
Get ready for the next generation of SAP business applications based on the Enterprise Service-Oriented Architecture (Enterprise SOA)

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 for several years as an SAP R/3 consultant and project manager in the Financials and Logistics areas. Robert joined SAP in 1999, where he worked on the integration of SAP software with Microsoft products using RFC, BAPI, and Web services technology. He then spent several years in the SAP NetWeaver Foundation group. He currently works in the Enterprise Services Infrastructure (ESI) group.

Your SAP implementation probably looks like most corporate computing environments — in other words, you are running the traditional types of business applications that are information silos, monolithic, and often painstakingly integrated. But we all recognize that every aspect of the business and therefore its automated processes have become significantly more dynamic. This evolution has driven the need for the supporting technology platforms to be correspondingly adaptable. As a project manager, system architect, or consultant, you’ve probably heard that in SAP the Enterprise Service-Oriented Architecture (or Enterprise SOA, for short) is designed to help you meet this challenge. But are you convinced that Enterprise SOA is ready for use? Do you understand what it is and how to take advantage of it? Perhaps you are worried that you might need to replace much of what already exists in your SAP environment. (To allay your fears, the answer to that concern is a resounding “no”.)

To answer these questions and more, this article introduces you to Enterprise SOA in SAP and the next generation of business applications that will be based on it. Briefly, Enterprise SOA is a conceptual architecture for SAP business applications and their development that is based on the Web services standard.¹ At its foundation is the familiar SAP NetWeaver platform with various technologies and tools added to support Enterprise SOA and Web service development. Enterprise SOA was first supported in SAP NetWeaver ’04. However, in this article I discuss many elements that are only supported in the next version (SAP NetWeaver 7.1).

Our tour of Enterprise SOA begins with an overview of its architecture and capabilities. Next I present some tools — specifically, the Enterprise Services Repository in SAP NetWeaver and ARIS for SAP NetWeaver — that support the new-to-SAP concepts of designing and modeling business

¹ A good reference is SAP’s blueprint for putting SOA to work — *Enterprise SOA: Designing IT for Business Innovation* (O’Reilly Media Inc., 2006), by Dan Woods and Thomas Mattern.

Terminology watch!

Enterprise SOA is the new name for what was initially introduced in SAP as the Enterprise Services Architecture (ESA). Because of this progression, be aware that both names are often used in parallel.

processes and their component objects. I then discuss the new SAP development paradigm based on business process modeling, which allows you to integrate applications and their basic components (such as business objects and services) at design time. I conclude by introducing you to the Enterprise SOA corner of SDN (<http://sdn.sap.com>), which is a valuable resource to help you begin taking advantage of the power and flexibility of Enterprise SOA.

As the business changes, so must the technology

Both business trend watchers and technologists agree that the modern company has for some time been transforming into an agile and dynamic enterprise that offers a variety of different business services. Not surprisingly, this operational shift demands the ability to quickly adapt both the business processes and their associated business applications. Companies can no longer afford to wait for months until the information technology (IT) staff and the supporting applications can respond to changes in the business.

Obviously, the next generation of business applications must be capable of rapidly accommodating frequent operational and process changes. The currently favored solution is the concept of building what are called “loosely coupled” systems. These types of systems consist of applications and reusable components that can be easily integrated or reconfigured because they are based on commonly supported communication standards — Web services, for example. The notion of system independence is

an underlying premise of loose coupling, which is defined in Wikipedia as follows:

Loose coupling describes a resilient relationship between two or more systems or organizations with some kind of exchange relationship. Each end of the transaction makes its requirements explicit and makes few assumptions about the other end. Loosely coupled systems are considered useful when either the source or the destination computer systems are subject to frequent changes.

In response to increased demand, the software industry has undertaken a major evolution of the underlying standards, concepts, and technologies that are used to build loosely coupled systems. At SAP, the foundation of this evolution is Enterprise SOA. Its history in SAP reflects some common industry challenges that led to this evolution and the ensuing technology breakthroughs that made progress possible.

Problem #1: The rise of business reengineering

Back in the 1990s, various management consultants and business leaders turned their attention to the concept of reengineering business processes.² These visionaries prescribed an innovative approach that combined complete business process redesign with information technologies that support rapid change, offering the potential to revolutionize how companies operate. Unfortunately, at the time, the technology was not entirely ready to support this transformation. In practice, most existing business applications either did not support or only partially supported complete end-to-end business processes. Many enterprise resource planning (ERP) systems only supported some aspects of the business cycle — for example, the financial, material management, sales, and distribution processes.

Perhaps the most flexible ERP solution at the time was SAP R/3 (in those days, Release 2.2 or 3.1).

² The classic reference on this topic is *Reengineering the Corporation: A Manifesto for Business Revolution* (updated by HarperCollins Publishers Inc., 2003, originally published by Reed Business Information Inc., 1993), written by Michael Hammer and James Champy.

Although both complex and customizable, even SAP R/3 didn't address every aspect that was required to support entire business processes. For example, at that time SAP R/3 was missing the ability to support relationships between customers and suppliers. This functionality arrived later in the SAP Customer Relationship Management (CRM) and SAP Supply Chain Management (SCM) solutions, which are now part of the SAP Business Suite.

Yet all of these systems were rather inflexible and difficult to integrate, which proved to be the next challenge.

Problem #2: The difficulties of system integration

To simplify the discussion, let's focus on the integration challenges that faced a typical SAP customer. To close the gaps between processes and systems, many SAP customers developed their own custom SAP-based solutions or purchased non-SAP applications that they then tried to integrate with their SAP systems. In theory, introducing more applications addressed the business coverage problem. In practice, this addition introduced more technological obstacles to achieving the reengineering vision:

- Business applications from different vendors (that is, SAP and non-SAP) supported different communication technologies and used different data formats, making system integration a challenge.
- The applications were often running on different hardware platforms and operating systems, which further complicated system integration.
- The mainframe and client-server applications were monolithic, each using its own central database. Related data was spread across and often duplicated in different applications across the system landscape. For example, SAP material master data existed in the SAP ERP, SAP CRM, SAP SCM, and SAP Advanced Planner and Optimizer (APO) solutions. Therefore, data maintenance across different systems was often manual, time-consuming, and error-prone, which resulted in data consistency issues.
- Business processes often involved entities both inside and outside of the company, resulting in the need to integrate processes and therefore systems across companies. For example, a collaborative business process, such as developing new devices, requires communication between the different entities that are involved in design, engineering, and production. Using the SAP Product Lifecycle Management (PLM) solution to support this business process would require integrating the systems of each participating entity.
- Because SAP systems were integrated with many non-SAP systems, common standards for cross-application communication and integration were lacking. Consequently, system integration problems were typically solved in different ways using different technologies.

In short, the existing technology platforms lacked the robust application integration capabilities based on shared standards that were needed to fully realize the vision of business process reengineering and agility. But fortunately the computer industry is continually evolving and a solution was on the horizon.

The solution: Services and Web services technology

In the late 1990s and early 2000s, a technology breakthrough in the computer industry occurred that would alleviate the widespread system integration issues. Standards and technologies were established for building a new type of software component called a service that allows you to integrate different applications using a set of common standards. Basically, a service enables interoperable communication between distributed applications. The advent of Web services (which are simply services that use Web standards to communicate) extends this paradigm so that different applications from different vendors can communicate using industry-defined standards. (For more information on Web services, see the sidebar on the next page.)

Using services as a fundamental system design principle is the foundation of what the industry now refers to as a service-oriented architecture (SOA). The combination of the Internet, XML-based protocols,

What is a Web service?

A Web service is a piece of platform/system-independent, self-contained, self-describing application functionality that is designed and operates according to open Internet standards. According to a designated design and communication model, one application makes a Web service available for use (referred to as the service provider) and another application accesses and runs the provided service (referred to as the service consumer). Therefore, you can use Web services for communicating between different applications and software components that are written in different programming languages and running on different software or system platforms.

To overcome application and platform barriers, Web services are based on the following widely supported industry standards:

- Standard communication protocols — Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP). For more on the HTTP specification, see www.w3.org/Protocols/. For more on SMTP, see <http://tools.ietf.org/html/rfc2821>.
- Standard data representation format — eXtensible Markup Language (XML) and XML using Simple Object Access Protocol (SOAP). SOAP is a protocol for exchanging XML-based documents via a network using a standard transport protocol, typically HTTP. For more on the XML specification, see www.w3.org/XML. For more on SOAP, see www.w3.org/TR/soap.
- Standard service description language — Web Services Description Language (WSDL). WSDL is the language that is used to describe the public interface of a Web service. You use WSDL to define the service contract specification, which contains the interface description that is defined by the XML-based document. SOAP describes how to invoke a Web service that is described by the WSDL document. For more on the WSDL specification, see www.w3.org/TR/wsdl.
- Standard discovery language — Universal Description Discovery and Integration (UDDI). UDDI is a platform-independent protocol that enables applications and software components to look up (or “discover,” in Web services terminology) available services in an XML-based service registry. In other words, UDDI is an industry initiative to standardize the discovery of Web services. UDDI defines a SOAP-based application programming interface (API) for querying centralized Web service repositories. For more on the UDDI specification, see www.uddi.org/specification.html.

If you want to learn more about Web services, many useful sources are available. The World Wide Web Consortium (W3C) Web site (www.w3.org) is a good starting point. It provides a comprehensive reference to the industry standards for Web services and much more. In addition, two recent articles in this publication describe Web services in more detail and show you how to build them: Arthur Wirthensohn’s article, “Extend the internal and external reach of your applications with ABAP-based Web services” (July/August 2005), and Dr. Willi Nüßer’s article, “Web services or RFCs — choosing the right technology for your SAP integration challenges” (May/June 2006).

and a flexible software architecture based on platform-independent services has begun to revolutionize the nature of business process management (BPM) applications. Before seeing how this transformation is taking place in SAP with Enterprise SOA, let's take a closer look at the key advantages of an SOA environment.

Advantages of a service-oriented architecture (SOA)

In any SOA-based environment, you deliver loosely coupled and interoperable application services based on Web services standards that operate independently of the underlying platform and programming language. The individual software components³ may be implemented in different programming languages. For example, one software component might provide a service that is implemented in Java and is accessed by another software component or service that is implemented in ABAP or C#. Industry standards (for example, WSDL for Web services) define the way in which the service is accessed. In this manner, each software component or service encapsulates and therefore hides the vendor-specific, language-specific implementation from the calling software component or service.

