Service-Oriented Architecture

SOA – WHAT DOES IT TAKE?

A CLOSER LOOK AT SUCCESSFUL SOA IMPLEMENTATIONS

THE BEST-RUN BUSINESSES RUN SAP™

SAP®

CONTENT

PREFACE

Service-oriented architecture (SOA) has begun to unleash considerable potential, enabling companies to implement and adapt business processes with unprecedented speed and ease. The world's best-run, forward-looking companies and organizations rely upon SOA to improve agility, enhance collaboration, and reduce operational costs.

This book, targeted at a broad audience that includes executives, managers, and specialists in both business and IT functional areas, presents the benefits of SOA and provides an overview of proven methodologies and best practices for the introduction of SOA. The key ingredients plus the adoption life cycle of SOA are illustrated through several extended examples of how SOA can address business challenges and bring value to organizations in a wide range of industries. IT specialists, including IT strategists, architects, and developers, are presented with an overview of the tools and technology aspects of SOA.

SOA is at the heart of SAP's commitment to help companies attain higher levels of flexibility, efficiency, and insight. SAP co-CEOs Léo Apotheker and Henning Kagermann underscore this commitment.

"In a flat world, companies with the most powerful and flexible business networks will continue to emerge as market leaders. To support best-run networks, businesses must employ flexible and open IT architectures that enable information and processes to span across company boundaries. Since SAP has completed its SOA road map, it provides the tools and methodologies needed to support the various types of organizations – of any size and industry vertical – that comprise a business network."

Léo Apotheker, Co-CEO, SAP AG

"From 2008 and beyond, we are focusing on three priorities: lowest possible total cost of ownership, continuous innovation, and easy consumption of new functionality. Our new enhancement packages – which are unique in the industry – provide customers the ability to continuously consume SAP innovation without having to go through a software upgrade, minimizing disruption. Companies can add new functionality incrementally and only when needed. Moving into the future, SAP product offerings will embrace SOA even more deeply to empower businesses and networks with new levels of flexibility, efficiency, and insight."

Henning Kagermann, Co-CEO, SAP AG

BUSINESS CASE AND IT MOTIVATION FOR SOA

Drawing on the real-time nature of enterprise services, companies are better positioned to outpace their competitors through the ability to rapidly adapt business processes in internal and external networks.

Service-oriented architecture (SOA) is designed to enable the rapid development and adaptation of innovative business processes in pace with changing business requirements. It enables companies to address their priority issues with unprecedented ease and achieve significant business benefits.

The Business Benefits of SOA

SOA offers compelling business benefits. It facilitates the creation and adaptation of business processes within and beyond the enterprise; that is, how easily business processes can be supported by software applications and how easily they can be modified in response to changes in business requirements. Furthermore, SOA-enabled applications also have a direct way of interacting with users. Enterprise services set new standards for the speed of business intelligence and insight for informed decision making by delivering key information as it is generated directly within the context of each transaction, even if these span multiple applications and physical locations.

Insight for Real-Time Business Networks

The most significant improvement provided by SOA is the delivery of unprecedented insight into business processes, complete with predictive analysis, enabling companies to shift attention and resources from transaction handling to the creation of real-time business networks. SOA also supports companies in consistently tracking and achieving regulatory compliance without additional effort, as shown in Figure 1.

Enterprise services set new standards for the speed of business intelligence and insight for informed decision making by delivering key information as it is generated directly within the context of each transaction, even if these span multiple applications and physical locations.

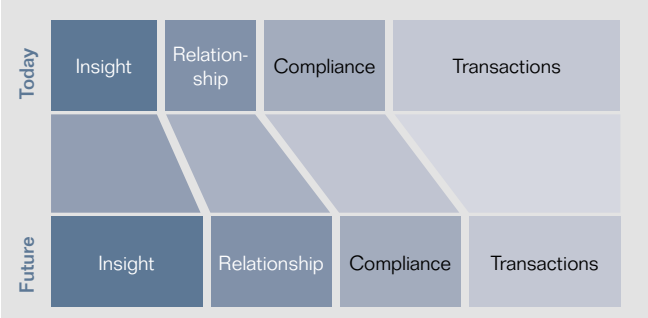

Figure 1: Shifting Attention to Insight and Business Relationships

Innovation and Collaboration for Jointly Achieving Competitive Advantage

With the business world getting "flatter," differentiation is becoming more challenging due to greater connectivity and "hypercompetition" that goes beyond leveraging the reduced cost of labor. The experience of forward-thinking companies around the globe indicates that the best response to hypercompetition is to innovate their business model, not just their products or services, by changing their fundamental value proposition, profit model, or resource utilization in a way that is not easy to copy and that naturally differentiates the organization.

This challenge calls for levels of collaboration that only business networks can provide – and SOA is the architecture for the services on which these networks rely. Leveraging the connectivity provided by enterprise services, companies are able to flexibly build better and stronger relationships with partners along the value delivery chain.

SOA closes the gap between business and IT, providing IT with the ability to create and adapt business processes at the speed of business change.

Simplification for Enhanced Ease of Use

SOA helps companies simplify their process models. For instance, companies can design role- and usage-specific interfaces that avoid clutter and lower the entrance barrier for data-gathering tasks. Another example is the use of enterprise services to shift from power users and human integrators to front-office self-services that exploit holistic data gathered across applications and technologies. As a case in point of a customer-facing user interface, consider the portal of an automotive company that enables a customer to check the current production status and delivery date of a car on order. When the customer uses the portal to change an item (such as seat upholstery) in the order, an enterprise service informs the customer of the resulting change in price.

SOA enables companies to deploy a combination of technologies that best matches the business model. In addition, SOA allows companies to move away from traditional user training and to shift resources to the modeling of user interfaces.

The IT Advantages of SOA

SOA offers significant IT advantages. The real-time nature of enterprise services accelerates how processes are executed. By leveraging the Internet as a ubiquitous, fast, and low-cost technology, enterprise services support fast application processing by eliminating latency and process disconnects.

An Architecture for Strategic Agility

Strategic agility – drawing on a combination of insight, flexibility, and efficiency – allows organizations to respond more effectively to shifting business challenges and opportunities. The IT organization plays a key role in enabling a company to achieve strategic agility. Leveraging SOA, the IT organization consistently delivers and connects the right enterprise services at the speed and quality the company requires for its business processes.

Landscape Modernization for Reduced Operational and Maintenance Costs

SOA makes it easier for companies to streamline and unify their software landscape. For example, a standard interface protocol can reduce complexity and facilitate the composition and consumption of services across multivendor landscapes. Such a protocol can enable companies to make better use of what they already have instead of forcing them to replace their legacy applications on a large scale. For mergers and acquisitions, customized enterprise services help to significantly accelerate the deployment of composite applications. Enterprise services can be employed to facilitate upgrades across the extended enterprise, while lines of business will no longer be left behind on older releases during massive global rollouts.

By reducing the effort and time required to build or redesign business processes, SOA has proved its worth for industrial corporations as well as for public sector organizations worldwide. SOA reduces the expense and effort of code development and maintenance and supports the reuse of enterprise services for faster solution delivery and cost savings.

Connectivity for Instant Processes

SOA enables organizations to exploit the connectivity delivered through enterprise services for greater agility and extended business reach. Use of existing Internet connections helps keep the cost of this connectivity extremely low. In addition, Internet connectivity thus attained is nonproprietary and interoperable, complies with industry standards, and provides global coverage. This high-speed connectivity increasingly encourages customers to replace their asynchronous hub-centric data flows with instant data access through the internal or external business network for fast and informed decision making.

IT and Business Alignment

SOA changes the way in which business and IT deal with each other, and the change in mind-set is significant. SOA helps a company's IT organization and the business units it serves speak the same language. The IT organization and business units can address business objects and processes jointly rather than focus only on technology. The IT organization provides the underlying architecture required to deliver and maintain services with business-level meanings. IT and business alignment is the key to the design, development, use, and reuse of enterprise services that are synchronized with business needs.

THE EVOLUTION TO SOA

The ongoing mass adoption of SOA, supported and enabled by SAP® business solutions, drives the growth of the enterprise services portfolio developed by SAP and the SAP ecosystem and partners. At the same time, feedback from customers helps SAP enhance the portfolio of enterprise services in line with current or emerging business requirements in a wide range of industries.

The Role of Enterprise Services

Within the SOA framework, complex business processes can be broken down into smaller process components and business objects, represented in software as reusable enterprise services. In a nutshell, enterprise services are highly integrated Web services combined with business logic and semantics; they can be accessed and used repeatedly to support business processes.

SAP delivers industry-focused bundles of enterprise services that are based on a design and modeling methodology. This methodology leads to harmonization of data models and business semantics. SAP customers – as well as their IT service providers, business partners, and suppliers – are encouraged to use the same methodology for breaking down business process components and business objects into new enterprise services, enhancing the harmonization of data models and business semantics. To make the resulting enterprise services available for reuse, they need to be collected, documented for clear communication and tracking, categorized, and maintained in a repository.

Based on Web service standards, which enables them to be consumed by every Web service client, enterprise services also incorporate nontechnical aspects. Simple data types are based on the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) Core Components Technical Specification (CCTS) standard. More complex types are defined in close cooperation with industry experts so that, for example, a date, a customer number, or an order has the same format in all enterprise services. Each enterprise service follows a clearly defined communication pattern; for example, query-response, request-confirmation, and so on. The communication pattern is also indicated by the service name to clearly identify its use.

From Web Services to Enterprise Services

Web services have been one of the most enthusiastically adopted technology standards in the history of the software industry. Based on XML and other open standards, Web services started off with an elegant and simple approach to self-description, using Web Services Description Language (WSDL), and to discovery, using Universal Description, Discovery, and Integration (UDDI). A remarkably harmonious standards development culture has evolved, adding a number of related standards for services, policy, versioning, security, reliability, and so forth. Developers can create Web services and have great confidence that they will work in any environment.

Overcoming the limitations imposed by the special-purpose interfaces of earlier generations of Web services, enterprise services are designed to master multiple interfaces, eliminating the need to learn a separate interface per use. To achieve this, enterprise services draw on global data types (GDTs) for consistency and uniformity, providing the standardization that is key to enterprise service reuse.

The problem now is to resolve the semantic challenge of Web services. Consider this example. You create an application (called a "composite" application) that uses customer information from three different enterprise applications. Each of these applications stores a name and address, but two use a different format for the address – one application uses a five-digit zip or postal code, the other uses a nine-digit code. Using this information on a read-only basis presents the challenge of deciding which information is authoritative. Although the information in the enterprise applications can be synchronized using a software solution such as the SAP NetWeaver® Master Data Management component, this situation is a prime contender for a customer information service that reads data from the authoritative applications and then becomes the single source of truth for such data.

An enterprise services repository comprises the design time functionality, while the services registry manages the information required to run an enterprise service.

Attempting to update this information provides a greater challenge. It is untenable for each composite application to have the intelligence to determine how to update each underlying enterprise application. Instead, what is needed is a customer information update service that accepts changes to customer records and then applies them, after approval, to all relevant enterprise applications. In this way the process of updating customer records is managed in one place.

The three underlying enterprise applications in this example remain within the boundaries of one company. However, this approach can be extended easily so that the customer information update service provides a service interface that enables customers to submit changes directly to their own name and address information, significantly improving both the accuracy of the data entered and the speed of the data update.

Enterprise Services Repository

An enterprise services repository is a central library where all enterprise services metadata is modeled and documented. Such a repository is used for centrally managing all enterprise services objects used in SOA application development. It provides governance for enterprise services and application design time, ensuring that enterprise services adhere to the enterprise services modeling methodology and align with standard data types and semantics. It provides information for the definition of enterprise services and metadata, central modeling of services, and the design environment. The repository enables the reuse of enterprise services and stores information for modeling and composing new applications.

Services Registry

The services registry, an integral part of an enterprise services repository, supports the publication, classification, and discovery of enterprise services for reuse in SOA-enabled applications. The services registry enables runtime governance by providing information about all services and services definitions in the SOA landscape, including references to WSDL files and to the locations of callable service endpoints.

SOA Governance Is Key

The term SOA governance refers to the processes, policies, and associated control mechanisms a company establishes in order to steer the adoption, implementation, and evolution of SOA in line with business needs. Governance provides SOA initiatives with structure, discipline, and control.

Consistently successful SOA calls for a holistic and evolutionary governance approach that encompasses people, processes, and technologies. SOA initiatives that rely on a tool-based governance approach alone are inevitably limited in scope. To ensure that the benefits of SOA are achieved, governance must expand beyond the core strategic alignment between business and IT. It has to encompass the full breadth and depth of ongoing transformation and growth of the company. SOA governance embraces well thought-out processes, guidelines, and procedures that need to be systematically developed and matured.

Governance is not just nice to have in SOA; it's a strategic imperative. You need good governance to unlock the full potential of SOA and to ensure the long-term success of your SOA initiatives. When SOA initiatives are done right from the start, they reduce the time required to achieve value. Competent implementation and ongoing management and control ensure that SOA delivers sustained benefits. Corrective actions and adjustments keep SOA initiatives on track. Composite applications that rely on SOA are designed, developed, and implemented efficiently and cost-effectively, while proven mechanisms encourage and support the reuse of enterprise services.

Provided that sound governance principles are adhered to from the outset, the reuse of enterprise services can deliver efficiency gains and cost reductions – immediately as well as in the long run.

Governance – Roles, Responsibilities, and Skills

SOA involves controlled changes in organizational structures as well as the definition of processes, roles, and skills. As people understand their roles in designing and implementing SOA solutions, they also understand the business usage of SOA applications. Principles for sound decision making as well as robust escalation procedures need to be in place.

IT and business organizations jointly define the policies that govern their interaction as providers and consumers of services, respectively. An SOA competency center can provide expertise and control. The board of such a competency center is run by the head of enterprise architecture and serves as the lighthouse that the organization turns to for guidance and advice. Enterprise architects are responsible for mapping business requirements and processes to IT services. Other SOA-centric roles, such as business process expert and integration expert, may come on board. In most companies, these roles can be filled by existing employees.

Design Time and Runtime Governance

The foundation for every successful SOA strategy is laid during design time. From the architectural bird's-eye view, services are identified, designed, modeled, and published – if they don't already exist – in an enterprise services repository. Design-time governance ensures that the right services are designed correctly and are available for runtime use.

SOA governance is about the right people making the right decisions in order to achieve and sustain business agility for competitive advantage.

As shown in Figure 2, design time encompasses the life-cycle phases of service identification, design, and implementation. The service provisioning phase transitions services from design time to runtime, exposing services to the entire organization by making these services available for productive use through an enterprise services repository. Runtime encompasses the life-cycle phases of service deployment, management, and analysis.

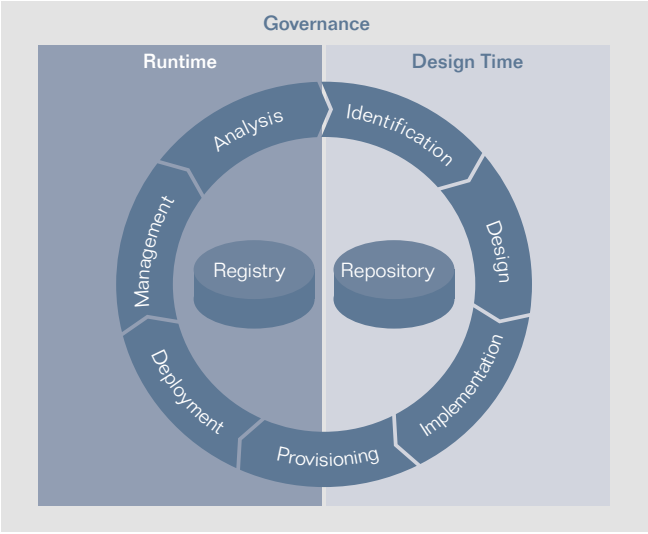

Figure 2: The Role of Governance in Design Time and Runtime

By providing preconfigured enterprise services and all of the elements needed for the design and development of additional enterprise services as well as relevant training, SAP gives companies everything they need for design-time as well as runtime governance.

How to Ensure Effective Design-Time and Runtime Governance

It is essential to have an environment that provides design-time and runtime governance for SOA in an evolutionary, incremental manner. You should always address existing services first and bring them under full governance control before you provision new services. Simple spreadsheets may suffice during the early stages of SOA adoption, but more powerful tools, enhanced functions, and broader skills are required to ensure adherence to service design principles as SOA matures.

As you move forward with SOA, a trusted advisor can provide invaluable guidance, helping you to establish and enforce effective design-time and runtime governance policies and to ensure that SOA deployments remain cost-effective, reliable, and manageable throughout their entire life cycle.

Golden Rules for Effective SOA Governance

- Plan for, continually monitor, and drive the reuse of enterprise services. Reuse doesn't just happen; it requires commitment and control. Keep assessing progress and be sure to check your "SOA scorecard" on a regular basis.
- Start with a readiness assessment. First of all, see what you already have in place before you move forward.
- Begin with small and simple steps that deliver fast results; get some quick wins, leveraging your current portfolio of applications and IT projects. Identify enterprise services that are ideal for reuse.
- Keep your service models as simple and well documented as possible. Documentation that is clearly written and easy to access enables and drives service reuse.
- Foster a culture of communication to overcome initial reluctance. Address responses such as "not invented by me" and "I can build a better interface myself" by emphasizing the benefits of reuse.
- Use an enterprise services repository to automate design-time and runtime governance, support life-cycle management, and facilitate reuse.
- Define a number of core competencies and associated roles to drive architectural development and organizational procedures and overcome functional silos.
- Build and actively participate in an enterprise services community so that the company can benefit from best practices and accelerate advancement along the SOA learning curve.
- Establish a change management process that prepares your organization for anticipated changes as well as for unexpected events. Make sure that enterprise services reuse (including the first instance of service reuse) is governed by a defined process.
- Establish effective escalation procedures. Opinions can and will differ. Set up an enterprise services design steering committee that is staffed by employees and stakeholders other than those of an SOA competency center.
- Don't reinvent the wheel. You are not alone out there. Trusted advisors and a community of SOA peers are willing and able to provide support and guidance – and share best practices with you.

