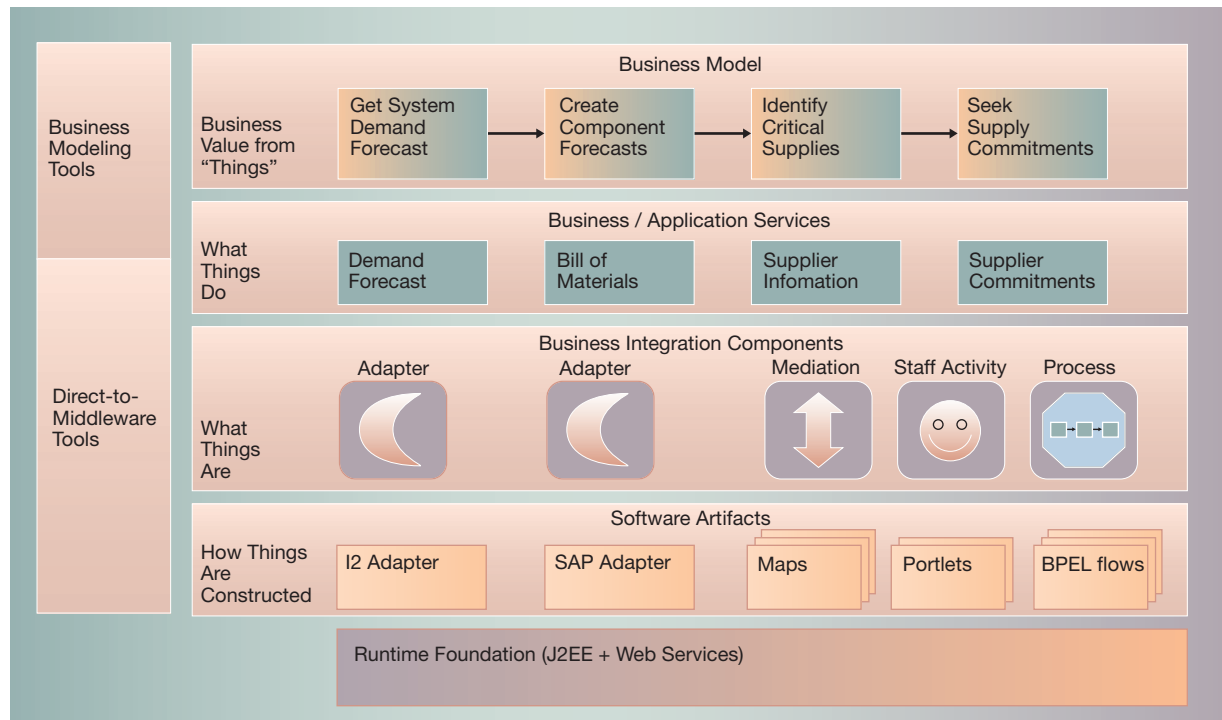
2. Information integration—Assemble operational and analytical data to provide an integrated view of business operations and to enable decision-making.
3. Application integration—Provide a mediated environment through integration of best-of-breed application packages offered by software vendors and customer assets.
4. Business-to-business partner integration—Enable business partners to collaborate and execute the business processes.
5. Process integration—Choreograph the interaction between users, applications, data sources, and

business partners using integrated business processes.

WebSphere Business Integration provides a set of capabilities for meeting the aforementioned business integration challenges.

Business integration capabilities. Business integration capabilities that characterize service-oriented architectures (SOA) for e-business solutions include the ability to integrate and manage business operation systems, enterprise information assets, business partners, and a collaborative network of decision makers to address specific business problems. WebSphere Business Integration embodies these capa-

Figure 2 Service-oriented business integration elements



bilities in an integrated framework of tools, platform, and content to offer rapid time to value for delivering integrated e-business solutions.

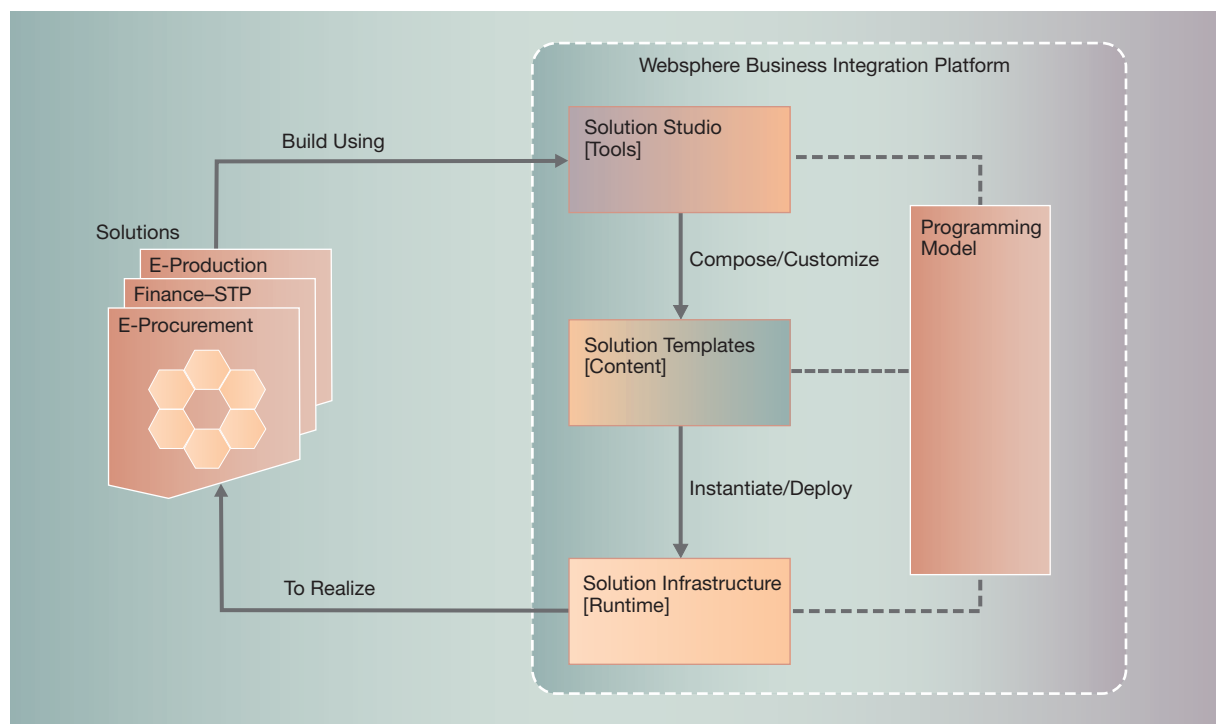
The WebSphere Business Integration capabilities enable the formulation of SOA for business integration solutions as illustrated in Figure 2.

Business models represent the choreography of one or more business or application services to address specific business problems. Such models help to express the business value from service-oriented solutions. Consider, for example, the supply-demand planning problem shown in Figure 2. The business model has a critical role in aligning the supply and demand within the supply chain and choreographs the business services, such as obtaining demand forecasts, creating component forecasts using the bill-of-materials, and collaborating with suppliers to obtain the supply commitment. The ability to model business operations in a technology-neutral manner is an important business integration capability, which is required to understand and transform business operations so as to realize business objectives (e.g., im-

prove business operations to realize shorter response times).