Another big advantage is the strict separation of the service implementation and its publicly available interface. The technical implementation of the service is irrelevant to the service consumer, which only needs to know about a stable interface that:

- Defines a communication endpoint, which the service provider publishes as a service description in a service registry
- Represents a contract between the service provider and service consumer, as defined by industry standards
- Supports a particular request format, as defined by industry standards

³ A software component is an encapsulated piece of functionality that can be used multiple times by other software components or services.

Therefore, instead of building a monolithic main-frame or client-server application, you implement a series of discrete services that can be reused in the future when you need to solve a new problem. Obviously this approach also provides the much-needed support for accommodating rapid changes to business processes and their supporting applications. Although SOA does not define the specific technology to be used to implement a service, the similarities between SOA and the Web services standard are obvious. In fact, many system architects generally identify SOA with Web services technology because of its wide acceptance.

SOA in SAP — an overview of Enterprise SOA

As you have seen, recent corporate trends mean that the next generation of BPM applications must be able to accommodate process agility and rapid change. SOA and Web services technology are widely accepted industry standards that have the potential to deliver on this vision today. To help you take advantage of this promise while maintaining your investment in your existing SAP environment, SAP introduced Enterprise SOA, which is based on these standards, as an open architecture for building adaptive business solutions.

Let's take a closer look at the Enterprise SOA infrastructure (see **Figure 1** on the next page). Starting with the existing applications, Enterprise SOA allows you to integrate the functionality of both non-SAP and SAP systems into service-based solutions. These systems can be running on the SAP NetWeaver platform, as represented by the Partner and SAP application boxes. They can also be running on other platforms, as represented by the Legacy Third Party and SAP R/3 system boxes. SAP NetWeaver is the Business Process Platform, which is the foundation in SAP for building and running business applications (both traditional and service-based). SAP NetWeaver also provides the Enterprise Services Repository, which stores the basic elements that are used to build applications and services. I talk more about these elements later in the article.

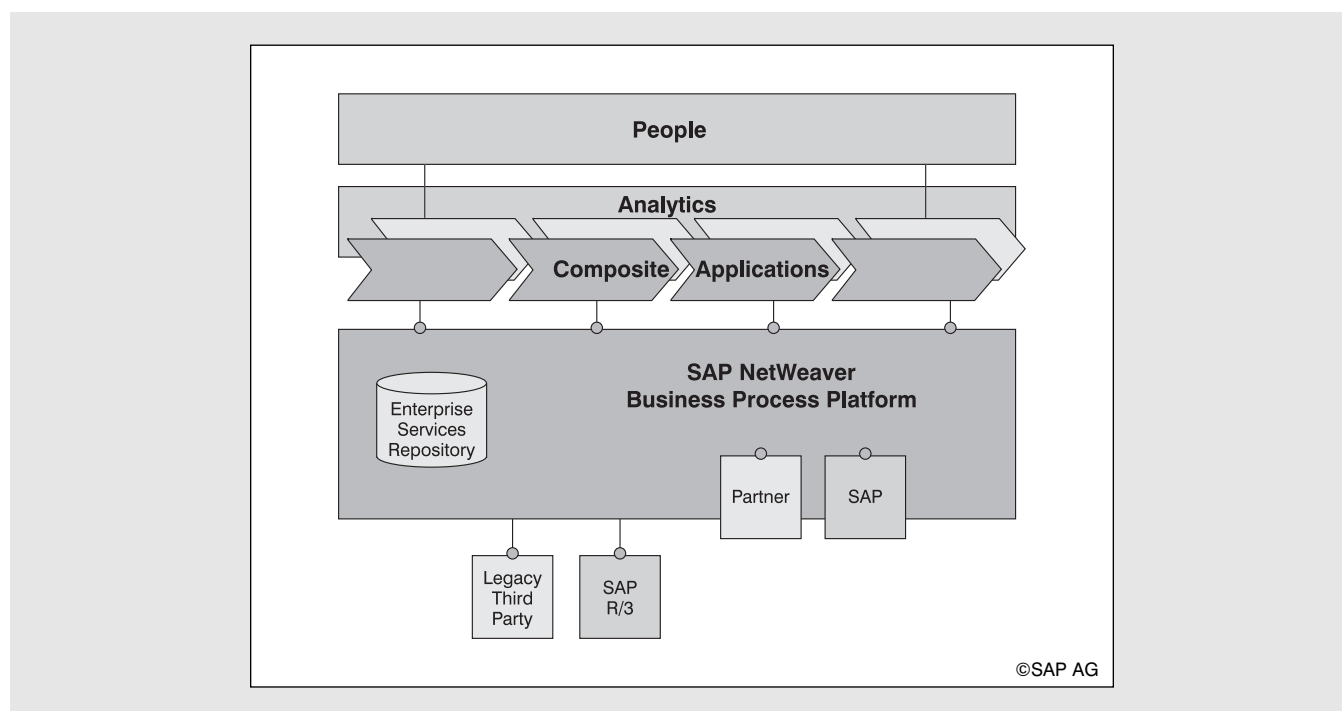


Figure 1 Enterprise SOA infrastructure

Notice the Composite Applications layer, which represents a concept that is new to SAP with the introduction of Enterprise SOA. Composite applications reuse parts of other applications or software components to build new applications. They allow you to separate the process logic from the business logic and create a reusable inventory of services that can be used to create new business processes. For example, you might use the tools in SAP NetWeaver to build a new service or user interface that reuses an existing service to deliver part of its functionality and then use SAP Enterprise Portal (EP) to publish them. This approach simplifies business process modification and optimization because modifying one application or software component automatically updates all processes that are using it. In addition, composite applications provide a bridge from the traditional SAP business application environment to the next generation of service-based applications.

Enterprise SOA also provides support for other fundamental capabilities (see **Figure 2**) that are critical for the effective delivery of service-based solutions,

including people productivity and analytics. As you can see, Enterprise SOA offers a powerful combination of technology infrastructure and productivity tools to convert your SAP environment to a service-oriented architecture. At its foundation is the objective to unify and simplify your business applications.

Don't worry that you will need to replace everything at once to begin taking advantage of Enterprise SOA. SAP recognizes that companies have invested significant time and money to buy, install, and configure applications that already solve many business problems. The migration is not intended to be a dramatic overnight move in a single step, but rather a managed and gradual step-by-step transformation of your existing applications. Enterprise SOA is designed so that you can build new applications designed around services to support new business processes, but also add services to existing applications or software components. New applications that are designed to take advantage of Enterprise SOA and existing applications that are extended by the addition of services can coexist in the same SAP environment, providing a smooth migration path.

Service delivery feature	Enterprise SOA capability
People productivity	Combine the role-based and pattern-based user interfaces of different business applications using SAP Enterprise Portal (EP), which enables you to implement cross-application and team-oriented work centers.*
Analytics	Integrate transactional content from different business applications using the model-driven design paradigm, which supports the complete modeling lifecycle from business process modeling to code generation. For example, SAP Business Intelligence (BI) supports the code-free composition and modeling of data from different business applications.
Service composition	Use model-driven tools for service composition (for new services) and orchestration (for existing services) to define new business scenarios, processes, and applications, which facilitates rapid business process changes. For example, you can quickly create a composite application that provides a new view combining existing data that is already delivered by other enterprise services.
Service enablement	Use the Enterprise Services Repository to define, implement, use, and build user interfaces for enterprise services in all supported scenarios.** This standard service infrastructure ensures that new enterprise services are created with the right granularity (in other words, the enterprise service provides all data that is required for the business process) so you can successfully use them to model business processes.
Business process platform	The SAP NetWeaver platform provides reusable functionality for all applications and establishes a common technical infrastructure for service enabling and business process composition.
Lifecycle management	Enterprise SOA provides a common lifecycle management platform that is valid for all SAP-provided and custom-built applications and services, which supports installation, configuration, change management, and support.
* A work center allows you to encapsulate a set of user interfaces and applications that can be called by a particular user. ** Services can be called either synchronously or asynchronously. You use synchronous services to perform business checks such as data availability, where the calling software component or service waits for a response before continuing. You use asynchronous services in processes that are separate business process steps, where the calling software component or service continues to operate without waiting for a response.	

Figure 2 Enterprise SOA service delivery capabilities

The conceptual architecture of Enterprise SOA

With the introduction of Enterprise SOA, the application architecture changes in the new service-oriented landscape. The conceptual architecture of Enterprise SOA (see **Figure 3**⁴ on the next page) is also referred to as the *Enterprise SOA stack*.⁵ Each layer is responsible for a separate set of tasks that collectively forms Enterprise SOA.

⁴ This diagram is adapted from the book *Enterprise SOA: Designing IT for Business Innovation* (O'Reilly Media Inc., 2006), by Dan Woods and Thomas Mattern.

⁵ For more information about the Enterprise SOA stack, see Dan Woods and Thomas Mattern's book *Enterprise SOA: Designing IT for Business Innovation* (O'Reilly Media Inc., 2006).

Starting from the bottom, the foundation of the Enterprise SOA stack is the *persistence* layer. Enterprise SOA introduces the concept of a distributed data repository that may consist of multiple databases, rather than a single database such as with traditional mainframe or client-server applications. In a distributed system landscape — that is, one that takes advantage of distributed services — each business application or software component may use a separate database. The advantage of a distributed data repository is that each business application or software component may be installed, configured, and used independently of other applications or components.

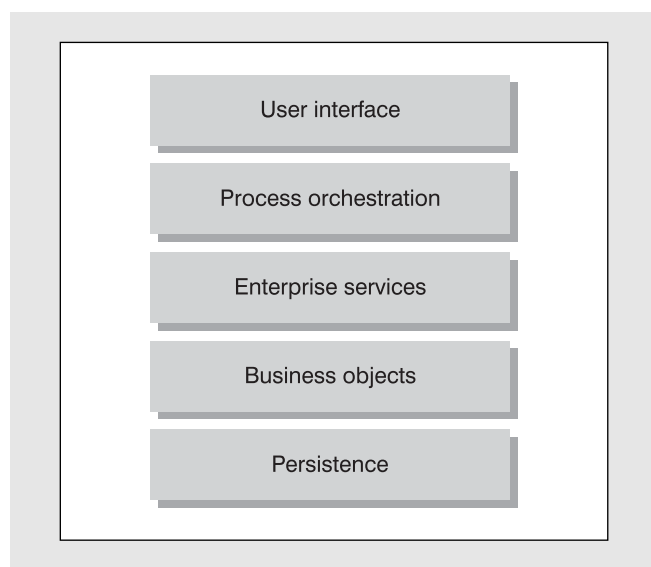


Figure 3 The Enterprise SOA stack

Note!

