Improve your business agility with enterprise SOA — a model-driven approach to flexible application development

by Robert Heidasch

Robert Heidasch
Enterprise Services Infrastructure,
SAP AG

Robert Heidasch holds an M.S. in Computer Science and a B.S. in Economics, Management, and Marketing from the Technical University of Opole in Poland. After receiving his degrees, he worked as an SAP R/3 consultant and project manager in the Financials and Logistics areas. He joined SAP in 1999, where he worked on SAP Project System and on the integration of SAP software with Microsoft products using RFC, BAPI, and Web services technology, and then in the SAP NetWeaver Foundation group. He currently works in the Enterprise Services Infrastructure (ESI) group. You may reach him at robert.heidasch@sap.com.

"Change is the only constant." So said the Greek philosopher Heraclitus in the fifth century B.C.E. That's certainly true in today's IT world — in fact, now more than ever. The process of developing software is changing, from using any number of obscure programming languages to modeling. Business processes are changing, from repetitive code operating in any number of monolithic programs to a single, reusable, distributed process. Even the need for quick and agile change is changing, and building agility into the business applications that support changing processes has become a challenging task.

Project managers, systems architects, and consultants alike are confronted with a disparity between a business process's description (e.g., specifications, documentation) and its actual implementation (i.e., a single business process running in a variety of business applications). All aspects of a business and its automated processes have become more dynamic in recent years. This evolution has driven the need for supporting technology platforms to adapt correspondingly.

To meet this need, SAP uses model-driven software techniques that make developing business process-oriented applications easier and faster. Enterprise service-oriented architecture (enterprise SOA) uses SAP's modeling approach to design processes and applications. Such models represent real, process-relevant entities in the implementation, make the entities more transparent, and allow you to close the gap between documentation and development. Enterprise SOA defines and presents these models using a different level of abstraction, which enables you to present specific aspects of the business applications and processes, such as a more dedicated view from a particular perspective, and to discuss various aspects of real, running business applications.

This article begins with an overview that explains how enterprise SOA affects the application development process, including the introduction to

the business process expert (BPX), whose purpose is to build a bridge between the business and IT specialists. Also discussed are both the concept and the key terms of modeling and model-driven development. This article introduces you to the modeling standards and tools supported in SAP NetWeaver 7.1 and shows you how to use them to build models of your business processes that you can use to create your business solutions.

The article then presents the different modeling layers used in enterprise SOA development (business process modeling, business process component modeling, and business object modeling) and the tools that you can use to create the models — for example, SAP NetWeaver and SAP Enterprise Modeling for IDS Scheer (formerly ARIS for SAP NetWeaver, which is how I refer to it throughout this article), SAP Enterprise Services Repository (ESR) in SAP NetWeaver, and SAP NetWeaver Process Integration (SAP NetWeaver PI) Integration Builder, which provides the business process execution language (BPEL) Process Editor. I also touch briefly on user interface (UI) modeling because it uses the business objects and enterprise services to compose a code-free UI. Finally, I talk about some of the keys to success with enterprise SOA development.

Now, let's look at an overview of enterprise SOA and how it changes the application development process.

Enterprise SOA architecture

The enterprise SOA solution offers model-driven applications that link business process definitions with a technical implementation on the familiar SAP NetWeaver platform, adding various technologies and tools to support Web services and model-driven development. Enterprise SOA is an architectural concept that describes the process of building business service-oriented applications, where the repository, business objects, enterprise services, process orchestration, and UIs for applications are distributed.

To support this architecture, enterprise SOA allows you to create business applications using a number of concepts, for example:

- Business process orientation, which focuses on the business process itself rather than the traditional functional enterprise structure (i.e., the structure of departments)
- Different levels of modeling that define business processes, business objects, services, and service consumers (e.g., UIs)
- Loose coupling of certain software components and applications that communicate using open standards, extending the flexibility of the business processes; for example, Web services using XML, Web Services Description Language (WSDL), Universal Description Discovery and Integration (UDDI), or Simple Object Access Protocol (SOAP)

From a technical perspective, the concept of enterprise SOA is based on model-based business process management (BPM; don't confuse this term with business process modeling), which the SAP NetWeaver platform implements and supports. SAP's business process-oriented solution enables the seamless combination of different process types with system-centric integration processes and human-centric composition processes:

- **System-centric integration processes:** The SAP NetWeaver PI solution technically delivers these processes and uses them, for example, for application-to-application (A2A) and business-to-business (B2B) integrations within SAP and third-party applications.
- **Human-centric composite processes:** SAP NetWeaver Composition Environment (SAP NetWeaver CE) technically supports these processes and uses them for human interactions and process collaborations.

Figure 1 shows you the SAP NetWeaver view of the business process.

From a development perspective, the modeling approach is completely different from traditional development approaches in which building new

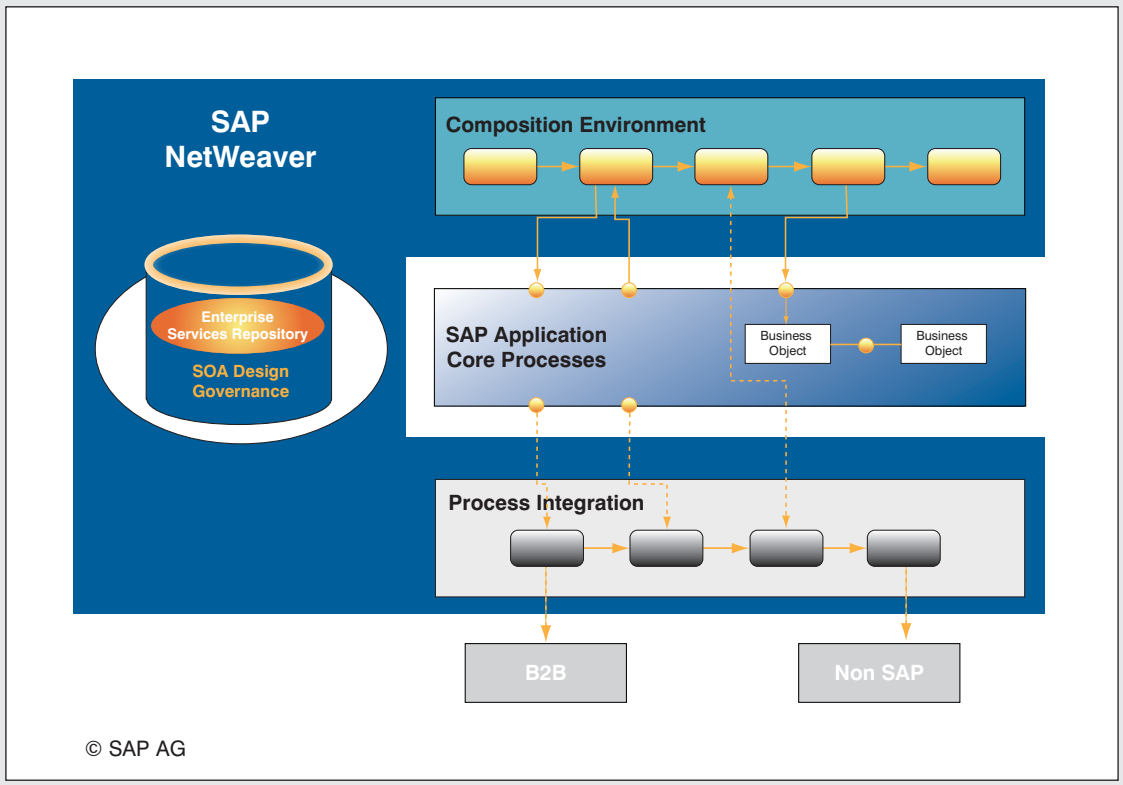


Figure 1 Supporting business process types with SAP NetWeaver

business applications requires a lot of written documentation (e.g., requirements, specifications, design specifications) to capture the business process description, software requirements, and system integration prerequisites. Models offer a clearer, more concise, and more accurate way to understand the overall process flow, as well as the particular steps, actions, and components used.