Business or application services capture the functional aspects of “things” that make up a solution. For example, the demand forecast business service in the supply-demand planning example obtains the demand forecast for specific products. The business operations model represents the choreography of business services that can be sourced or used either within the enterprise or outside the enterprise. The demand forecast business service in the above example is an internal service provided by a packaged application, whereas the enterprise suppliers provide the supply-commit service in the supply-demand planning model. A heterogeneous set of software components, representing the “things” in a solution, implement the business or application services. These components help to realize the business process, mediate the business-to-business interaction with suppliers, enable human interaction as warranted, and provide the necessary semantic and technical adaptation to enterprise information systems such as packaged or legacy applications and data

Figure 3 Websphere Business Integration architectural pattern



sources. The ability to integrate business or application services within an enterprise with external services offered by business partners or service providers is basic to business integration. The business integration components in turn are constructed using J2EE** (Java 2 Platform, Enterprise Edition) and Web services artifacts as shown in Figure 2.

WebSphere Business Integration also offers additional capabilities that are relevant for managing the business performance of service-oriented solutions. We mention these capabilities for completeness but do not discuss them in the paper.

WebSphere Business Integration platform

The parts of the WebSphere Business Integration platform are introduced in this section.

Architecture. Understanding customer requirements, formulating an approach to address these requirements, devising and designing an architecture for a solution, and integrating technology components to build the solution is the traditional integration cycle that customers and system integrators alike

practice. The solution experience under this traditional approach has been a time- and skill-intensive process of realizing a solution that involves a complex quilt of distributed and heterogeneous components. Further, the integration often leaves a customer with a highly customized but inflexible solution that is difficult to change and manage. For the system integrator, each solution experience tends to be a one-time effort that is neither replicable nor efficient in terms of asset reuse across solution instances.

Figure 3 illustrates the WebSphere Business Integration architectural pattern that brings together the principal architectural elements of the platform in realizing business integration solutions. The WebSphere Business Integration architecture and platform look anew at building business integration solutions to address the key value drivers of return-on-investment, time-to-market, and flexibility. WebSphere Business Integration addresses the return-on-investment value proposition by offering solution templates that prepackage domain-specific content (e.g., e-procurement solution templates). Solution

templates reduce the time-to-market by enabling the quick construction and customization of solution instances using the solution studio, an integrated suite of tools. Furthermore, these business integration solution instances execute on a consistent open-standards-based solution infrastructure that provides the necessary flexibility to adapt to change without extensive software programming.

The WebSphere Business Integration programming model serves to unify the tools, the content, and the runtime elements of the architecture; that is, the tools and the runtime are faithful to the programming model, and the content is expressed in terms of the programming model software artifacts. In the next subsection, we describe the WebSphere Business Integration programming model and the runtime.

Programming model. The WebSphere Business Integration programming model supports assembling business integration solutions from a set of integration components. The concept of an integration component provides a common abstraction over a variety of business function implementations. An application adapter or a Business Process Execution Language (BPEL) process choreography script, for example, can realize a component. Business-integration solution developers assemble a new solution from a palette of components rather than by writing any code. The programming model identifies a set of component kinds that handle specific aspects of a typical business integration solution (e.g., mediation, adapter) and provide the building blocks for pattern-based construction of those solutions. These component kinds in turn implement one or more business or application services. The following subsections describe these concepts in more detail.

Components and component assembly. The Component Declaration Language (CDL)² provides the model for representing and assembling the “application components” that make up a business integration application. A WebSphere Business Integration component is described using a *<component>* declaration, which identifies the *<type>* of the component and captures details about the specific *<implementation>* of that type, quality-of-service, and *<policies>* applicable to the component and references to other components that take care of administration and monitoring. The *<type>* refers to services provided by the component as well as services required from other components. Figure 4 illustrates the application of the concepts for a simple process integration example with three components, one rep-

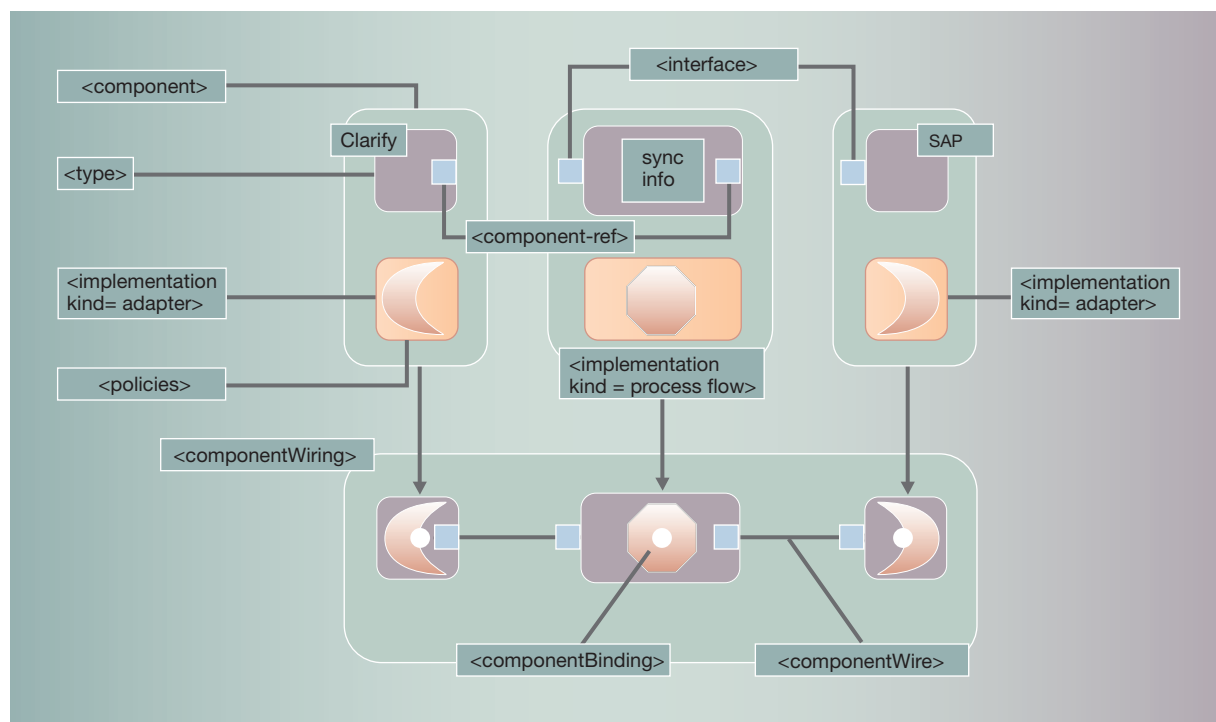
resenting an adapter to an SAP** system, another an adapter for a Clarify system, and the one in the middle a process flow that describes how to propagate events reporting updates in one system into the other one. On the type level, the Clarify component declares that it requires another component to receive the events it might produce, and the SAP component declares that it can provide operations to update the underlying application. The process component type offers an interface that can receive Clarify events and requires another component to provide SAP update requests. The implementation section for those types declares the implementation kind and captures the necessary meta-data for parameterization of the required adapters and the BPEL script that describes the process flow. Policy parameters may determine, for example, the transaction mode used for the end-to-end interaction.

The separation of type, implementation, and policy declarations facilitates a development process that completes the component specification over time in different phases of solution development. In top-down development scenarios, for example, users initially define “abstract” components by declaring services provided and required by the component type and can decide on the specific implementation later. The definition of policy parameters occurs either early in the development process or during deployment of a solution. Developers might also select predefined components with “opaque” implementation from a catalog of reusable assets and focus on the capabilities and constraints of the component expressed in its type and policy declaration only.