The distributed data repository has a certain level of data redundancy for performance and other reasons. Therefore, Enterprise SOA provides optional data aggregation and distribution solutions. SAP Master Data Management (MDM) allows you to manage distributed master data across applications and systems. SAP Business Intelligence (BI) allows you to consolidate data from different applications and systems. These solutions enable you to create a logically normalized information model and a physically distributed set of data repositories.

On top of the persistence layer are *business objects*, which are the basic elements of an application in Enterprise SOA. A business object encapsulates semantic data (such as material or customer data) and provides methods for its manipulation. For example, you might have a business object named Material, the purpose of which is to collect material-related information data. Independent of the implementation, business object methods can be considered abstractly

as operations. One or more related operations form a service that can be accessed from outside of the application via its service interface. Think of business object methods as the elementary building blocks of the executable business logic.

Moving up the stack, each business object exposes one or more services, or *enterprise services* as they are called in Enterprise SOA.⁶ The role of an enterprise service is to provide a specific piece of business-related data or functionality that can be used to compose a business process. Enterprise services are designed to be reusable in multiple processes, not just in the exposed functionality. For example, a material availability check process that is part of one enterprise service can be used by any other process or enterprise service that needs to check the availability of a particular material. Both business objects and enterprise services are defined in the Enterprise Services Repository, which is a central shared repository for the entire SAP environment. I discuss this tool in more detail in the next section.

As we near the top of the stack, the *process orchestration* layer provides a mechanism for defining the logic that controls a business process. In other words, you define the mapping that coordinates the business objects and enterprise services within a business process as part of building a business application. This mechanism is called process orchestration because it allows you to integrate elements into a business process. In essence, business processes use a workflow to coordinate and integrate the process steps that are made available by enterprise services.

Finally, at the top of the stack is the *user interface* (UI) layer, which is where Enterprise SOA offers a significant advantage. Enterprise SOA actually uses enterprise services to build UIs. In other words, you use a specialized modeling tool to create a UI that in turn uses one or more enterprise services to obtain the required data. You design and create a UI using SAP Visual Composer, which provides UI patterns as building blocks so you can drag-and-drop objects

⁶ In implementing Enterprise SOA, SAP added support for some proprietary features that extend the Web service standard to provide added power and flexibility. SAP therefore uses the term *enterprise service* (instead of Web service) to reflect this divergence, which I discuss in more detail later in the article.

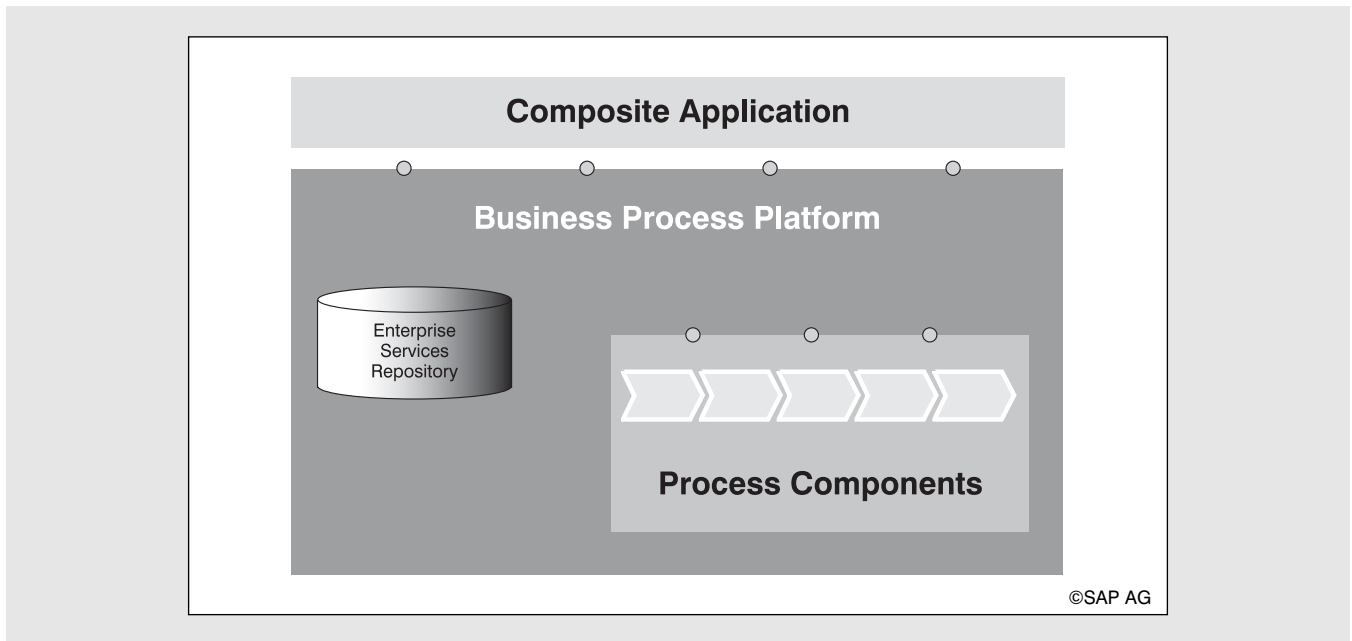


Figure 4 Business Process Platform

Note!

There is a difference between front-end and back-end process orchestration. Front-end processes are conversational and require collaboration and interaction with users. Enterprise SOA provides tools such as guided procedures to assist with front-end process orchestration. Guided procedures are a set of predefined steps that perform a particular action such as searching for and attaching a document. Back-end processes do not normally require any user interaction. Therefore, back-end process orchestration is concerned with long-running, primarily asynchronous processes that perform operations without user interaction, such as the Business Process Management functionality in SAP Process Integration (PI).⁷

in order to build “code-free” UIs. Visual Composer automatically generates the executable code, which is then deployed and run on the SAP NetWeaver Application Server. This tool significantly accelerates and simplifies UI development, as well as extends the potential pool of UI developers. Because you can design and create a UI without any technical or programming knowledge, even business analysts can develop a new application.

Now that you understand the conceptual infrastructure of Enterprise SOA, let’s examine how it is realized in SAP, and specifically in the SAP NetWeaver platform.

SAP NetWeaver provides the Business Process Platform

The central element of Enterprise SOA is the Business Process Platform (**Figure 4**), which is the technical infrastructure necessary for creating and running service-oriented business applications. It supports the concept of reusable business objects and process components that are used in different business processes.

⁷ SAP Exchange Infrastructure (XI) has been renamed SAP Process Integration (PI) in SAP NetWeaver 2004s to more accurately reflect its functionality.

Enterprise SOA tool	Purpose
Enterprise Services Repository	Defining, modeling, and storing the definition of business objects and enterprise services
Visual Composer	Developing user interfaces for applications using a modeling approach
ABAP Workbench	Developing enterprise services in ABAP-based systems
NetWeaver Developer Studio	Developing enterprise services in Java-based systems
Enterprise SOA application	Purpose
Process Integration (PI)*	Integrating processes across applications
Enterprise Portal (EP)	Integrating user interfaces across applications
Business Intelligence (BI)	Consolidating and reporting data across applications
Master Data Management (MDM)	Consolidating master data across applications and distributed data repositories
* Known as Exchange Infrastructure (XI) prior to SAP NetWeaver 2004s.	

Figure 5 Enterprise SOA tools and applications in SAP NetWeaver

From a technology perspective, SAP NetWeaver is the foundation of Enterprise SOA and therefore also provides the Business Process Platform. Support for Enterprise SOA was initially available in SAP NetWeaver '04, although it has been extended in subsequent releases. SAP NetWeaver provides the tools for developing and maintaining service-oriented applications, as well as the supporting system integration applications for Enterprise SOA (see **Figure 5**).

Enterprise SOA offers another advantage with respect to its modeling approach to process and application design. Normally, building a new application requires a lot of written documentation (requirements specifications, design specifications, and so on) to capture the business process description, software requirements, and system integration prerequisites. Enterprise SOA offers a more efficient approach. Instead of writing huge documents, you simply use specialized modeling tools such as the Enterprise Services Repository or ARIS for SAP NetWeaver to create a series of models. These models always reflect both the real business process and the application implementation because the business applications are generated from them. Some parts of the application (for example, proxy classes that are created from a WSDL description) are generated automatically by the

various tools. Other parts of the application require manual work to complete the implementation, such as developing the classes that implement a service. The model-driven paradigm applies to building enterprise services as well as composite applications, as I discuss later in the article.

Before we move on to examining the provided tools, we have one more conceptual aspect of Enterprise SOA to consider — the difference between a Web service in general and an enterprise service in SAP.

Web services become enterprise services

As the foundation for the next generation of business applications, Enterprise SOA seeks to foster closer alignment between the requirements of the business and IT services. However, the underlying SOA concept is based on Web service standards that are somewhat lacking in business semantics. Therefore, SAP extended the concept of a Web service in Enterprise SOA by adding business semantics — in other words, data types that define the meaning of particular business-related data elements. Web services that are created in SAP and specific to

Enterprise SOA are called enterprise services (ES) to reflect this minor divergence.⁸

Enterprise services and Web services are virtually the same in most ways. Like Web services, you can use enterprise services as independent building blocks. You can incorporate enterprise services into a new or existing business process regardless of their implementation or the underlying applications or software components. Also like Web services, enterprise services have separate service interfaces and service implementations that conform to Web service standards. The service consumer (for example, a user interface, an application, or another enterprise service) is therefore isolated from changes in the application or software component that provides the service — in other words, the implementation of the enterprise service.

While based on Web services, enterprise services provide additional enterprise-related information and business functionality. Enterprise services:

- Have high granularity and aggregate reusable elements with specific business value
- Support the execution of specific business tasks, such as the automatic execution of enterprise business scenarios
- Are optimized for the effective development of business processes and composite applications

Unlike Web services, enterprise services use global data types to simplify data exchange. Global data types, which are defined in the Enterprise Services Repository, establish globally available definitions of individual data elements. This approach guarantees that the data that is sent by one enterprise service will be understood by another enterprise service because a single centrally stored definition is used by all participants. Otherwise, each client must handle the data transformation separately, which is error-prone. Therefore, using global valid data types significantly simplifies enterprise service composition and orchestration.

Another advantage is that enterprise services are built on business objects, which collect important data and functionality that embody the unit of business.

⁸ The concept of enterprise services is described by Dan Woods in *Enterprise Services Architecture* (O'Reilly Media Inc., 2003).

Note!

The advantage of using global data types does not apply when integrating non-SAP systems that use different data definitions.