SAP has SOA-enabled SAP Business Suite applications and offers a growing portfolio of industry-focused enterprise services bundles to deliver a rich set of services that can be used to build composite applications.

Standardization Is Imperative

Standardization is key to cutting costs and effort as well as to reducing complexities in the SOA arena. When properly employed, standards not only dramatically reduce the effort and cost associated with integration, they also cut the cost of building, operating, and managing enterprise services. These benefits can be achieved through a combination of technology standards and semantics standards that define the information flow through and among the applications that consume enterprise services.

A typical IT organization may spend over a third of its budget on integration issues. To a large extent, this expense is caused by a lack of standard semantic definitions. SAP addresses this lack by SOA-enabling its existing solutions and creating industry-focused enterprise services bundles to deliver a rich set of services that can be used to build composite applications. SOA leverages these prebuilt enterprise services by bringing them together in a way that efficiently addresses and resolves business problems. The portfolio of enterprise services from SAP keeps growing, making it easier for companies to leverage these for faster time to value in their SOA initiatives. The primary objective of SOA is that every business application used by a company – and, eventually, every application used by a company's business partners – is enabled by enterprise services or by composite applications that are composed from enterprise services.

Without standards for semantics, composite applications are burdened by the need for complex semantic mapping – especially when enterprise services are combined for collaborative business models that span multiple companies. Standards for semantics can be defined accurately only in a business context, which requires the participation of individual or collaborating businesses plus their implementation partners and SAP. With these parties at the table,

SAP delivers the technology stack – complete with enterprise services that reflect lessons learned – that shortens the learning curve and accelerates the adoption of SOA.

new sets of services can be properly defined and incorporated into products in a way that increases technology and semantic interoperability, reduces the need for complex integration, and further expands the scope of services available to construct new solutions.

To make the creation of composite applications as easy as possible for customers and partners, SAP leverages technology standards, such as Java Platform, Enterprise Edition 5 (Java EE 5), for modeling and development. SAP provides value to customers by treating the development and use of standards as an evolving process, rather than as a set of static definitions that just need implementation.

SAP Strategy and Technology for SOA

The SAP approach to SOA supports a business-driven software architecture that goes beyond SOA fundamentals by adding business semantics. SAP combines the technologies, standards, and approaches of SOA with a common business language that SAP co-defines within the SAP® ecosystem in the form of enterprise services.

SAP helps organizations radically increase the scope of enterprise services and dramatically decrease the effort involved in creating and deploying these services, ensuring faster time to value in enterprise-wide SOA initiatives. SOA-enabled business solutions and enterprise services from SAP reflect what SAP has learned and continues to learn from its customers.

SAP provides customers with software, tools, and technology that support the design, development, identification, and consumption of enterprise services. The breadth of offerings from SAP addresses the full life cycle of SOA projects – from development to deployment and ongoing management – making the transition to SOA easy and affordable. SAP offerings underscore SAP's commitment to helping its customers reap the benefits of SOA.

Business Process Platform

SAP helps organizations establish their own business process platform by delivering the software and technology, including the SAP NetWeaver technology platform, SOA-enabled business software solutions, and reusable enterprise services. With this software and technology, customers can align their IT landscape with their specific business requirements. With their own business process platform, organizations can compose applications using enterprise services, orchestrate business processes and events, manage enterprise information, and deliver applications and content to users more quickly and cost-effectively. Companies can adopt SOA at their own pace, extracting new value from their existing applications while building the composite applications to transform the way they do business.

SAP NetWeaver enables applications and services to connect with each other, regardless of the semantics of the information being processed. SAP NetWeaver preserves existing technology investments in SAP solutions, supports the integration of non-SAP applications, leverages technology standards, and helps customers realize the benefit of new standards-based technologies.

Part of the SAP NetWeaver technology platform, Enterprise Services Repository (ES Repository) contains a description of each enterprise service along with the process models that show how each service fits into business processes. The SAP NetWeaver Composition Environment (SAP NetWeaver CE) offering is the development environment for orchestrating enterprise services and for building SOA-enabled solutions.

SAP Business Suite

In 2006 SAP announced its strategy for the SOA enablement of SAP Business Suite applications. The Enterprise Services Workplace (ES Workplace) site, which is accessible via the SAP Developer Network (SDN) site at www.sdn.sap.com, documents the associated services. Enhancement packages for the SAP ERP application (version 6.0) and other SAP Business Suite applications are planned for release twice yearly. These include enterprise services bundles, which are collections of related enterprise services that enable a specific set of business processes by providing access to data and functionality. This strategy allows SAP customers to rely on a stable product as a foundation while receiving innovative new functionality in the form of enterprise services for changing needs.

Enterprise Services Bundles

Developed in collaboration with the SAP ecosystem and partners, enterprise services bundles deliver a rich set of services that can be used to build composite applications. These bundles are available for industry-specific and cross-industry processes. The services included in these bundles target real-life needs and are easy to implement, enabling your IT organization to score "quick wins" for your company. Delivering new business functionality with minimum effort and lead time, these services help to lower total cost of ownership (TCO) and accelerate time to value.

SOA Training Courses

Drawing on in-depth experience in SOA, the SAP Education organization provides a fully fledged SOA curriculum. Instructors share their insight into the methodologies and governance processes that underpin the design, implementation, and reuse of enterprise services. For example, SOA-centric SAP training courses build the skills that your SOA specialists need through hands-on guidance in the design, development, and governance of enterprise services that are aligned with business needs.

Development Support and Sourcing

For custom development and in-house projects, SAP helps customers build the required skills base through knowledge transfer and R & D support. If required, SAP consultants provide thought leadership and guidance as well as hands-on assistance in the creation of enterprise services tuned to your specific needs. Within the SAP ecosystem, peers provide SOA expertise and are willing to share their insight. Optionally, you can outsource the development of enterprise services to SAP or to third-party providers.

SAP Ecosystem and Partners

SAP has created an ecosystem of partners and customers to guide the evolution of its solutions, products, and standards. This ecosystem includes:

- Industry value networks that engage business executives of each industry
- An initiative for engaging companies that create the core for enterprise computing
- The Enterprise Services Community program for designing semantically compatible service definitions in cooperation with businesses and independent software vendors (ISVs)

SAP continues its long-standing participation and leadership in standards bodies that develop technology standards as well as standards for semantics. SAP supports industry-standards initiatives around service definitions, collaborating with external standards organizations to help turn those service definitions into standards.

Start small, think big. That's the proven recipe for successful SOA projects. Building on experience, your company can move on to more ambitious stages of SOA adoption.

Within the SAP ecosystem and partners, a growing community of organizations from all industries and from the public sector is engaged in collaboration that drives co-innovation to the mutual benefit of all community members. The ecosystem enables organizations to harness and share collective knowledge, best practices, and a wealth of resources for greater efficiency, lower costs, faster results, and reduced risk in initiatives.

The Best Way Forward for SOA

You are well-advised to start your journey to SOA with a small pilot project. As you gather experience, you can then gradually expand across and beyond functional and corporate boundaries. However, even for the pilot, getting it right the first time requires a sound approach that involves the right people and encompasses all of the right things that need to be done at the right time.

Exploration

In the exploration phase, your company can find its bearings in SOA adoption through a hands-on approach that involves a limited number of enterprise services.

Enterprise architects assess the needs of the company's business organizations, addressing the most pressing challenges and identifying the highest payback opportunities for enterprise services. This initial assessment reflects upon what is already available and what gaps in SOA readiness – in terms of technology as well as human resources and skills – need to be closed before the company is ready to develop and deploy its first enterprise services.

A realistically scoped pilot project serves as the proof of concept for the company's SOA approach and its underlying design principles. The proof of concept is typically run by a small team (acting as an SOA incubator in subsequent phases) that makes up the core of the internal SOA community.

Upon completion of the exploration phase, the company's SOA initiative will have delivered a first track record of success through a number of robust enterprise services, instilled trust, secured senior management sponsorship, created SOA awareness across the company, demonstrated how to set the levers for enterprise service reuse, and provided a road map for the further spread of SOA within the company.

Following the exploration phase with a "sandbox" approach as a proof of concept (for example, using the SAP Discovery system), you may arrive at the point where you have to upgrade your system environment to the SOA level. A typical approach would be to upgrade to SAP ERP 6.0, including the latest enhancement package. To facilitate this step, SAP offers an upgrade service that supports the technical upgrade to SAP ERP 6.0 at minimal cost.

Consolidation

Building upon the experience gained in the exploration phase, the consolidation phase aims at spreading and growing your company's SOA adoption in terms of breadth and depth. The objectives shift to use and reuse of enterprise services, increasing organizational learning, and establishing controls and more formalized methodologies. More enterprise services are now being brought onboard and documented for efficient reuse. Transaction communication patterns and interface patterns are defined to ensure the behavioral integrity of these services. Cross-application process integration, such as business-to-business processes, tends to be a key item on the agenda during the consolidation phase. ES Repository, delivered with SAP NetWeaver, begins to take shape.

Enterprise services delivered with enhancement packages for SAP solutions enable companies to quickly and easily build composite applications that support their business processes. In addition, customers are free to adapt these predefined services and identify, design, and model additional services in line with their business process requirements. Published through the design-time ES Repository, SAP-supplied as well as custom-built services are gradually made available to a larger audience within a small number of "pilot" business units. As the number of enterprise services developers and published services increase, the need for life-cycle management, design methodologies, planning, control, and quality management for SOA becomes more evident and must be addressed. Robust design-time governance is needed to ensure that policies are adhered to and pitfalls, such as "Wild West SOA," are avoided. Policies should be established to enable more effective and widespread reuse of enterprise services through the growing ES Repository.

The consolidation phase typically delivers a significant number of enterprise services, drawing on multiple applications leveraged in a small number of business units. This phase provides a solid basis of design-time automation and paves the way for further SOA intensification within and beyond corporate boundaries. Last but not least, it yields valuable experience to be shared with others in the greater SOA community.

Intensification

Having successfully completed its first set of SOA projects, your company's internal SOA community seeks to enhance its professionalism and visibility. Processes are harmonized with growing ease. The evolution of SOA is proactively driven through a formalized approach that involves all stakeholders in senior management, IT, and business units.

The use of enterprise services is gradually expanded to other units. New roles, such as repository keeper, services portfolio manager, and so on, and new tasks are added in pace with the growing scope of SOA adoption. The increasingly formalized SOA competency center provides guidance and expertise, helping to resolve any issues or disputes among stakeholders.

The intensification phase assures the adoption of enterprise-wide life-cycle SOA governance, including automated runtime governance. Service execution is monitored continually by means of specialized tools, ensuring that issues are detected before mission-critical enterprise services are adversely affected. Rapid fault detection, identification, and resolution functions are established and utilized to maximize service availability and performance.

By delivering hundreds of enterprise services for use by multiple business units, the intensification phase provides fully fledged services life-cycle management, including a service design methodology, an enterprise services reuse policy, and the prerequisites for the implementation of a cost-allocation policy.

Drawing on in-depth experience, the company's SOA community is ready to contribute meaningful ideas to external community definition groups and community advisory groups, thus enhancing the pool of SOA best practices beyond corporate boundaries.

Strategic Deployment

SOA adoption consistently and routinely delivers value to business operations. Naming conventions, data types, transaction communication patterns and interface patterns, and the model-driven composition of enterprise services are under governance control. The SOA focus remains on the evolution and broadening adoption of enterprise services aligned with the imperatives of change. Enterprise-wide SOA governance will have reached a high degree of maturity. A governance board is set up to take control of the SOA evolution. A mutually agreed upon cost-allocation policy for enterprise services is in place.

Your company's SOA practices and associated governance are benchmarked against industry best practices. At this stage, the company's SOA community is a fully fledged member of the global SOA community, sharing best practices with other members. An enterprise-wide business process platform is in place to integrate the company and its supply and demand partners along the value chain.

The strategic deployment phase delivers on the ultimate goals of SOA: enhanced business agility, greater cost-effectiveness through services reuse, and significantly reduced integration costs.

Keeping Your SOA Initiative On Track

If the right approach to SOA adoption goes wrong in spite of meticulous planning, what can you do to keep your SOA initiative on track? The following recommendations may help:

- Identify and implement key performance indicators (KPIs) and methods – such as satisfaction surveys for users, customers, or employees – that enable you to continually check the current status of an SOA initiative against defined targets.
- Step back and identify the point at which corrective action is likely to deliver the greatest value. Use KPIs to prioritize remedial efforts.
- Remember that you are not alone. Share your experiences – positive or negative – with other members of the SOA community to create awareness, share best practices, learn from each other, and avoid known pitfalls.

IDENTIFICATION AND USE OF COMPOSITE APPLICATIONS

To achieve operational excellence as well as competitive differentiation, companies need applications that enable them to flexibly build and adapt business processes in pace with changing needs.

Classical business applications have helped companies to significantly improve their operational performance in core disciplines such as supply chain management, accounting, and human resource management. Designed primarily for company-internal use, these applications have enabled companies to achieve consolidation, efficiency, control, and uniformity. However, since classical business applications follow the "built to last" paradigm, these goals are reached at the expense of flexibility. Companies tend to pattern their business processes in line with the functionality these applications provide. This lack of flexibility limits a company's chances of winning the race for competitive differentiation. Even in terms of operational improvement, these applications have largely reached their saturation level.

Innovating with Traditional and Composite Applications

So what are the options? For the slow-moving processes in your company's business, traditional business applications are and will remain the right choice. However, for the fast-moving lines of business that depend and thrive on agility, service-based solutions are hard to beat. For instance, if your company faces frequent changes in its distribution channels or wants to pioneer an innovative line of business (such as a novel offering in the telecommunications sector), a service-based solution delivers the required degree of responsiveness to change.

Both traditional and composite applications can be used side by side, allowing your company to innovate different lines of business at different speeds.

Business model innovation is often more important than traditional product innovation. While customers don't buy individual products per se but have come to expect complete solutions or even entirely new experiences, companies realize they cannot do everything themselves and must engage in different types of business networks.

Faced with the speed and scope of global change, companies need software solutions that are architected and built to adapt. This paradigm shift (see Figure 3) is at the heart of SOA, enabling companies to achieve operational excellence as well as competitive advantage. Classical business applications do not encourage the establishment of novel or flexible business processes customers need to gain competitive advantage. In addition, classical applications are self-contained and assume that all data and all business logic are owned by the application.

From a customer's and end user's perspective, there are requirements that classical business applications cannot meet, such as:

- A single intelligent user experience across system and enterprise boundaries
- The ability to cross functional, system, and company boundaries
- The presentation of the entire enterprise or its extended value chain – including customers, suppliers, and channel partners – as a whole rather than in its separate parts
- Processwide as well as enterprise-wide system and human collaboration
- The ability to assemble additional applications from existing multisource content
- Adaptability in line with business needs identified by business analysts

Only when these requirements are met can companies experience the levels of agility, innovation, insight, collaboration, co-innovation, and diversity they need to stay ahead in the face of global competition.

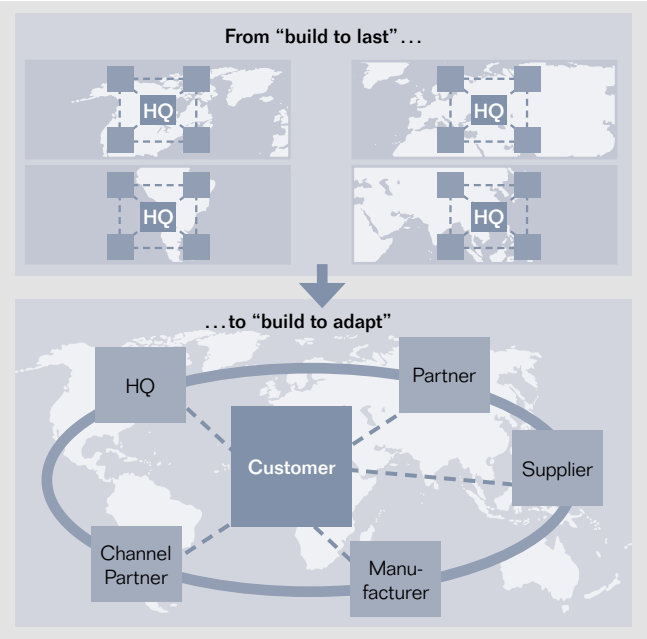

Figure 3: The Shift to Business Solutions “Built to Adapt”

Leveraging Enterprise Services for Composite Applications

SOA adds enterprise services as new software components to an enterprise application environment. For example, an enterprise service can be set up to retrieve information from a business application or to translate data between different application-specific formats. Enterprise services operate in real time, cutting out the latency in classical applications that impairs the performance of business processes. An enterprise service can be simple or complex and can be reused. Multiple enterprise services can be combined or concatenated to form a more complex service.

Enterprise services enable a company to move from a set of static business applications to a flexible composite application that helps it achieve its goals.