After they have all the components they need to implement a business process application, application assemblers “wire up” those components into an application. The wiring deployment descriptor establishes the identity of deployed component instances and matches services provided by components to services required by others (as defined in their respective type declarations).

Component wires should not be confused with process flows as used, for example, in BPEL processes; whereas process flows define sequencing rules for the coordination of components, component wires simply do the matchmaking between service-requesting and service-providing components; they describe potential interactions that could take place between the connected components, not rules for if and when they should happen.

Figure 4 Component declaration elements



Component kinds and integration patterns. The WebSphere Business Integration programming model identifies a set of component kinds that support common implementation patterns for business integration applications. Business-integration solution builders decompose the overall solution into a set of components that complement each other in executing the overall integration solution.

Process assemblers, for example, also use one or more process-kind components to script the overall process flow of the new business integration solution. Process flow components decompose complex requests into a sequence of processing steps that they delegate to other components. They are implemented using BPEL processes; their component type interface declares dependencies on services provided by partners identified in the underlying BPEL process.

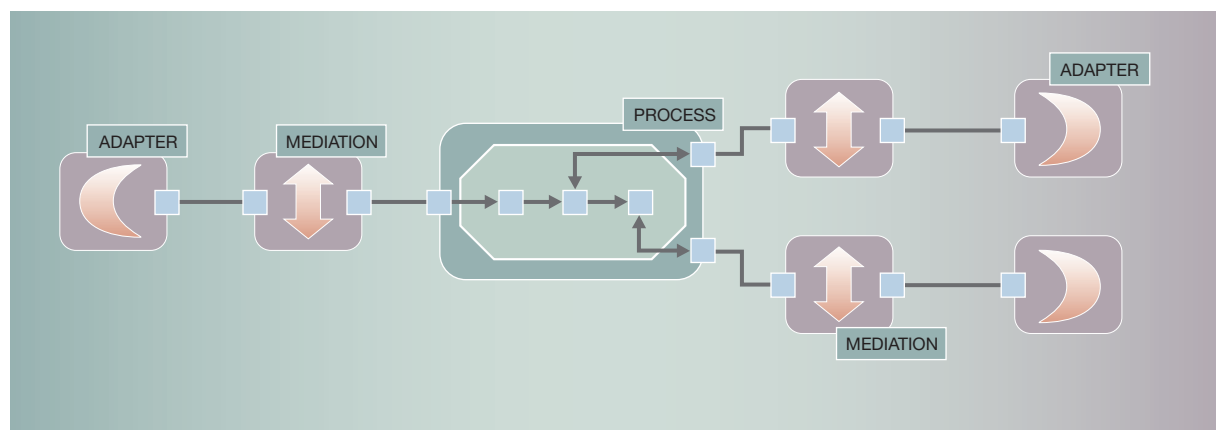
We also need one or more adapter-kind components that facilitate interaction with some external application component; this can be a legacy application accessed through an application adapter, or a service offered by an external partner that is accessed

through a Web services gateway or via some legacy business-to-business interaction protocol (EDI [electronic data interchange] or RosettaNet). Adapter components can propagate events produced by the underlying application onto the business process context, or they can propagate requests originating in the business process onto the relevant external components.

EJB[™]-kind (Enterprise JavaBeans[™]-kind) components can implement any new business logic within the J2EE programming model. Note that the adapter-kind component implements interactions with any non-J2EE legacy assets, such as the Customer Information Control System (CICS[®]) and PL/I applications.

In many cases, the process component delegates processing of its activities to adapter-kind or EJB-kind components. However, in some cases, human interaction is required to perform a process step, and a staff-activity-kind component facilitates delegation of such a task to a human “worker.” The staff-activity component uses work-list management ser-

Figure 5 Component kinds and patterns



vices to notify the responsible person and provide the required process context for the individual to perform his or her task.

Because business process composition is very much about integrating existing components that were not made to interact with each other, another essential component kind that process assemblers need is the mediation-kind component. Mediation-kind components facilitate the interaction between requesting and providing components. They implement a variety of interaction patterns, including publish-subscribe and multicasting event propagation as well as the transformation of in-flight requests or request routing based on request content and context parameters.

Rule-kind components support externalization of common business rules from the implementation of other components and thereby enable explicit management of those rules without affecting the realization of other components. Rule management operations include activation or deactivation of rules and modification of rule parameters. Rule-kind components can be used in combination with simple mediation-kind components, which act as rule set selectors, based on request content and context.

With this “starter kit” of component kinds, process assemblers can decompose their application into adapter, EJB, and staff-activity components whose interaction is eventually coordinated by process-kind components, with mediation-kind components interposed to enable their interaction, and rule-kind com-

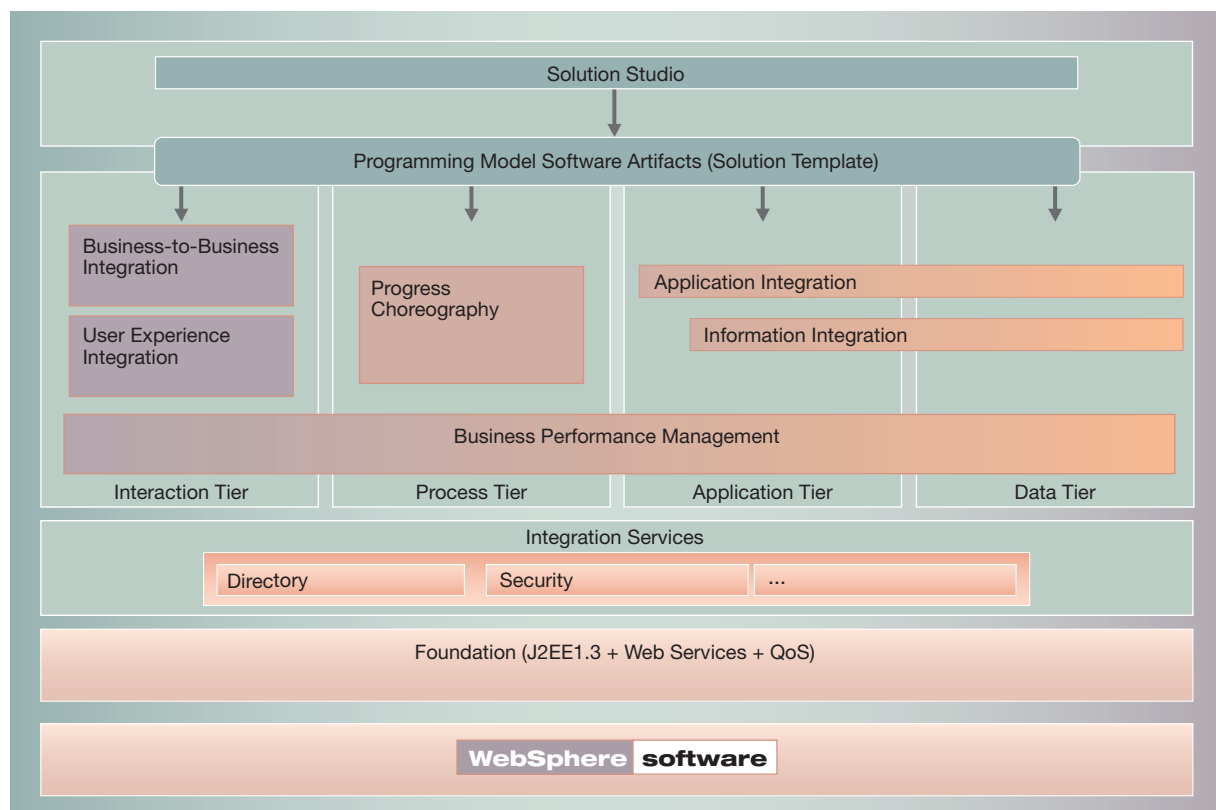
ponents maintaining business rules that govern process flow routing decisions (see Figure 5).