A business object (for example, a customer or material) is a business-specific functional unit that can be reused to create service-oriented applications. Therefore, the same definition is used everywhere that the business object is used.

Aside from the clear advantages, developing an enterprise service is a challenging task for a variety of reasons. Defining the nature of an enterprise service is relatively straightforward using the provided modeling tools. However, completing its implementation entails some manual effort (that is, writing some code), which requires a skilled application developer. See **Figure 6** on the next page for more on the challenges of developing enterprise services.

Now that we have finished our conceptual tour of Enterprise SOA, let's take a closer look at the supporting tools in SAP — specifically, the Enterprise Services Repository and ARIS for SAP NetWeaver.

The Enterprise Services Repository (ESR) for Enterprise SOA

As you have learned, Enterprise SOA enables you to build new or extend existing business applications in SAP using independent building blocks that communicate using standard protocols. These building blocks (business objects, enterprise services, and process components) can be reused or modified as needed and typically provide an industry-specific piece of functionality. SAP provides enterprise services for the SAP Business Suite, although you can also create your own.

All Enterprise SOA building blocks, both SAP-supplied and custom-built, are stored in a central

Requirement	Considerations for developers
Robustness and reliability	An enterprise service may potentially be shared and accessed by many processes and users across the SAP landscape. One enterprise service that is unavailable or producing errors can cause problems for a particular business process or, if centrally used, even an entire business application. Therefore, the stability of a business application depends heavily on the stability of the software components that implement its enterprise services.
Scalability	An enterprise service may potentially need to be capable of supporting thousands of calls in parallel with a reasonable response time. This expectation is especially high for enterprise services that are centrally used by many clients, such as ones that provide master data. Therefore, performance is critical for an enterprise service.
Security	An enterprise service must guarantee high security by providing an authentication mechanism that enforces data confidentiality and end-to-end message integrity. This requirement is especially critical in cases where everyone who starts a business process is not authenticated to maintain (or even view) certain company-sensitive data.
Reliable data sources	An enterprise service must provide the same functionality and data to all clients, regardless of whether they are Web-based clients, other related enterprise services, or a composite application. Therefore, it must utilize data sources that will ensure its stable behavior. Meeting this requirement is a challenging task when you are adapting existing monolithic applications to operate in a service-oriented environment.

Figure 6 Challenges of developing enterprise services

shared repository that is called the Enterprise Services Repository (ESR). This tool is conceptually part of the Enterprise Services Infrastructure (ESI).⁹ Business experts use this tool to model business objects to meet business requirements, while developers use this tool as a starting point and reference for the technical implementation of the defined building blocks. In other words, the ESR is the design-time tool that is the central place where the Enterprise SOA element definitions are stored and maintained. The ESR supports enterprise service modeling (that is, design using model-based development tools) that is based on reusing existing data types, message types, and operations. Establishing a central repository increases the adaptability and transparency of the modeling solution.

From a historical perspective, the ESR is the next generation of the SAP PI Integration Repository.¹⁰

⁹ As part of Enterprise SOA, the ESI is a coordinated set of technologies, infrastructure, and tools for designing and building enterprise services in SAP. Technically, the ESI is part of the SAP NetWeaver platform.

¹⁰ The Integration Repository is the design-time repository that is provided with SAP PI (formerly known as SAP XI), which is part of the SAP NetWeaver platform. For more information, refer to the SAP Help Portal at <http://help.sap.com>.

Figure 7 describes the types of Enterprise SOA elements that you define and store in the ESR. To model enterprise services, you can either use existing or define new data types and service interfaces. As the design-time repository for enterprise services, the ESR contains everything you need to design enterprise services and other business elements. For example, the ESR also stores how an enterprise service is related to business objects that implement it.

In addition to being the design-time repository, the ESR serves the following purposes:

- **Directory of enterprise services:** You can use the ESR to find enterprise services, determine what they do, and transport their definition to the development tools in SAP NetWeaver. The ESR stores the definition of an enterprise service, but not its implementation.
- **Repository for metadata and data types:** The ESR also contains these types of low-level data that enable you to define service interfaces that are then used to define enterprise services. These elements

Enterprise SOA element	Description
Data types	Basic elements that are used to construct enterprise services, business objects, and interfaces. For example, you might define data types such as city, ZIP code, street, and house number that are used to construct a customer or supplier address.
Message types	XML entities that are the design-time representation of messages that describe the communication messages that are sent by enterprise services.
Operations	Synchronous or asynchronous actions that enterprise services can perform at runtime. For example, you might define a material availability check operation, which is a synchronous action that is performed by the purchase order completion step in a process.
Service interfaces	Metadata descriptions of messages and operations that are used at runtime by enterprise services. One example is the enterprise service interface that is used to generate a WSDL document.
Integration objects	Integration objects and mappings that were introduced in the SAP PI (formerly SAP XI) Integration Repository and are also used in the Enterprise SOA solution to facilitate flexible business process integration. For example, you might define a mapping to orchestrate the operations of two different enterprise services that are used in the same business process.
Process models	Business process models that you add to the ESR from the ARIS for SAP NetWeaver modeling tool.
Business objects	Identifiable business entities such as a material or a sales order that can have a logical relationship with other business objects. Business objects are used to ensure orderly development and generate enterprise services.

Figure 7 Enterprise SOA objects defined in the ESR

are used to generate the ABAP and Java proxies¹¹ that enable enterprise services to communicate (for example, via SAP PI) and to implement enterprise services in the development environment.

- **Integrated view of business process modeling:** The ESR treats the business process definition as an object using the service definition and treats its orchestration as a business process using the interfaces and the service definition.¹² This feature supports consistency checks that

allow for early problem detection, even as early as the design phase.

A quick tour of the ESR

After the ESR is installed on the server, you start the tool from the Windows Start menu. You use the object navigation list on the left side of the screen to select an object and you use the function buttons (which are

Note!

The ESR has not yet been released and is planned to be available in SAP NetWeaver 7.1. This section of the article and the screenshot in **Figure 8** (on the next page) provide a sneak preview of the ESR, which is currently being localized in English.

¹¹ For more information about ABAP and Java proxies and their generation, go to the SAP Help Portal at <http://help.sap.com>. Go to the SAP NetWeaver 7.0 (2004s) documentation and navigate to Application-to-Application Processes → Core Development Tasks → Developing Business Logic → Design → Designing Interfaces and Proxy Generation.

¹² The ESR provides an integrated view of the various business process modeling aspects: business view, integration scenario view, executable business process view, mapping and routing view, process components view, enterprise services view, and data types view.

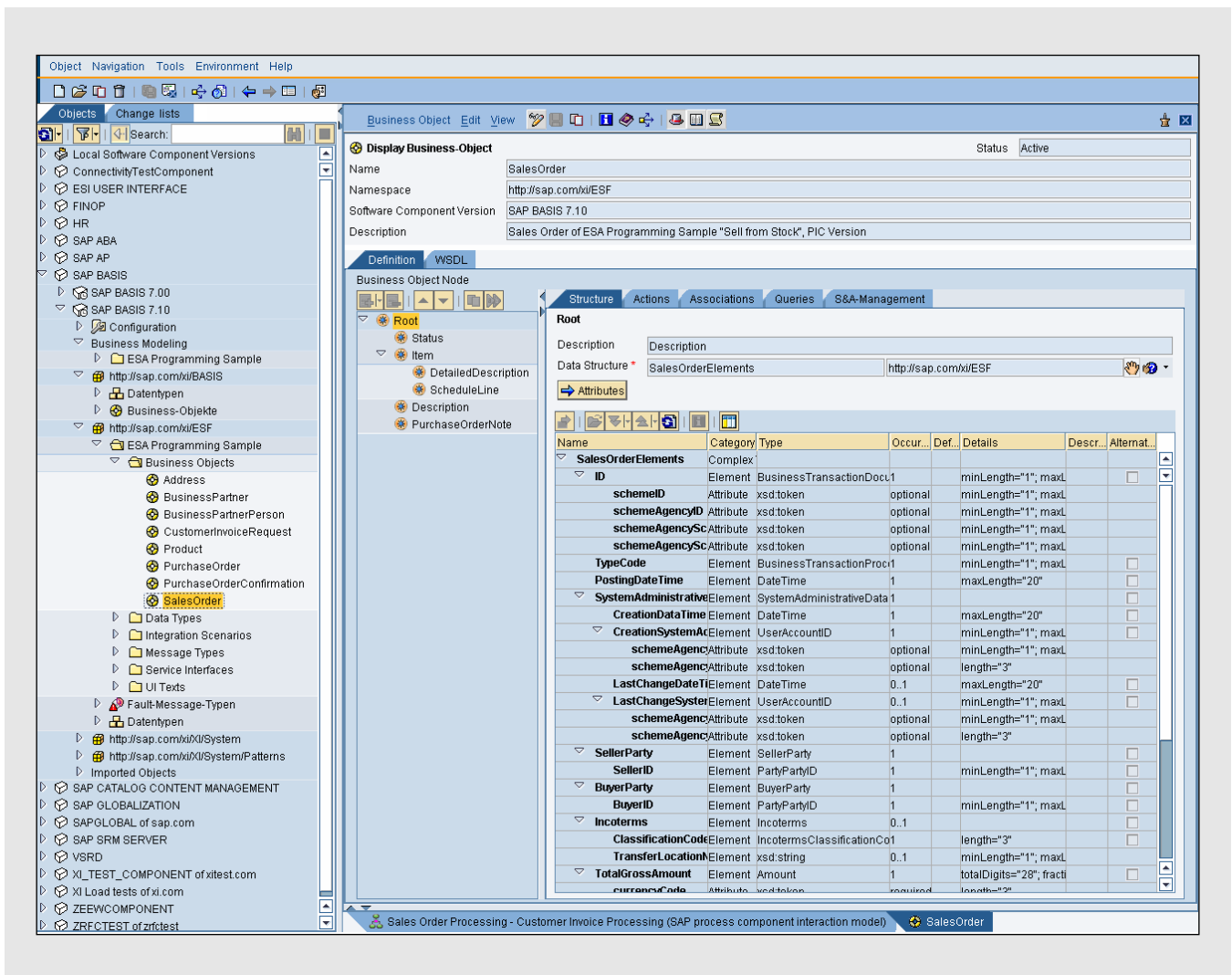


Figure 8 Viewing a business object in the ESR

located in the toolbar under the ESR menu) to maintain the elements of the selected object. The screenshot of ESR in **Figure 8** shows the definition for the SalesOrder business object, which is used in the SalesOrderProcessing business process model. Its definition includes structural information such as the element name (in the Name column) and the data type used (in the Type column).