The business networking opportunities offered by service-enabled applications are compelling. For example, imagine two airlines whose customers' experiences suffer from occasional overbooking and whose accounts suffer from costly excess capacity at other times. Instead of opting for a costly merger or a hostile takeover (which may make the "winner" unpopular with the "loser's" customers), both airlines can strengthen their brands by flexibly sharing capacities when required. Let's assume that each airline has invested heavily in a flight booking system that is incompatible with that of its potential partner. Naturally, both airlines refuse to scrap their flight booking system, citing the imperatives of investment protection and the cost of training or perhaps even local language support.

And that's exactly where SOA and its enterprise services come in. Instead of requiring the rollout of a completely new flight booking application, the only step these two airlines have to take is service-enabling their existing application environments. Enterprise services can then be leveraged to integrate both flight-booking applications via a composite application that delivers a "single version of the truth."

What Is a Composite Application?

A composite application is defined as a packaged application that sits on top of other applications and reuses their functionality through service calls. As shown in Figure 4, a composite application makes up the top layer in the SOA application environment. This layer interacts via enterprise services with the underlying application layers. A key element of SOA within a company or its business network, a composite application is typically presented to users via a browser-based interface.

Enterprise services provide ease of integration. In a composite application, business process steps represented by enterprise services in ES Repository can be put together quickly and easily to build an application that enables a business process. When different or additional capabilities are needed, customized applications and enterprise services can be added or modified to support these capabilities.

When companies heavily modify their classical business applications to create or support a new process, they inevitably cut themselves off from the update path for these applications. One of the advantages of a composite application is that updates in the underlying application pool don't require any change to the composite application itself. Enterprise services simply hook into the current version of each application to deliver up-to-date functionality.

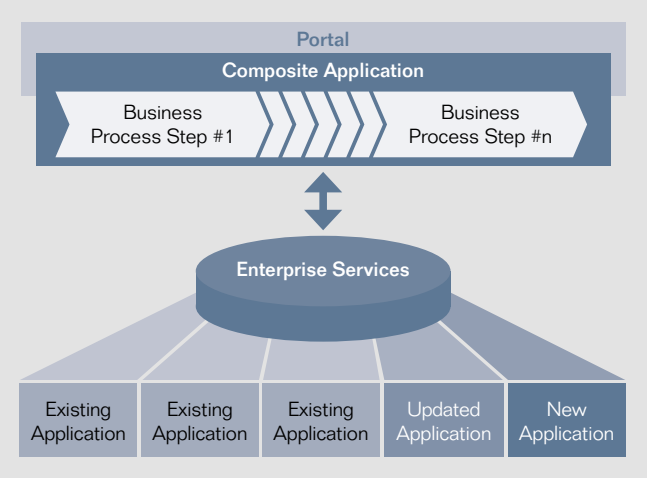

Figure 4: Composite Application in an Application Environment

And what happens to those nifty niche applications a company doesn't want to give up because they provide useful, sometimes even critical, functionality? Via enterprise services, they are simply integrated with the composite application for needed functionality as well as investment protection.

Key Characteristics of Composite Applications

Composite applications share the following characteristics:

- Composite applications draw on services provided by other applications or components. They combine available service operations with new application logic, user interfaces, and business process orchestration. These services can be provided by SOA-enabled applications from SAP Business Suite, enterprise services bundles, or service-enabled third-party solutions that provide self-contained enterprise functionality accessible via the Web services standard.
- Composite applications are process-centric, supporting business processes that span one or more underlying applications or components that are often data-centric. These composite applications may also include steps from third-party or customer-proprietary software components.
- Composite applications are user oriented. They enable workflow consistency and support cross-functional as well as cross-system collaboration. They commonly provide a user interface but they may automate process steps, thereby eliminating the need for user interaction.
- Composite applications are integrated with back-end applications via enterprise services. These enterprise services provide the connectivity that brings disparate applications together and enables the consolidation of data at the user interface level.

- Composite applications often need to abstract from the components on which they are based in order to run in heterogeneous landscapes as well as with multiple releases of the same back-end applications without customers having to upgrade the main application. A particular requirement is that composite applications can cope with the existing configurations of the underlying components.
- Composite applications require the service enablement of the underlying applications or components. However, the applications on which composite applications are built often do not have interfaces that composite applications can plug into. As a result, additional development as well as shipment and delivery processes are required to add those interfaces to the underlying components or applications.
- Composite applications have their own life cycle. They can be built, packaged, deployed, and upgraded independently of the underlying applications by SAP or ISVs. Consequently, composite applications establish a new logical application tier. This type of loose coupling also requires that there be no shared volatile states and no transactional locks between the composite application and its underlying components, thus imposing the requirement of a stateless service-oriented communication protocol.
- Composite applications rely on a very small number of standard protocols – or, ideally, on a single standard protocol – rather than on a multitude of proprietary ones. The idea is to reduce complexity by using a target protocol that does not overburden the composite application with protocol mappings.

Composition Approach

SOA is the open IT architecture that facilitates integration and collaboration for business agility while ensuring the reuse of existing assets for cost-effectiveness. As a key benefit delivered by SOA, business semantics for building composite applications combine enterprise services, which may run on different back-end applications, with business processes. In the airline example mentioned above, this implies that both parties have "implicit knowledge" about the business details of the seat-sharing transaction. Thus a booking reference number can be translated from one system to the other and passenger records can be exchanged.

While standard cross-industry applications fulfill most standardized business requirements, the composition approach is superior in scope – particularly when it comes to building applications for a specific industry sector, country, or region. The ability to penetrate highly vertical markets by providing an architecture with which ISVs can build composite applications is also a key driver for composite applications.

Bringing It All Together: The Integration Platform

No company starts its SOA initiative completely from scratch. What companies need is a platform that ties in with their existing IT infrastructure and that integrates their existing suite of solutions, composite applications, partner solutions, and custom-built applications with their newly developed enterprise services portfolio. SAP NetWeaver is a prime example of such an integration platform.

An integral part of SAP NetWeaver, SAP NetWeaver CE is dedicated to building and hosting composite applications. A composite application running on SAP NetWeaver CE invokes its underlying services from a different back-end application remotely. That composite application can be conveniently integrated with and published via a portal. Alternatively, the composite application can be built without a user interface (and thus without a portal); for example, in application-to-application (A2A) and business-to-business (B2B) scenarios.

Choosing the Right Composition Environment

Composite applications empower companies to drive innovative business processes and obtain more business value from their IT investments through the seamless integration of existing functionality and assets. ES Workplace enables business process experts and IT solution architects to collaborate on defining and creating composite applications. For example, business process experts and solution architects can use ES Workplace to search for existing enterprise services that support specific business process steps. ES Workplace supports browsing through services alphabetically and searching by general business context, industry, or solution map. This functionality helps business experts define the steps of a business process in a way that is clearly understood by solution architects who, in turn, are in charge of finding the right technology to implement the appropriate composite application that can enable the process steps.

Composite application flexibility is not only about defining and using components but also about using a platform that gives software vendors and customers the required flexibility to create their own solutions that are independent of SAP release cycles and offerings.

SAP NetWeaver CE provides integrated support for composite applications and the processes these applications enable. It helps developers and business process experts increase their productivity through a standards-based environment that:

- Allows the IT organization to create composite applications faster and at a significantly lower cost
- Accelerates business process composition
- Increases developer productivity through enhanced and integrated tools
- Lowers the SOA entrance barrier through a fast and convenient download with installation timelines and automatic updates
- Reduces the cost of development, training, and support
- Enables the porting of Java applications to the latest Java platforms

As part of SAP NetWeaver CE, the SAP Composite Application Framework (SAP CAF) tool provides the functionality for creating business object models and for generating the service operations required for composite applications.

The Value of Composite Applications

Composite applications help companies achieve their flexibility, productivity, and cost-effectiveness goals by enabling them to efficiently add, replace, remove, or rearrange steps within their business processes. By providing a process context and an overall process view that intelligently crosses system boundaries, composite applications facilitate the implementation of collaborative scenarios along the value chain. At the same time, composite applications provide flexibility and a fair degree of independence from underlying applications.

Composite applications give companies the flexibility they need to better meet their business objectives.

IT organizations face the challenges of consolidating and innovating at the same time. Composite applications enable the IT organization to master both challenges. Even when a legacy application used by a composite application is switched off, existing business processes remain enabled within the context of the composite application. Enterprise services can be used to tap into whatever alternative application is provided. This approach significantly reduces the effort and complexities of innovating in a nonconsolidated IT landscape.

Business Value

Composite applications are inherently designed for the adaptation or creation of business processes, enabling companies to achieve competitive differentiation by engaging in collaboration and by responding quickly and effectively to new situations – including unplanned or unforeseen events. Composite applications enable users to participate in automated business processes that help them instantly detect and handle exceptional situations. These applications are rendered directly into the user's work context to enhance process efficiency and productivity. When multiple applications fed by different applications are used, the composite application unifies all of these underlying applications so that the end user does not experience any boundaries, disconnects, or discrepancies among them.

IT Value

The composition approach enables the IT side to keep core enterprise resource planning (ERP) applications intact while innovating on top of them. Composite applications draw on enterprise services that are decoupled from specific physical servers and that can be reassigned dynamically to the resources – including servers,

Composite applications provide innovation without disruption. Core applications, such as ERP, remain fully operational during the development of composite applications.

networks, and storage – that provide the best fit for real-time business and operational requirements. In the development arena, the business semantics defined by SOA provide a high degree of consistency, helping to significantly reduce the need for data-structure mapping and improve development productivity.

Time to Value

A key part of the value proposition of composite applications is the reuse and combination of existing assets of the customer's system landscape for the implementation of new business processes. To achieve this reuse, composite applications have to integrate into the existing landscape without requiring either a major upgrade or a modification of the existing back-end applications; that is, the composite application is not invasive. The reuse of existing functionality contributes significantly to the goal of short time to value because existing functionality doesn't need to be implemented again. Shorter time to value means that companies can go after windows of opportunity that might be closed to them with traditional approaches.

Time to value is reduced further by the accelerated pace at which composite applications are delivered for use in the production environment. This acceleration is achieved through simple configuration, easy extension, a straightforward installation process, and seamless integration with the existing solution landscape – without the need for additional hardware or software instances – helping companies keep TCO in check.

Composite Application Proof Points

The following examples illustrate the value that composite applications can deliver to companies faced with massive change.

Power Company

Environmental legislation in most industrialized countries calls for an increase in the production of renewable energy. For most major power companies, green energy makes up a considerable proportion of the "fast-moving" part of the business. The decentralized nature of facilities that produce green energy, compounded by natural fluctuations in power output, make it difficult for a power company to gather, validate, and deliver power-capacity data. Reasonably accurate capacity forecasts pose an even greater challenge. And then there are changes in government policies that result in changes to subsidies and tax breaks granted to producers of green energy.

That's exactly where a composite application comes in. While the "slow-moving" mainstream power business is supported efficiently by standard business applications, enterprise services that feed into a composite application enable a power company to reliably gather green-energy data from disparate sources and to embrace change in the fast-moving green-energy part of its business.

Challenge: A power company needs timely information on its current and anticipated future capacities of green energy – a task that is extremely difficult to fulfill with error-prone manual and semiautomatic methods. Data sources typically include SAP solutions as well as a number of non-SAP applications plus nonstandard databases run by the company's smaller suppliers of green energy. They may even include arrays of solar panels operated by individuals who merely have a separate electricity meter that relays data to the power company in real time. Collecting complex data from disparate sources such as these and then aggregating and validating all relevant data requires a tremendous amount of effort plus time-consuming communication.

Objectives: Green energy is popular; a growing number of customers are prepared to pay a premium for "clean" power. That's why a power company wants to keep customer-facing information up-to-date at short refresh intervals without tying up scarce resources. To sharpen its competitive edge in a highly competitive and sometimes volatile market, a power company wants to deploy a solution designed to ensure the quality and currency of data; to automate the tasks of data gathering, aggregation, validation, and presentation; and to ensure sustained regulatory compliance.

Solution: Such a power company would need to develop an automated solution designed to address these issues quickly, reliably, and cost-effectively. SAP and partners are willing and able to help the company build a composite application that leverages SAP NetWeaver to integrate the underlying SAP and non-SAP applications and databases. Drawing on enterprise services supplied by SAP plus customized interfaces required to feed validated power-capacity data from all of the disparate green-energy sources, this composite application would enable a power company to publish its regular- and green-power capacities via its portal.

Benefits: This composite application would enable a power company to resolve a number of data aggregation and data quality issues, automatically providing reliable and timely power-capacity information in compliance with pertinent regulatory requirements. The ability to apply the same pattern by reusing proven enterprise services would make it easy for the power company to benefit from windows of opportunity by integrating additional green-power capacities with its composite application. This flexible approach is more than likely to pay dividends: a power company could easily partner with other suppliers – or commercial buyers – of green energy to build economies of scale.

Mobile Phone Company

Let's take a closer look at a company that suffers from disparate approaches to approval workflow for financial and nonfinancial transactions. This could be any company in any number of industries, such as insurance, professional services, banking, or high tech. For our example, we've picked a mobile phone company. What does this company need to make its approval processes more efficient while reducing operating costs and outlays for implementation and training?

Challenge: Dispersed data, disaggregated functions, protracted tasks, and a multitude of user interfaces make it cumbersome for employees to apply for approvals when they purchase anything from office supplies to hotel accommodations. A complex layer of processes is supported by the company's legacy solution, which is difficult for most employees to use. In addition, the company is reluctant to make any modifications to the legacy solution; the software is coded in a proprietary language, making it difficult to implement changes and tying up expensive resources. To overcome these challenges, the company needs a solution that supports employees with an easy-to-use approval workflow that can be modified quickly should the company's policies or workflow change.

Objectives: A common approval solution would provide support for the end-to-end approval process, from initiation and validation to approval and payment. Key requirements of the solution include a standard look and feel as well as the consistent application of policies and business rules. The solution also has to meet the needs of different audiences. The company wants to see time and cost reductions. The IT department is looking for a solution that can be adapted quickly, easily, and cost-effectively. Employees want something that's easy to use.

Solution: The approval solution could be implemented as a composite application built on SAP NetWeaver. Reusable and easily adaptable enterprise services could enable the composite application to draw on multiple back-end applications across a number of different technology platforms. The solution could be available to employees via the SAP NetWeaver Portal component. Role-based access determines the data, interface, and functions that a user sees. For example, a financial administrator can set up a vendor and create a financial posting for payment. Workflow rules automatically determine when a requisition has to be approved and then forwards the requisition to the appropriate manager.

Benefits: The composite application approach enables the company to consolidate a number of transactions across a range of systems into a single portal-based solution that's easy for employees to learn and use. Paper-based steps are almost completely eliminated, needed only to resolve exceptional situations. Built-in business rules and user-friendly interfaces enable the company to save money on employee training and eliminate errors associated with manual data entry. Reports provide managers with real-time information for informed decisions. Such an approval solution would meet any company's requirements for usability, maintainability, scalability, and adaptability.

SOA ADOPTION LIFE CYCLE

SAP provides the knowledge, services, and tools your company needs to design and execute an SOA strategy that meets the company's business goals – with appropriate governance and quality measures built in.

SOA describes the strategy of enabling business processes by means of services and of reusing these services within the enterprise. One of the first steps in your SOA journey is to identify business processes that can be easily and accurately enabled and optimized via such services to deliver tangible benefits and measurable returns.

You need sound governance practices in place to maintain your portfolio of services in a services repository. You need to ensure effective reuse of these services and to gradually expand these services so that additional business processes can be optimized. Successively implemented, these process improvements are part of a long-term strategy rather than a one-time solution. That's why the adoption life cycle plays a key role in SOA success; it provides the "big picture" through a road map that enables all stakeholders – including senior management – to understand, assess, and make decisions regarding an SOA transformation project.

Make or Buy – or Both?

The concept of SOA and the underlying platform put customers in a position to "make and buy," enabling them to better tailor solutions to their specific needs. This combines the benefits of off-the-shelf software, such as support, maintenance, standards, and integration, with the flexibility to meet highly specific business needs essential for core differentiation. However, companies that want to reap the benefits of SOA in a predictable time frame need to be prepared to invest in SOA technologies and software products. Proven solutions as well as guidance and support by a trusted advisor can help companies mitigate risks and accelerate the time to value of their SOA initiatives.

SAP has enabled SOA in its portfolio of solutions in order to provide its customers with "one-stop shopping" for SOA. SAP has enhanced the SOA concept by adding business semantics delivered through enterprise services. To save customers the cost and effort of building their own services "from scratch," SAP supplies enterprise service bundles via enhancement packages for SAP solutions. These industry-focused bundles enable companies to conveniently select the services they need to deploy in support of the applicable business processes.

SAP's approach to SOA extends the scope of SAP Business Suite applications by offering customization options that go far beyond the scope of traditional parameter-oriented customizing. The composition approach based on productized enterprise services from SAP is a very cost-effective way of leveraging the advantages of SOA-enabled applications in an evolutionary manner. Composite applications open up a new dimension of flexibility for building tailored solutions while simultaneously leveraging the benefits of standardized business applications.

What Roles Are Involved in an SOA Project?

Considering the number of different views and individuals involved, it is paramount to agree on a common plan and path to achieve the goals of your company's SOA initiative. Such a common denominator can typically be provided in the form of a framework or methodology.

Depending upon your company's specific needs along its SOA road map, a number of SOA-centric roles exist within cross-functional teams. These roles can be filled by existing employees or by adding an "SOA flavor" to existing roles.

The transformation to SOA does not imply that each and every business process of a company is optimized by a service-enabled solution, such as a composite application. It means that there is a business process platform in place that offers the company the **option** of addressing business requirements via service-enabled solutions and complementing SOA-enabled SAP® Business Suite applications.