Enterprise SOA development phases

The modeling approach in enterprise SOA consists of the five phases shown in **Figure 2** (on the next page):

1. Business process analysis
2. Business process design
3. Implementation (business applications)
4. Testing, monitoring, and controlling (measurement and evaluation)
5. Optimization

Phase 1: Analysis

In this phase, the project leads and BPXs review the business issue, try to portray the business processes, assess their importance, and describe them. These tasks include analyzing the current environment, including existing and proposed business processes and their interactions with other business processes.

The analysis process should consider business strategy, governance documents, and other process information. Therefore, the analysis should first aim to understand your business and identify the particular business processes, their elements, and their competitive advantages.

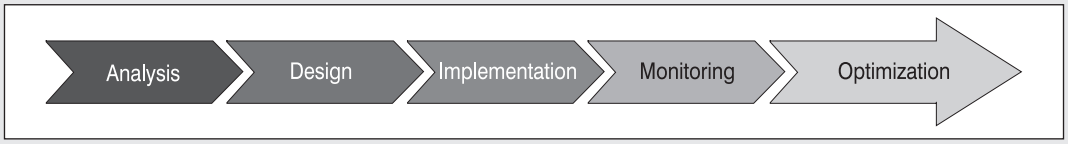


Figure 2 A phased development process for enterprise SOA

Analyzing the business processes in enterprise SOA enables the BPX to synchronize business strategy and business applications. It also allows the BPX to diagram the structure of the business architecture, bringing business executives, analysts, and technology specialists together with clearly defined responsibilities.

Phase 2: Design

In this phase, the BPX uses the results of the analysis phase to create a model of the business process. The tools should support the creation of an abstract business process representation: a business process model and its particular elements.

The BPX uses modeling language to transform the business process description into a business process model, also called a "business flow model." Then, the BPX defines the structure of the business components, analyzes different business integration scenarios, and tries to define the relationship between specific business components. Some modeling tools support this specification phase, such as the ARIS for SAP NetWeaver[1] tool, which is integrated into ESR. (ESR shipped in 2007 with SAP NetWeaver PI 7.1 and SAP NetWeaver CE 7.1. ESR is an evolution of SAP NetWeaver Exchange Infrastructure (SAP NetWeaver XI) Integration Builder.)

[1] For more information on ARIS for SAP NetWeaver, see the enterprise SOA page in SDN at www.sdn.sap.com/irj/sdn/nw-process-modeling → Enterprise Modeling → Enterprise Architecture → SAP Enterprise Modeling Applications by IDS Scheer. ARIS for SAP NetWeaver is available in SAP NetWeaver 7.0.

The BPX defines the required enterprise services and business objects for particular business processes, which are then used to handle the business-related information (messages). All of these modeling and creation activities are done in ESR, which is the central enterprise SOA designing repository.

Finally, the BPX uses the models to generate the particular interfaces, proxies, and implementation classes that the developer must manually implement in the business process implementation phase, which is described next.

Phase 3: Implementation

In this phase, the BPX and the developer transform the business process definition, its models, and its particular elements into an operating business application (ABAP-based or Java-based business application). For a successful implementation, you must accomplish the transformation without losing information; otherwise, the business process model won't reflect the real business process implementation, which is often a problem in a business process-oriented world.

To aid the business applications developer, enterprise SOA supports the generation of interfaces, proxy classes, and code skeletons from the enterprise SOA definition in ESR, allowing a new, more flexible way to implement business processes. This functionality is the essence of enterprise SOA. The particular enterprise service and business object implementation classes and functions are developed either in an ABAP-based system using ABAP Workbench or in a Java-based system using SAP NetWeaver Developer Studio (NWDS).

In parallel, the developer, using the business object and enterprise services definitions, can start creating the relevant UIs using SAP modeling tools such as SAP NetWeaver Visual Composer (referred to in this article simply as Visual Composer) and SAP Web Dynpro (Web Dynpro).

Phase 4: Monitoring

This phase is for testing the functionality and performance of the newly implemented business objects and enterprise services. This phase is where business process monitoring comes into play (go to SAP Help Portal at http://help.sap.com → SAP NetWeaver → SAP NetWeaver 7.0 (2004s) → SAP NetWeaver 7.0 Library → Administrator's Guide – Technical Operations Manual → Administration of SAP NetWeaver Systems → PI (Process Integration) → Monitoring), allowing you to measure the implemented business processes running in different business applications or components. Business process monitoring gives you component reliability and availability (from Monitoring, click on Component Monitoring in SAP Help Portal). This is important, because you can only complete the whole business process if all of its components are available and performing.

Requirements typically demand ongoing, proactive management of business processes to ensure complete control. SAP NetWeaver PI continually monitors the target business processes and provides prompt warnings if the current values (the results of process execution) deviate from your planned values (monitoring-relevant values that you can set). Continual monitoring closes the gap between corporate strategy and its operational implementation.

Phase 5: Optimization

This phase, though often overlooked and underestimated, is important from the design-time and runtime perspectives. In this final phase, the BPX, business expert, and IT expert should try to make the business process as efficient and effective as possible. This should be reflected in the optimization of particular models, for example, business process models that describe the business process flow and executable code (you may need to regenerate the interface and the proxies) from particular models.

Business process optimization also means improving resource use and avoiding side effects in process duration or quality. For this reason, business process optimization is not a single task you can do at the end of business process development; instead, it is a continual task that you should repeat each time something in the system changes.

Now that you understand the five phases of business process development, let's look at some of the benefits that accrue with enterprise SOA and find out more about the duties and capabilities of the BPX.

Some benefits of enterprise SOA-based modeling

One benefit of enterprise SOA-based modeling is that various tools can automatically generate some parts of an application (e.g., proxy classes, which can function as a substitute to control access to other objects, can be created from a WSDL description), making the implementation much easier. However, other parts of an application may require manual work to complete the implementation, such as developing the classes that realize a service. In any case, the use of a model-oriented solution helps to simplify the development of business processes and business process-oriented applications and, most importantly, make them more agile because they use the same data definition (e.g., the one provided in ESR) to generate the various application elements.

In enterprise SOA, SAP, like other BPM vendors, introduced the role of the BPX. The person assigned to this critical role closes the gaps between the business requirements and the underlying technology. In other words, the BPX bridges the domain gap between the business and technology specialists.

For example, the BPX validates the business processes with the end users and, using modeling tools, creates business process models and business

object models. In "traditional" development methodology, the business user validated the business requirements via mock UIs, test cases, and so on. The BPX also begins to compose new scenarios by selecting from the business objects already defined in the ESR and creating interfaces and model UIs using tools such as Visual Composer. Once the business process is designed, the developers assume the responsibility for writing the necessary code for individual pieces, such as the business object implementation class, to complete the implementation.

The modeling tools and decentralized development process of enterprise SOA dramatically alter future development. They allow a new group of users (e.g., business analysts and BPXs) to develop process-oriented business applications. These changes allow for better business process understanding and a smoother development process flow. (For more information on the role of the BPX, see the sidebar on the next page.)

Let's start our discussion of modeling with a short description of modeling terminology and then move deeper into the modeling approach by describing the particular models in enterprise SOA and their roles.

Modeling — a new development paradigm

First, let's define the terms model and modeling, which are the basis of business process-oriented development.

- A **model** is an abstraction of the real world, and a business process model is an abstract representation of a particular business process. A model hides certain characteristics that are irrelevant for a given purpose, while emphasizing others. Models can be more efficient and powerful than standard programming languages alone (e.g., ABAP, Java, C#, C++) because you use special languages (e.g., Domain Specific Languages [DSL]) for them. These languages provide a rich medium for describing different aspects of the "real" problem (e.g., different aspects of the business process: the business process model, business component model, etc.) and the relationships among these aspects, making them more efficient than source code.
- **Modeling**, from a business perspective, is the art of describing business objects, operational rules, and user roles and relating them to business processes using descriptive language. You can define business process modeling as the art of describing business processes using specific modeling languages or notations (e.g., BPMN or BPEL) to present activities that cause business-related actions.

Model-based development increases the effectiveness, efficiency, and agility of the software development process by creating artifacts (e.g., objects, function modules) that you can reuse when building new software applications. However, you can only achieve this reusability if you synchronize the model and the executable application code.