The component assembly model is useful in constructing solution templates through the specification of points of variability in the component assembly and prefabricated component implementations that are part of an assembly.

Runtime. The WebSphere Business Integration runtime consists of software infrastructure elements that host the programming model software artifacts in a business integration solution. Figure 6 shows the logical tiered architecture of the runtime elements. The user experience integration and the information integration aspects of the runtime shown in Figure 6 are capabilities provided by the portal and the data management portfolio of products respectively, which can be used in conjunction with WebSphere Business Integration.

The runtime elements include the user experience integration for rendering role-based executable content, the partner integration for business-to-business interactions with the business partners of a firm such as customers and suppliers, and process choreography for the choreography of applications in the context of the business process. They also include application integration for connectivity and mediation of application data, information integration for integrating operational and analytical data sources, and business performance management for monitoring the business operations, detecting solution exceptions, and launching appropriate recovery. The run-

Figure 6 WebSphere Business Integration run-time logical view



time also features a number of common integration services, such as the directory and security services, relevant to all the runtime elements.

A description of the runtime elements follows:

- **User experience integration**—The user experience integration runtime element enables users with specific roles to view monitoring dashboards, review work lists, and interact with processes to execute specific work items, to analyze and view business information, and to access one or more applications. Essentially this component renders the content based on who is accessing the content (role) and where they are in the business operations (context sensitive). The WebSphere Portal family of products provides this element.
- **Partner integration**—The partner integration runtime element facilitates interaction between businesses across enterprise boundaries. This runtime element supports a variety of transports (e.g., Hy-

perText Transfer Protocol [HTTP], MQSeries*), delivery interchange protocols (e.g., Simple Object Access Protocol [SOAP], EDI), and business protocols (e.g., SWIFT, RosettaNet). Partner integration serves as a business gateway in a business integration solution and uses the trading partner profile to determine the most appropriate business channel (i.e., business protocol and quality of service and protection) to use. The WebSphere Business Integration Connect product provides this runtime element.

- **Process choreography**—The process choreography runtime element is responsible for executing the business transactions defined by the processes and provides transaction management, including compensation for interactions with resources that are not transaction-compliant. Process choreography essentially knits together processes executing in a distributed infrastructure, such as workflow engines and applications that encapsulate process, to provide an end-to-end view of a business

process. The WebSphere Business Integration Foundation provides the process-choreography element.

- **Application integration**—The application integration runtime element executes the message flows that are required for brokering between applications and providing a mediated environment for business process integration. There are many different message models (publish-subscribe, request-reply, and so forth) and topologies (point-to-point, hub-and-spoke) that apply to the implementation of this runtime element. The WebSphere Business Integration Foundation, the Message Broker, and the adapters collectively realize application integration.
- **Information integration**—The information integration runtime element provides the persistence for business artifacts. It also provides the federation and replication of data and optimizes the placement of data (caching). Additionally, the information integration element provides access to structured and unstructured content (e.g., data in different media forms such as faxed documents). IBM data management technology realizes the information integration runtime.
- **Business performance management**—The business performance management runtime element transcends all tiers of a solution. All components in the solution generate events based on build time annotation of the parts of the solution (e.g., generate an event upon receipt of an advance ship notice from a supplier and then use this information to assess supplier flexibility). The business performance management runtime element can dynamically express interest in specific event types emitted by components in the solution. It also provides a mechanism to store the events, correlate events to detect business situations, populate dashboards based on the roles of users, and initiate human or automated actions corresponding to events. The WebSphere Business Integration modeler and monitor products realize the business performance management runtime.
- **Foundation**—The WebSphere Business Integration Foundation relies on the J2EE/Web services standards implementation and the quality-of-service capabilities provided by WebSphere Application Server. For more details on the Foundation, see Reference 3.

Building the solution

The WebSphere Business Integration approach to building solutions for business integration problems

is model-driven. There are three echelons of business integration models: business, information technology (IT), and deployment.

The business operation and observation model in the business echelon is a platform-independent representation of the operations of a business. This model represents the principal business activities and entities, the operating policies, the resources that are engaged in the business operations, and how a business measures its performance. The platform-independent model undergoes progressive refinement and transformation to a platform-specific model in the next two echelons.

The service-oriented programming model in the IT echelon is an assembly of business components and services generated as part of the transformation of the business model. Software artifacts that implement these components and services are packaged and distributed to a solution platform topology defined by the nonfunctional requirements and constraints of the solution environment.

The solution deployment model in the deployment echelon represents the packaging and distribution of software artifacts.

A number of tools operate on these models, at all three echelons of the solution life cycle, to support team and role-based solution development. The solution studio refers to the integrated collection of these tools. The solution studio integrates these tools through an open-source tool framework, namely, Eclipse.⁴ Eclipse features the Eclipse Modeling Framework (EMF), an implementation of the Meta Object Facility (MOF[®]) standards, which enables the integration of meta-models in the three echelons. The integration of the meta-models in turn permits the tools to exchange solution artifacts.

The next subsection provides an overview of the business integration models. It is followed by a description of the tools that constitute the solution studio.

Business integration models. The business echelon models take a line-of-business perspective that is of interest to business analysts and architects. A typical use of such models is in the business transformation stage where one develops an understanding of the business operations in terms of their to-be state. The business transformation analysis includes the simulation of the business operations models and iterative refinement of the models through interac-

tions with subject matter experts in the business operations. The IT echelon model is where the solution begins to take shape in terms of software components and services. This echelon is of interest to solution and IT architects who analyze the business echelon models to arrive at the manifest of software components for solution assembly. The deployment echelon model factors the environment in which the business integration solution has to execute and is of interest to systems architects and integrators.

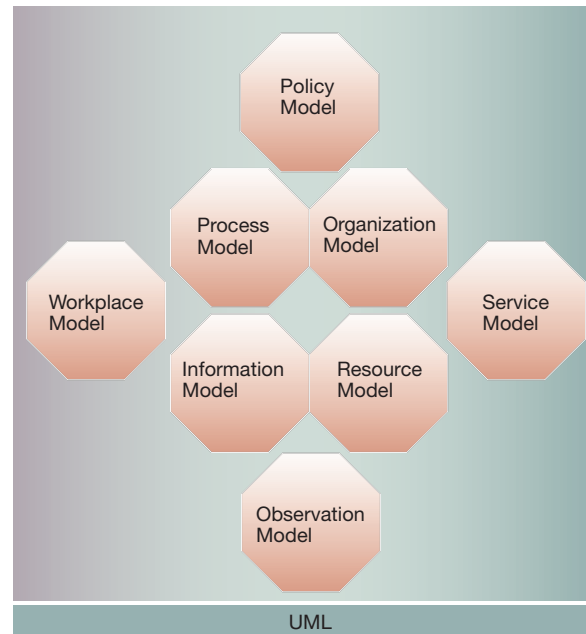
Business echelon: Business operation and observation models. The business operation and observation models are platform-independent characterizations of how a business operates and how to monitor and manage business performance. They are a collection of models based on Unified Modeling Language (UML**) standards (see Figure 7).