As you can see, the ESR screen is divided into four areas:

- The object list in the left pane (Objects) is used for finding and navigating to objects.
- The object header in the top pane (Display Business-Object) identifies the selected object.
- The object definition navigation list in the middle pane is used for choosing which object characteristics to display. The Definition tab displays the structure of an object and the WSDL tab shows its WSDL definition.
- The object definition detail area in the right window pane shows the object characteristics. For example, when you select Definition, you see tabs to display the structure, actions, associations, and queries for the selected object.

The entities that are stored in the ESR are linked for easy navigation, from graphical process models down to enterprise services and then to business objects, interfaces, and data types. You expand the hierarchy to display and manage elements. So, to navigate the ESR screenshot, I first selected the SAP release in the object list (SAP BASIS 7.10), followed by the URL for the application (<http://sap.com/xi/ESF>), then the graphical process model (ESA Programming Sample), and finally the SalesOrder business object.

The real strength of the ESR is that it provides graphical process modeling and specification capabilities at design time. You create a model of an object, which the ESR then uses to generate the defined object. Let's look at an example of what happens for a business object such as SalesOrder. A business object is implemented in a business object provider class, which also implements the generic interface that is used for all service interfaces. In reality, an application developer transports the ESR model of the business object to the appropriate system (SAP NetWeaver AS ABAP or AS Java) in order to generate the business object provider classes. The ABAP-based or Java-based system provides the development tools in SAP NetWeaver for writing the code that implements the required functionality. The developer might also need to write additional code, such as a class that implements the database operations for storing the business object in the underlying database.

Modeling tools streamline business process design

Enterprise SOA offers several business-friendly modeling tools that simplify both the design and development of enterprise services. These tools provide an easy-to-use, graphical interface that masks development complexity:

- ARIS for SAP NetWeaver supports business process modeling.
- Enterprise Services Repository (ESR) supports business object and enterprise service modeling.

- Visual Composer supports UI development and integration with back-end systems.

These tools are designed to work together to foster end-to-end business process modeling. For example, you use ARIS for SAP NetWeaver to build a business process model that includes business elements (business objects or enterprise services) that are defined in the ESR. Similarly, you use Visual Composer to create a graphical UI model and attach it to existing enterprise services (which are also defined in the ESR) that provide business data.

In this section, I focus on the capabilities of ARIS for SAP NetWeaver, since we have already discussed the ESR and previous articles have covered Visual Composer.¹³

What is ARIS for SAP NetWeaver?

Enterprise SOA uses the ARIS platform¹⁴ for creating business process models that are designed to support business process management (BPM) projects. Specifically, ARIS for SAP NetWeaver is the business process design solution that is integrated with the SAP NetWeaver platform. It leverages the core competencies of SAP and IDS Sheer to provide comprehensive business process design, execution, control, and monitoring.

ARIS for SAP NetWeaver provides a specialized environment for business process modeling. Among other business-friendly features, it offers a graphical drag-and-drop environment for process model design that supports industry standards for business-oriented notation. (For more information, see the sidebar

¹³ For more information about Visual Composer, see the articles “Get started creating SAP Enterprise Portal iViews with Visual Composer — a purely model-driven, code-free development approach” and “Advanced techniques for enhancing your SAP Enterprise Portal iViews with Visual Composer — a purely model-driven, code-free development approach,” both of which appeared in the November/December 2005 issue of *SAP Professional Journal*.

¹⁴ The ARIS product line is developed by IDS Scheer AG. For more information, go to <http://www.ids-scheer.com/international/english/products/53961>. For more information about ARIS for SAP NetWeaver, click on the More Information link in the “ARIS Software” section, and then scroll down and click on the SAP Solutions link in the “Areas of application” section.

What is Business Process Modeling Notation (BPMN)?

In the past, business experts developed business process models in dedicated design tools such as Microsoft Visio. These tools were completely separate from the representation of the process in the development platform that was used for implementation and execution. Models were manually transformed from one environment to the other, which was often a difficult and error-prone task. As the process evolved, this separation of design and implementation usually resulted at some point in mismatched process definitions in the two environments.

The Business Process Modeling Notation (BPMN) standard is designed to combine different modeling notations and viewpoints in a way that is understandable for both business experts (business specialists, business process experts, and so on) and IT experts (developers and business process administrators).^{*} Its main goal is to close the gap between the business process definition and its implementation. At design time, a business expert creates a draft of the business process using BPMN, typically in a visual design tool that supports this standard. A business process expert or a developer then implements that model in the appropriate technology on the same platform. At runtime, the business process administrator monitors its execution and resolves any problems.

^{*} Other examples of modeling notation that you may encounter are UML Activity Diagrams, UML EDOC Business Processes, IDEF, ebXML BPSS, Activity-Decision Flow Diagrams, RosettaNet, and Event-Process Chains (EPC).

above). The process models that you create in ARIS for SAP NetWeaver are seamlessly integrated into the SAP NetWeaver platform. These process models are saved in the ESR in SAP NetWeaver, which is where you model and create the business elements that are used in the process. The business element definitions are then transported to the appropriate system (SAP NetWeaver AS ABAP or AS Java). There, developers use the associated development environment (ABAP Workbench or NetWeaver Developer Studio) to implement the defined business elements. Overall, this approach significantly reduces

the time needed for and associated cost of business process design and system implementation. The result is faster turnaround in reaction to business process changes — from design to implementation and monitoring.

The ARIS platform consists of the following modules:

- The ARIS Strategy Platform allows you to analyze business processes, their alignment with corporate objectives and strategies, and their efficiency and effectiveness.
- The ARIS Design Platform allows you to design, model, simulate, and optimize business process models and role-based representation of process content. This module is necessary for integration with the ESR.
- The ARIS Implementation Platform allows you to configure SAP systems, link business processes and SAP transactions, and model and manage business rules.

Note!

In addition to being integrated with the ESR, ARIS for SAP NetWeaver also provides interfaces to SAP Solution Manager and SAP PI in order to provide links that simplify configuration.

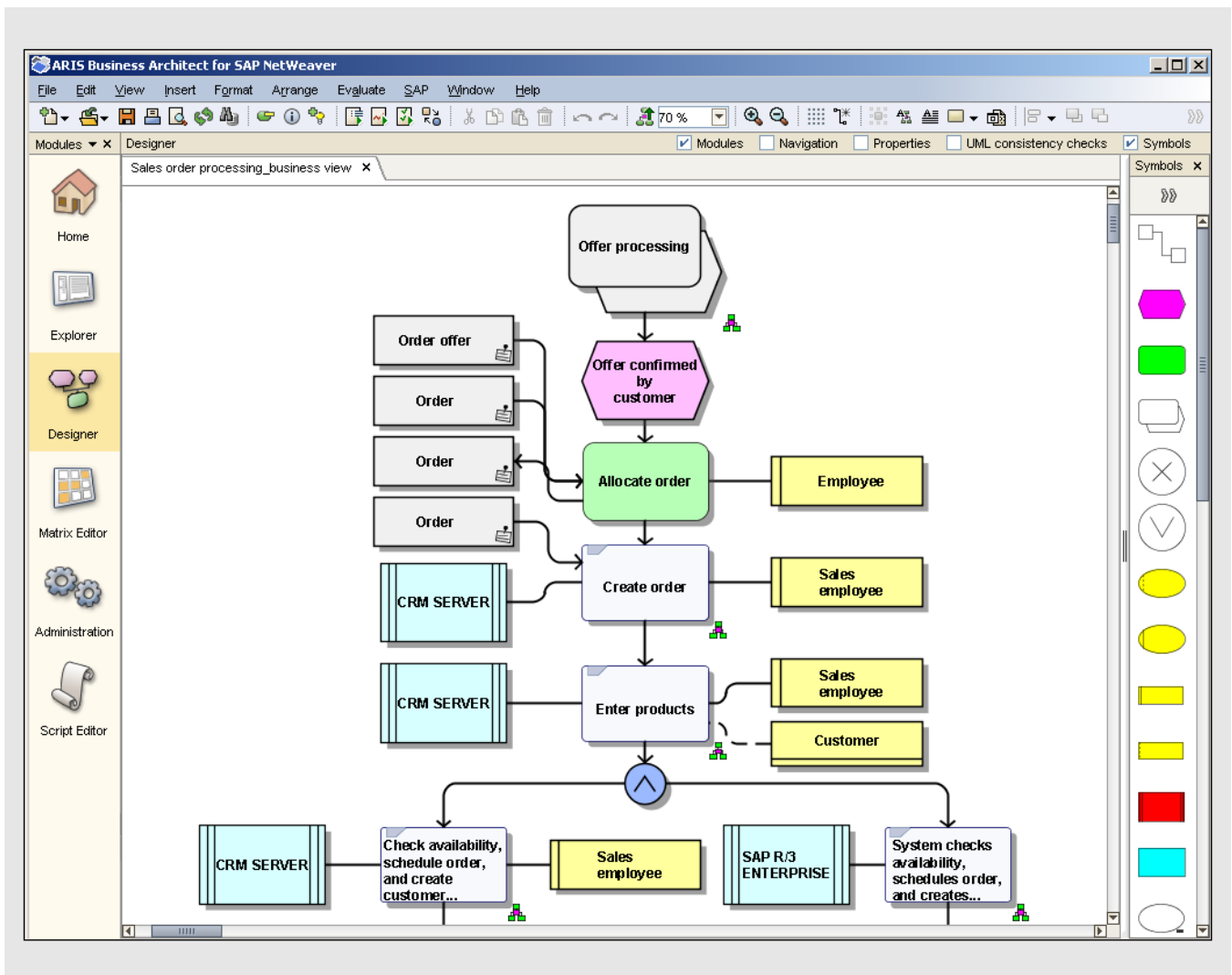


Figure 9 Sales Order Processing model in ARIS for SAP NetWeaver

- The ARIS Controlling Platform allows you to monitor business processes, including monitoring, analysis, and early-warning functionality.

These modules are individually available so you can choose the right combination for your needs. Some modules (such as the ARIS Strategy Platform and ARIS Design Platform) are designed for business people because they support business strategy and model development. Other modules (such as the ARIS Implementation Platform and the ARIS Controlling Platform) are primarily intended for use by technical staff such as developers and system

administrators that implement and monitor the execution of business processes.