> ***Note!***
>
> A model is not actually executable code. Therefore, one of the most important development steps is generating code from models and "metamodels" (meaning "model metadata"). The code generation tool must recognize the architecture, the platform, and the development process required to generate meaningful code. Visual modeling tools support the generation of source code from models. Generating source code from graphical visualizations is particularly valuable, especially when they contain higher-level abstractions (e.g., business processes, entities, processing protocols). SAP tools that use this technology include Web Dynpro, Visual Composer, and SAP Guided Procedures (SAP GPs).

The BPX in model-based development

Many business projects fail because of insufficient communication between business and IT people. Business process knowledge used to be the domain of the business specialist; whereas, IT knowledge was the domain of technology professionals and IT specialists. Typically, business people defined the requirements and described the processes in terms that the technical staff misunderstood. What has changed for business process-oriented applications? What changes enable businesses to increase agility and profit from the new service-oriented solutions?

You need to define a new communication channel. In practice, you need people who understand both the business processes and the underlying technology; people who can define new business processes and adapt existing ones, build selected components using, for example, integrated modeling tools, and then deploy and run the new processes; people who possess the skills that are the domain of the business specialist and the IT specialist: a new development role — *business process expert*.

In enterprise SOA, the business process modeling tools that are integrated into both the development and application operating environments create the need for and empower the BPX. Model-based business process development requires understanding business process requirements and technologies well enough to define the business process model and implement the business process. An incomplete or "buggy" business process model can invalidate your implementation.

BPXs are individuals or groups of people who bridge the domain gap between IT and business professionals. They need to understand the language, behavior, and needs of both groups and play the mediator and translator roles. The BPX must grasp the requirements of the system landscape and the criteria for selecting software technologies. He or she should also understand the business-related issues, such as the organization's culture, strategy, and legacy, as well as the business needs of single departments and the organization.

In addition, the BPX has to have experience with existing solutions, strong communication skills, and modeling knowledge. This means, in practice, that the BPX needs the talents of a diplomat to be familiar with politics and the abilities of a teacher to reduce complex facts to simple ones using the appropriate translations, metaphors, and comparisons. These skills are fundamental to gain the respect and trust of the involved parties and complete the project.

The BPX is not the person to write the requirements or specification document for IT, but rather is more involved in defining business process solutions — actually building the operational model that is an abstract, electronic representation of the business process. This means that the predesign specifications may become obsolete because the model is more current or accurate in the long run.

Today's new business applications are based on business process models and, therefore, one of the most important skills for the BPX is modeling knowledge, both generic business modeling skills and SAP-specific tool skills. Common ways of acquiring this type of knowledge include learning about modeling languages (e.g., Business Process Modeling Notation [BPMN]) and their practical uses. The BPX role is comparable to the developers' role, using object-oriented programming with XML-based languages.

Modeling term	Definition
Business object	A basic building block that describes business-relevant entities (e.g., purchase order, customer, materials) and their operations (e.g., business object life-cycle methods)
Business process	The relationships and interactions among multiple business objects and business roles, constrained and directed by business rules
Business role	A group of activities that a business application can perform; assigned to a person (e.g., employee, sales manager, sales representative, inventory manager), allowing the execution of particular actions
Business rule	A statement that defines or constrains some operational aspect of the business; intended to assert the business's structure or to control or otherwise influence its behavior
Enterprise data	The data that the organization generates and operates from the business perspective; typically captured and stored in one or more application systems, such as SAP ERP
Metadata	Data used to describe business objects, roles, rules, and processes; makes the definitions independent of the underlying platform
Standards and methodology	The particular modeling technology (e.g., BPMN, BPEL)

Figure 3 Generic business process modeling terms

Figure 3 contains some of the generic modeling terms that SAP has adopted when talking about modeling in enterprise SOA.

Enterprise SOA uses a variety of different forms of modeling: from business object modeling, enterprise service modeling, and business process modeling, through business process orchestration modeling, to UI modeling. **Figure 4** presents the different business process-oriented modeling abstraction layers that enterprise SOA supports.

Enterprise SOA supports two methods of development: top-down (i.e., outside-in) and bottom-up (i.e., inside-out).

- **Top-down:** Using this approach to design your business processes and business objects means that you start from the highest-level design (the business process model) and then dive into the particular process components to define models and interactions. Next, you implement the required business objects or enterprise services in the particular development environment you're using (ABAP-based services in the ABAP Workbench and Java-based services in NWDS).

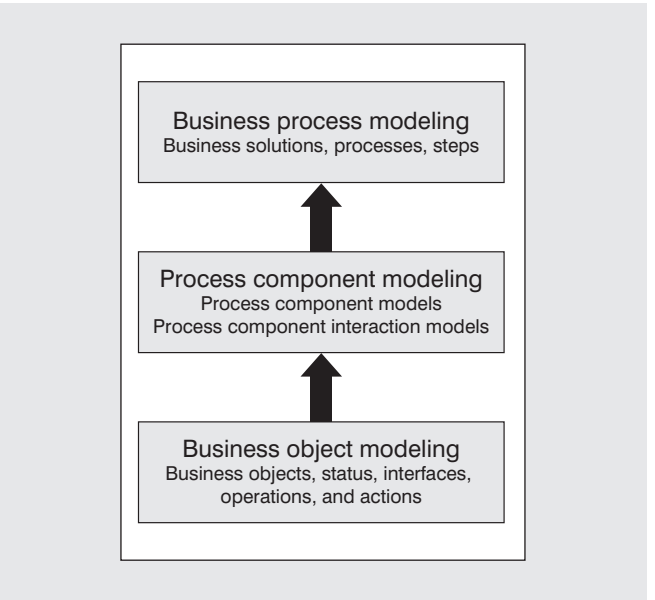


Figure 4 Understanding the abstraction layers of modeling business processes

Type	SAP tools	Intended audience
Business process modeling	ARIS for SAP NetWeaver	Business specialist, BPX
Process component modeling	ARIS for SAP NetWeaver	BPX
Business object modeling and enterprise service modeling	ESR	BPX

Figure 5 Defining the model types and SAP modeling tools

- **Bottom-up:** Employing this approach (also called the decomposition or breakdown of the business process) means starting with the business objects and operations (e.g., provided by the BAPI or remote function call [RFC] function) that you use to generate the enterprise services. First, you design the loosely coupled business objects, enterprise services, and components that have clear and intuitive interfaces. Then, you use them to compose new or use existing business processes to get the benefits of enterprise SOA at a relatively low cost and risk.

A dedicated modeling tool supports each of the modeling layers in enterprise SOA (see **Figure 5**).

In addition, enterprise SOA uses different forms of UI modeling. For example, you can design and create UIs using modeling tools — tools that use graphics-oriented notations to compose new UIs. Typical examples of such tools are Visual Composer, Web Dynpro, and SAP GP.[2] For example, using modeling information, such as business objects and enterprise services definitions from ESR, Visual Composer allows "code-free" UI development, which significantly accelerates and simplifies the process. From a technical perspective, Visual Composer generates the application code from the modeling information and the UI model; you can then use SAP NetWeaver Application Server (SAP NetWeaver AS) to deploy the code. Such tools allow you to design and create UIs without programming, expanding the group of potential UI developers; even business analysts using existing services or business objects that provide business-relevant data can develop a UI application, deploy it, and run it with SAP NetWeaver AS.

Now, let's take a closer look at the particular modeling layers that can help you successfully adopt particular business processes. I begin with a discussion of business object modeling and an exploration of process component modeling and business process modeling, and then briefly present UI modeling.

Business object modeling

For enterprise SOA, the basic element in the application architecture is the business object. It encapsulates semantics-related business logic, the required business data, and the appropriate methods to enable you to maintain and manipulate that data. The business object model describes the business object. Defined in ESR, this model has the following characteristics:

- **Business object attribute structure:** The structure of the data elements that are grouped into nodes (see the business object node structure in **Figure 6** on the next page)
- **Business object attribute type:** The business-related data type that defines the attributes of a particular business object
- **Business object service interface:** A service operation that defines the business object method (also known as an operation)

You use business-related attributes, which are standardized data types that define particular attributes, to describe the business object (e.g., sales order ID, customer ID, customer name). In other words, you use the data type as the formal description of the structure and representation of the data.