Process model: The process model provides a functional description of the business activities that constitute a business operation (e.g., e-procurement business operation includes processing of a request-for-quote (RFQ) as a business activity). The description includes the information that flows in and out of the business activity (e.g., RFQ business document) and the resources that are consumed in the execution of the business activity (e.g., facsimile machines). In addition, the model also identifies the business services that are engaged as part of the activity (e.g., credit check), and the business users with specific roles who participate in the business activity (e.g., all buyers in a particular geography).

Information model: The information model captures the business artifacts (e.g., RFQ business document) and the business events (e.g., acknowledgment of the receipt of a quote) in a business operation. The model supports an arbitrarily nested business information hierarchy. Business artifacts can be grouped into categories (e.g., planning documents) where a category can contain other categories (e.g., production planning) and many information types (e.g., master production plan). Additionally, the information model covers business events; the business event content specification follows the same approach as that of other business artifacts in the model.

Resource model: The resource model characterizes the resource requirements as well as the specific resources. Resource requirements are expressed in terms of the role that the resource will play in the business operation (e.g., buyer), quantity of the re-

Figure 7 Business operation and observation models



source that is required (e.g., one), and the relevant resource constraints (e.g., buyer specialized in purchasing computer supplies). A resource, in contrast, is described using a number of attributes, such as the resource identifier (e.g., John Doe), relevant cost attributes (e.g., labor cost per hour), and the roles that the resource can play (e.g., buyer, local purchasing lead). The resource model is used to model human resources, automated resources (e.g., accounting system), and consumable resources (e.g., lubricants).

Organization model: The organization model captures the structure of the organization involved in the business operation. The model represents organizational units, such as an enterprise (e.g., IBM Corporation), the divisions in an enterprise (e.g., Research Division), the departments within a division (e.g., Procurement), and the resources owned by the department (e.g., John Doe). Additionally, the organization model also represents the location of the various organization units (e.g., organization unit: IBM; location: Armonk, NY).

Service model: The service model represents the business services that are consumed (e.g., Digital Subscriber Line [DSL] provisioning service offered by a service provider that is used by a telephone-company-

fulfillment business operation) as well as those that are exposed by a business operation (e.g., order placement self-service). The service model characterizes both the service provider (e.g., provider type: DSL; provider interface: provisioning) as well as the service (e.g., service specification: place order; service implementation: fulfillment activity).

Workplace model: Workplace refers to the role and context-specific display that a human user employs to participate in a business operation. For example, a buyer role would interact with a procurement workplace to access work items in an RFQ process, applications such as the vendor catalog, and relevant business dashboards to monitor and act upon performance related to the RFQ process. The workplace model represents the human-facing tasks (e.g., approve RFQ task) and the views that constitute the workplace (e.g., list of RFQ and associated actions such as edit). It also includes the user-interaction state model that specifies the behavior of how a user can interact with a view and cycle through the views, and any applicable content management preferences (e.g., taxonomy of workplace content).

Policy model: A policy is an element of guidance for a business operation.⁵ It is a four-tuple: when is a policy applicable (precondition), where is it applicable (scope), what is the desired outcome (measurable intent), and what is the implication of meeting or violating the policy (business value)? Policies are broadly applicable in various aspects of a business operation, such as to represent decision or condition logic in a process model (e.g., “gold” customer orders can skip a credit check) and resource constraints in a resource model (e.g., buyers for certain commodities must have a certain minimum skill level).

Observation model: The observation model is concerned with the monitoring and management of business operations. This model specifies the correlation of business events that define business situations (e.g., an order for a gold customer has arrived, and there is insufficient inventory—these are two events that, when correlated, define a customer-satisfaction business situation). Furthermore, the model specifies the necessary actions associated with business situations (e.g., notify customer relationship manager) and the key performance indicators that are used to monitor business operations (e.g., fill rate for gold customers).

IT echelon: Service-oriented programming models.

The IT echelon represents the business integration solution as an assembly of components and services. The earlier subsection on the programming model provides details on the service-oriented programming model for business integration.

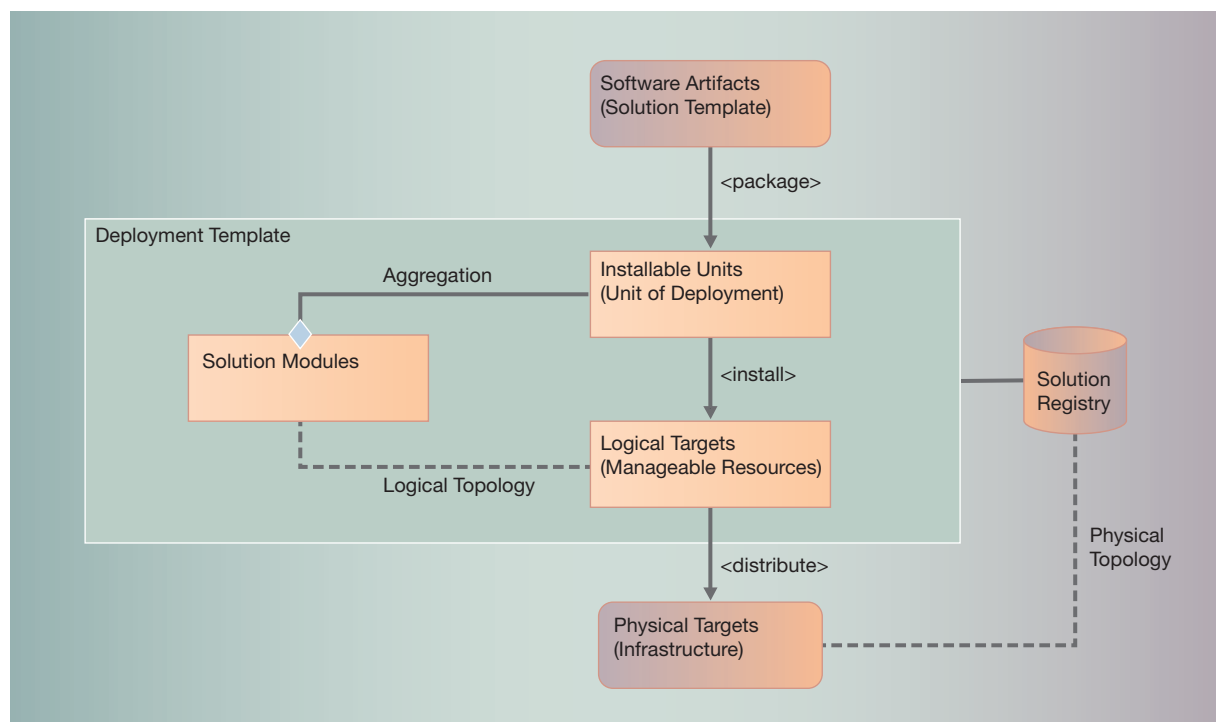
Solution and IT architects derive the IT echelon models from the business echelon models using architectural analysis and specific architectural styles. The transformation of business echelon models to IT echelon models is aided by e-business patterns, best practices, and the practical experience of solution and IT architects. Essentially, the business echelon models map to the component kinds and integration patterns described earlier. The BPEL and J2EE/Web services artifacts implement these component kinds.