Capabilities of ARIS for SAP NetWeaver

After ARIS for SAP NetWeaver is installed on your desktop, you start the tool from the Windows Start menu. To open one of the ARIS platform modules, you select an icon in the left pane (Modules). **Figure 9** shows a business process model (in this case, Sales Order Processing) in the design

view of ARIS for SAP NetWeaver. The Designer icon is highlighted in the left pane, reflecting that this example shows the ARIS Design Platform. ARIS models support top-down or outside-in development, so you can create high-level business process models that drill down into more detailed process models.

The blocks in the process model represent the steps of the business process. You create a model by dragging predefined symbols from the right window pane (Symbols) onto your model. Each shape represents a business step (for example, Allocate order, Create order, and Enter products) and the assigned business objects (for example, Employee, Sales employee, and Customer). When you are finished working for the moment or when the model is complete, you save the business process model in ARIS for SAP NetWeaver. If the ESR is integrated with ARIS for SAP NetWeaver, the model is transferred into the ESR. You can also work offline without the ESR being integrated.

ARIS for SAP NetWeaver integrates industry-standard methods and frameworks for business process modeling. Business Process Modeling Notation (BPMN) is the notation that is used to define business process models.¹⁵ Business Process Execution Language (BPEL) is the notation that is used to define Web services and their interactions.¹⁶ Support for these standards simplifies business process design and development because everyone uses the same language. Business people use it to design and model the real business process and IT people use it to develop the required business elements (business objects, enterprise services, and so on). (See the sidebar on page 118 for more information.)

Enterprise SOA fosters a new development paradigm

Perhaps one of the most challenging problems in software development is managing the development

process, especially when multiple groups collaborate on a single application or product. To alleviate this problem, Enterprise SOA employs business objects, enterprise services, and process components to encapsulate and decouple the individual pieces of functionality in a business application.

Because Enterprise SOA is based on Web services technology, it is possible to separate the technology that is used to specify and assemble the components from the technology that is used to implement them. Like Web services, Enterprise SOA uses WSDL to specify how an enterprise service is invoked and what it returns. The WSDL document (or service contract) is equivalent to an interface specification; it does not define the service implementation. Therefore, an application component that is implemented as an enterprise service can be easily modified and redeployed without affecting the service consumers.

As a result, distributed software development in a managed environment is both possible and a tangible benefit for the development process. Enterprise SOA has improved the component packaging that defines assumptions, dependencies, behavior, resource consumption, and other characteristics. Consequently, enterprise services may provide metadata that can be used to automate component discovery, selection, licensing, installation, deployment, assembly, configuration, testing, and even monitoring. Two services can interact if their service contracts are compatible. Therefore, as long as the developers agree on a consistent contract format beforehand, services can be developed independently and assembled later using the process orchestration capabilities of Enterprise SOA.

Process orchestration also increases development and integration agility. This key Enterprise SOA functionality is part of SAP PI, which is used to transport messages between applications and software components. Using process orchestration, you can easily integrate modified services and systems into existing business processes and solutions. Further, process orchestration also provides assistance in solving service and system integration problems by providing mapping of business data between enterprise services and applications.

¹⁵ For more information about BPMN, go to the Object Management Group Web site at www.bpmn.org.

¹⁶ For more information about BPEL, go to the developerWorks section at www-128.ibm.com/developerworks/library/ws-bpel.

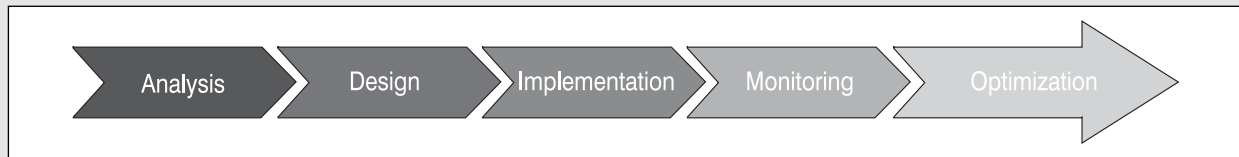


Figure 10 Phased development process with Enterprise SOA

Given these differences from a traditional development project, a phased development process (see **Figure 10**) is inherent with Enterprise SOA projects:

- In the *analysis* phase, the project leads, architects, and business experts analyze the business process, assess its importance, and describe them.
- In the *design* phase, the business process experts create a model of the business process in ARIS for SAP NetWeaver and define the required business elements (business objects and service interfaces) in the ESR.
- In the *implementation* phase, the architects and business process experts with strong developer skills implement the required elements (service interfaces, proxies, and implementation classes) and build new components that are required by the business application.
- In the *monitoring* phase, the business process experts monitor the execution of the business process using SAP-provided tools and ARIS for SAP NetWeaver.
- In the final (and often overlooked) *optimization* phase, the business process experts should seek to make the business process as efficient and effective as possible.

So, for best results with Enterprise SOA, developers should focus on building common services that can in turn be used by business people (for example, business process experts in models) or middleware (for example, automated intelligent clients that are

used in application-to-application or business-to-business systems) to compose business processes. In addition, developers should seek to create a library of common business elements (business objects and enterprise services) for business process experts to use for modeling a business process — for example, business objects such as employee or sales order.

Where to learn more about Enterprise SOA

SDN (<http://sdn.sap.com>) is the best place to find more information about Enterprise SOA. Click on the Enterprise SOA option in the navigation pane on the left to go to the Enterprise SOA home page (see **Figure 11** on the next page), which offers several areas of interest:

- **Adoption Program:** Professional services and accelerators (content, tools, templates, and samples) that help you develop an Enterprise SOA transition plan for your SAP environment.
- **ES Workplace:** Destination to explore, test, and use enterprise services, including those provided by SAP.
- **Discovery System:** Configured environment including sample composite applications where developers and architects can learn about Enterprise SOA and SAP-provided solutions.
- **ES Community:** Collaborative, cross-industry program that brings together thought leaders to

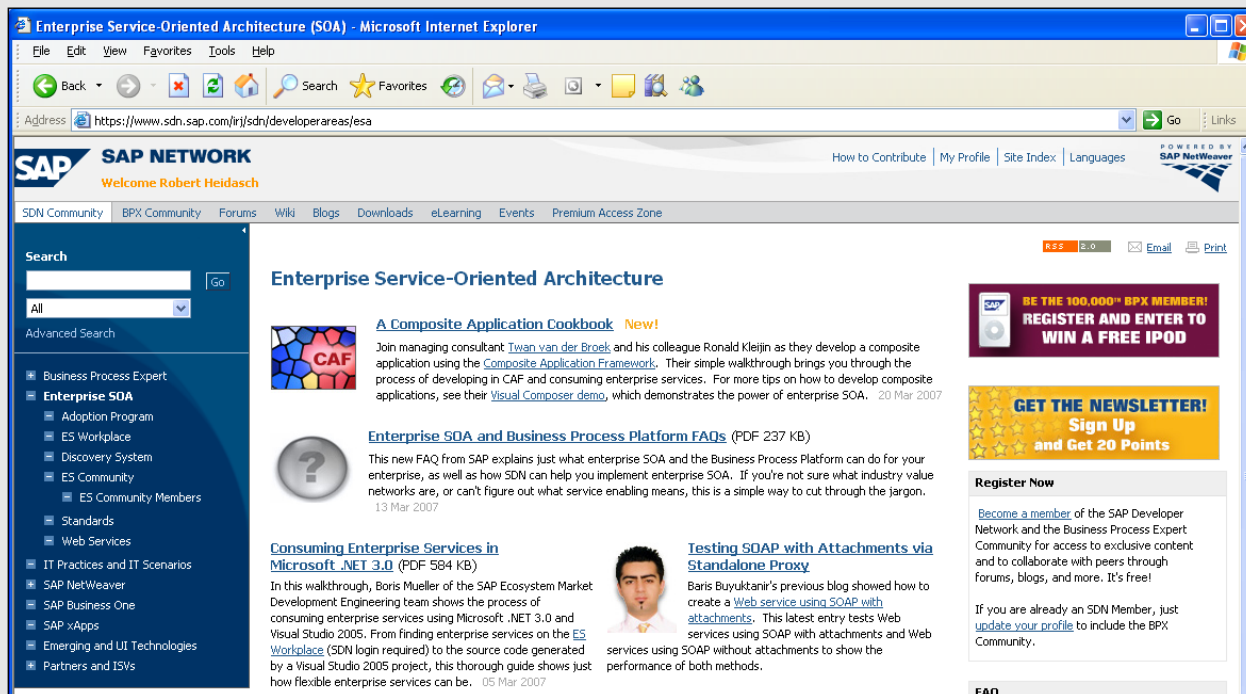


Figure 11 The Enterprise SOA home page on SDN

share ideas and innovations in enterprise services and provide feedback to influence Enterprise SOA.

- **Standards:** Overview of SAP-supported standards and SAP's involvement in the standards community.
- **Web Services:** Knowledge center for building and consuming Web services and enterprise services.

Note!

The Adoption Program and ES Workplace options are only available to registered members of SDN. If you are not familiar with SDN, registration is free and provides community access as well as additional exclusive content.

Let's take a closer look at the ES Workplace page (see **Figure 12**), which is only visible to registered SDN members. This area is an extremely useful resource to help you begin taking advantage of Enterprise SOA. Here, you can explore the SAP-provided enterprise services that are available for the SAP Business Suite. To get started, click on the Start browsing now! link (which is circled in the figure) in the "Discover, Evaluate, and Consume Enterprise Services" section at the top of the screen to access the public SAP Enterprise Services Repository. The "How-to Guides" section on the right side of the screen is another great reference. Click on the How to Use the ES Workplace link to learn more.

