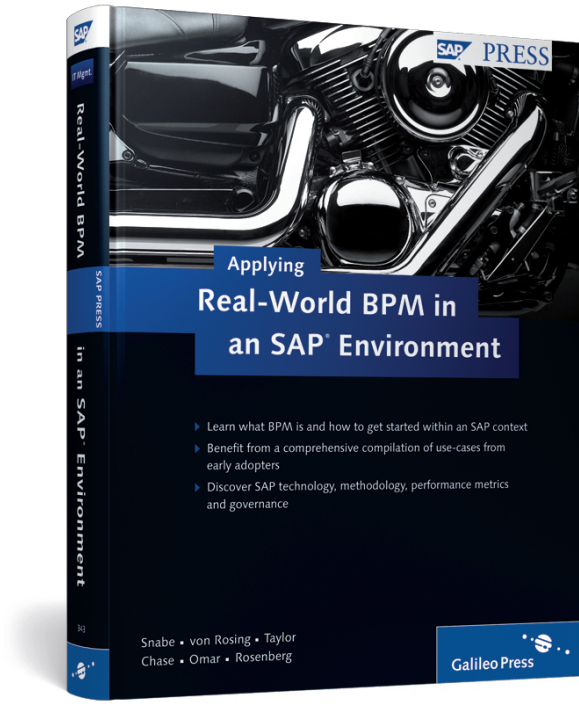
SAP PRESS

Ann Rosenberg, Greg Chase, Rukhshaan Omar,
James Taylor, and Mark von Rosing

Applying Real-World BPM in an SAP® Environment



www.sap-press.com

Galileo Press

Bonn • Boston

Contents at a Glance

Contents

1.5 New Form of the Business Model Concept

With the components development, *logic* and *value* became key words in the literature on business models. In recent years IBM Global Services and the IBV (IBM Center for Business Value) have developed an innovative business model approach that includes business model design, business model innovation, and business model transformation. The IBM business model approach is called the Component Business Model (CBM). This was the first time somebody not only used a general method to identify core competencies (resources and capabilities), partner networks, value proposition, customer segments, and relationships, and thereby cost and revenue, but used a logical representation and technique to map the enterprise on a single page. This CBM approach can be used to analyze the alignment of enterprise strategy with the organization's capabilities and investments, identify redundant or overlapping business competencies/capabilities, analyze sourcing options for the different components (buy or build), prioritize transformation options, and create a unified roadmap after mergers or acquisitions. The model is organized as business components along columns and "operational levels" along rows. Business components are defined as large business areas with characteristic skills, resources, processes, and competencies.

The three operational levels (depending on industry) are planning, monitoring, and execution. They separate strategic decisions (planning), management checks (monitoring), and business actions (execution) on business components. This new approach to business modeling took the concept of the business model to a higher and more strategic level. A split began to form between business model approaches, where many of the business modeling approaches continue to focus on functional component requirements without paying sufficient attention to the other nonfunctional issues. The result is a final product that is unsatisfactory and fails to comply with the strategic business objectives of its users. Therefore, the perspective of business model innovation and transformation is not jointly unified.

The other development approaches focused on resources (assets), capabilities, and thereby competencies, which are combined competencies, which a company needs to plan, create, and realize value, in both an effective and efficient way (see Figure 1.6).

In order to plan, create, and realize value in an effective and efficient way, a company should identify the key value drivers of innovation (see Chapter 3), which should yield rewards rather than extra cost in building such competencies. Understanding which processes and initiatives it takes to design, innovate, and

transform one's business competencies to match the vision and strategy that is needed should not only help a company gain cost improvement, but differentiate advantage as well.