Deploy echelon: Solution deployment models. The software artifacts from the IT echelon models are packaged, installed, and distributed in the deploy echelon. Figure 8 shows the fundamental elements of the deployment models.⁶

The software artifacts are packaged into installable units. An *installable unit* is a unit of deployment and consists of a descriptor and the packaged artifacts. For example, an RFQ process component in the e-procurement solution can be an installable unit. The installable unit is installed onto a *logical target*, which is typically one of the runtime elements described earlier (e.g., process choreography). The set of logical targets in a solution defines the logical topology for a solution. Solution modules group installable units (e.g., RFQ process component and the associated adapters and mediations that make up a solution module); a solution module is also an installable unit. The deployment template refers to a predefined set of solution modules, installable units, and logical topology with customizable points of variability. A solution template, which is a composition of components and services in the IT echelon, can be associated with one or more deployment templates. The deployment model, the instantiation of the deployment template, consisting of specific installable units and their association with logical targets, is registered in a solution registry. This facilitates the management of the solution over its life cycle (e.g., changes and fixes applied to a solution over its life cycle).

A logical target in the deployment template can map to one or more physical targets in the infrastructure

Figure 8 WebSphere Business Integration deployment model



(e.g., process choreography can be realized in application servers A and E). The distribution of solution modules to the physical targets is based on the mapping between the logical and the physical topologies. This mapping is part of the solution registry.

Solution studio. The business-integration solution studio provides a set of tools in support of defining the models described in the previous subsection and for transforming model elements between the modeling levels—from BPM-type (business process management type) business models to UML-style IT models and their translation into runtime artifact models that execute in a J2EE runtime.

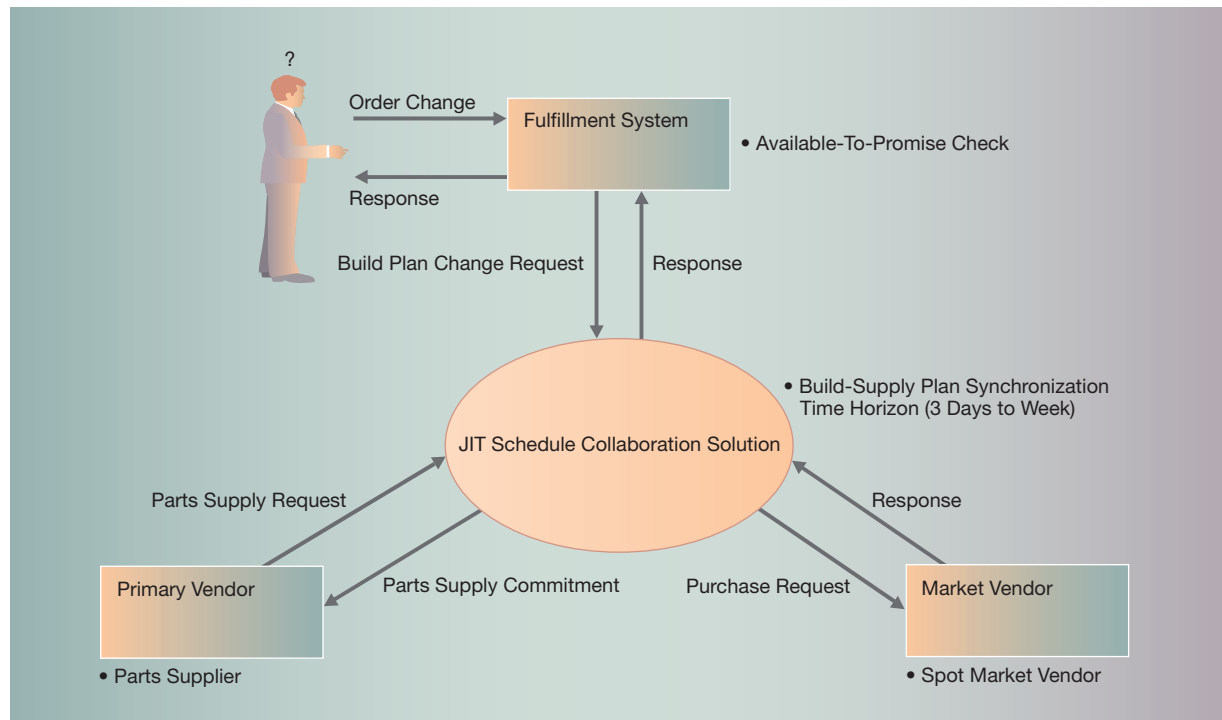
The tools stack supports a spectrum of user roles and skills, from business architects modeling business processes and information manipulated by those processes, to IT architects modeling application logic and underlying data, to application developers implementing J2EE components that complement existing assets, to application testers.

To support the different requirements and views on the business-integration solution development prob-

lem of those user roles, the solution studio provides a set of tools supporting specific user roles. All of the tools are on the Eclipse framework and share the same integrated development environment (IDE); “perspectives” with the appropriate set of tools address the needs of different user roles. (Note that Eclipse 2.x does not have explicit role support.) Eclipse supports scenarios in which each role is performed by a different individual (as might be the case in a large enterprise), as well as a situation in which one individual is responsible for the end-to-construction of a business integration application potentially using a subset of the tools only (as might be the case in a smaller enterprise).

The WebSphere Business Integration modeler covers the business analysis space. The modeler consists of editors for all of the business-level models described in the previous subsection. The business integration models also carry annotations such as the simulation parameters describing expected behavior of the elements of the business operation model. Execution of that operation model can then be simulated and analyzed.

Figure 9 Example of solution context



The IT architects can optionally transform the WebSphere Business Integration modeler business-operations-model artifacts into Rational Rose* (or XDE*, the Extended Development Environment) UML models to add more details to the model. The UML models would include templates for Use Case Diagrams representing a high-level view of business objectives and primary tasks performed by the business process, Activity Diagrams representing workflow processes, Class Diagrams representing the information manipulated by Rational UML tools with State Diagrams describing the life-cycle states of those information entities, and so forth. IT architects would refine those templates, adding details of process behavior and improving the information model beyond the initial, conceptual model provided by the analyst.

The translation of the business operation model to a runtime model can be either directly from the WebSphere Business Integration modeler or via the UML modeler path. In most cases, that model will be a starting point for further refinement using the direct-to-middleware tools in WebSphere Studio.