For instance, the planning and building of service-enabled applications typically involve the following SOA-centric roles:

- **Business process experts** have strong business knowledge and establish the connection between business and IT by delivering requirements.
- **Application experts** know the functional domains and the existing services in their areas.
- **Security and compliance experts** define the boundaries of security policy domains for user identities, access control to SOA resources, and regulatory compliance.
- **Technical architects** convert business and application expertise into a concrete architecture.
- **Technical experts** take care of process, people, and information integration and provide the required landscape and technology.
- **Composition experts** implement the actual application.
- **Software developers** provide additional software functions.

A Methodology for Accelerating the Adoption of SOA

Within the framework of SAP's approach to SOA, the SAP Services organization has developed a methodology designed to accelerate time to value in SOA initiatives. The methodology for accelerating the transformation to SOA provides a road map for SOA adoption in the context of SAP products and technologies. This methodology focuses on an evolutionary SOA adoption approach plus related SAP products, services, and concepts. It is intended for customers that are driving, planning, or executing an evolution to SOA; in other words, an SOA transformation.

This methodology distinguishes between the following fundamental adoption phases within the overall transition to an SOA-based landscape (see Figure 5):

- Strategy and governance
- Landscape development and operations
- Business solution planning and transformation
- Building and implementation of a solution design
- Solution management

The first two adoption phases are not related to the planning and building of a specific solution. Rather, these phases are related to the overall IT landscape of a company. The implementation of service-enabled solutions in the last three phases is a key building block within the overall methodology.

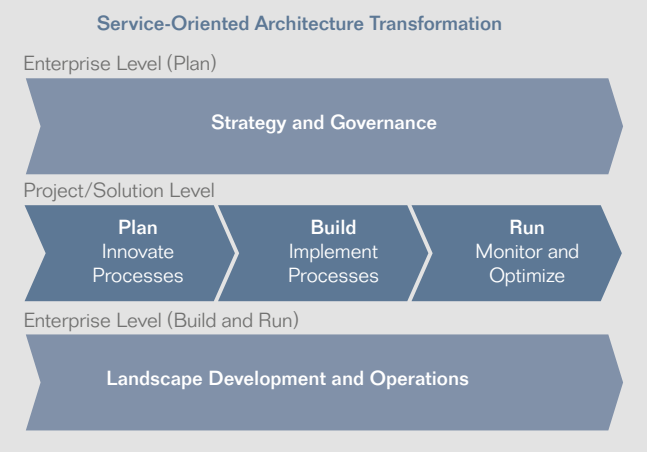

Figure 5: Transformation to Service-Oriented Architecture

On the service provisioning side, this methodology enables companies to leverage enterprise services supplied by SAP for building a solution tailored to their specific needs as well as adapting SAP-provided enterprise services and building customized enterprise services. On the service consumption side, this methodology supports an evolutionary approach to SOA-enabled solutions. This evolutionary approach implies that core "platform" processes can draw on business applications, such as SAP Business Suite applications, while, in parallel, companies are enabling innovative processes via their own service-enabled solutions, such as composite applications.

When using this methodology, your company may choose to perform activities in parallel, set altered focal points, or choose distinct entry points in the form of different adoption patterns. For example, you may already have established the required foundation and may have started with the implementation of SOA-enabled solutions to show the value of SOA. On the other hand, your company might not feel the immediate need to invest in long-term enterprise architecture planning and could start by setting up its own business process platform.

The following sections describe the phases of this methodology in more detail.

Strategy and Governance

At the enterprise level, strategic planning serves to define and assess the overall SOA strategy of your company, prepare the organization for the transformation to SOA, and define a high-level and enterprise-wide reference and target architecture (enterprise architecture). At the same time, a customized governance framework needs to be established that will manage the utilization of the business process platform on which the company's SOA initiative is founded.

The resulting enterprise architecture includes a description of the steps, standards, and guidelines required to take the processes and IT infrastructure of the company from the current condition to a future state that best supports the company's business strategy. SAP has defined its own enterprise architecture initiative and has created a framework to support enterprise architects from a methodology point of view.

The SAP Enterprise Architecture Framework methodology for SOA includes accelerators, guidelines, checklists, and examples for use by SAP consultants, SAP partners, and SAP customers during the execution of enterprise architecture engagements. It is part of an overall SOA transformation methodology.

SAP Enterprise Architecture Framework incorporates The Open Group Architecture Framework (TOGAF) standards plus a series of extensions designed specifically to support the implementation of SAP solutions and to complement current SAP practices. TOGAF comprises a detailed method plus a set of supporting tools for developing an enterprise architecture.

Another strategic building block for the transformation to SOA is a customized governance framework encompassing all of the right things that need to be done at the right time by the right people. Even in pilot projects, getting SOA right the first time requires a sound governance framework that supports the reusability of enterprise services and safeguards the provisioning of semantically harmonized service interfaces across the system landscape. The governance framework spans design-time governance (developing enterprise services and service-consuming applications) as well as runtime- and operations-oriented governance, such as the monitoring of service consumption.

Effective SOA governance is built upon the following elements, as depicted in Figure 6:

- Governance management, including the organizational structures, skill sets, and procedures that are aligned with the specific needs of the company
- Automation and life-cycle management required to support good governance and to achieve the ultimate goal of automated SOA governance
- A design and modeling methodology that spans all phases of service design, harmonization, and implementation
- Community building to support the continuous sharing of ideas and best practices, inside and outside the company, for faster time to value and sustained success

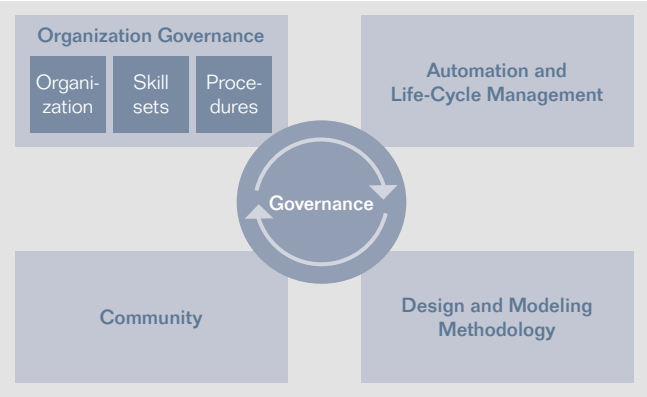

Figure 6: The Elements of Governance

Companies are advised to focus on areas where service-oriented architecture and highly standardized interfaces deliver the greatest return – that is, where processes span functions, applications, and enterprise boundaries.

Landscape Development and Operations

SOA is changing the way companies think about utilizing and upgrading their IT infrastructure. This is because SOA can be implemented incrementally – implying the end of costly "rip and replace" infrastructure overhauls. For example, numerous enterprise IT managers, as well as enterprise architects, business process managers, IT development teams, consultants, and system integrators, are putting in place development "sandboxes" in order to gain familiarity with the creation of composite applications. Using the sandbox approach as a proof of concept, companies can learn how to consume enterprise services and build composite applications to map their business processes at their own pace – without disrupting their business environment. This approach helps to answer the "make or buy" question by delivering insight into which services are available and ready to run, that is, supplied via predefined enterprise services; which services can be adapted to the company's specific needs; and which need to be developed from scratch.

Once you are confident that your SOA initiative is headed in the right direction, your company is free to progress gradually to fully fledged SOA for its business processes. From a landscape development perspective, the planning and implementation of a business process platform is a key milestone of an overall SOA transformation.

A business process platform serves as the foundation for an SOA-based system landscape. A business process platform is the unified environment that companies implement to perform their core business processes efficiently and to reorganize, extend, and create new business processes flexibly and at a predictable cost across the IT landscape. The overall design of a customer's business process platform is derived from the strategy and landscape development phases. While a customer will continue to use an

existing ERP solution for traditional tasks, such as inventory management or invoicing, SOA will be applied where the principles of SOA deliver value; for example, in a business environment that is subject to frequent changes in supply chain partnerships. It's up to the customer to draw the line – the technology supplied by SAP spans both traditional as well as composite (service-based) applications.

SAP helps organizations establish their unique business process platform by delivering ready-to-execute software for business processes, reusable enterprise services that enable business process steps, and the technology for composing and deploying software that enables flexible business processes.

As shown in Figure 7, the key elements of a business process platform enabled by SAP are:

- SAP NetWeaver, including SAP NetWeaver CE, as the technology, composition, and integration platform
- SOA-enabled applications from SAP Business Suite
- Enterprise services supplied with enhancement packages for SAP solutions; used for accessing business objects of process components

The integration functionality of SAP NetWeaver can also be used to integrate non-SAP applications and SOA infrastructures.

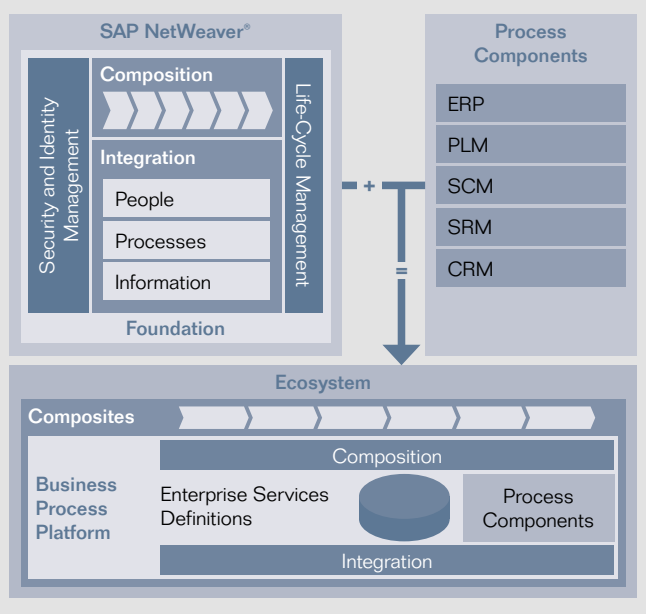

Figure 7: The Elements of a Business Process Platform

The SOA adoption methodology discussed in this section utilizes proven best practices that facilitate and accelerate the introduction of a business process platform. For companies that want to benefit from SOA-enabled SAP Business Suite applications and enterprise services supplied with enhancement packages, SAP offers a special "upgrade factory" that supports the technical upgrade to SAP ERP 6.0 at minimal cost.

Besides describing the fundamental building blocks of a business process platform and the way to implement one, the methodology also outlines reference architectures for typical use cases of SOA-enabled solutions and their relevance in the overall architectural planning process.

Operational aspects are covered by the Run SAP methodology road map referenced in the SOA transformation methodology, which is discussed in the "Solution Management" section.

Business Solution Planning and Transformation

With your strategic and landscape foundation in place, you can start to leverage SOA by implementing SOA-enabled solutions in your business process platform. Today, organizations face a growing need for software that supports the execution of frequently changing processes and leverages stable process execution in packaged business applications. During the solution planning phase, business process management helps to innovate business processes and to translate conceptual business process models into SOA-enabled solutions (see Figure 8). The combination of business process management and SOA is an ideal way to face this growing demand for flexibility. This is exactly the point where SOA helps organizations to align IT with the business.

The methodology described in this section is included in the SAP Business Process Management service developed by SAP Services. This methodology addresses the planning phase of a customer's SAP solution project, with the focus on business solution planning and transformation. The aim is the improvement of customer business processes based on industry-specific SAP content and the mapping to the corresponding implementation options within the context of initial architectural guidance. This methodology adds value by simultaneously addressing business process and architectural topics, also known as the "tandem approach" of process and solution architecture consulting. It replaces the sequential way of addressing business and IT issues.

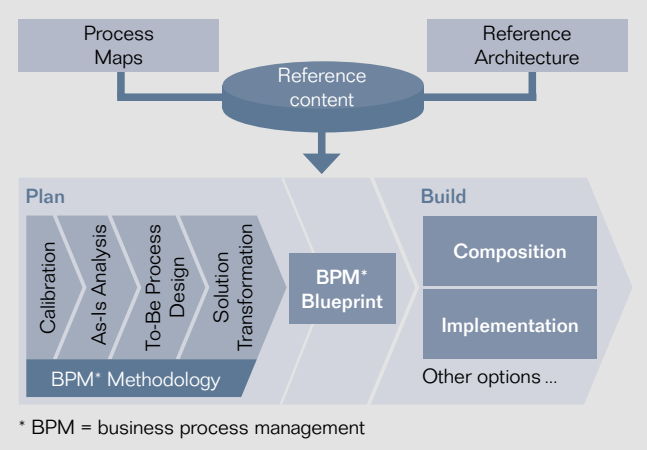

Figure 8: Plan and Build Phases of the Transformation to Service-Oriented Architecture

Building and Implementation of a Solution Design

The purpose of this phase is to build solutions running in a business process platform and to implement the enterprise services needed to support your company's business solutions and processes. SOA-enabled solutions help your company set up customized processes using highly standardized interfaces while leveraging its existing investment in packaged business applications. The SAP Consulting and SAP Custom Development organizations offer a range of services for the planning, design, and implementation of customer-specific solutions based on SOA standards. Additionally, SAP Consulting and SAP Custom Development offer deployment services for industry-specific scenarios, such as order-status tracking for the wholesale industry. SAP Consulting offers a service based on the SAP Discovery system for SOA that includes the development of a lightweight composite application.

A development handbook from SAP (see the "Additional Information" section) provides valuable guidance during this phase. Reflecting customer experience and feedback, this handbook describes the end-to-end development process in an SOA environment at an intermediate level.

Solution Management

The purpose of this phase is to run and optimize an SOA-enabled system landscape in the most efficient and effective manner. SAP provides a specialized methodology – Run SAP – for running SAP solutions. This methodology includes specific aspects of SOA and running SOA-enabled solutions.

Building a Services Portfolio

In addition to drawing on the predefined enterprise services supplied with SAP enhancement packages, customers are free to design and develop their own enterprise services. SAP Consulting and SAP partners can provide guidance and hands-on involvement during the development phase.

Business-driven SOA is based on a clear understanding of business strategy and objectives. Business processes need to be defined and engineered on the basis of this understanding. This approach determines which enterprise services are appropriate to a particular business process, which services are ready for reuse, which enterprise services need to be developed, and the appropriate granularity of an enterprise service ("service cut").

Ideally, the enterprise services design and modeling methodology will integrate business processes and business content while also harmonizing Web service technology with business semantics. It takes good design-time governance to ensure that the tools and procedures used by this methodology adhere to architectural

Use the enterprise services builder function in ES Repository to model, define, and manage your objects. You first need to create your models in order to determine which design objects you require for your application and then define these objects in the repository.

guidelines and standards. The design and modeling methodology needs to ensure that new enterprise services are consistently designed, developed, modified, and deployed in conformance with enterprise-wide standards. This is critical for SOA success. The methodology incorporates formal review cycles with clearly defined milestones.

Typical stages of enterprise service design are:

- Identification of service granularity. As business processes and business objects are disaggregated, this step identifies the granularity of an enterprise service. Architectural principles, naming rules, and the need for harmonization help to ensure a consistent cut.
- Initial design. In this step the raw design for the service is defined so that it maps to the business object model as well as to the interface and communication patterns in the SOA architecture.
- Definition of global data types. The enterprise service interface is defined based on standard global data types (GDTs are the basic building blocks of data). Specific GDTs are selected for use in the service interface. This step assures cross-enterprise harmonization of data in enterprise services.
- Final design. The design and interface definition, including element structure and all details, are finalized.
- Insertion in ES Repository. The new enterprise service is added to ES Repository where the service is precisely defined and documented, complete with metadata and other information, such as usage-related keywords and location of the reusable service.

SAP Services provides guidance to help companies close the gap between business and IT strategies.

The "design by contract" principle must be applied in the design and implementation of services. Essentially, this means services need to be designed and implemented as documented so that the service consumer can rely on the runtime behavior of the service as agreed in the contract between the service provider and service consumer. This avoids problems at the consumer side resulting from undocumented but "supported" behavior that implicitly becomes part of the contract between the service provider and service consumer.

Resources for Accelerating the Adoption of SOA

SAP has service offerings as well as methodologies, documentation, tools, and software that can help you speed up the adoption of SOA at your company.

SAP Consulting Services

The SAP Consulting organization provides service offerings that support the full adoption cycle. These services include a comprehensive set of accelerators, such as content, tools, and templates, that support the execution of an SOA road map at your company. Tuned to the specific needs of the customer, this road map covers all information needed for the transformation of your IT and business landscape to SOA.

Starter Kit – SOA Adoption Made Easy

Available free of charge to SAP customers, a starter kit for the adoption of SOA contains all relevant information (including the methodology described above) for supporting customers during their adoption of SOA. This kit delivers knowledge building blocks plus content focused on value and benefits, architecture, governance,

Begin with small and simple steps that deliver fast results. Achieve some quick wins by leveraging your existing portfolio of applications and IT projects.

composition of services, and ongoing operations. In addition, the kit illustrates a proven SOA adoption methodology for project managers as well as hands-on developer guidance. The SOA customer cases and references included in the kit enable you to benefit from the experiences of others.

SAP Discovery System

Another useful resource for SOA is the SAP Discovery system. Offered by SAP in collaboration with selected hardware partners, this system is designed to help customers and partners map their path to the successful adoption of SOA. The system can be used as a "sandbox" for initiating a business process platform, enabling you to take your first steps towards SOA as a proof of concept. SAP Consulting offers a workshop that complements the SAP Discovery system for SOA. Conducted as a one-day interactive session by SAP enterprise architects, SOA architects, and other subject matter experts from SAP, this workshop provides guidance in the areas of general SOA principles and practices as well as in SAP-specific processes and SOA-enabled solutions. The workshop includes a live demonstration of SOA supported by SAP technology utilizing the SAP Discovery system.