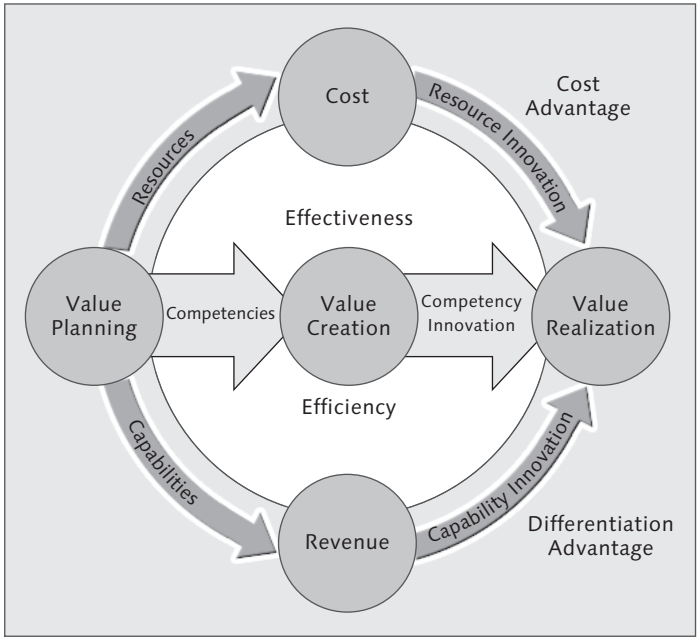


Figure 1.6 Competency – Value Model (von Rosing, 2009. "Business Value Management")

To gain a differentiated advantage, one or more competitive strategies should be chosen. Becoming a leading-edge company occurs by taking offensive or defensive action to create pioneering competencies, and therefore the prime position in an industry or the market, in order to cope successfully with the competitive forces and choices the peers in the industry, have chosen and consequently generate a superior return on investment. When the principles of competitive advantage strategies (from Porter [Porter, 1998]) are applied, there are two basics types of competitive advantage:

- Cost leadership (low cost)
- Differentiation

Both can be more broadly approached or narrowed to be more specific, which results in the third viable competitive strategy: focus.

A competitive advantage exists when the firm is able to deliver the same competencies and benefits as its competitors, but at a lower cost (cost advantage), or deliver competencies that exceed those of a competing organization (differentiation advantage). Thus, a competitive advantage enables the organization to create superior competencies, and in this manner value, for its customers and superior profits for itself. Cost and differentiation advantages are known as positional advantages since they describe the firm's position in the industry as a leader in either cost or differentiation.

However, contrary to the rationalization of Porter, contemporary research from Kim Chang in 1997, 1998, and 1999 has shown evidence of firms practicing a successful mixture of low cost and differentiation strategy. Research literature by Prajogo [Pajogo, 2007] state that firms employing the hybrid business strategy (low cost and differentiation strategy) outperform the ones adopting a single generic strategy. Sharing the same view point, Charles Hill argues in his paper, "Corporate strategy and Firm Performance," that a successful combination of these two strategies will result in a long-term competitive advantage. As an example, combining these two strategies in one's competencies is successful, when combining a market segmentation strategy with a product differentiation strategy is an effective way of matching your firm's product strategy (supply side) to the characteristics of your target market segments (demand side).

However, combinations such as cost leadership and differentiation in one's competencies are hard (but not impossible) to implement, due to the potential for conflict between cost minimization and the additional cost of value-added differentiation. To achieve a competitive advantage, the firm must perform one or more value-creating competency activities in a way that creates more overall value than do competitors. Superior value is created and realized through lower costs or superior benefits to the consumer (differentiation). In this case a company needs to define on a high level how competitive advantage is created. After the strategies of cost and differentiation leadership are chosen, many writers [Robert M. Dibrell; C. Clay; Kim, Eonsoo Nam; Daeil, Allen; R. Helms, M. Takeda; M. White C.; Stimpert, J.L.] argue that specific and multiple business strategies need to be specified and applied in order for companies to carry out the chosen strategies. According to the resource-based view [Wernerfelt, B., Hoopes, D.G.; Madsen, T.L.; Walker, G], in order to develop different underlying competitive advantage strategies that supports cost advantage and differentiation advantage, the firm must apply the strategies to the resources and capabilities, and in doing so the competencies of the company. As illustrated in Figure 1.6, both resources (assets) and capabilities, which are combined competencies, need to be innovated in order to create and

realize value, both in an effective and efficient way. Without applying a cost and/or differentiation approach to one's competencies, the competitors simply could replicate what the organization is doing and any advantage could quickly disappear.

Let's go into the details of each of the capability/resource innovation elements in the competency – value model. We will start with resources.