[2] You can find more information about SAP GPs on the SAP Help Portal at http://help.sap.com → SAP NetWeaver CE → SAP NetWeaver Composition Environment Library → Developer's Guide → Developing and Composing Applications → Designing Composite Processes with Guided Procedures.

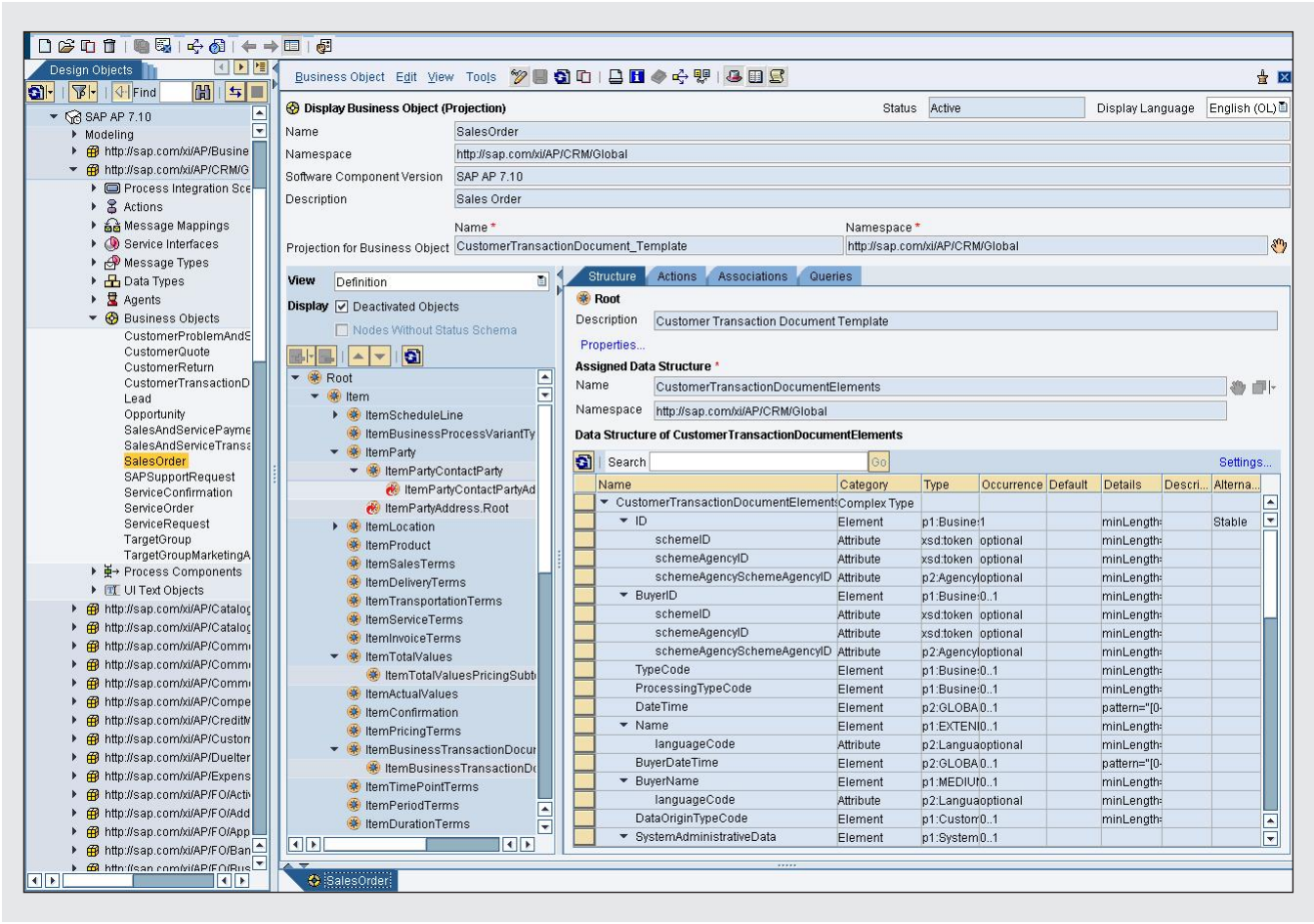


Figure 6 Displaying a sales order business object node structure

To homogenize the data types, SAP defines consolidated data types, called *global* data types (GDTs; for example, sales order ID) that represent the business data in widely used Web and business applications. Therefore, all business objects and business object service interfaces use the same pool of GDTs, significantly simplifying the harmonization. In other words, the same semantic description is used to define the same attributes in different business objects and business object service interfaces.

The business object service interface, described using WSDL, enables you to access the business object data, maintaining and manipulating it. Each business object defines operations — an abstract definition of the smallest callable functionality that uses the input and output parameters. Therefore, to model service interfaces and enterprise services, you use existing elements from the business object model that are described by metadata and data types.

> ***Note!***
>
> From a process integration perspective, you can access the business object's data through the business object service interfaces (also known as enterprise services) that define its methods.

Technically, a business object is a tree of business object nodes with one root node. This tree represents a set of semantics-related attributes. For example, the sales order business object consists of the sales order header data (root node) and the sales order item data (one or more sales order items). **Figure 6** is an example screen shot of the sales order business object node.

If you look at business object development with regard to the particular development phases, you can say that in the design phase the BPX performs these tasks:

- **Define data types:** Define the business-related data types that characterize the particular business object attributes
- **Define business object model:** Define business object nodes, their associations, and their service interfaces

After the development phases, which are all defined in ESR business object models, you can translate the business object service interfaces and data types into the specific programming language. The implementation phase (see the section "Phase 3: Implementation" on page 6) is where you use the business object model to create an environment-specific (development language-specific) representation of the business object. The business object model generates the code skeleton of a service provider class, and the business object service interface generates a proxy class that enables communications via this interface. The ESR browser and generator have been integrated into NWDS for the generation of Java classes and proxies and into ABAP Workbench for the generation of ABAP classes and proxies.

Developers browse the ESR content in particular development environments, generate the code skeletons, and implement the code skeletons of the service provider classes with additional program code, if required. The ABAP or Java system executes both generated and manually adapted extended classes and service provider classes without further ESR access.

> ***Note!***
>
> ESR is a design-time repository; it stores the business object and the enterprise services' definitions, but it does not contain the implementation (the executable code). The implementation uses ABAP Workbench for ABAP-based enterprise services or NWDS for Java-based enterprise services, and it stores the associated code separately from its definition in SAP NetWeaver AS for ABAP or SAP NetWeaver AS for Java, respectively.

Next, let's look at how you model the components of a business process and their interactions.

Business process component modeling

Each business object and enterprise service belongs to a business process component, which is the reusable building block of a business application that combines certain business functionalities (e.g., sales or purchase functionality). One process component can combine many business objects, which is ideal if they are related. This level of modeling is process component modeling, which shapes these components and their interactions. (SAP Business ByDesign and SAP Business Suite use different component granularity, so their component models may differ slightly.)

A model that reflects the interactions among process components is called the "process component model," and a model that reflects the interactions between two process components is called the "process component interaction model." For example, the process component model focuses on the view of the component and its internal relations. **Figure 7** (on the next page) presents the sales order process component model. In the process component interaction model, the particular components are presented as blocks (e.g., the sales order processing element) as in **Figure 8** (on page 15). The focus is the interaction between particular components.