END-TO-END DEVELOPMENT PROCESS IN AN SOA ENVIRONMENT

This chapter is an extract from *Enterprise SOA Development Handbook* available from SAP (see the "Additional Information" section). While this chapter offers an overview of the end-to-end development process in an SOA environment, the handbook provides more detailed information and in-depth development guidance.

This chapter takes a closer look at the technology and tools involved in the end-to-end development process in an SOA environment. As such, it addresses the interests of IT architects, strategists, developers, and operations personnel.

The first section focuses on life-cycle management aspects. The next section explains how a process is composed. This is followed by a discussion of service provisioning and service consumption. The last section deals with the representation of services through an enterprise service bus.

Life-Cycle Management

This section presents the building blocks of an SOA development landscape, the development infrastructure, the creation of custom and partner applications, and the transportation and productization of applications developed in an SOA environment.

Initial Considerations

When planning the development landscape for their SOA initiative, companies need to address a number of issues. For instance, companies need to distinguish between the provisioning and the consumption of services. In the programming arena, the ABAP™ programming language and Java differ in terms of development functionality. Standard business applications deployed at the customer's site handle the bulk of the workload, while custom developments serve to enhance and extend the functionality of these standard applications in order to meet a customer's business requirements. This has implications for the transport landscape. The team size, the number of development objects, and numerous other factors impact the setup of the development infrastructure. And, last but not least, security issues in the development and production landscapes need to be addressed.

SAP Tools and Technology

The following building blocks are characteristic of an SOA landscape.

SAP NetWeaver Composition Environment. SAP NetWeaver CE makes up the SOA service-consumption layer. A variety of different functional components can be added to the SAP NetWeaver Application Server component for Java. The SAP NetWeaver Developer Studio tool is also included for Java application development.

SAP NetWeaver Process Integration. The SAP NetWeaver Process Integration (SAP NetWeaver PI) offering provides, among other things, the functionality of an enterprise service bus, which facilitates SOA-enablement of legacy applications, cross-component business process management, asynchronous service invocation, and service orchestration. Refer to the "Enterprise Service Bus" section for details.

Enterprise Services Repository. ES Repository is the place where the service provisioning process starts. ES Repository can be deployed together with SAP NetWeaver PI or SAP NetWeaver CE.

Services registry. A services registry serves as a directory for all services available in a system landscape. Service providers publish their services to the services registry once the services are deployed and ready for use. Service-consumer applications retrieve service definitions from the registry and construct client coding accordingly.

The following building blocks are mandatory not only in an SOA development landscape but also in other scenarios.

SAP Solution Manager application management solution. This building block integrates and extends a comprehensive tool set needed for the efficient implementation, administration, and operation of SAP solutions. You need only one SAP Solution Manager in your landscape. Unless you already have an instance of SAP Solution Manager in place, you should set one up along with your SOA landscape.

System landscape directory. A system landscape directory (SLD) serves as the central provider of system landscape–related information for many SAP applications. In an SOA development landscape, ES Repository retrieves information about software products and components from an SLD. Therefore, if you do not already maintain an SLD in your landscape, you have to activate one along with ES Repository.

The following building blocks are optional and serve to meet customer-specific requirements and provide valuable support in an SOA development or productive landscape.

Development infrastructure. A development infrastructure helps to improve the coordination of team development. The development infrastructure for SAP NetWeaver supports Java SOA application development; this includes an integrated development environment (IDE) that supports applications developed with Java EE and SAP CAF as well as server-based application development using the SAP NetWeaver Visual Composer tool. For ABAP-based development, a transport management system should be used. With an enhanced central transport service (CTS+), both ABAP and Java developments can be handled synchronously.

If you want to test-drive SOA without touching your productive landscape, SAP has two options for you: Enterprise Services Workplace and the SAP Discovery system.

Back-end applications. SAP and non-SAP applications can be used as back-end applications as long as they are SOA-enabled. These applications are indispensable for the provisioning of enterprise services.

Enterprise portal. In most cases it makes sense to run composite applications in an enterprise portal. Since SAP NetWeaver Application Server for Java comes with an enterprise portal runtime by default, there is no need to maintain an additional enterprise portal in the development landscape.

All building blocks mentioned above are functional components. It is not necessary to run each of them in a separate system. Depending on computing resources available, you can flexibly combine many functional components in a single system.

Custom and Partner Applications

Software development in an SAP context starts with maintaining the records of the software products and the underlying software components as well as the versions to be developed in the system landscape directory. Product metainformation in the SLD is required for development life-cycle management at the vendor's site and for software life-cycle management at the customer's site. Once the product information is maintained in the SLD, the actual application development can start. Depending on whether service-provider applications or service-consumer applications are going to be developed, the process differs as follows:

Service-provider application. The starting point of an enterprise service implementation is ES Repository. Following the outside-in approach, begin by modeling the interface of the service to be implemented in ES Repository. Once the service interface is defined, the actual implementation in the back end can take place. According to the service definition in ES Repository, service pro-

vider proxies for communication and data conversion can be generated. The service will be deployed when the implementation is done. To make the service entity known in the landscape, publish the service in the services registry so that the service can be used.

Service-consumer application. To implement a service-consumer application, first retrieve the service definition from the services registry. Depending on the service definition, a proxy for the service consumer can be generated. The application resulting from the development will be packaged for shipment to customers for deployment in productive landscapes.

Productization

Depending on the purpose of the development, there are two ways to ship your software products. Either the software is intended for production use in the same system landscape, or it is meant for delivery and deployment in systems residing in different landscapes. Both forms of delivery and deployment have something in common. All content types belonging to the same application version have to be consistently transported or packaged for delivery. This also applies to the deployment of an application. The content types must be deployed or updated.

Internationalization and localization of software applications have many aspects, such as multilanguage support and adaptability and extensibility of the applications. To meet the challenges of software adaptability and extensibility, various architectural means can be considered, depending on the requirements of the individual applications. With regard to multilanguage support, SAP development tools for both ABAP-based and Java-based programming models facilitate the separation of text resources from program coding by means of text pools. Program code references user interface texts via symbolic names. The actual texts and their translations for different languages are stored in text pools. In this way, translation processes take place independently.

An SOA-enabled application consists of multiple entities in terms of service providers and service consumers, as well as the underlying applications that carry out the service-providing and service-consuming processes. Theoretically, all these entities can be updated or upgraded independently, as long as the service interfaces remain stable. Service interfaces consist of two parts: the service definitions, which are fixed during design time; and the service endpoints, which are determined by the runtime environment. If new service-component versions come with incompatible changes to the service definitions, you have to take into account that the service-consumer applications will require synchronous upgrades or adaptations in order to work properly with the new service components. Furthermore, hardware changes during the upgrade of service components often lead to changes in host names and thus changes of service endpoints. In this case, service-consumer applications have to be reconfigured according to the new runtime environment.

Additionally, you have to take into account whether component upgrades change role definitions or assignments. Such changes must be documented in order to enable the realignment of the authorizations in the production landscape.

Process Composition

This section explains how a process is modeled and implemented in an SOA environment. Technically, a business process is defined as a set of linked activities that create value by transforming an input into a more valuable output. SAP differentiates the business process definition further by establishing two basic subprocess types: core processes and composite business processes.

By nature, core applications deliver the functionality that runs core business operations: from financials, controlling, and human resources to materials management, and from procurement and

sales order management to supply chain and customer relationship management. Support for core processes is delivered via SAP business applications as part of SAP Business Suite, including applications such as SAP ERP, SAP Product Lifecycle Management (SAP PLM), SAP Supply Chain Management (SAP SCM), SAP Supplier Relationship Management (SAP SRM), and SAP Customer Relationship Management (SAP CRM). These applications are exposed as reference content in ES Repository and SAP Solution Manager. This provides process insight and transparency into what is covered within these business applications, bringing discoverability and design visibility. This also provides insight into how the content can be used to extend a core set of highly standardized business operations supported by SAP applications.

Business process management as a technology (or a software application) provides IT organizations with a framework and tools to compose, model, deploy, execute, and monitor processes that include human and system tasks or that span different business applications and require a broad set of integration functionality.

SAP Tools for Process Modeling

The SAP NetWeaver technology platform provides functions for modeling business processes at different levels of abstraction. Conceptual process modeling sustains a wide range of process analysis projects. These drive process harmonization and standardization, the aim being the achievement of a high degree of process excellence. This works very well for the stable core processes of an organization. However, the need to integrate with individuals, business partners, and third-party systems, and, above all, to remain innovative, requires methods for supporting the translation of functional business requirements into technical specifications of process execution.

Guided procedures. Part of SAP NetWeaver CE, guided procedures (GP) focus on human interaction and process collaboration. Process modeling with guided procedures does not require specific development skills. It is targeted at business experts who are familiar with business processes and who use guided procedures to model processes according to specific requirements in the enterprise. GP design time enables business experts to access all tools that enable process modeling. Using GP design time, business experts can create, delete, and edit the elements that build a GP process; create and maintain GP process templates; and organize the objects into folders.

SAP NetWeaver Business Process Management (SAP NetWeaver BPM) component. Delivered as part of SAP NetWeaver CE, SAP NetWeaver BPM offers functionality for modeling, connecting, compiling, deploying, and maintaining composite applications. These functions are split into building blocks: process composer, process server, and process desk.

The process composer in SAP NetWeaver BPM supports all stages of process modeling, from high-level definitions of the process down to the enrichment for the actual development into deployment and execution. Life-cycle aspects, such as versioning, transport, and so on, are fully supported through SAP NetWeaver CE. Process composer facilitates easy graphical design and process composition based on standard Business Process Modeling Notation (BPMN) via "drag and drop" functionality.

SAP NetWeaver Business Rules Management (SAP NetWeaver BRM) component. Business rules represent the policies and guidelines that drive operational business decisions. Examples include rules for validation, rate calculations, decisions, recommendations, and personalizations. SAP NetWeaver BRM is a set of pure Java components that provide management and execution functionality

as well as rules modeling. Rules can be "involved" in a process model in multiple ways. With the process composer in SAP NetWeaver BPM, you can model rules in the context of a business process; with the rules composer in SAP NetWeaver BRM, you can model rules for consumption by composite applications.

Cross-component business process management (ccBPM) is delivered with SAP NetWeaver PI, and it orchestrates the message choreography between systems and applications using "status" interactions. Workflows and processes are modeled by different users in different project or product stages. A process editor is used to model cross-component processes. Deadlines and exceptions with respective handlers are also supported. ccBPM thus enables the modeling of system-centric integration processes, including A2A and B2B interactions and integration with legacy or third-party applications.

SAP Business Workflow tool. Core application transactions delivered as part of SAP business software are based on standard business practices and are designed to satisfy rigorous demands for integration, legal compliance, and processing integrity. SAP Business Workflow is an integral part of SAP applications, including SAP Customer Relationship Management, SAP Supplier Relationship Management, and SAP ERP. The tool incorporates workflow features directly in the respective business application to support these types of mission-critical processes. Additionally, the deep integration of SAP Business Workflow with functionality for organizational management in the SAP ERP Human Capital Management solution as well as with standard SAP reporting tools enables companies to capitalize on their investments made in other SAP solutions.

To support the portability and interoperability of business process definitions, SAP embraces, contributes to, and implements industry standards widely adopted in the SAP customer base. These standards include Business Process Execution Language (BPEL), BPEL4People, Web Services Human Task (WS Human Task), and BPMN.

Process Modeling Approaches

Software developers need to understand the concepts of object-oriented programming, inheritance, SQL statements, class diagrams, and so on. For a business process expert, the equivalent understanding is process modeling. This term describes the art of "carving out" all aspects of the business in a descriptive language that allows the successful building of an application, incorporating all critical processes and, ideally, all day-to-day business activities. Modeling is based on different approaches, each having its pros and cons. The approach used depends on whether the process modeler's objective is an overall strategic model or a detailed tactical model.

Top-down approach. This approach focuses on the overall process. It starts with the definition of the overall business requirements that define the framework of the business process model. Then the model describes how a requirement is fulfilled and split into separate business processes. Each of these processes has specific activities that include specific services and tasks. The goal of the model, however, is to depict a broader view of a business process fulfilling a given requirement rather than to show how a specific activity is performed. This broader view helps business analysts and managers see the overall process, where improvements are needed, and whether elements are missing. However, a top-down model, which does not focus on subprocess characteristics and activities, does not allow you to see how the subprocesses and their activities really behave.

This recommended approach typically involves the following sequence of steps:

1. Analyze business requirements
2. Specify process information about the (collaborative) business process and composite applications
3. Think about exception handling
4. Determine business objects
5. Describe the user interfaces (interactive steps)
6. Describe the required services

Bottom-up approach. In contrast to the top-down approach, the bottom-up approach starts with the definition of the specific activities of the process. The goal of the model is to show how a specific activity is performed to produce a value at the end of the process. This detailed view helps developers and system architects make the process work. However, a problem occurs when all these details have to be combined to form the overall model picture, or when it comes to defining the key requirements that the general model should fulfill. As a result, this approach should be applied if there are any extensions to the process, or if a composite application is necessary – for example, when a new user interface is required because the business process itself has to be extended.

Inside-out approach. Unlike the top-down and bottom-up approaches, which are vertical types of modeling, the inside-out approach is horizontal. It defines key processes in the overall process and then complements them with other processes. This approach may be helpful to modelers in different areas if neither of the vertical approaches is appropriate. Like the other approaches, it also has disadvantages; for instance, it may be difficult to define what the key processes are.

The result of the steps described above would typically be a diagram like the one in Figure 9. This diagram illustrates an approval process modeled by the process composer tool in SAP NetWeaver BPM.

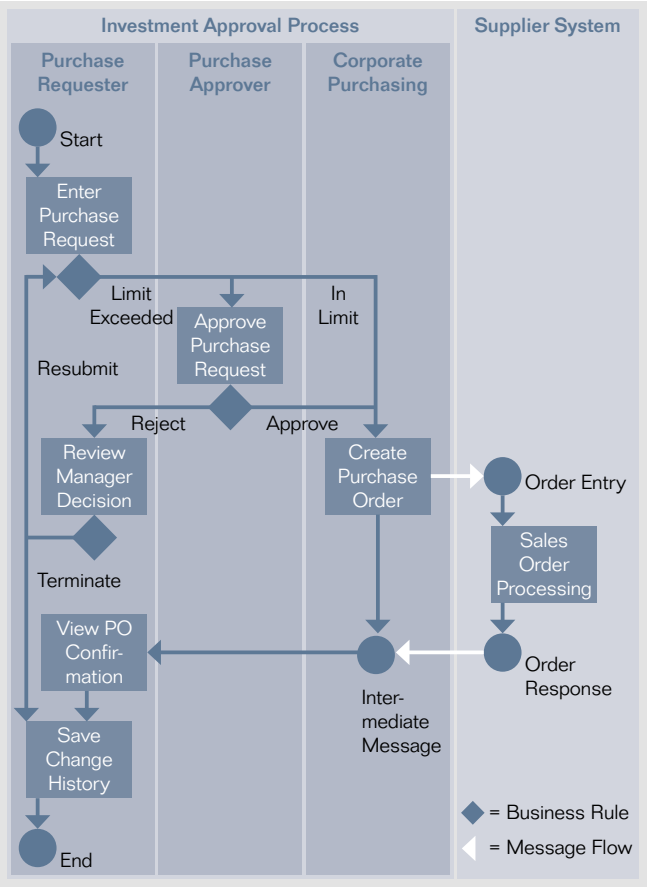

Figure 9: Example of an Approval Process Model

Service Provisioning

The service provisioning phase transitions services from design time to runtime, exposing services to the entire organization by making these services available for consumption via ES Repository.

Modeling Entities of an Enterprise Service

Service modeling in ES Repository is based on a set of predefined modeling entities and patterns that ensure that all services are always defined and named in the same way. Following these modeling principles and patterns provides the homogenous granularity of your services. Figure 10 illustrates the meta model of an enhanced enterprise service.

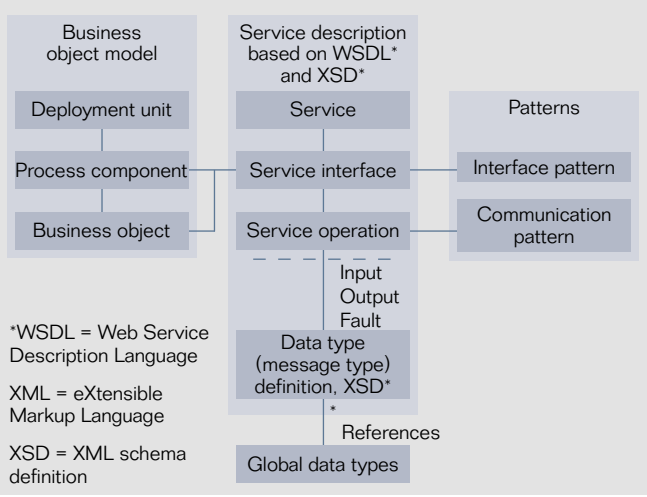

Figure 10: Meta Model of an Enhanced Enterprise Service

ES Repository includes the following modeling entities:

Business objects describe the data of a well-defined and outlined business area. Business objects are defined free of business functionality redundancies and therefore serve as the central point and basis for modeling and defining services. Business objects represent a set of entities with common characteristics and common behaviors, which, in turn, represent well-defined business semantics. A sales order is an example of a business object.

Process components describe self-contained parts of a value chain and group business objects together; one business object belongs to exactly one process component. Process components contain data and services for accessing that data and thus form reusable modules of larger applications. Access to data is modeled by service interfaces and operations. Sales order processing is an example of a process component.