Clicking on the Start browsing now! link displays the index for the SAP Enterprise Services Directory (see **Figure 13**). To make it easier to navigate to the solutions that are best-suited for your environment, enterprise services are organized logically by business

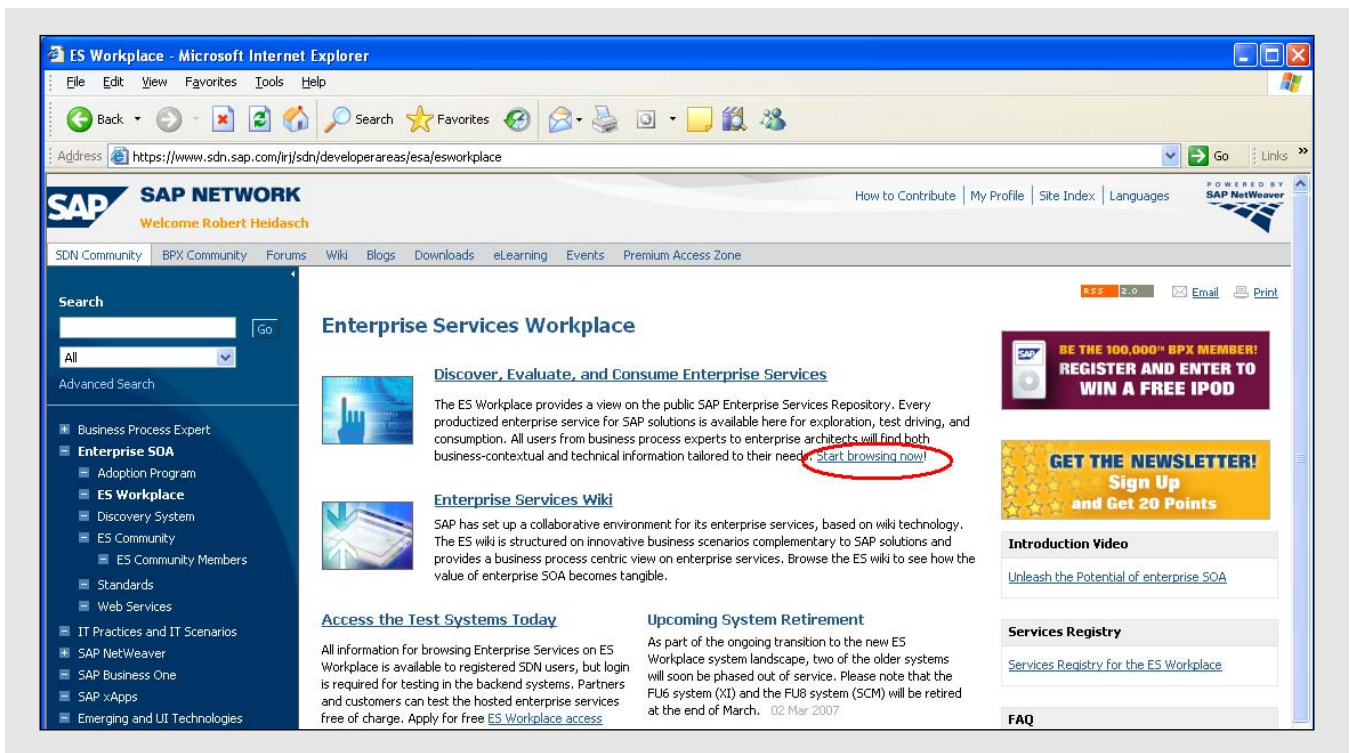


Figure 12 ES Workplace page on SDN

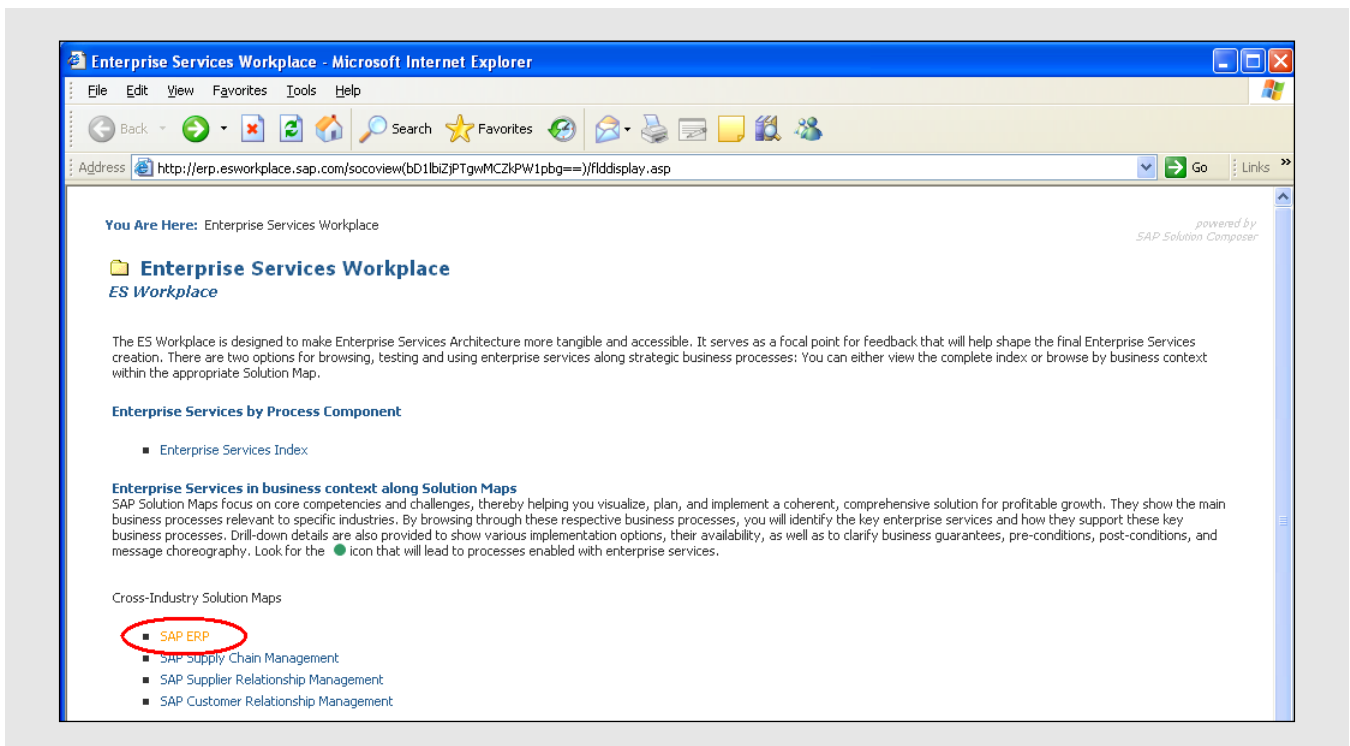


Figure 13 Industry-specific solution maps in the ES Workplace

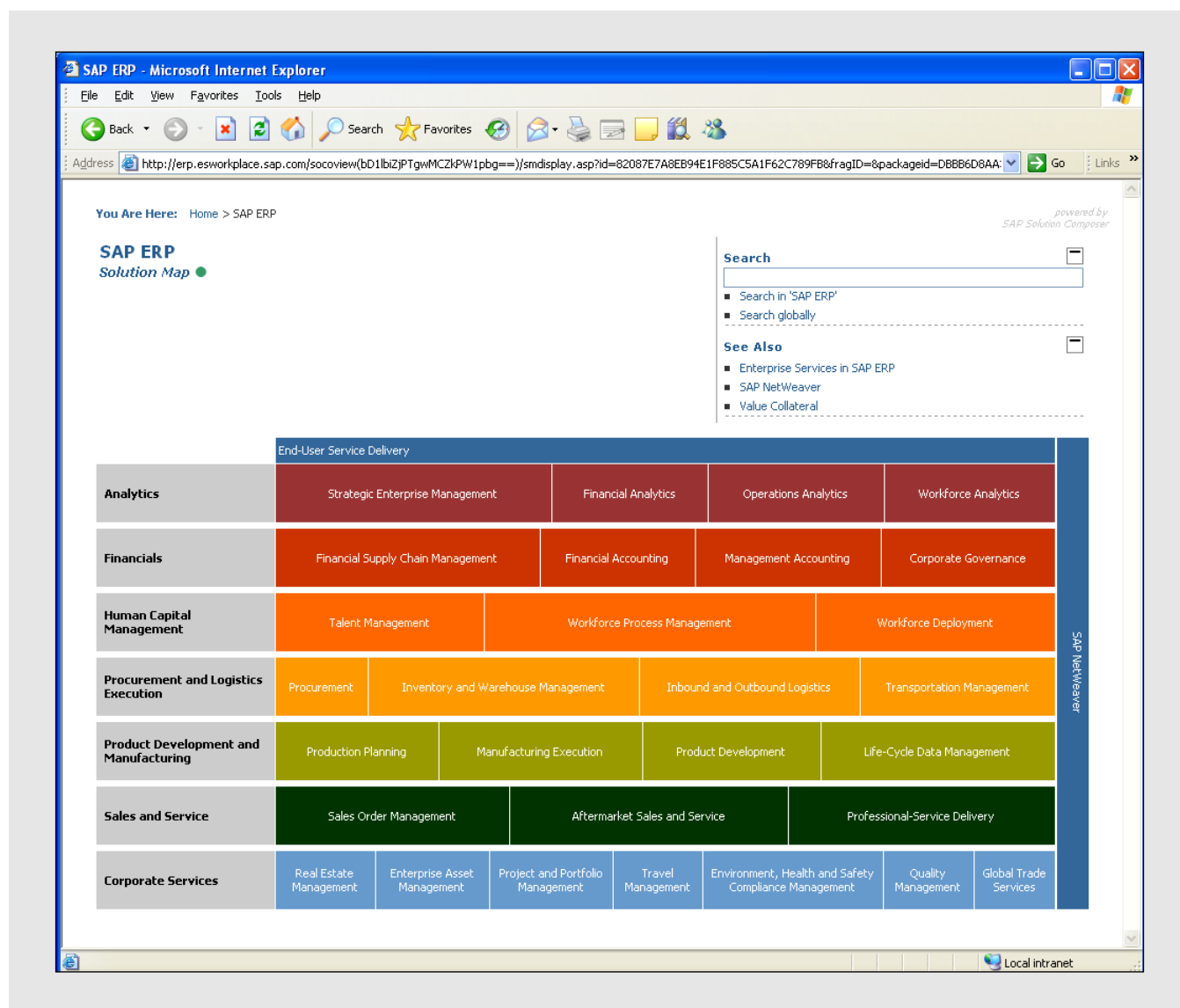


Figure 14 SAP ERP solution map in the ES Workplace

area into cross-industry and industry-specific solution maps. For example, see the SAP ERP link (which is circled in **Figure 13**) in the “Cross-Industry Solution Maps” section in the center of the screen.

Let’s go deeper into one of the solution maps. Clicking on the SAP ERP link takes you to the solution map for SAP ERP (**Figure 14**). Look at the “Sales and Service” section near the bottom of the screen. If you click on the first box in the row (Sales Order Management), you can browse the enterprise

services that support the sales order functionality of the SAP ERP solution.

For example, **Figure 15** shows an overview of the business process for creating and monitoring the different types of sales orders that are supported by SAP ERP. The “See Also” section on the right side of the screen provides useful links such as “Scenario and Process Component List” and “Enterprise Services Index” for navigating to other related elements that are provided by SAP.