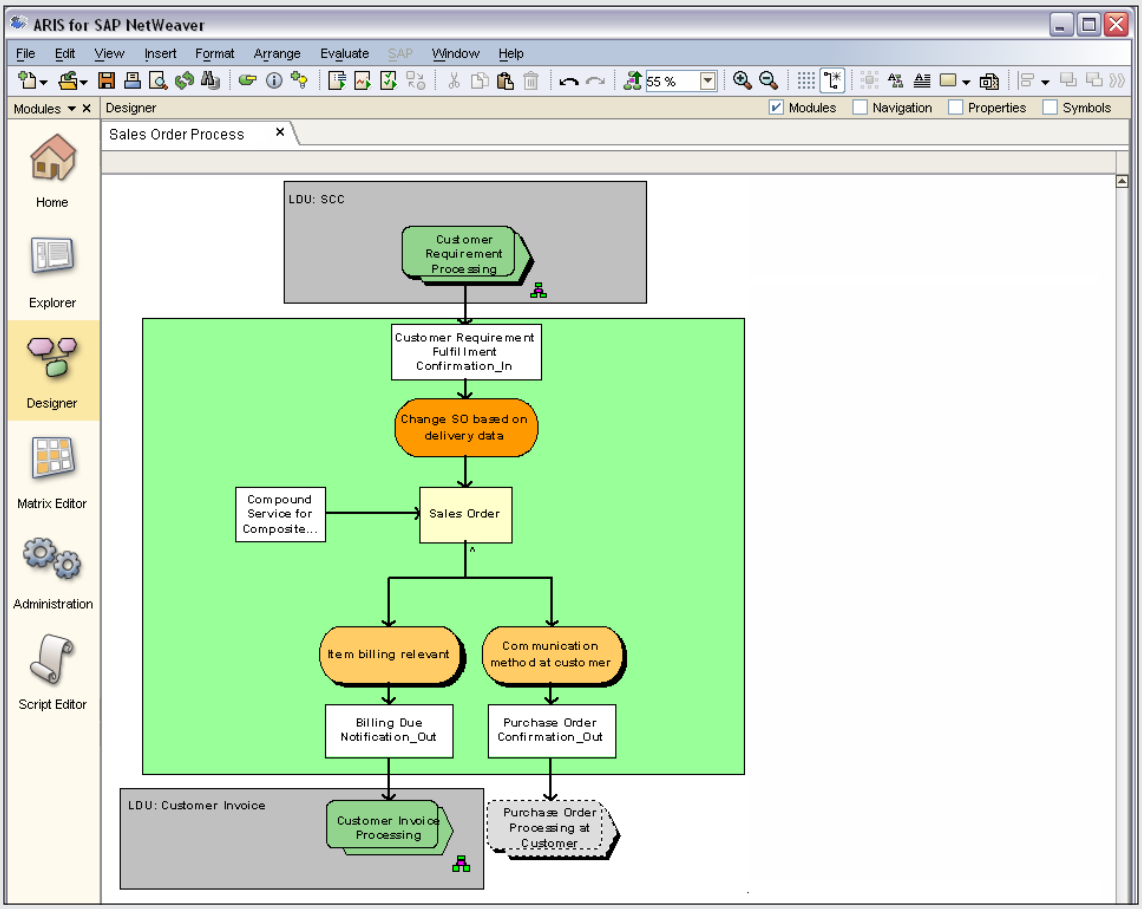


Figure 7 Modeling a sales order business process component

Both models describe the end-to-end business process composed of business process components that typically represent a specific functional area within the enterprise (e.g., the sales order in the sales area). The process component model and process component interaction model are abstract models that describe how a group of process components work together.

The process component model describes how you define and build enterprise services to perform the work of process components (for a sample business process component model in ARIS for SAP NetWeaver, see **Figure 7**).

The process component model focuses on defining a particular process component; it illustrates the business objects (e.g., sales order), process agents (i.e., input agent, Customer Requirement Fulfillment Confirmation_In; or output agent, Billing Due Notification_Out), and their related service interfaces and operations. Process component modeling allows better integration during the development process because a less formal mechanism — a model — integrates and communicates the requirements and specifications. A model is an abstract, programming language-independent solution that allows you to define components and the interactions among them without any development language concerns.

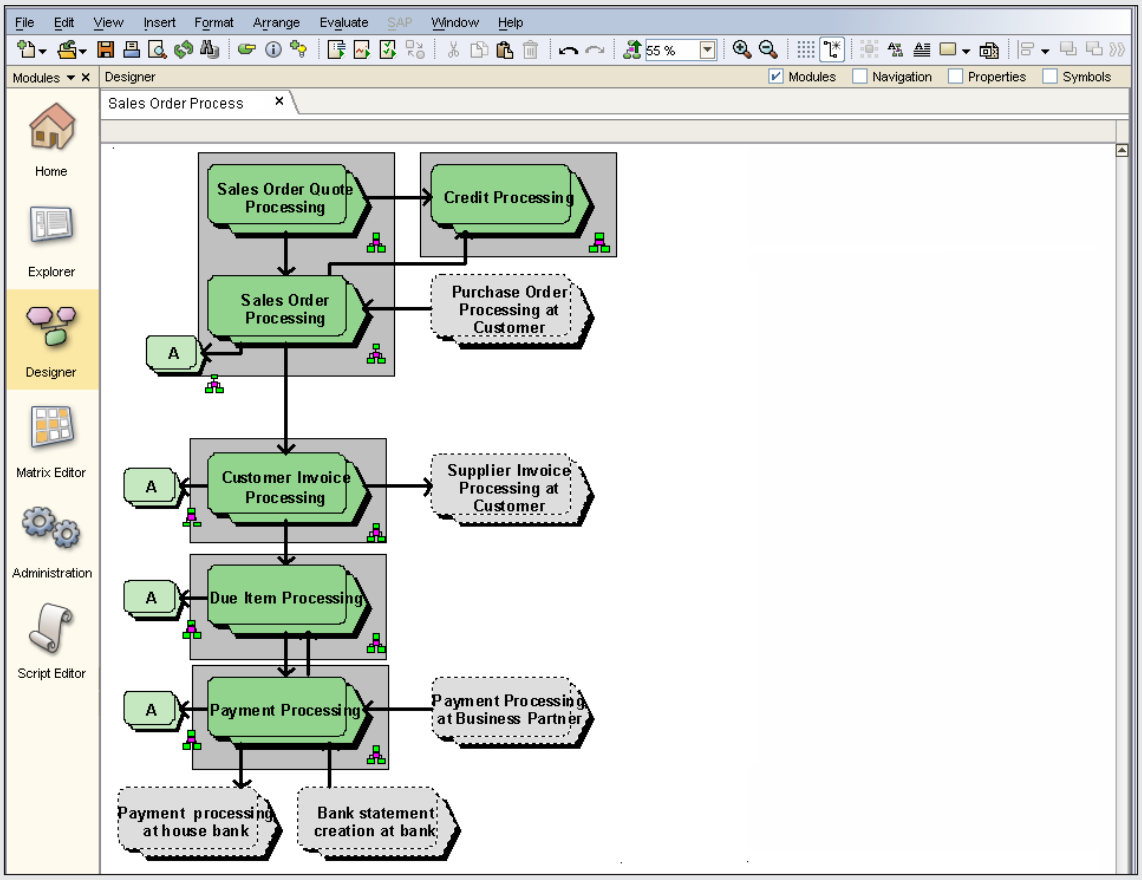


Figure 8 Modeling a business process component interaction

The process component interaction model describes the interactions between process components that communicate with each other; for example, **Figure 8** shows the process component interaction model in ARIS for SAP NetWeaver.

The process component integration model defines the interactions between two process components. In other words, this model describes the business objects that participate in the process, service interfaces, and operations, and the messages (e.g., Web services messages) that support a particular business process.

From a technical perspective, SAP NetWeaver PI, as the enterprise SOA middleware solution, is used to communicate between particular components.

Note!

SAP NetWeaver PI uses the business object, enterprise services, and data types' definition (modeled and provided in ESR) and supplies the tools (e.g., Integration Repository from Integration Builder) that provide a graphical process editor for defining integration processes. Integration Directory in the Integration Builder provides configuration wizards to help you adapt integration processes to your system landscape.