1.5.1 Resources

Resources (also called assets) can be categorized as tangible, intangible, and human. Tangible assets can be physical, such as plants and equipment, or financial, such as cash. These are the types of assets that are usually identified and accounted for in financial statements under the category "assets." Intangible assets are nonphysical and nonfinancial assets such as patents, brands, copyrights, trade secrets, market research findings, relationships with customers, knowledge in databases, and relationships with vendors. They are usually not identified in financial statements but can be excellent sources of profits. For example, a patent or trade secret that gives a firm exclusive access to a product or process may allow the firm to be the only one producing a product with certain characteristics, thereby making the product highly differentiated and profitable. For a while, the copyright for Intel's microcode allowed the firm to offer differentiated microprocessors to makers of personal computers. Human assets are the skills and knowledge that employees carry with them [Afuah, 2003]. As shown in Figure 1.6, for the most part resources are associated with cost, and therefore resource innovation leads to cost advantage. Resources are the organization-specific assets that are useful for creating a cost advantage; however, certain resource/assets can be applied to differentiation advantage as well, all which few competitors can copy or acquire easily. Resources are inputs into a firm's process, such as capital, equipment, and the skills of individual employees, patents, finance, and talented managers. With increasing effectiveness, the set of resources available to the firm tends to become larger. Individual resources may not yield to a competitive advantage. It is through the synergistic combination and integration of sets of resources that competitive advantages are formed. The following are some examples of such resources:

- Patents and trademarks
- Proprietary know-how
- Employee relations and commitment
- Employee morale

- Installed customer base
- Reputation of the firm
- Brand equity

1.5.2 Capabilities and Abilities

As important as resources/assets are, it usually takes more than resources and assets to offer value to customers. One such additional value is a firm's capabilities.

Capabilities

Capabilities refer to the firm's ability to utilize its resources efficiently. A capability is the capacity for a set of resources to interactively perform a stretch task or an activity. Through continued use, capabilities become stronger and more difficult for competitors to understand and imitate. As a source of competitive advantage, a capability should be neither so simple that it is highly imitable, nor so complex that it defies internal steering and control [Schoemaker and Amit, 1994]. As shown in Figure 1.6, capabilities are associated with revenue potential for the most part; therefore capability innovation should lead to any form of differentiation that would give the company a differentiated advantage.

An example of a capability is the ability to bring a sustainable product to market faster than competitors. Such capabilities are embedded in the processes and activities of the organization, and should be documented from main processes group to sub processes, and thus are difficult for competitors to replicate. The firm's resources and capabilities together form its distinctive competencies. The competencies are mapped on business model level, as described in Part III, Chapter 13. In order to develop one's competencies (resources and capabilities) actively, a firm must:

- Analyze as-is and to-be competencies
- Define process value drivers
- Implement process measurement
- Define continuous improvement of processes
- Develop process performance metrics
- Innovate processes
- Initiate a process governance model

The active planning, creation, and realization of such competencies strengthens the firm's brand, reputation, enables innovation, meets customer sustainability needs, builds employee relations, productivity, efficiency, quality, all of which can be leveraged to create a cost advantage or a differentiation advantage.

We can conclude that since the goal of a business model is to make money, a task that must interest an organization is the pursuit of a business model. Therefore, the firm's attention must focus on the types of capabilities and resource/assets that are most likely to develop the critical core competencies the organization needs to create and realize the planned value, in order to ensure that the business model is profitable. Furthermore, a firm must acknowledge which capabilities and resource/assets are non-core competencies that most likely to ensure low cost, e.g. standardization and automation, which the organization's business model needs in order to be competitive and profitable.

Abilities

A firm needs to have the ability to convert its resources and assets into competencies that create value (internal and external). Customers will not scramble to a firm's doors simply because the firm has modern resources and assets such as plants, geniuses, and patents. The firm has to use the plants, the geniuses, and the knowledge embodied in the patents to offer customers something they value. Patients do not buy patents or skilled scientists from pharmaceutical companies; they buy medicines that have been developed by skilled scientists using knowledge embodied in patents. Assets must be converted into something that customers want. A firm's ability or capacity to turn its resources into customer value and profits is usually called a competence or competencies. Competences usually involve the use or integration of an organization's capabilities and resources/assets. Logic's ability to quickly turn its "cores" into products that customers want is a competence, which can be either core or non-core competencies. Intel's ability to develop microprocessors that exploit its copyrighted mircrocode and that are compatible with its installed base of microprocessors is a core differentiating competence. So is Coca-Cola's ability to turn its secret formula and brand into a product that many customers perceive as being preferable to its rivals' products [Afuah, 2003.].

Because the goal of a business model is to make money, a question that must interest an organization is: What types of capabilities and resource/assets are most likely to develop the critical core competencies the organization needs to create and realize the planned value and to ensure that the business model is profitable?