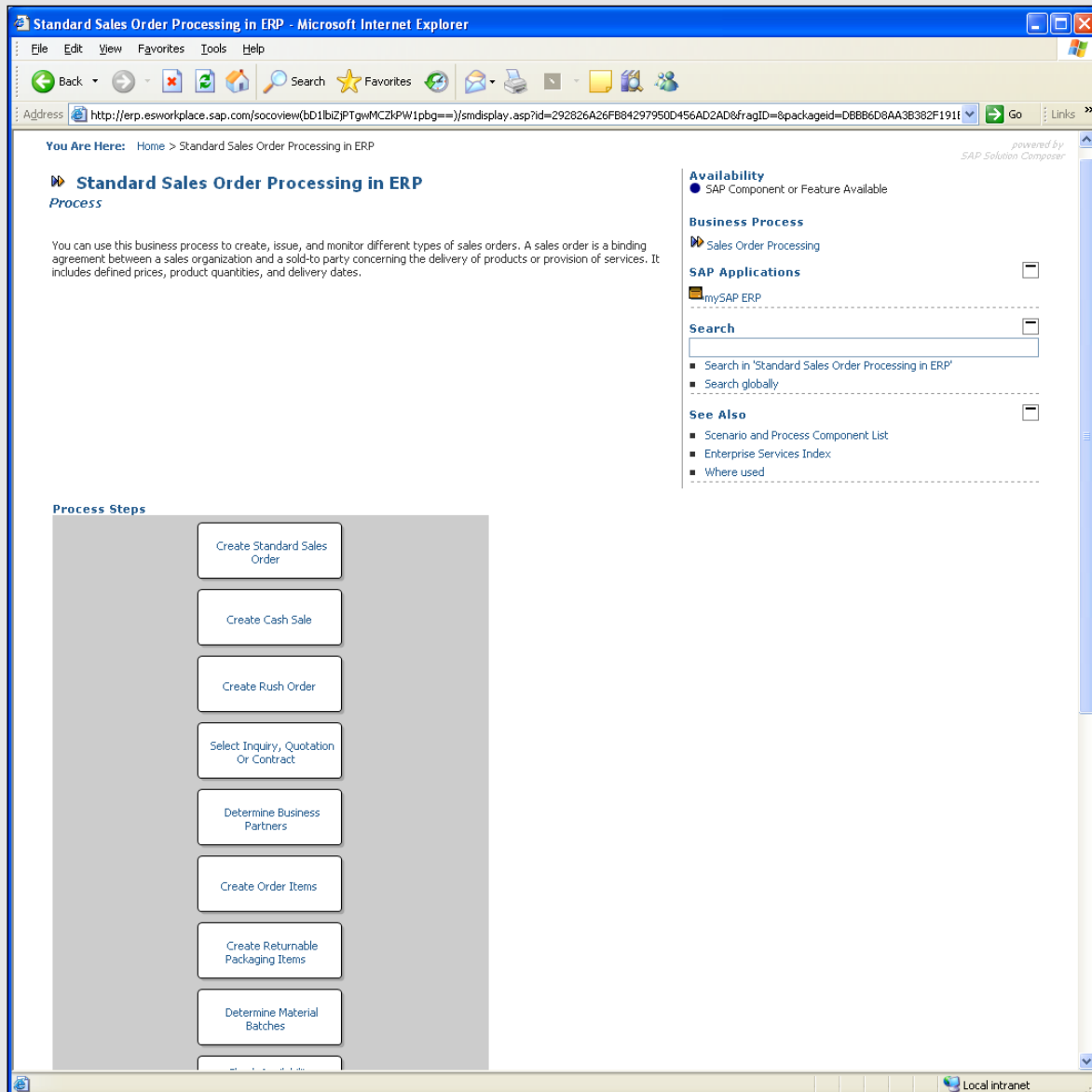


Figure 15 Sales Order Processing business process for SAP ERP

The path to Enterprise SOA — design versus adoption

When considering your SAP-delivered options for migrating to Enterprise SOA, you have two paths from which to choose:

- With Enterprise SOA by design, you implement the new SAP service-oriented business solution that is code-named A1S. SAP is introducing this on-demand solution through a phased launch starting with a set of early-adopter customers.
- With Enterprise SOA by adoption, you migrate

your existing SAP environment to a service-oriented solution by implementing SAP-provided enterprise services for the SAP Business Suite. Support for this option is available starting with SAP Business Suite 2005.

In both cases, SAP NetWeaver is the platform that supports application integration using enterprise services. SAP Enterprise Portal (SAP EP) supports user integration. SAP Master Data Management (SAP MDM) and SAP Business Intelligence (SAP BI) support data aggregation and distribution. SAP Process Integration (SAP PI) supports process integration. In other words, you use SAP EP to integrate the user interface across applications, you use SAP MDM and SAP BI to integrate data across applications, and you use SAP PI to integrate processes across applications via message exchange.

Enterprise SOA by design

AIS is a new business solution that is built using business objects, enterprise services, and process components. SAP has designed AIS specifically for fast-growing midsize companies with limited IT resources. The business objects, as reusable units of business functionality, are used to build enterprise services that work together. The enterprise services are implemented by process components. The process components and enterprise services collectively form the Application Platform, which is running on SAP NetWeaver.

With Enterprise SOA by design, enterprise services are created “outside-in,” which means that the enterprise service interfaces are designed and modeled from scratch to meet the needs of the business processes. The enterprise service interface is designed using Enterprise SOA business modeling tools and then implemented in either ABAP or Java.

This approach is best-suited for automating new business processes and building new SAP NetWeaver-based business applications.

Enterprise SOA by adoption

Enterprise SOA by adoption consists of publishing

enterprise services for existing SAP Business Suite applications. These services can be either SAP-provided or custom-built. In this case, the applications continuously adapt to the service-oriented landscape by “adopting” new enterprise services. While an existing well-functioning application provides a service-oriented interface, the business applications are still largely monolithic units.

Anecdotal evidence suggests that this composite solution works and that the combination of monolithic applications and service-oriented design produces a viable Enterprise SOA implementation. You simply construct a service-enablement layer between service consumers (the client applications) and service providers (the existing monolithic applications). In order to implement a particular function as an enterprise service, you need to define what is referred to as a service wrapper in order to encapsulate the service implementation. A service wrapper is simply a function that implements the service interface and internally calls the existing functions in the monolithic application. Service-enabling existing functionality is a reasonable option for achieving a flexible and agile platform that can respond quickly to changing business processes. You only need to add a small piece of code (that is, the service wrapper) to an existing application in order to provide new service-oriented functionality.

This approach allows existing business processes to take advantage of enterprise services and provides an attractive option for implementing new business processes that are based on your current business applications such as SAP Business Suite. SAP-provided enterprise services for SAP Business Suite are available on SDN. Go to the Enterprise SOA page and click on the Adoption Program link.

In business applications such as SAP Business Suite, you can also use existing BAPIs¹⁷ and IDocs¹⁸ to build enterprise services. Both BAPIs and IDocs are

¹⁷ Business Application Programming Interfaces (BAPIs) are stable, standardized methods that are provided in an SAP system to access business objects.

¹⁸ Intermediate Document (IDoc) is an electronic format designed for transferring business data between systems. The agents that participate in the data transfer use the same IDoc format, which is defined by the IDoc Interface.

designed to be used as encapsulated functions, which makes them similar to enterprise services. Therefore, the enterprise service interface can be generated automatically from the BAPI or IDoc definition. In this scenario, the enterprise service is created “inside-out,” which means the implementation exists first and an enterprise service is generated from it.

Benefits of Enterprise SOA

The main objective of Enterprise SOA is to help SAP customers transform into agile and dynamic business process-oriented and service-oriented enterprises. Enterprise services offer the promise of cost-effective adaptability — faster restructuring of existing business processes, quicker implementation of new business processes, and the ability to rapidly accommodate these changes in the supporting business applications. But the real advantages of Enterprise SOA are the bottom-line benefits to your business:

- Leverage your SAP investment by restructuring existing applications into business-related software components that are better able to support your business processes. Implementing new business processes faster can give you a competitive edge with the ability to deliver new products or services sooner.
- Combine existing systems in new ways to adapt quickly to changing market conditions and business innovation, promoting differentiation.
- Improve development efficiency and turnaround by adopting industry standards and technologies for building loosely coupled service-oriented systems.
- Increase development productivity by leveraging the benefits of a service-oriented architecture:
 - Repeatability, with services that are efficient to develop, deploy, and maintain across the organization
 - Extensibility, with services that can be reused as building blocks for other services or service aggregation

- Maintainability, with services that can be changed once in a central place for all affected business processes
- Supportability
- Reduce development costs by building new software components and applications on top of existing ones, reusing existing enterprise services, and reducing the need for inter-application connections.
- Streamline business process design and development with business process modeling tools that are integrated into the development and operational environment.

With multiple migration paths available, you can begin gaining the benefits of Enterprise SOA in your SAP environment with minimal risk almost immediately.

Conclusion

As you have seen, Enterprise SOA introduces a new dimension to the SAP business application landscape. It establishes the foundation for building software components and composite applications to implement business process applications and revolutionizes how business and IT work together to develop new solutions.

Before the availability of Enterprise SOA, IT staff developed configurable (but still hardwired), one-to-one integrations of different applications (mySAP ERP, mySAP CRM, mySAP SCM, and so on) to meet business process requirements. As you know, this approach requires a huge effort to integrate the various business solutions and applications. It demands a team of experts with different skills (including highly skilled business analysts, technical resources, and experienced consultants) to translate the business requirements into configurable software functionality. Not surprisingly, this task is usually expensive and time-consuming.

In contrast, Enterprise SOA is a model-driven solution that supports the entire modeling lifecycle

from business process modeling to code generation. Business people use their own language and modeling tools that are integrated into the development environment to describe business scenarios and business processes, and then refine them using predefined business objects, business roles, and enterprise services. Developers or application specialists define and implement these business elements to establish a library of reusable objects. Ultimately business analysts can readily create composite applications that use enterprise services to support a defined business scenario, often with moderate or no technical assistance or intervention.

In short, Enterprise SOA is a flexible and effective solution for any SAP customer that builds, deploys, and maintains business processes. These tasks can now be accomplished with significantly greater agility and cost efficiency than with the customizable-yet-monolithic business applications such as SAP R/3. To make the migration easier, Enterprise SOA supports a progressive transformation path for your existing SAP business applications. Therefore, I hope this article has convinced you to begin moving your SAP architecture to Enterprise SOA in the near future and to use it to transform the future of SAP in your environment.