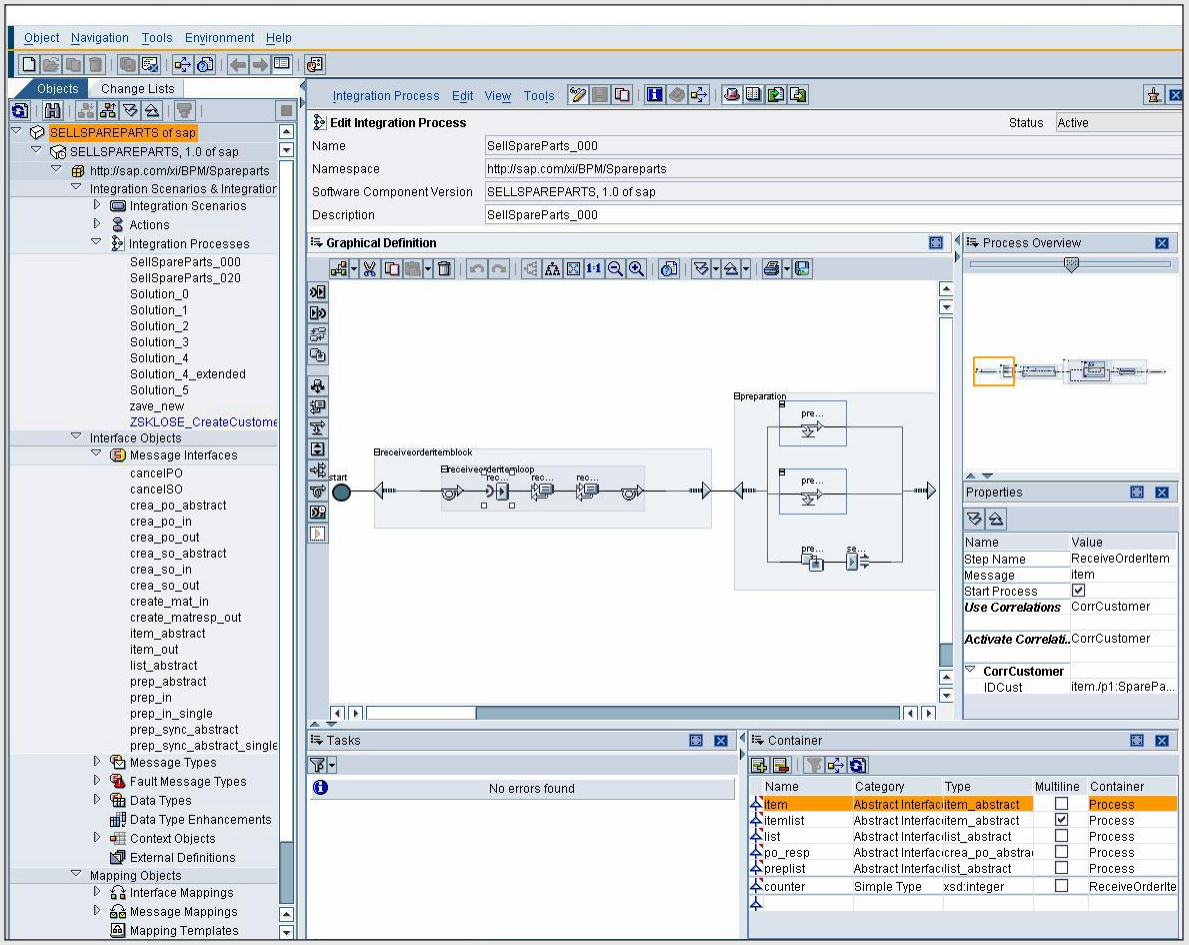


Figure 9 Using the BPEL Process Editor

Therefore, you use the process component integration model as a basis to model the integration scenarios and processes relevant to this business process. To model integration scenarios and processes, you use Integration Repository from Integration Builder, which is part of SAP NetWeaver PI.

ARIS for SAP NetWeaver also supports BPEL models (for more information on BPEL, see the sidebar on the next page). This functionality defines models that orchestrate Web services in ARIS for SAP NetWeaver and then import them into Integration Builder. After the importing, you can add more information to the model (e.g., to add metadata, to maintain exception-handling and deadline-monitoring; for more information, see the "Model a Business Process in ARIS for SAP NetWeaver and Import BPEL in Exchange Infrastructure" article on SDN at http://sdn.sap.com → SDN Community → SAP NetWeaver Capabilities → Business Process Management → Business Process Modeling → Enterprise Modeling → SAP Enterprise Modeling Applications by IDS Scheer).

In addition, Integration Builder provides the BPEL Process Editor shown in **Figure 9**, which allows you to compose and edit integration processes from the service perspective. The process editor provides the

Business Process Execution Language

The Business Process Execution Language for Web Services (BPEL4WS, or BPEL, for short) is a language created to describe the orchestration of Web services. It uses stateful synchronous or asynchronous messages that can be exchanged in a long-running interaction process. Synchronous messages are used when the client waits for an answer before performing the next actions; asynchronous messages are used when the client does not wait for the answer. Instead, the client performs the next steps without delay after sending an asynchronous message.

Web services are the building blocks for BPEL. In enterprise SOA, enterprise services play the role of Web services. Therefore, BPEL is good at executing and orchestrating services, but not at defining business-related human interactions. SAP works together with IBM on extended standards, for example, the BPEL4People standard (see the "WS-BPEL Extension for People — BPEL4People" white paper) to support future human interactions. Also, the interoperability standard, which introduces sub-processes and nested processes into the BPEL4WS standard — BPEL-SPE (see the "WS-BPEL Extension for Sub-processes — BPEL_SPE" white paper), ensures encapsulation and the reuse of business processes and its parts to support other business processes.

The primary benefit of BPEL is that it provides a high level of abstraction. Business analysts can compose executable business processes by pointing and clicking in modeling environments. BPEL does not enable the BPX to actually write programs. It provides an abstraction that allows the developer to implement flexible model-based applications more effectively. In other words, the BPX creates a model in a comfortable modeling environment, which can automatically generate a "skeleton" of executable code that the developer uses as a starting point in his or her own comfortable development environment to finish the implementation. In addition, BPEL extends the formality and transparency of your business process modeling, delivering visibility into particular business operations and rules. (For more information, see the paper "Using BPMN to Model a BPEL Process" by Stephen A. White on the BPMN Web site. For more information about BPEL4WS, see the IBM Web site.)

graphical editing area (see the Graphical Definition pane in the middle of the screen) that allows you to edit the integration process. The BPEL Process Editor also provides an overview of the whole integration process (see the Process Overview pane to the right) and the property editor (see the Properties pane just below the overview), which allows you to maintain the properties of a particular object or element selected in the Edit Integration Process area at the top.

To summarize, the process component models use the business object, the enterprise services, and the data types' definitions modeled and provided in ESR, which you use to build business solutions and business processes. For example, you might have a business process component model (e.g., sales order process) that incorporates all of the above-mentioned example elements (the sales order business object, particular services, and process components).

The process component models generate the process interaction models and build integration processes. By using the modeling solution, you can employ different levels of abstraction and different aspects or views of business process component modeling. These views are a kind of documentation that you can use to present particular components and their interactions, and to develop executable integration processes that run on the SAP NetWeaver PI platform.

Now, let's move to the top level of abstraction, business process modeling (also known as business flow modeling).

Business process modeling

In enterprise SOA, a business process model describes each business process. This model is created in the specification step ("Phase 2: Design" on page 6) using the documents resulting from the previously performed business analysis step ("Phase 1: Analysis" on page 5). Typically, the BPXs, who create the business process model with the business specialists, analyze the particular business process flow.

The main use of the business process model is to describe how and in which order to perform particular actions that reflect what happens in the real world. The business process model is a map that describes how the business process looks. In enterprise SOA, you define the business process model in ARIS for SAP NetWeaver using BPMN. (For more information on this notation, see the sidebar to the right.)

> ***Note!***
>
> You can use ARIS for SAP NetWeaver to import or synchronize predefined business process models as reference content from SAP Solution Manager.

Figure 10 on page 22 shows a simple example of a business process model created with ARIS for SAP NetWeaver.

After you have created the required models (e.g., business process models and business object models) and started to implement the business-related functionality, you may start to compose the required UIs in parallel, as a part of the implementation phase. I briefly describe which SAP tools support this development step.