Deployment units group process components that interact with each other and that are to be installed together on a system. Deployment units show the type of interactions needed between the process components within the deployment unit.

Service operations are entities that perform specific tasks on a business object, for example, creating, updating, or deleting a business object. The operation is a specification of a function with a set of message types assigned as the signature. An operation is assigned to exactly one business object, whereas a business object can have multiple operations. Depending on the type of access required to the data or business object, the operations can be asynchronous (for A2A or B2B communication) or synchronous (for access from other components of the same application). An example of an operation is "find sales order by order ID."

Service interfaces are named groups of operations. A service interface belongs to exactly one process component, whereas a process component can contain multiple service interfaces. Service interfaces specify offered (inbound service interfaces) or used (outbound service interfaces) functionality. When you model the operations and service interfaces, you use predefined patterns to ensure that the naming and definition of your services are unified. These interface patterns are derived from the access type needed and cover the majority of use cases. Thus, service interfaces, operations, and message types are always modeled in the same way.

Interface patterns define naming rules for operations and service interfaces based on the business object and its node structure, grouping rules for the operations in the service interfaces, naming rules for the message types, and rules for the message choreography of an enterprise service interaction.

Interface patterns for A2A interaction model the message exchange between two process components with the help of synchronous and asynchronous operations. These patterns can also be used for B2B communication. For example, a "request-confirmation" pattern describes semantics and naming conventions for an interaction in which a change or update is requested and a confirmation is expected to be returned.

Global data types are standards-based data types with business-related content. GDTs ensure harmonization of the hierarchical signature structures of operations and the structure of the business object nodes. They represent business-related subject matter that is described by a specified structure. If this semantic subject matter occurs in a business object node or a B2B or A2A operation, it is always characterized by the same GDT. This helps to ensure the uniformity of data types across all business objects, interfaces, and operations.

Since business information is represented through business objects, the best way to provide a service is to start with that business object. A business object needs to be defined according to the pertinent international standards (ISO 15000-5 and UN/CEFACT CCTS).

Defining Service Contract Business Content

The definition of an enterprise service message type (which is the main part of the contract between two communication partners) follows a clear procedural model. The goal of this exercise is a structured derivation of business content based on the underlying integrated business object model.

The scenario model provides a good understanding of the business context that defines consistency boundaries and describes the business capabilities used for the interface. The required message types can be derived from this model. Every process component interaction model describes one particular interaction path in more detail. It is enhanced by the business entities (business objects) that communicate with each other. Accordingly, they show which interfaces, operations, and message types are used in the communication.

Adapting or Extending Enterprise Services Provided by SAP

When enterprise services delivered by SAP do not fully meet your business requirements and you need to adapt them, be sure to modify or enhance them in such a way that there are no versioning conflicts and that the modified or enhanced services remain stable after an upgrade or update of the original enterprise services. To enable this, SAP provides mechanisms for creating modification-free enhancements and for unifying all possible ways of modifying or enhancing repository objects.

To enhance standard SAP applications, you can use business add-ins (BAdIs). For example, to enhance a service operation in ES Repository, you need to generate an enhancement proxy and implement the required business logic in your back-end application. You implement this business logic in an enterprise service–specific BAdI.

When you enhance data types in ABAP, you need to use ABAP proxy generation to create proxy objects for the data-type enhancement in the application. The field from the enhancement is added automatically to the WSDL as an optional field.

There is no such enhancement concept in Java; therefore, you need to regenerate proxies for service interfaces that reference data-type enhancements. The generation function recognizes the elements and attributes of the enhancement but does not handle them separately. The generated objects do not contain any information about which classes or attributes originate from the data-type enhancement. There are, therefore, no separate Java objects for enhancements. Java proxy generation can only generate proxy objects from service interfaces. If more than one data-type enhancement exists for a data type, the generation function simply creates proxy objects for all data-type enhancements.

In ES Repository you have to create the object enhancements in a separate customer-specific software component version that is based on the SAP software component version in which the object to be modified is defined. You also need a separate namespace, which ensures that your objects will not be overwritten by SAP objects when new objects are imported into your application or during an upgrade. To be able to change objects in the underlying software components, these objects need to be marked as modifiable in the customer software component version.

Implementing Services

SOA principles imply independence of the underlying platform and programming language since SOA is designed to work with any open standard – it is a means of creating loosely coupled services that allow for separating the service description, its implementation, and its binding. Thus, you have the freedom to choose your preferred programming language and development environment when implementing services.

With the outside-in approach described above, you first model your service definitions and create design objects in ES Repository independent of the language that will be used afterwards to implement the actual business logic of the operations. The decision of which language to proceed with becomes more important in the next step when you go from metadata definition to the language-specific definition with proxy objects. These definitions can be built using either ABAP or Java.

Reusing Existing Functionality

If you want to integrate preexisting external services into your business scenario, you first need to import them into ES Repository where all of your other services – from SAP and third-party providers – reside. Then you publish these services to the services registry once the services are deployed and ready for use by service-consumer applications.

Providing a Service Based on Existing Functionality

When you already have the functionality needed for your business purposes implemented in a back-end application, you may only need to expose that functionality as Web services, thus enabling your legacy application for SOA development. This approach of service-enabling an existing functionality is known as inside-out development of services. Note that by providing services in this way, you lose the benefits of governance, homogeneity, and so forth, which are provided by the outside-in approach.

In Java you can create an inside-out Web service by exposing a session enterprise bean or a pure nonabstract Java class. SAP NetWeaver Developer Studio adds the required Java EE 5 annotations for Web services accordingly. You can also use a service endpoint interface (SEI) to define which methods of the implementation bean to expose as Web service methods. This SEI is a Java interface that represents the Web service contract definition and enables you to separate that contract definition from the Web service implementation.

In ABAP you can expose the following as Web services: remote function call (RFC)-enabled function modules, function groups that contain an RFC-enabled function module, and BAPI® programming interface. To provide the Web services, you use a service wizard. The Web service properties are defined in a preset, selectable profile. The values assigned through a profile can be changed in the object navigator tool of the ABAP development workbench. You can assign features, such as the authentication level, to the service definition in abstract form. The technical details of these features are specified during Web service configuration.

If you need to call services of a non-SAP or legacy SAP application that cannot otherwise be made available as Web services directly (for example, because they do not support the SOAP protocol), you can use brokered service-based integration. An integration broker can be used to enhance point-to-point Web service calls by forwarding the Web service call to services from various systems in a heterogeneous system landscape.

Service Consumption

There are three major entry points for discovering enterprise services provided by SAP. These are illustrated in Figure 11.

Enterprise Services Workplace. ES Workplace is part of SAP Developer Network and provides detailed and differentiated navigation and search functionality. You can find enterprise services via solution maps, process components, integration scenarios, and enterprise service bundles, or you can use a free-text search. Although you can search for enterprise services and explore their details without additional authorizations (other than your SDN user), you need to request a free-of-charge user in order to use the services. Note that you cannot use these services productively and you have no way of evaluating concepts like enhancements to enterprise services, for example.

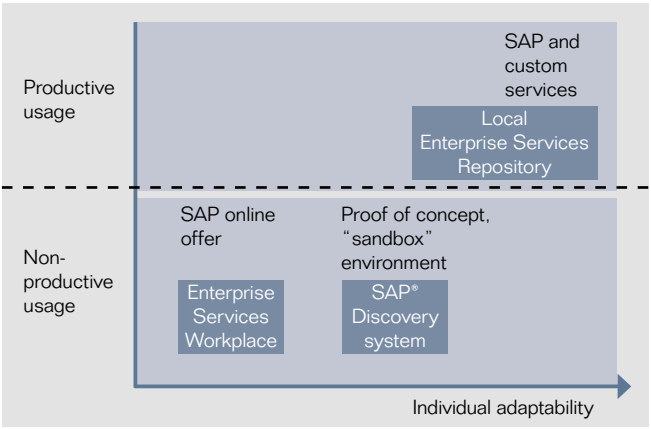

Figure 11: Availability of Enterprise Services

SAP Discovery system. SAP Discovery system is a fully configured SAP landscape for SOA innovation and exploration. Using this system, customers and partners can test-drive SOA in their own environment, get hands-on experience, and interact with solutions and current technologies, including SAP ERP, SAP NetWeaver, and enterprise services. In contrast to ES Workplace, where services are hosted for you by SAP, the SAP Discovery system goes one step further and brings the services into the customer landscape. While this setup does not allow productive use, you can change existing services, create enhancements, and so on.

Enterprise Services Repository. You can use ES Repository in your landscape to discover enterprise services. This is the only way to productively use enterprise services for your development. Since not all descriptive elements and search taxonomies from ES Workplace are contained in ES Repository, it is recommended that you search for a service in ES Workplace first, before taking a closer look at it via ES Repository.

Finding Services for Consumption

First of all, you need to distinguish between service definitions and service endpoints. For definitions, ES Repository is the place to go. Here you can find all services along with their process components, integration scenarios, relevant process component interaction models, service interfaces, and so on. These models help you to understand what a service is supposed to do, to which business domain it belongs, and which relationships it has to other services, process components, or business objects. Service endpoints can be found in your services registry. The registry contains all configured endpoints, and it is integrated into SAP development tools, allowing it to connect directly to one of your configured services during development.

Service Consumption with SAP NetWeaver

SAP NetWeaver offers several tools for service consumption with differing scopes. According to specific requirements, services can be included on different layers.

Service layer. When building application logic in ABAP or Java, you have a choice of numerous methods for reusing functionality based on services. For classic programming, you can generate consumer proxies in both languages. This approach encapsulates communication and lets you concentrate on your own source code to use the actual functionality. On the Java side, SAP CAF provides additional means to easily create application logic for composite applications. Existing services can be consumed to persist application-specific business objects created with SAP CAF ("remote persistence") or to simply call required functionality.

User interface (UI) layer. In general, you base user interfaces on your application logic, which itself refers to the required services; however, you can include services on the UI level as well. SAP NetWeaver Visual Composer and the Web Dynpro development environment for both Java and ABAP offer this functionality. This includes services based on your own programmed business logic, SAP CAF services, and direct usage of enterprise services.

Process layer. In most cases you will build your processes based on services from the service and UI layers. While both artifacts themselves can include services, SAP process modeling tools also support direct service consumption. Again, you can include enterprise services directly as well as services based on SAP CAF or plain Java or ABAP.

Regardless of which tool you use for consumption, services should only be consumed via the central hub in your landscape: ES Repository with the services registry.

Best Practices for Service Consumption

You can include services on different layers in your application. Which layer is most suitable depends on your use cases. For example, if you need a user interface for a service, you can either consume the service in the user interface directly or use SAP CAF as a mediator. If you collect data during a process and you need to send this to a service, you may include the service in your process directly.

A general recommendation is to consume all reading services with a user interface while integrating writing services on the process level. If you are unsure of what to do, the following approach may help:

- Wrap the required services with SAP CAF to reach back-end abstraction
- Consume SAP CAF services in user interfaces to provide application-specific screens
- Build your process by integrating specific user interfaces and consuming other SAP CAF–wrapped services

SAP tools provide direct connectivity to ES Repository for importing the latest service definitions. Although the tools also allow you to import services directly, for example, based on a WSDL file, we strongly recommend not using this option (see below for reasons). However, you still have to differentiate between two aspects: design time and runtime. Also, security issues need to be addressed.

Design-time service definitions should be imported only from ES Repository or the services registry. This rule is based on a number of reasons. For instance, it helps to avoid double work since you get an overview of what already exists in your landscape. You can be sure that you are working on the latest versions. And you use only those services that were accepted for use in your landscape.

Design-time administrators can restrict developers' access to the various sources. Consequently, not everybody can read information from the registry, and – what's more important – not everybody can publish services. When integrating services, you should make use of logical destinations whenever possible. This yields several benefits. Endpoints can be changed without affecting the application. Parts of an application that belong together can be administered centrally. Shifting from development to testing to productive systems is easier.

Runtime. For runtime you should refer only to endpoints that are published in your services registry. This means you can be sure that you are using the latest versions of the services and accessing the correct applications.

Security. Developers should aim at inheriting the security settings of service providers via WSDLs and the included Web Services Policy Framework (WS-Policy) declarations. When service providers use mechanisms for user ID propagation, such as single sign-on (SSO) tickets or Security Assertion Markup Language (SAML) token profiles, configuration is reduced to a minimum and only a single trust relationship (X.509 certificate based) is required between the service provider and the consumer. As a prerequisite, the identity information exchanged during the SSO process needs to be shared between service providers and consumers. This task can be handled by common user stores or by deploying identity management solutions such as the SAP NetWeaver Identity Management component.

Concerning consumer-specific security aspects such as front-end authorizations, auditing, or threat and vulnerability management, developers should adhere to the relevant security development guidelines for their platform and reuse security application programming interfaces (APIs) offered by their front-end platform as much as possible. To ensure a high security level, it is recommended that you strictly separate UI and business logic, implementing security checks in the business logic part via APIs and using a standardized UI framework such as Web Dynpro.

Wrapping Services with SAP Composite Application Framework

There are three major use cases that require wrapping services with SAP CAF before consuming them in your application (see Figure 12). These use cases – back-end abstraction, simplification, and enhancement – differ in the relationships between the original service interface and the application-facing interface defined by SAP CAF application service.

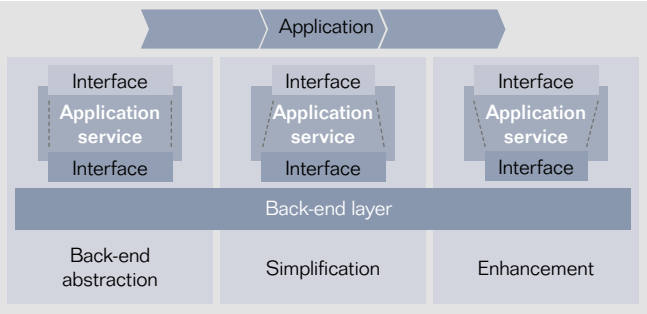

Figure 12: Service Wrapping with SAP® Composite Application Framework

Back-end abstraction allows a consuming application to remain unchanged even when the underlying service implementation and interface changes.

To achieve back-end abstraction you can use SAP CAF. For this purpose, you need to create an external service, which wraps the service you want to use. You then create an application service with the required interface, which should be exactly the interface of the called external service in the case of plain abstraction. Inside the application service you simply map the application-facing interface to the external service's interface. Your application refers to the SAP CAF service, which wraps the back-end call. This makes it transparent for the application how the underlying service's interface is defined and how the service is called.

Simplification of a service interface with SAP CAF includes back-end abstraction. Again, you need to refer to the external service you need for your application. You need to define an application service that will be used by your application. In the case of simplification, the application service's interface will be simpler than the native service interface – you expose only the essential parts of the service to the consuming application. Inside your application service you need to map both interfaces, and you might have to take care of the unused fields of the native interface. In straightforward cases you can achieve simplification by using a virtual interface for any required Web service.

However, there are two aspects to consider. A virtual interface belongs to a specific service – by simplifying a service with SAP CAF you reach back-end abstraction. Also, a virtual interface can handle only default values for excluded parameters; with SAP CAF you can add program logic between your application-specific interface and the service's interface.

Enhancements that are application-specific and not generic should be implemented by means of SAP CAF. Again, you need to define an application service with the application-specific interface, and you need to refer to the actual service you require for your application. You now implement the communication between the two interfaces. This time the application-facing interface is "bigger," which means simply transporting values between them, in some cases, and enhancements based on the additional parameters in other cases.

User Interfaces

The fastest way to come up with a user interface for a service is to use SAP NetWeaver Visual Composer. Although SAP NetWeaver Visual Composer is intended to build UIs for productive usage, you can also use it for prototyping. This gives you an idea about which tool will be the right one for your actual development and can act as a means to align with end users and stakeholders.

SAP NetWeaver Visual Composer is a modeling tool, while Web Dynpro comprises fully fledged development environments following the model-view-controller paradigm. Each approach comes with some native restrictions. While modeling is rather easy to learn and no sophisticated programming skills are required, this approach is limited in terms of functionality. Programming, on the other hand, allows a wider range of functionality but requires more sophisticated skills.

Another option is to create UIs based on SAP Interactive Forms software by Adobe. Adobe's tool for designing forms (Adobe LifeCycle Designer) is integrated into the design time of Web Dynpro and guided procedures. This provides a high degree of automation – such as replicating existing data structures to the forms. Adobe LifeCycle Designer also allows existing forms to be imported; these forms can then be connected to Web Dynpro or guided procedures.

An enterprise service bus provides an abstraction layer on top of an implementation of an enterprise messaging engine. It does not implement a service-oriented architecture but provides the functionality with which one may be implemented.

Enterprise Service Bus

SAP NetWeaver enables IT organizations to use standards-based Web services to form new and innovative business solutions that meet their changing business needs quickly. In particular, SAP NetWeaver provides SOA middleware that facilitates communication between disparate applications. From a logical point of view, SOA middleware consists of ES Repository and a services registry, an enterprise service bus (ESB), and SOA management tools. SAP NetWeaver PI is the SAP implementation of SOA middleware and, in particular, of an enterprise service bus. With this SOA middleware, IT professionals can rely on an open and standards-based platform to accelerate business integration.

In the SAP environment for SOA, the ESB is the communication infrastructure that is responsible for the integration of different systems in a system landscape. The service bus combines the providers and consumers of services, the functions of a service runtime that enables service-based communication, and the functions of an integration runtime that provides mapping, routing, and business process management. In this way, scenarios can be supported in which the consumer and provider do not know each other and have different technical prerequisites. In cases where either the consumer or the provider is a legacy application that is not SOA-enabled, adapters must take on the task of bringing both parties together; that is, the task of creating a protocol conversion between the Web service protocol and the respective protocol of the legacy application.