1.6 The Logic of a Business Model Framework Based on Competencies

As we have discussed, business models are vital for business design, innovation, and transformation. But putting business innovation and transformation into practice with the right operating model requires executives to think differently, not only about the construct of the organization but also about the interrelationships of the assets they rely on to provide value to their customers. Business models offer a proven approach to driving a critical core competencies focus, both internally and externally. Internally, competencies help organizations rethink the leverage they can achieve with the assets and capabilities they own. Externally, competencies help organizations source specialized abilities they cannot feasibly create themselves.

Combining these types of business innovation and transformation allows organizations to redefine their competitive positions in the face of the sweeping changes in their industries while simultaneously achieving the competing benefits of scale, flexibility, and efficiency. In Figure 1.7 we see an example of a general business model with its business competencies, which are the modular building blocks that make up a business model.

Business Administration				Business Operations			
General Administration	Human Resource Management	Information Technology	Operations Support	Business Development	Operations	Distribution	Marketing, Sales, and Service
Strategic Planning	Organizational Planning	IT Planning	Operations Support Planning	R&D Planning	Operations Planning	Distribution Planning	Segmentation Planning
Legal & Regulatory Affairs	Recruitment	Deployment	Assets	Product Design	Component Manufacture	Scheduling	Selling
Information Analysis	Administration	IT Business Management	Quality	Research	Operations Procurement	Order Fulfillment	Market Analysis
Project Management	Benefits	Risk & Compliance	Environment & Health	Production Setup	Product Manufacture	Transportation	Channels
Finance	Performance Evaluation	Information Management	Sourcing & Procurement	Intellectual Property	Inbound Inventory	Import & Export	Brand Management
Facility Management	Compensation	Service Delivery	Safety & Security	Product Deployment	Product Assembly	Distribution	Customer Account
Accounting	Education	Development	Equipment & Plant	Content	Refining	Finished Goods Inventory	Customer Acquisition
Travel Management	Payroll	Support & Relationship	Data Management	Product Maintenance	Packaging	Costing	Servicing

Figure 1.7 General Business Model Combining Business Competencies (von Rosing, 2009. "Business Value Management")

Each competency of the business model encompasses seven dimensions (see Figure 1.8):

1. The competency purpose and service — the logical reason for the competency's existence within the organization, as defined by the service, and thereby value it provides to other competencies.
2. Each competency conducts a mutually exclusive set of activities to achieve its business purpose and thereby create value. This is where processes are interconnected.
3. To create value, competencies require resources — the people, knowledge, and assets that support their activities and processes.
4. Each competency consists of capabilities that use different resources, and therefore company assets.
5. Each competency is managed as an independent entity, based on its own governance model to ensure performance and value realization.

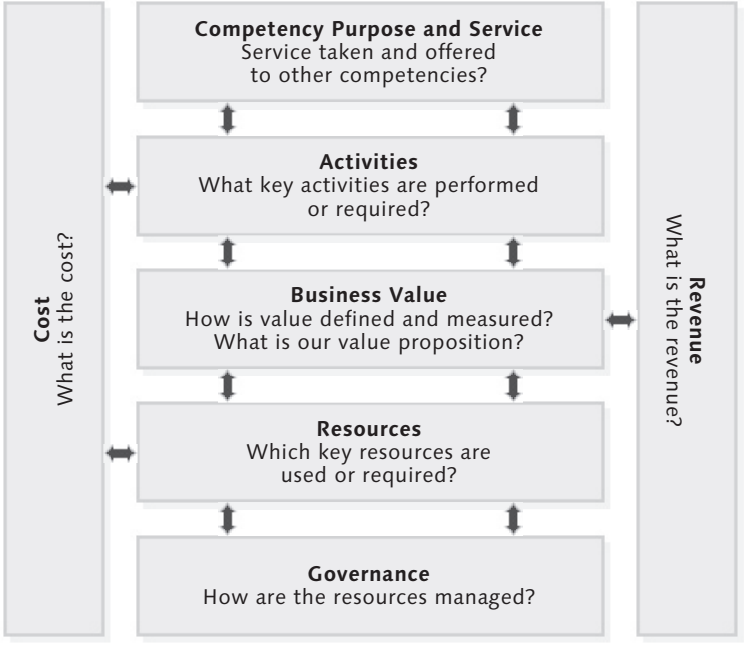


Figure 1.8 Seven Dimensions of a Competency

6. The different competencies each have a cost that can be either measured, optimized, or developed, depending on whether it is a core or non-core competency. A ground rule here is that cost reduction should for the most part only be done within non-core competencies.
7. The revenue a competency generates can be measured, developed, and innovated depending on whether it is a core or non-core competency.