Modeling the UI

An enterprise SOA solution offers significant improvement in the building and modeling of UIs. Enterprise SOA actually uses business objects and enterprise service definitions to build UIs. The new model-based solution offers you a specialized modeling tool — Web Dynpro or Visual Composer — to create a UI that in turn uses one or more enterprise services to obtain the required data.

Web Dynpro provides the infrastructure (called the Web Dynpro Foundation) that allows you to design and implement UIs and standardized UIs from reusable building blocks (called Web Dynpro patterns). Visual Composer uses reusable UI building blocks together with business object models to generate UIs. Web Dynpro follows the model-view-controller (MVC) pattern, which separates the model data (model), the visible screen output (view), and the control logic (controller) that processes the user input. (For more information about generating UIs in Visual Composer and Web Dynpro, go to SDN at www.sdn.sap.com → SDN Community → SAP NetWeaver Capabilities → Development and Composition → User Interface Composition.)

> ***Note!***
>
> The UI building block is an element that enables you to use standardized interfaces between UIs and business objects. With this approach, the UI accesses the business logic only via business objects and enterprise services, allowing you to decouple UI logic and business logic. So, it's possible to use any combination of UI building blocks, business objects, and enterprise services. UI designers can create many different UIs that present different data from the same business object; these UIs can then be used to provide business data for different groups of users.

Business Process Modeling Notation

The Business Process Modeling Notation (BPMN) is a language that both business and IT experts (e.g., business specialists, BPXs, developers, and business process administrators) can understand. In the past, business experts developed business process models with dedicated tools such as Microsoft Visio, which were separate from the process representation in the development platform required to implement and execute the process. The model transformation, often a difficult and error-prone process, was manual. This separation of design and implementation usually resulted in mismatched definitions as the process evolved.

The BPMN representation models the business process at the business modeling level. For example, the BPMN captures the process at the highest organizational level; details the process steps at the next level; and then itemizes the steps between applications at the next level, component and interaction models (see "Business process component modeling" on page 13). One of the main goals of BPMN is to close the gap between business process definition and its implementation. At design time, the business expert, using BPMN (or a visual design tool that supports this BPMN standard) creates the initial draft of the business process, and the developer, who could also be the BPX, implements it in a particular technology. At runtime, the business process administrator monitors its execution and eventually solves any problems that occurred during business process execution.

BPMN is a standard notation that attempts to resolve a variety of different modeling notations and viewpoints currently in use; for example, see following modeling notations: Unified Modeling Language (UML) Activity Diagrams, UML Enterprise Distributed Object Computing (EDOC) Business Processes, Integrated Computer-Aided Manufacturing (ICAM) Definition Languages (IDEF), Electronic Business eXtensible Markup Language-Business Process Specification Schema (ebXML-BPSS), Activity-Decision Flow Diagrams, RosettaNet, and Event-Process Chains (EPC).

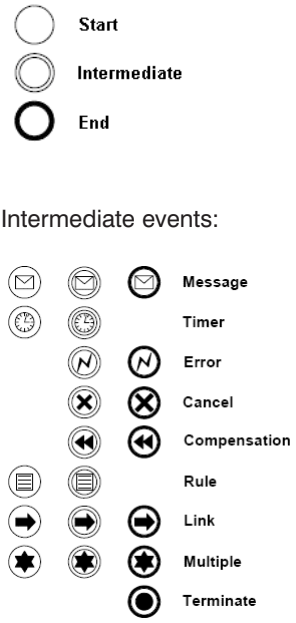
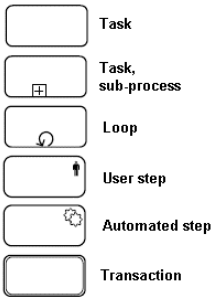
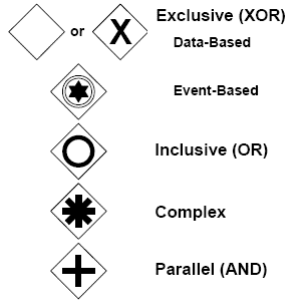
This notation defines a wide branch of modeling elements (called "Business Process Diagrams" [BPD] based on flowcharting techniques) and allows the creation of graphical modeling objects (types and graphical objects). The BPX uses graphical modeling objects to model particular business processes. Modeling objects are divided into the following four categories:

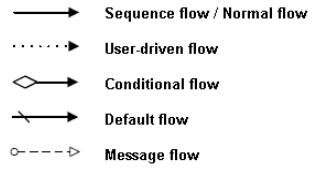
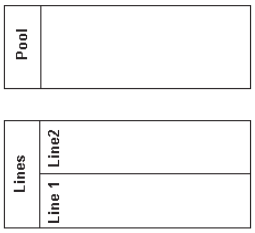
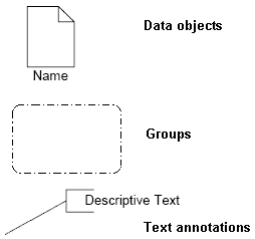
- **Flow objects:** Events, activities, and gateways that control divergence and convergence in the sequence flow
- **Connection objects:** Sequence flows, message flows, and associations that define the flow of messages among particular process participants
- **Swim lanes:** Pools and lines that represent a mechanism used to organize particular activities into separate visual categories (e.g., illustrate different functional capabilities or responsibilities)
- **Artifacts:** Data objects, groups, and annotations that provide additional information about the business process

The figure presents a short overview of the main BPMN modeling objects, their graphical representations, and short descriptions.

Continues on next page

Continued from previous page

BPMN modeling objects	Description
	Events indicate that something has happened that has an impact on the business process flow (e.g., a trigger). Events mark when the particular process starts (start event), ends (end event), or changes (something has happened that influences the process flow — an intermediate event; the intermediate event may occur between the start and end events). Start and intermediate events can be triggered in multiple ways (e.g., message, timer, error).
	A task is a business process-related activity; it is a generic term that describes every bit of work that a company performs. The BPMN defines different types of tasks (e.g., “normal” tasks, sub-processes, loops, user-driven tasks [user steps], automated steps, and transactions). The sub-process is a task that contains more detailed modeling elements (sub-processes) that describe a lower level of business process details.
	Gateways control the particular business process flow by allowing it to diverge and converge (e.g., branching, forking, and joining of the flow). Internal marks, such as exclusive (XOR), inclusive (OR), complex, and parallel (AND) marks, indicate the control type.

BPMN modeling objects	Description
	Sequence flow (also called “normal” flow) describes the normal sequence of the business process activities. Message flow presents how the messages flow (e.g., sent, received) between two business process participants (which are represented using pool and line modeling objects, see below). Association is used to associate additional information to particular modeling or flow objects.
	Pools represent business process participants. The pool combines the different business activities a specific business participant processes. Lines vertically or horizontally organize and categorize business activities.
	Data objects, groups, and annotations represent business process-related information that doesn’t affect the flow. They are typically used for additional documentation (e.g., text annotations) or better analysis (e.g., groups).

BPMN is also the modeling notation used for SAP’s new BPM development, which includes the business process modeling of human-centric and system-centric workflows. The new BPM is code-named “Galaxy” and will be available in the next release of SAP NetWeaver CE, version 7.11. (For more information about the SAP “Galaxy” solution, go to SDN at http://sdn.sap.com → Blogs → and search for the “Workflow in SAP NetWeaver” blog.)

In addition, BPMN allows you to visualize other XML-based languages designed for business process execution (e.g., BPEL). So, you can automatically map the BPD modeling object to BPEL. Mapping to BPEL provides a formal mechanism for converting business process models to the BPEL document, which enables you to formally specify business processes and interaction protocols.

You can convert the models created using BPMN into BPEL, the language that describes Web services orchestration (enterprise services based on Web services). This functionality integrates business processes into the SAP NetWeaver PI solution (see the section “Business process component modeling” on page 13; for more information about BPMN, see www.bpmn.org/.)