This process integration based on SOA middleware ensures that differences between purely service-based scenarios and integration scenarios can be kept to an absolute minimum. The service bus enables you to connect core application processes with business partners as well as the composition environment. The various connectivity options of the service bus are supported by different Web services–based standards.

SAP delivers business content that contains all integration knowledge for a cross-component business process that can be specified during the design phase. This content is stored and managed in ES Repository and is also referred to as ES Repository content.

Figure 13 provides an overview of the components that make up an enterprise service bus.

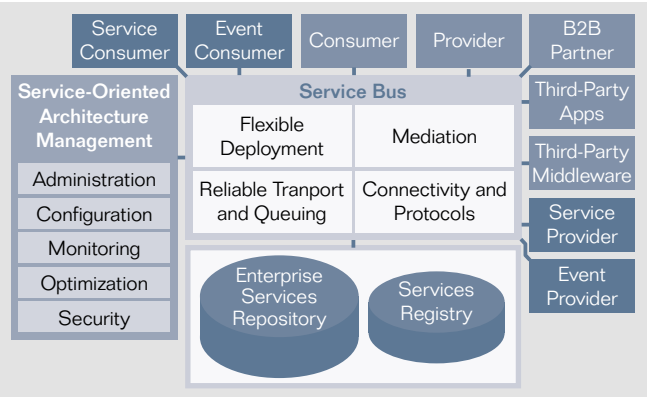

Figure 13: Enterprise Service Bus

Key characteristics of an enterprise service bus include:

- Support for synchronous and asynchronous messaging
- Support for standards
- Ability to wrap non-Web service–enabled (legacy) applications and expose them as managed and reusable services
- Mediation functionality, with the ability to add additional mediation functionality for message enrichment

- All routing patterns including dynamic routing (content-based routing)
- Data mapping and transformation
- Ability to handle high-volume messaging scenarios
- Interoperability between consumers and providers with different bindings, such as SOAP and HTTP as well as proprietary protocols
- Support for message-level as well as transport-level policy-based security

Point-to-Point Versus Mediated Scenarios

In a point-to-point scenario, messages are sent directly from a sender application to a receiver application. In a mediated scenario, a message is sent from a sender application to middleware software, where it is processed and then forwarded to the receiver application. The integration server is the central instance of the middleware software in this case, and it provides process integration services such as routing and mapping. The integration directory enables central configuration of integration broker–mediated communication for A2A and B2B scenarios. It enables you to connect non-SAP and SAP applications to each other in different versions and implemented in different programming languages, such as Java and ABAP.

At the center of the process integration infrastructure is XML-based communication that uses HTTP. Application-specific content is transferred in messages in user-defined XML schema from the sender to the receiver using the integration server. Senders and receivers that exchange messages using the integration server are separated from one another. This separation makes it easier to connect systems that are technologically different. Every system that can exchange messages with the integration server can also exchange messages with all other systems that are connected to the integration server.

The mediation functionality of the enterprise service bus provides services that are essential in a heterogeneous and complex system landscape:

- You can define mappings between data and structures of different technical and semantic types and execute them at runtime.
- You can dynamically specify the destination application of a message according to various parameters at runtime (routing).
- You can split messages; for example, send all messages with specific attributes to an additional application as well as the actual target application.
- You can convert messages from other formats into the XML message format of the process integration environment and vice versa by means of suitable adapters. For example, you can convert RFC-based calls into XML-based messages and communicate with external applications.
- You can monitor the message flow and processes with central monitoring tools.

Performance Considerations

SAP NetWeaver PI provides three techniques to better support high-volume scenarios.

Direct connections allow process integration to skip runtime traffic when it adds no value. A central Web services configuration for direct communication between two applications without using the integration server is possible.

Packaging is basically intended to group small messages and process them in parallel queues to improve throughput. You can improve the performance of message processing in SAP NetWeaver PI by grouping together asynchronous messages in packages and then processing each message package. The asynchronous messages are not processed individually but as a package in one logical unit of work (LUW). Depending on the size and number of messages in the queue, this procedure can increase performance significantly.

Additional advantages include:

- Multiple messages are processed in one dialog work process in one service call. Context switches for mapping, routing, reading the message header, and so on can be reduced, and mass operations on the database are possible.
- Programs required for processing messages are loaded only once for each package.
- Messages can be loaded more quickly.
- Fewer logons are required for processing the messages.

Local processing in the advanced adapter engine (AAE), the successor of the adapter engine, supports a wide range of scenarios, skipping unnecessary internal steps and boosting message speed and throughput. Message processing is executed only on the AAE from one adapter to another without the involvement of the integration engine. The AAE provides mapping and routing locally. The primary objective of using local processing in the AAE is to increase the performance of message processing by eliminating the need for the ABAP stack during the process. By bypassing the ABAP stack, message processing still keeps existing monitoring and support functions.

Connectivity

In direct communication, two applications use the Web service runtime to communicate with each other directly without using middleware software. Direct connections increase performance. The drawback is the continued maintenance and support of the point-to-point connections.

These activities can be centralized by using SAP NetWeaver PI. The necessary configuration settings for the involved applications can be made centrally in the integration directory. The configuration settings are propagated to the back-end applications by means of cache notifications.

Be aware that high performance is based primarily on proper integration design

Configuring Connectivity to the Back-End Application

Connectivity between the integration server and back-end applications is enabled by adapters. Depending on the standards and protocols supported by the back-end application, different adapter types have to be used. For example, the intermediate document (IDoc) adapter converts XML-based and HTTP-based documents to IDocs, whereas the file/FTP adapter enables file applications to connect to the integration server.

To configure connectivity, you have to specify the adapter type for a connected back-end application and then specify the available adapter attributes. You define these settings in the communication channel object in the integration directory. To specify the back-end application for which you want to configure connectivity, you have to assign the communication channel to the corresponding communication component that represents the back-end application.

In a next step, you define the adapter type, and thereafter, the available attributes for the selected adapter type. The latter step depends on the adapter type. For particular adapter types, you can also specify security settings. For example, when using a SOAP adapter, you can define settings for message encryption or decryption.

Configuring Routing

To configure routing, you have to specify the relevant receiver and interface determinations in the integration directory.

When a message is exchanged by the integration server, it is sent to one or more receivers. The integration knowledge that is necessary for the message to "find its way" to the right receivers is contained in the corresponding receiver determinations and interface determinations in the integration directory. These objects are defined for specific key attributes contained in the message header. This ensures that, for a particular incoming message, the correct receiver determination or interface determination object is found to determine the further path of the message.

A receiver determination contains information on the receiver applications to which an incoming message (on the integration server's side) should be routed. An interface determination further specifies the receiver interfaces and, if applicable, the mapping that should be applied to map the incoming data structures to the receiver interface. Receiver and interface determinations can contain conditions that define the receivers and receiver interfaces dependent on the content of the message (known as payload-based routing).

SOA-Enabling Legacy Applications

With SAP NetWeaver PI, customers can make legacy functions available as standardized services, starting with process modeling in ES Repository and the design of the necessary service interfaces, data types, as well as mappings and communication channel templates, which contain preconfigured settings for connectivity.

110 Use Cases

USE CASES

This chapter examines a customer use case that illustrates how the end-to-end development process described in the sections above delivers value in a real-life industrial environment. The first use case, based on the experience of an SAP customer, is presented as a working tour that touches upon many of the components described in the previous chapters. The examples that follow this customer case study illustrate other potential use cases and business drivers for SOA adoption in a range of industries.

Building and Running a Composite Application to Improve the Customer Service Experience

The customer, a manufacturer in the industrial machinery and components industry, was looking for ways to optimize the customer service experience by accelerating the time to resolution of maintenance and repair calls. In the past, the customer service experience was impaired through information disconnects. For instance, customers were frustrated when they had to describe a technical fault to a (usually nontechnical) hotline assistant on the phone – and when the maintenance technician arrived on the scene, often with the wrong parts, and inevitably had to return at a later date. Repeated maintenance trips and customers' downtime cost both the manufacturer and its customers precious time and money.

The SAP approach to SOA has delivered a composite application that enables the manufacturer to overcome information disconnects in the customer service call area. Today, the manufacturer leverages a composite application that helps the company improve its agility, enhance collaboration, and reduce operational costs.

When a fault occurs in the equipment supplied by the manufacturer, the customer's employee logs on to the manufacturer's customer service portal and conveniently selects the applicable fault condition from the list in a drop-down box. A list of defined faults qualifies the repair and maintenance case, making it easier for the manufacturer's service technician to retrieve the right spare parts from the warehouse. Unnecessary trips with the wrong spare parts in the service technician's van are now a thing of the past. The manufacturer's composite application for the customer-facing maintenance and repair process delivers value to both the manufacturer and its customers. It supports service technicians in "doing it right the first time" and at the same time helps to improve customer satisfaction and retention.

How was this achieved? Through composition! Core processes, such as fault reporting, were linked with context processes, such as fault resolution tracking and analysis. The seamless integration of all processes involved in maintenance and repair is the result of the consumption of enterprise services via a composite application. While the data of the manufacturer's core processes remains in different systems, the representation of all relevant data is fully aligned with the manufacturer's use case. Instead of triggering 20 or more applications, the entire business process relies on a single application to get the job done.

Beneficial spin-offs of this SOA approach include fault statistics that help the manufacturer improve the quality of its goods and services as well as the planning of maintenance intervals based on verified statistics.

Choosing the Right Blend of Services

The manufacturer's SOA project started at the top – with an assessment of the business process. This was followed by an assessment of the technical requirements that had to be met so that the business process could be modeled. Business process experts from SAP Consulting performed a gap analysis that matched the standard service bundles delivered through SAP enhancement packages with the manufacturer's business blueprint. This analysis confirmed that the vast bulk of the manufacturer's business process needs were addressed effectively by standard SAP services; the "gap" identified by the analysis consisted of a number of additional services that were not available out of the box.

When faced with the "make or buy" question, the manufacturer decided to extend a number of standard services (taken from SAP enterprise service bundles) and to develop some additional services in-house. The development of other services was outsourced to SAP.

Service Development and Composition Made Easy

Since the standard services supplied in bundles via SAP enhancement packages are developed in ABAP, the manufacturer also used ABAP to extend selected standard services and to create additional custom services. Like the standard services, these extended and custom services draw on data from the SAP ERP back end. The SAP NetWeaver Composition Environment was used for process composition in Java. The associated user interfaces were developed in Web Dynpro. The SAP NetWeaver Application Server for Java provided the enterprise portal runtime for the manufacturer's composite application (see Figure 14). This portal runtime saves the manufacturer the effort of maintaining a separate enterprise portal in the development landscape.

Service development and provisioning in this project strictly followed the SOA governance model.

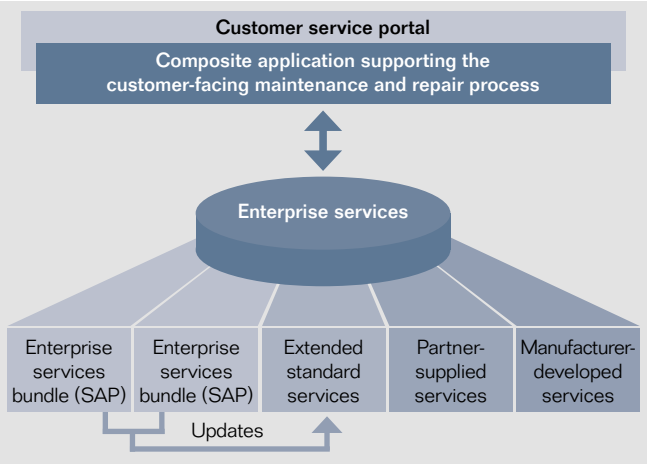

Figure 14: Customer Service Portal Supported by Enterprise Services

The manufacturer used the development infrastructure in SAP NetWeaver, that is, Java, for composite development. The manufacturer's ABAP development, versioning, and transport were supported by convenient ABAP tools such as transport management.

While the initial deployment of the development environment in SAP NetWeaver requires a certain degree of coordination and alignment effort within the manufacturer's IT organization, the payoff far outweighs this effort. For example, software maintenance requires much less labor than in the past. The manufacturer's IT organization confirms that the time to value for new service versions has been reduced considerably. For extended versions of standard services, the IT organization can be assured that the next update of the underlying standard services connects smoothly to the manufacturer's custom extensions.

Simplification of Transport Management

The manufacturer's composite application involves both ABAP- and Java-based software components plus a portal. When this composite application went live, **all** of its components had to go live at the same time. In theory, the manufacturer could have ensured this synchronization through an organizational procedure. However, the potential risk of media breaks, human error, and disparities among underlying applications prompted the manufacturer to look for a smarter option: an enhanced central transport service that connects all of the components of the composite application to accomplish a single transport, eliminating media breaks and requiring only minimal human intervention. Now the manufacturer's IT staff uses the CTS+ cockpit for going live with the entire composite application in a single step; that is, all of the components of the application go live at the same time via the usual transport paths.

Robust End-to-End Business Process Monitoring

The system landscape directory spans both the technical landscape and the business landscape. While the technical landscape provides for software development, the business landscape is characterized by one or multiple composite applications, each of which may span a number of different technical systems.

The service landscape directory serves as lookup pages in which specific items, such as each system's IP address and physical location as well as the components included in each system and the dependencies between components, are registered and kept current. This information is of vital importance any time two or more physical systems within the technical landscape communicate with each other.

On the business landscape side, different composite applications may rely on different applications. For example, let's assume that there are three applications – A, B, and C – that support two composite applications, I and II. Composite application I uses application A and B; composite application II uses application B and C. While this information may be present in the heads of software developers, chances are pretty good that it's not registered anywhere else. That's exactly where the SLD comes in; the manufacturer's customer service application stage can be monitored seamlessly across all of the applications involved. The SLD thus helps to generate a consistent business landscape view for SAP Solution Manager diagnostics. The information provided by the SLD enables SAP Solution Manager diagnostics to determine whether a fault is due to a technical defect or to an inconsistency in dependencies among components, whether a wrong change or transport was performed, or whether the application simply hangs for some unknown reason.

Each composite application can be mapped to SAP Solution Manager for testing. SAP Solution Manager makes it easy to track the degree of completion of each application. Once the composite application goes live, SAP Solution Manager can be used to check which business process spans which applications. If a fault occurs, the end-to-end process monitoring diagnostics of SAP Solution Manager pinpoint the application in which the fault originates. The key benefit of these diagnostics, supported by the SLD, is the high degree of transparency and rapid fault resolution they provide in spite of the complexity and degree of distribution of the underlying system landscape.

Spanning the plan, build, and run phases, SAP Solution Manager helps the manufacturer align IT management and software development efforts with the life cycle of the composite application. During the plan and build phases, SAP Solution Manager is used for testing the development results and validating progress. During the run phase, diagnostics in SAP Solution Manager serve largely for automated end-to-end-monitoring.

Naming Service for Distributed Software Development in Java

While the ABAP world is centralized, Java is open – and thus prone to naming clashes. The object naming service provided by the system landscape directory helps to ensure that unique names are used in Java. Due to its distributed nature, Java development greatly benefits from unique naming. Today, the manufacturer's software developers use unique names, complete with prefixes, that are predefined at whatever depth is required. The unique naming provided by the naming service applies to object names as well as to Java packet names.

Fault Resolution or Change Request?

The service desk function that comes with SAP Solution Manager enables the manufacturer's IT support team to register, track, and resolve issues reported through support calls and to close the ticket associated with each call. Initially, a call has to be qualified. Is the caller reporting a fault or does the reported issue constitute a change request? How much cost and effort does this change request involve? The manufacturer uses SAP Solution Manager to map the entire change request process, complete with approval cycles. The manufacturer's composite application is a customized software product based on standard SAP solutions. If the change request relates to customer-specific parts of the composite application, the manufacturer's external IT service provider takes care

of the change. If call analysis at the service desk indicates that a fault or a change request refers to the SAP part of the composite application, the customer's IT staff simply forwards the ticket complete with all its history to the appropriate SAP support staff instead of accessing the SAP Service Marketplace extranet and entering the fault, from scratch, into the maintenance portal.

Getting on the Enterprise Service Bus

Enterprise services, coded in ABAP or Java, are centrally maintained in the manufacturer's ES Repository. The enterprise service bus is used for representing these services. The bus is the architectural pattern that brings together the providers (the manufacturer's back-end applications) and the consumers (components of the manufacturer's composite application) of these services. Synchronous as well as asynchronous processing takes place via the enterprise service bus.

ES Repository, the services registry, the system landscape directory, and diagnostics in SAP Solution Manager are constituent parts of the manufacturer's enterprise service bus. Another constituent part of the manufacturer's ESB is CA Wily Introscope. Designed for root-cause analysis, this tool allows IT staff to define measuring points and to take a look inside an application. It can also be used for latency tracking (that is, when the processing of an application call exceeds a predefined time threshold). CA Wily Introscope agents feed diagnostics data into SAP Solution Manager.

Results

Delivered through an integrated enterprise portal, the composite application described in this case story makes the manufacturer's maintenance and repair process more responsive, more agile, and more cost-effective.

SOA helps companies leverage the connectivity provided by enterprise services for business processes at Internet speed.

Connectivity, Simplification, Landscape Modernization, and Collaboration Use Cases

The following use cases illustrate how SOA can help companies address priority issues with unprecedented ease through connectivity, simplification, landscape modernization, and collaboration.