It is important that when determining the boundaries of the competencies, you consider the various dimensions, not just one or two, even though one or two might be the drivers, for cost cutting, value creation, and/or governance. All of the competencies are highly collaborative, working with other competencies both inside and outside the company. Collaboration is accomplished through the exchange of services, activities, and inputs and outputs for all competencies. When a competency requires an input to complete a particular activity, it procures it as a service from another competency. That way it can access the full range of inputs it requires. This competency will in turn provide an output that other competencies can use as their input. Predefined service-level agreements — covering such aspects as formatting, timing, quantity, quality, payment, and provisioning — set the standards for all of these transactions.

Business competencies derive much of their advantage from two related but distinct traits: The loose coupling of links between competencies provides flexibility, adaptability, and responsiveness, whereas the cohesion of activities within each competency provides efficiency and enhanced quality.

1.6.1 Flexible and Free Connection of the Competencies

Interaction and connection between competencies is characterized by flexible, loose, and free coupling. Instead of "hardwired" inflexible links based on proprietary or customized connections, competencies interface through clearly defined service boundaries, forming and breaking connections as they initiate and respond to service requests to each other. Flexible coupling also relies on a common sharing principle, so that even incompatible underlying systems can be joined based on competency value-added communication. This aspect of competency sharing and thereby development gives organizations much more scalability in the services they provide and use among each other and more flexibility in deciding whether to source a competency within the firm or outside it. In either case, the competency requesting an input in terms of service is indifferent to how that service is

implemented. Please note that we are not talking about SOA services, but how competencies give value-added services to each other.

From the process perspective this would be competency activities that are illustrated in a process model as value-added chain diagrams. Value-added chain diagrams are used to illustrate and identify those competency activities within the company that are directly involved in the creation of a company's added value. These functions can be interlinked by creating a competency and function sequence and thus a value-added chain. Such a value-added chain diagram not only enables you to express a subordination of competency activities; it can also display the competency functions' links to the business model and thereby the organizational units and information objects. From the outside a competency is a "black box" whose inner workings are irrelevant. The possibilities of developing business competencies and building on the value-added services they can share with each other is a great potential for value creation and realization within an organization (refer to Chapter 2).

1.6.2 Consistency and Union of the Competencies

Internally, competencies deliver scale and efficiency gains through consistency, the union of similar competency activities from across the organization into a business competency group. To achieve cohesion, each competency activity must belong uniquely within one competency group, with no duplication within or between competencies (because they should share these competencies as described in Section 1.6.1). An added benefit of bringing these competency activities together in a business model competency group is to expose the relative performance discrepancies between competency activities and others that are not performing and thereby create high costs for executing the competency. In this area there are great possibilities for developing, optimizing, and innovating competency activities for cost cutting and value creation and realization within an organization (see Part III, Chapter 14, for more information).

1.7 Organizing Business Competencies

Business modeling provides a framework for organizing competencies by accountability level. By employing such a framework, executives can begin to envision how current business activities might function as an interlocking set of modules.

Categorizing activities by business competency yields a high-level view of competencies according to the type of value they provide to the enterprise. Different firms in different industries model their competencies differently, but in every case, each activity should line up under a particular competency.

Examples of competencies are HR, operations, distribution, business development (see Figure 1.6 and Figure 1.7).

Assigning each activity to one of three accountability levels — strategic, tactical, and operational — can also help executives begin to flesh out the business competency development vision/roadmap (see Figure 1.9). The level of a given competency should be intuitive, although exceptions exist. The three accountability levels are defined as follows:

1. **Strategic**
   Competencies at this level provide strategic direction, planning, and corporate policy to other competencies. They also facilitate collaboration with other competencies. These strategic competencies provide the business actions that drive value planning in the enterprise.
2. **Tactical**
   These mid-tier competencies serve as control, monitoring, checks, and balances between the strategic and operational levels. They monitor performance, manage exceptions, and act as gatekeepers of assets and information. These tactical competencies provide the business actions that drive value planning, monitoring, and governance in the enterprise.
3. **Operational**
   These low-level competencies provide the business actions that drive value identification and creation in the enterprise. They process assets and information for use by other competencies or the customer.

The three accountability levels imply different priorities. At the operational level, for example, the emphasis is on keeping people fully occupied and productive. From a technology standpoint speed of data entry and real-time availability are important. Contrast this with activities related to the strategic tier, where such high-level activities as planning and launching new products are handled. This level houses a small number of people who have a very large impact on shareholder value.