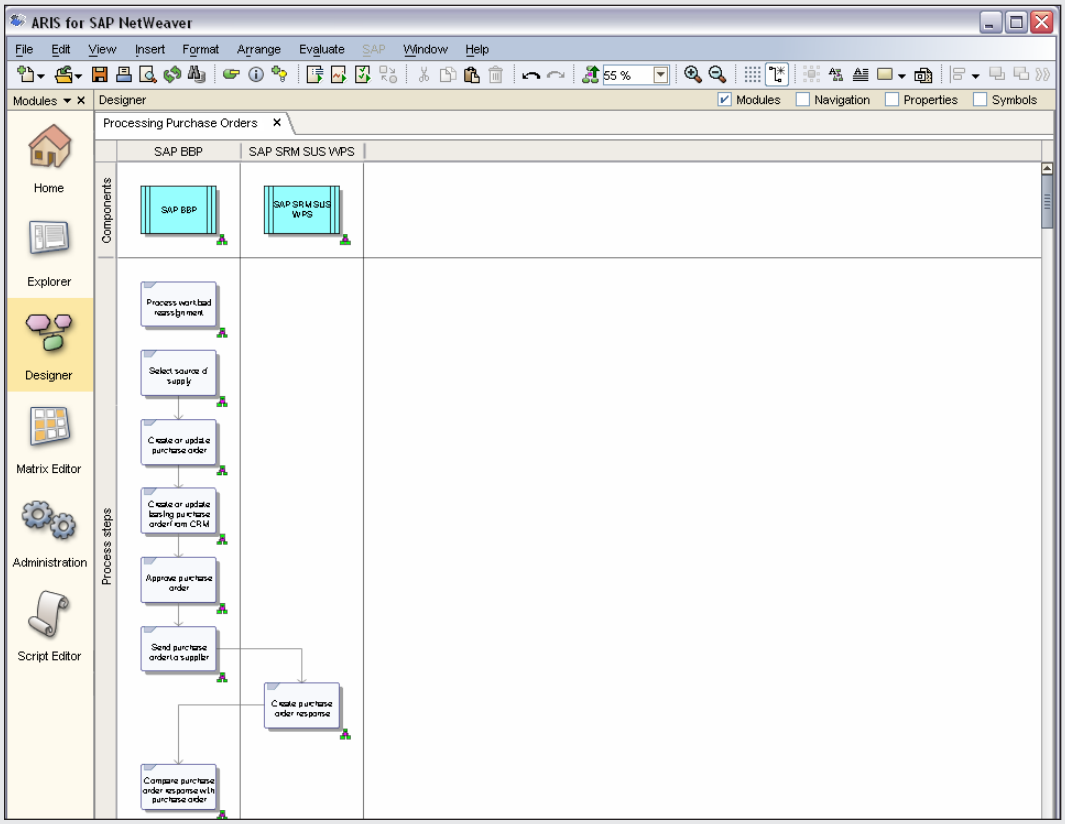


Figure 10 Modeling business processes

Visual Composer uses a pattern kit that enables you to model from UI building blocks, allowing you to drag and drop objects for the "code-free" building of UIs. Visual Composer automatically generates the executable code, which is then deployed in SAP NetWeaver AS and initiated using SAP NetWeaver Portal functionality.

SAP NetWeaver Portal provides the common infrastructure that is used to install and access UIs. Web Dynpro is the design and runtime environment of the UI building blocks. Visual Composer lets you model (without writing code) the particular UIs using standardized UI building blocks. In other words, the particular components (e.g., SAP NetWeaver Portal, Web Dynpro, and Visual Composer) form the technical foundation of the UI solution in enterprise SOA.

Note!

For more information about Visual Composer, see "Get started creating SAP Enterprise Portal iViews with Visual Composer" and "Advanced techniques for enhancing your SAP Enterprise Portal iViews with Visual Composer" (*SAP Professional Journal*, November/December 2005).

The use of a UI modeling tool significantly accelerates and simplifies UI development, and extends the potential pool of UI developers. Rapid prototyping and mock UI development allow you

to validate the requirements with the business user and emphasize the reuse of modular business objects and enterprise services. In addition, you can design and create a UI without any technical or programming knowledge; even business analysts can do it.

Advantages of model-based solutions

The enterprise SOA solution offers model-driven applications that link the business process definition with the technical implementation. Such models represent real process-relevant entities of the implementation and make them more transparent because you use different levels of abstraction to define and present the models. These abstraction layers (**Figure 4**), in turn, allow you to put forward specific aspects of the model (a better, more dedicated view that helps to provide a particular perspective) and to present different aspects of real, running business applications.

The model-based development or solution is a big advantage over "standard" development (design, specification, documentation in text, and manually entered program code). Some key elements that guarantee success in business-related modeling are the following:

- Using models as an abstraction, the business-related model can focus on a particular aspect of a business application, for example, a high-level business process view (i.e., business process model), a software component view (i.e., business process component model), and a business object view (i.e., business object model).
- Employ the model information to generate other models and executable code. For example, use the business object model to generate the service interfaces and the skeleton code for function modules.
- Use of the defined data types to describe business objects and service interfaces designed in ESR (reusing particular business-specific data types increases business object interoperability; they use the same metadata — data definition — to define the same business data in different models).
- The UI applications are built using modeling tools (e.g., Visual Composer), published by the UI developer, and consumed by the end user in a role-based portal — SAP NetWeaver Portal.

In addition, from the modeling perspective, enterprise SOA supports two methods of development: top-down and bottom-up. Therefore, in practice, the BPX may use either a top-down or a bottom-up approach. (For more information on these two methods, see **Figure 4** and **Figure 5** and their descriptions on pages 10 and 11, respectively.)

BPXs often use a top-down approach, starting business process modeling by defining high-level, abstract, business process model elements and then drilling down to the lower-level sub-processes and elements. Business people think in term of business processes; they understand the business processes and try to adapt and implement them in business applications. Often, designing and modeling business objects and their operations are done in parallel; this means the BPXs and developers define models, implement their services, and then assign them to particular business process steps.

As business process-oriented applications, the next generation of business applications uses the new modeling language. The BPX is using a new kind of language — modeling language — while the developer uses programming languages. The model

> ***Note!***
>
> This modeling solution has some problems. The separation between modeling and programming languages may result in an out-of-synch condition among documents, specifications, diagrams, models, and programming code. Therefore, the code-generation step from the model should be redone every time that the model definition changes.

designer (typically, the BPX) defines the business process using process flow models that follow standardized BPMN. Process flow models are used on the business level. The process flow model shows activities, flows, and gateways that control the flow. BPMN is supported by ARIS for SAP NetWeaver.

There is a lot more to be said about modeling and model-driven development in the SAP environment. The best place to go deeper into this subject is on SDN (http://sdn.sap.com) under the Business Process Management section (see SDN Community → SAP NetWeaver Capabilities → Business Process Management); you can find many interesting articles about SAP business process modeling solutions.

Conclusion

Enterprise SOA introduces a new dimension of business process-oriented and model-driven development. In this new paradigm, business people use a dedicated modeling language to describe business scenarios and business processes; then, they refine them using defined business objects, roles, and enterprise services. Business specialists and BPXs model all these elements, and developers and application specialists then implement them (the executable code is in part automatically generated). For example, the business objects that define particular business processes and their orchestration do not require interaction from technical experts.

This is possible because enterprise SOA is a model-driven solution that supports the entire modeling life cycle, from business process modeling to code generation. Using this functionality, business analysts may create business processes, design software components, and model particular business objects, enterprise services, and business-related data types.

In addition, different modeling tools may work together; this means that you can, for example, use ARIS for SAP NetWeaver to display or maintain (an import-and-synchronize functionality) predefined business process models as reference content in SAP Solution Manager. You can also use ARIS for SAP NetWeaver to create BPEL models and import them into SAP NetWeaver PI Integration Builder for Web services orchestration purposes.

Enterprise SOA is a flexible and effective solution that is used in new business applications (e.g., SAP Business ByDesign application, a new model-driven and business process-oriented application). This is accomplished with significantly greater agility and cost-efficiency than in the previous, monolithic, customizable business applications, such as the SAP R/3 system. Enterprise SOA allows a progressive transformation of existing SAP business applications (e.g., SAP Business Suite) toward a powerful service-oriented solution. I hope this article helps you understand enterprise SOA so you can transform your current SAP environment into a business process-oriented, model-driven solution in the near future.