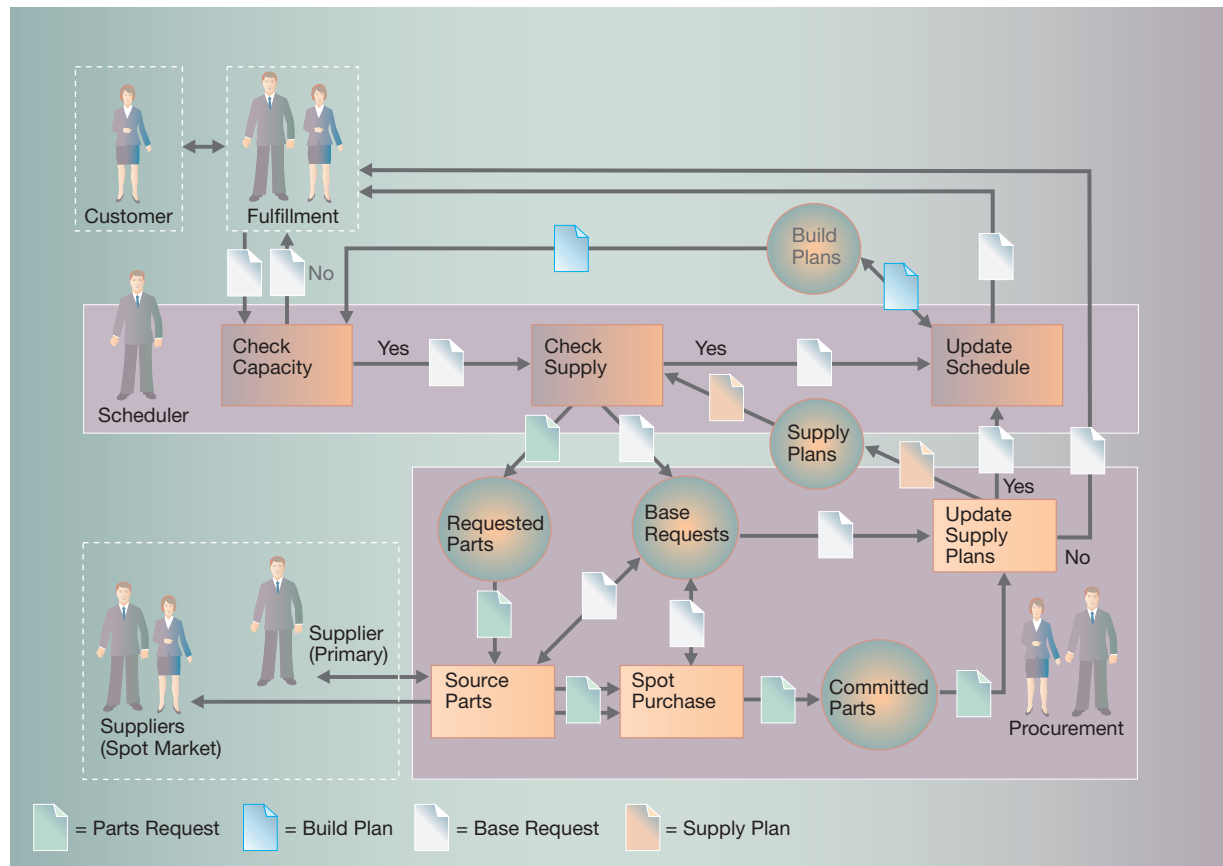
WebSphere Studio provides a set of editors that allows users to manipulate the programming model artifacts; for example, there is a graphical editor for BPEL process choreography models. In addition, WebSphere Studio supports debugging of the solution artifacts in a WebSphere sandbox environment; further details on WebSphere Studio can be found elsewhere.⁷ At the end of the solution development process, deployment tools pack the solution artifacts in a solution module; a detailed discussion of those tools is outside the scope of this paper.

Solution scenario

The solution scenario is an e-production supply chain problem.⁸ The following subsections describe the solution context and the associated business integration models.

Solution context. Consider an electronics value chain in which a customer contacts an electronics manufacturer to change an existing order (order quantity or specific items in the order). The manufacturer checks the current build plan and discovers an in-

Figure 10 Business operation model for JIT solution



Source: Business Transformation Solutions, IBM Research

ability to fulfill the customer order. This discovery triggers a just-in-time (JIT) scheduling collaboration involving the assessment of production capacity as well as interaction with suppliers to procure the materials necessary to meet the customer request. If the primary supplier cannot meet the request, then the policy is to purchase the materials in the spot market. Figure 9 shows the solution context, defining the functional scope of the solution and the external participants in the JIT scheduling collaboration problem. The electronics manufacturer is seeking a solution to this problem in order to increase responsiveness to its customers, make better use of production capacity, increase agility to identify alternative forms of supply, and dynamically evaluate a supplier's ability to meet demand.

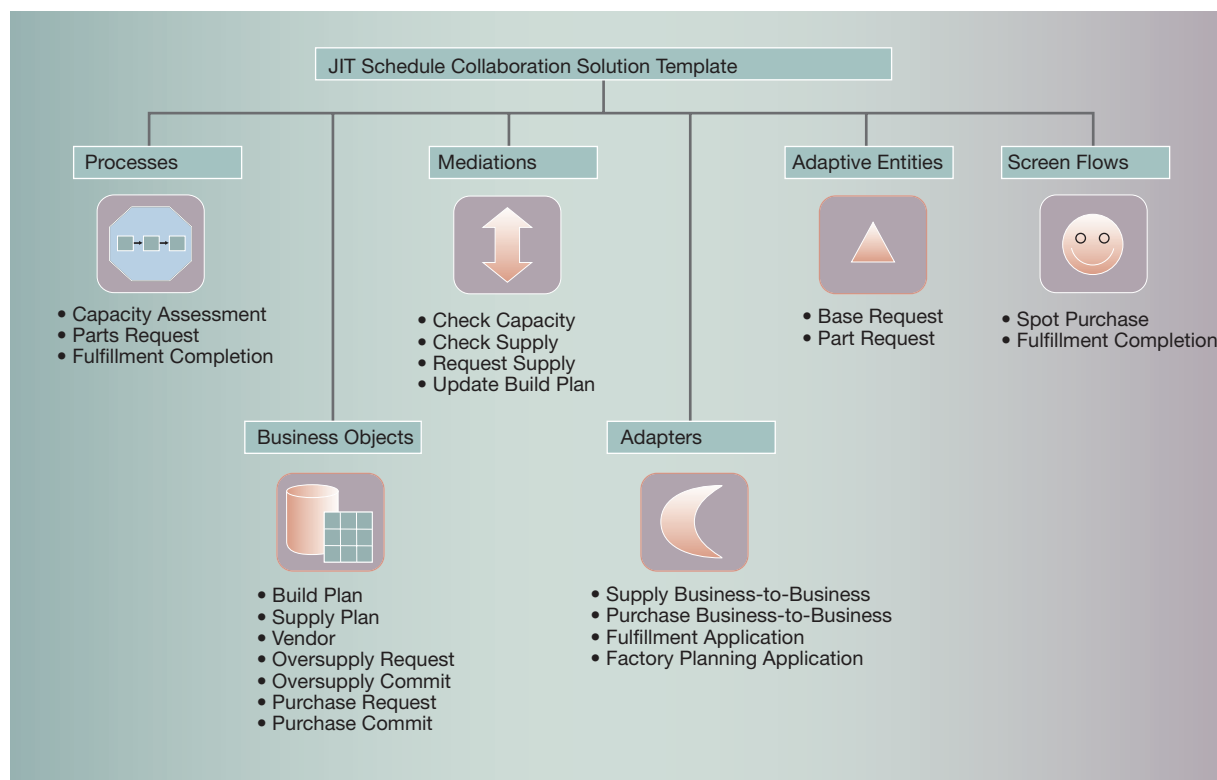
Business echelon modeling. The WebSphere Business Integration modeler models the business op-

erations for the JIT scheduling collaboration. (See Figure 10. It is an illustration and not the actual display from the modeling tool.) The business operation model identifies the business activities (rectangular symbol), the organization role players involved in the activities (person symbol), and the business documents (document symbol) that flow through the system, including where documents are staged and asynchronously used within activities (circular symbol).

IT echelon modeling. The analysis of the business operation models reveals the necessary business integration components and services. As mentioned before, this analysis typically involves a combination of human input from an experienced architect as well as automated discovery using e-business patterns.