Connectivity Use Case: Inventory

Out-of-stock situations cause customer dissatisfaction and lost revenue, especially for retailers. An SOA-based information system can connect retail locations, enabling employees to check in real time whether the requested item is still in stock so inventory can be shared "on the fly."

Challenge: Out-of-stock situations, especially for high-margin items with low stock levels, can cost a considerable amount of time and money and often require a cumbersome and time-consuming manual process to resolve. Reports used in the process may not contain the most up-to-date data, causing additional errors that require manual approaches to resolve, such as phone calls, faxes, or on-site visits. In addition to lost revenue, out-of-stocks can damage a company's reputation.

Objective: The objective is to overcome the lack of reliable information and enable access to information about the availability of items across retail locations. The business case for an IT solution is driven by the need to keep popular items available in spite of low inventory levels, to reduce the workload of and the pressure on employees, to minimize reporting and inventory errors, and to eliminate the cost of gathering and maintaining redundant data.

SOA helps to reduce complexity, enabling companies to "take the fat" out of business processes and make them leaner – and faster.

Solution: Enterprise services can provide any-to-any connectivity at the store level on a variety of different devices, including hand-held devices that can read and process data from RF tags. When employees in inventory, sales, and customer service can access the same information through a role-specific interface, they can quickly and easily get to the information they need for the specific step in the sales and service process. In addition, employees can be alerted proactively when in-store stock levels fall below a defined threshold.

Benefits: When store employees have real-time access to current inventory information across all retail locations, they can respond to customer requests and provide superior service, which improves customer satisfaction and retention. In terms of system maintenance and change readiness, enterprise services can provide a much more flexible and cost-effective approach by replacing numerous application interfaces with a single, easy-to-use interface.

Simplification Use Case: Master-Data Entry

In many companies, the creation of new master records is a slow, cumbersome, and error-prone process. Let's consider the creation of a master-data record for a new supplier – a common process across many industries. It starts with a purchasing agent who delivers the new supplier's information on a piece of paper to a clerk who enters this data into a software application. Then a power user may validate the data and cross-check it manually against other applications to see if there's a duplicate record anywhere in the system. Let's see how a composite application can help bring a new supplier's master record online in minutes, reduce the need for manual intervention, and increase data quality.

Challenge: In our example company, multiple back-end applications and databases resulted in a data entry process for master data that was complicated and time-consuming. In addition, it required a small group of highly skilled and highly trained "super users" to enter and validate master data. Information about new suppliers came to these users in a variety of ways: e-mail, spreadsheet, Microsoft Word file, or phone call. Typically the information was incomplete and didn't match up with what the software application required. Delays occurred when super users had to contact the originator of the information to get the missing information about a new supplier. Additional process steps, such as checking for duplicate data records, further delayed the availability of new supplier master records across the company's business units.

Solution: A composite application provides simplified, role-specific user interfaces that help to improve the speed, quality, and consistency of master record creation. This approach moves the data entry process from super users in the back office to purchasing agents in the front office. To improve the speed and quality of the data being entered, simple role-based user interfaces can help. Designed for ease of use, each interface provides exactly the view of the data that the purchasing agent needs to enter. Data entry fields (such as address, phone number, and bank details) are self-explanatory and don't require additional on-screen help, documentation, or training. The role-based interfaces access data in a single unified master database, so there's no need for employees to manually check for duplicates. Now super users are available to handle exceptions and perform other value-added activities.

Benefits: Role- and usage-specific interfaces reduce the skill and knowledge needed to enter supplier master data. Automatic checking and built-in rules catch errors when they occur, which means that the master data entered is accurate from the start, eliminating major problems down the road. When a purchasing

With SOA, service execution can simply be rerouted when applications are consolidated or retired.

agent has a question, the exception-handling workflow automatically alerts the super user, who can step in and research and resolve the problem. In addition, the cross-media usage of services (online, offline, mobile, portal, B2B, voice, and so on) eliminates information disconnects and provides a high degree of collaboration.

Landscape Modernization Use Case: One Face to the Customer

Mergers and acquisitions are a fact of life. What is also a fact is that the acquiring company must decide what to do with duplicate as well as unique product lines brought into the mix by the acquired company. Often the acquiring company wants to keep acquired product lines alive while extending the reach of its sales operations to increase its win rate and to achieve significant growth at home and in emerging markets. This challenge is faced by companies in most every industry – including manufacturing, consumer products, high tech, insurance, banking, professional services, and communications.

Challenge: In this example, let's consider a midsize supplier of cosmetics and hair care products that acquired a former competitor in a neighboring state. The acquiring company wanted to achieve quick time to value before the start of the holiday season and leverage synergies immediately. In addition, the company wanted to consolidate and update its IT environment without impacting customer-facing processes.

Solution: Using a composite application, the company created a portal that enabled sales reps to sell from the product lines of both companies, even though the product data is physically stored in two different systems. The semantically abstracted enterprise services on which this composite application is built enable the user-facing functionality to "sit on top" of legacy IT landscapes in the short term while these two landscapes are being consolidated in the long term.

SOA delivers self-service processes that cut administrative costs across functional and organizational boundaries.

Benefits: Customers experience one-stop shopping and can talk to one sales rep. This approach increases customer retention, shortens the sales cycles, and reduces the cost of sales. A single easy-to-use interface eliminates the need for additional employee training and cuts down on order-entry errors. On the IT side, the service-enabled solution can be reused following the consolidation of the IT infrastructure, reducing total cost of ownership.

Collaboration Use Case: Self-Service

Government agencies at all levels – national, state, county, and city – are looking for ways to improve the delivery of services to their constituency while reducing administrative costs. Pressured by work schedules and stymied by limited service hours and distance from government service centers, citizens often neglect to apply for licenses and permits, such as animal licenses, fishing and hunting permits, and home improvement permits, resulting in lost revenue for the affected government agencies. In response to this challenge, government agencies are providing self-service Internet portals that enable citizens to process applications online, check the status of applications, and receive licenses or permits. Filing out an application involves the same sequence of steps, making this process a prime candidate for enterprise services.

Challenge: As citizens of a city, state, and country, we are familiar with the process of applying for a license or permit. An applicant appears at the government office and fills out a form. An agency employee checks the information provided on the form and hands it to another employee who is responsible for entering the data into a stand-alone software application. The applicant returns home to wait for the permit, but instead receives a letter from the agency requesting additional information. Eventually the applicant receives the license or permit. However, in many instances, citizens take the risk of paying a fine and simply don't apply for the necessary license or permit.

Solution: When an agency has a portal, whether on-site in a kiosk or available via the agency's Web site, an applicant can fill out an application online. An enterprise service can populate the form with data about the citizen from the agency's database. Standard data validation routines check that all the required data has been entered, and the data is correct. Another service can process online payment of the fee for the license or permit. Depending on the type of permit, the applicant can print the permit immediately, or the agency can perform additional steps and either e-mail the permit as an attachment or send the permit by physical mail.

Benefits: Citizens and agencies benefit from this approach – as does the environment. Citizens save time and money by not having to travel to the agency or waiting to receive a permit. Agencies can reduce or redeploy staff, trim administrative costs, and earn more revenue. An easy-to-use online process and immediate results encourage citizens to comply with regulations rather than risk fines.

CONCLUSION

SOA makes it easier for companies to operate more efficiently, to gain business insight that drives better-informed decisions, and to boost flexibility. SAP is committed to making SOA easier to adopt – enabling companies to quickly differentiate themselves and adjust their business processes across the business network.

SOA – Here Today and Here to Stay

Companies across all industries as well as public-sector organizations around the globe have adopted SOA for enhanced connectivity, IT simplification, nondisruptive landscape modernization, and collaboration within and across their boundaries. SAP and the SAP ecosystem and partners provide the software, technologies, knowledge, and expertise that enable these companies and organizations to achieve faster time to value in their SOA projects.

SAP Is Committed to SOA

SOA is at the heart of SAP's commitment to help companies and organizations attain higher levels of flexibility, efficiency, and business insight. Delivering on this commitment, SAP has a growing portfolio of SOA-enabled solutions and offers industry-focused enterprise services that deliver rich functionality companies can use to build their own composite applications needed to embrace accelerating change.

SAP NetWeaver Composition Environment provides the tools and runtime environment for building and managing composite applications. It's one of the first integrated offerings on the market that was developed right from the start specifically for composite applications. SAP also offers proven tools, such as SAP NetWeaver Visual Composer and SAP Composite Application Framework. The easy-to-navigate Enterprise Services Workplace helps you find existing services, while the SAP Discovery system provides a hand-on environment in which you can try out SOA-enabled solutions and modify enterprise services. Additional support services, workshops, and training are available from SAP Services, while opportunities for sharing experiences with SAP customers, developers, and business process experts are available in the Enterprise Services Community program.

Making SOA Easier

On their journey to SOA, companies will come to a critical junction, a place where SOA is no longer an abstract theory, but something tangible they can employ in their organizations today. By taking the next step, they can quickly adapt their business processes to address the accelerating pace of change in their respective markets – a feat that was very difficult in the past because any process or functionality change had to be hard-coded into the back-end application. Today, companies are free to leverage design-time and ready-to-use enterprise services for faster time to value in their SOA initiatives. Through composite applications that follow this design-time process and draw on enterprise services, IT organizations set themselves up for rapid innovation on top of – and by reusing – existing IT assets.

This is the perfect time to get started with your SOA project. Armed with the guidance provided here, drawing on SAP solutions, and supported by experts from the SAP ecosystem and partners, you are ready to make SOA a reality for your company.

Using SAP's approach to SOA, companies can:

- Quickly adapt business processes. SOA enables the IT organization to separate interface and process definitions from the underlying applications. This results in faster implementations and cost-effective upgrades, deployed as needed and without the business interruptions of "big-bang" approaches.
- Innovate without disruption. Most significantly, SOA enables SAP partners and customers to deploy innovative solutions that take advantage of existing IT systems and other assets. With SOA, companies can run new processes on top of existing applications.
- Extend and automate their value network. SOA allows companies to model and automate the terms of business relationships, so they can extend business models and value chains with optimum speed and transparency.
- Accelerate the time to value of their SOA initiatives. SAP products and the collaborative nature of the SAP approach to SOA reduce risks and make SOA a cost-effective option.
- Outsource tasks. SOA facilitates access to external expertise and services, allowing companies to connect to external partners while focusing on core competencies, retaining visibility into and control of critical processes.

Enterprise services make it easier for companies to achieve their goals of strategic agility, differentiation, and accelerated growth. SAP is committed to helping customers achieve these goals faster through SOA-enabled business software solutions and a rich set of enterprise services for a wide range of industries.

What Approach to SOA Do Companies Typically Take?

A study conducted jointly by SAP and University of St. Gallen, Switzerland, concludes that business-driven SOA initiatives deliver results superior to those of technology-driven SOA initiatives.[1] The study also concludes that success in SOA projects requires the alignment of IT with business. SAP acts upon this conclusion by providing SOA-enabled solutions, complete with industry-focused enterprise services and a design-time process that pave the way to accelerated SOA adoption. Design-time best practices shorten the SOA learning curve and help companies consolidate and govern IT along business priorities as well as optimize IT operations without disrupting the business.

1. Dr. Stephan Aier and Tobias Bucher, *Kritische Erfolgsfaktoren initialer SOA-Projekte – Ergebnisse einer empirischen Studie [Critical Success Factors in First-Time SOA Projects: Results of an Empirical Study]* (University St. Gallen, January 2008).

ADDITIONAL INFORMATION

Documentation

Enterprise SOA Development Handbook:
www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/40db4735-02f9-2a10-b198-a888a056bb67

Enterprise Services Enhancement Guide:
www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/c0bb5687-00b2-2a10-ed8f-c9af69942e5d

Web Sites

SAP Developer Network: www.sdn.sap.com
SAP ecosystem and partners: www.sap.com/ecosystem
The Open Group Architecture Forum (TOGAF):
www.opengroup.org/architecture

ACKNOWLEDGEMENTS

The following individuals have contributed to this book:

Dirk Ammermann,
Thomas Eckert,
Sindhu Gangharadan,
Raimund Gross,
Rudolf Hois,
Wulff-Heinrich Knapp,
Jewgeni Kravets,
Carsten Linz,
Randi Polanich,
Itai Sadan,
Ulrich Scholl,
Michel Serié,
Goran Stoiljkovski,
Daniel Stollenwerk,
Thomas Volmering,
Michael Zimpel,

and all contributors to the *Enterprise SOA Development Handbook*.

GLOSSARY

B **business object**

A business object is a logical object of significance to the business. It represents a class of entities with common characteristics and common behavior describing well-defined business semantics. Business objects are used to model a business process and represent a specific view on business content.

B **business process management (BPM)**

BPM has its roots in workflow management. Workflow management applications are primarily concerned with the execution of activities in a predefined but revisable order called workflow. BPM, in contrast, addresses a broader scope; it comprises methodologies, modeling techniques, and tools to define, simulate, and validate process flows. The term is embraced by the major vendors of service-oriented architecture (SOA) middleware as well as by analysts.

B **business process platform**

A business process platform is the combination of a technology infrastructure and business applications (fundamental business functionality provided by highly reusable parts of SAP software). It can include enterprise services as well as the infrastructure to deploy and manage enterprise services and to create composite applications. SAP partners and customers can use these components to create their own business process platform, adding composite applications to enable a specific business solution.

C **composite application**

Sitting on top of other applications, a composite application cuts across functional silos. It generally adds its own business logic according to business requirements and provides specialized user interfaces to fully support a business scenario.

C **composition environment**

A composition environment serves for modeling, connecting, compiling, deploying, and maintaining composite applications that support business processes. The SAP NetWeaver® Composition Environment offering exemplifies the tools and runtime environment for developing, running, and efficiently managing composite applications.

C **Core Components Technical Specification (CCTS)**

CCTS is a specification driven by United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). It defines meta models and rules necessary for describing the structure and contents of conceptual and physical or logical data models, process models, and information exchange models.

E **enhancement package**

An enhancement package is a collection of new and improved functions for SAP® Business Suite applications, such as the SAP ERP application. By means of these packages, customers can take advantage of ongoing business innovation while keeping their core applications stable. Customers can configure these optional enhancement packages in a completely modular fashion, by switching on only the desired features.

E **enterprise service**

An enterprise service is a Web service that has an enterprise-level business value. Enterprise services are typically an aggregation of Web services of finer granularity combined with simple business logic; they support a business process step in a reusable way. Enterprise services provide business semantics; they are structured according to a harmonized enterprise model based on process components, business objects, and global data types. They ensure stable interfaces for the future and are well documented, complete with dependencies, configurations, and so on.

E **enterprise service bus (ESB)**

This software architecture construct is typically implemented by technologies that provide basic services for more complex architectures via an event-driven and standards-based messaging engine (the bus). An ESB generally provides an abstraction layer on top of an implementation of an enterprise messaging system. ESB does not implement a service-oriented architecture but provides the features with which one may be implemented. Although it is a common belief, ESB is not necessarily Web-services based.

E **Enterprise Services Repository (ES Repository)**

ES Repository is a central repository for modeling enterprise services – defined by customers, SAP, and partners – and storing their metadata. ES Repository is an integral part of future releases of the SAP NetWeaver technology platform.

G **global data type**

Global data types are based on CCTS, which allows for the definition of generic data types (for core components) and data types for a specific vertical industry. An example of a generic global data type is "sales order," which can be extended for the needs of a specific industry.

P **process component**

A process component is a building block of a service-enabled solution. It is a modular, context-independent, reusable piece of software that exposes it functionalities as services. A process component contains at least one business object.

S **SAP Discovery system**

In collaboration with selected hardware partners, SAP offers the SAP Discovery system. This system can be a "sandbox" where customers take their first steps towards SOA, using it as a "proof of concept" and to identify gaps in their approach. With the SAP Discovery system, developers and enterprise architects can try out the composition of new applications using enterprise services in a test environment. The SAP Discovery system delivers a preconfigured SOA landscape that grants customers and partners immediate access to the latest software and tools available from SAP as well as a comprehensive set of sample business scenarios.

S **SAP Enterprise Architecture Framework**

The SAP Enterprise Architecture Framework methodology for SOA enables an enterprise to transform from its current or "as-is" architecture to a strategic or "to-be" architecture. Enterprise architecture can be thought of as an equation. In its purest form, the equation is: enterprise architecture = strategy + business + technology. It answers the question of how you align IT with business and with strategy.

S **service**

A service is functionality exposed via fixed interfaces and using open standards. A service consists of a contract (constituted by the interfaces' definitions), one or more interfaces, and an implementation.

S **service-oriented architecture (SOA)**

SOA is a distributed software model within which all functionality is defined as independent Web services. Within SOA, these Web services can be used in defined sequences according to business logic to form applications that enable business processes. SAP delivers a comprehensive portfolio of SOA-enabled solutions that can help businesses of all sizes transform their existing IT infrastructures into more flexible business process platforms.

T **The Open Group Architecture Framework (TOGAF)**

Devised by The Open Group, TOGAF is a framework, including a method and a comprehensive set of supporting tools, for the development and governance of an enterprise information architecture.

W **Web service**

A Web service is a standards-based way of encapsulating the functionality of an application that other applications can locate and access. For example, a Web service could be the deletion of an order record in a sales management application.

A Web service is available over the Internet and uses a standardized XML messaging system, such as SOAP. It is not tied to any operating system or programming language, is self-described via a common XML grammar, and is discoverable via a standardized mechanism, such as Universal Description, Discovery, and Integration (UDDI). The original and rather simplistic definition of a Web service did not include the last two requirements.

W **Web Service Description Language (WSDL)**

WSDL is an XML-based description of Web services.

50 092 434 (08/10)