Figure 11 shows the solution template for the JIT scenario. The CDL, introduced earlier in the subsection

Figure 11 JIT solution template



“Components and component assembly,” describes the business components constituting the solution template. The JIT solution involves three process component kinds, namely, capacity assessment, parts request, and fulfillment completion. At the IT echelon level, BPEL specifies each of these process components (see Reference 9 for more details on BPEL). In addition to the process components, the solution template also contains business artifacts such as the base request business entity. A UML state chart, part of the Rational tools as mentioned previously, specifies the life cycle of the base request business entity.

Deploy echelon modeling. The components and software artifacts that make up the JIT solution (as shown in Figure 11) are packaged into solution modules and then associated with a logical solution topology. Figure 12 shows the resulting deployment template for the JIT scenario. The deployment template is based on a logical topology that consists of a user interaction logical target (electronics workplace), a business-to-business gateway (electronics

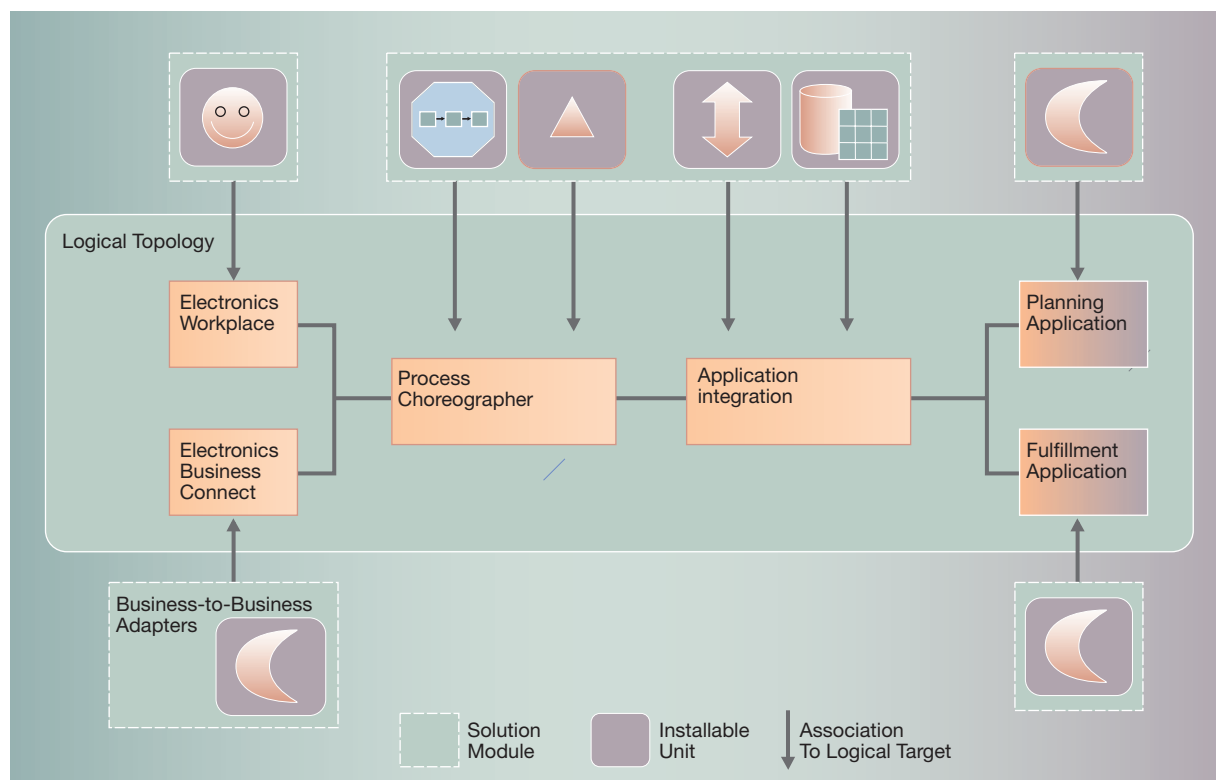
business connect), a process integrator (process choreographer), an enterprise application integrator (application integration), and two application environments (planning and fulfillment applications).

The solution modules are associated with these logical targets. The deployment template shown in Figure 12 consists of five solution modules: one solution module consisting of all the user interaction components; another comprising the process, adaptive entities, mediations, and business objects; and three solution modules consisting of the individual application and business-to-business adapters.

Summary

WebSphere Business Integration is a software platform for integrating internal enterprise systems and business partners and managing the business operations of the value chain. Service-oriented solutions to business integration problems can be composed by using WebSphere Business Integration to form

Figure 12 Deployment template for JIT scenario



an ensemble of integration components whose assembly is guided by integration patterns. WebSphere Business Integration will support the solution life cycle through the solution studio—an integrated suite of tools that operate on models at business, IT, and deployment echelons. The sharing of model information between tools across the echelons is possible through meta-data integration in Eclipse. The WebSphere Business Integration runtime relies on J2EE and Web services standards. The business integration solution templates offer a mechanism to assemble prefabricated components and services to quickly customize and generate solution instances. WebSphere Business Integration delivers rapid time to value to customers and system integrators through an integrated framework of tools, content, and runtime.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., SAP AG, Object Management Group.

Cited references and note

1. The capabilities and features described here include current and future aspects of the WebSphere Business Integration family.
2. M.-T. Schmidt, *WebSphere Business Integration Programming Model*, IBM Software Group Technical Report (March 2003).
3. E. Herness, R. High, and J. McGee, "WebSphere Application Server: A Foundation for On Demand Computing," *IBM Systems Journal* **43**, No. 2, 213–237 (2004, this issue).
4. Eclipse, www.eclipse.org.
5. *Organizing Business Plans: The Standard Model for Business Rule Motivation*, Version 1.0, The Business Rules Group (November 2000), www.BusinessRulesGroup.org.
6. C. Draper et al., *Solution Module Business Object Definition*, IBM Autonomic Group Technical Report (May 2003).
7. F. Budinsky, G. DeCandio, R. Earle, T. Francis, J. Jones, J. Li, M. Nally, C. Nelin, V. Popescu, S. Rich, A. Ryman, and T. Wilson, "WebSphere Studio Overview," *IBM Systems Journal* **43**, No. 2, 384–419 (2004, this issue).
8. F. Wu et al., *A Business Process Management Solution Storyboard*, White Paper, IBM Research (2003).
9. M. Kloppmann, D. König, F. Leymann, G. Pfau, and D. Roller, "Business Process Choreography in WebSphere—Combining the Power of BPEL and J2EE," *IBM Systems Journal* **43**, No. 2, 270–296 (2004, this issue).

Accepted for publication December 11, 2003.

Kumar Bhaskaran *IBM Software Group, Route 100, Somers, New York 10589 (bha@us.ibm.com).* Dr. Bhaskaran is the chief architect for business integration solutions in the Software Group. He has been working in the area of service-oriented technologies for e-business solutions and has been actively involved for the past fifteen years in the design and development of enterprise business solutions. He has a Ph.D. in industrial engineering and operations research from Rensselaer Polytechnic Institute.

Marc-Thomas Schmidt *IBM Software Group, Route 100, Somers, New York 10589 (mts@us.ibm.com).* Mr. Schmidt is an IBM Senior Technical Staff Member, was a lead architect for IBM's WebSphere Business Integrator, and is a member of the Software Group Architecture Board. He has been working in the area of business process management for many years, focusing on workflow management, message brokering, integration of message-based and object-oriented technologies, and more recently applications of BPM (business process management) technologies for implementing Web services. He is currently working on applications of meta-information management for the semantic integration of Web services.