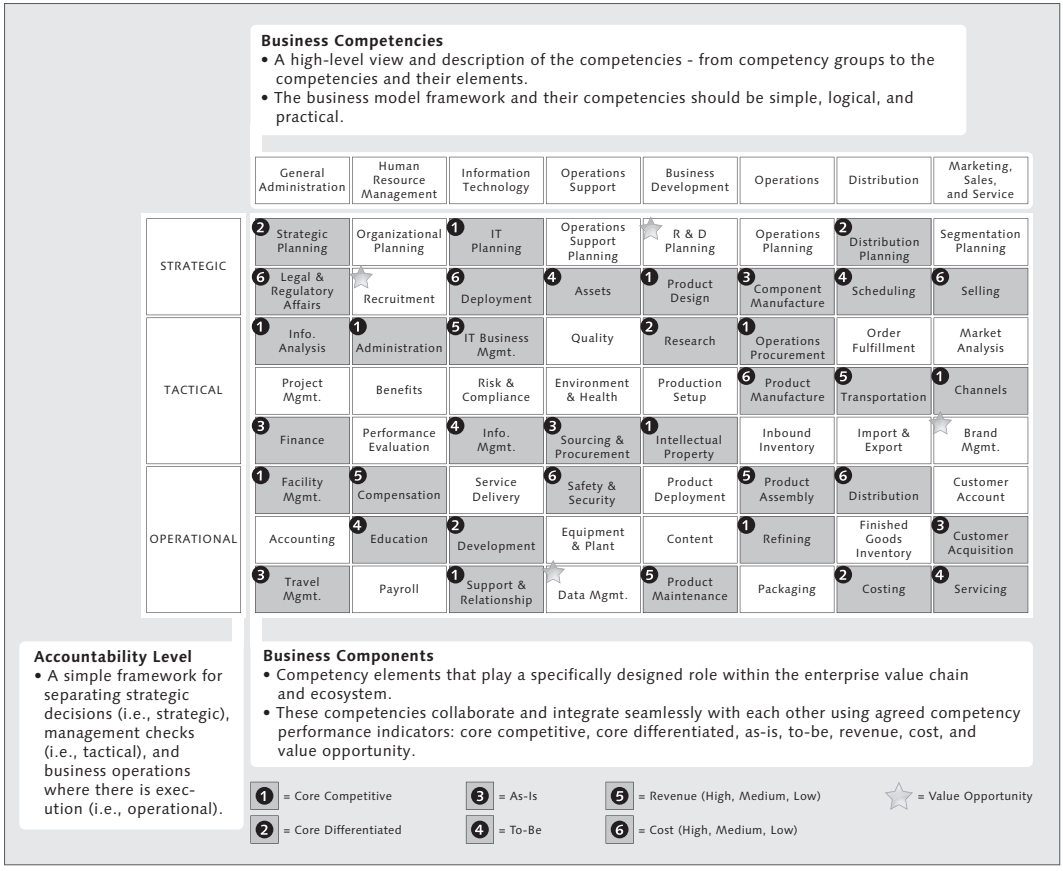


Figure 1.9 Business Model with the Three Accountability Levels (von Rosing and von Scheel, 2010. "How to Identify, Plan, Create and Realize Value")

Launching a new product requires collaboration among several elements, including marketing, risk, finance, regulations, and credit. Input from all of these stakeholders is needed to make the launch a success, so workflow is a key requirement. From a technology standpoint, activities typically require people (resources) and capabilities to discern patterns and trends from rich, multidimensional data, usually stored in a data warehouse. Systems at the strategic level are not designed for speed of data entry, but rather for ease, breadth, and depth of analysis. Real-time interfaces are not needed, because data is often months old and processed in batches.

To drive as much revenue, value creation, and realization as possible from business model competency development, only core competitive and core differentiated competencies across the firm are aggregated. It is an organization's CCCs that enable an organization to outperform its rivals. These competencies with the attached seven dimensions should, when automated and supported with an IT system, be treated as own practice. Far too often such competencies are automated with the IT system's Best Practices, and therefore their uniqueness and differentiation can potentially be destroyed.

Whereas the IT system Best Practices are vital to cut costs, for example, fast implementation, fewer mistakes, standardization, and less risk, because it is proven to work, this can't be applied in the area of CCCs that enable core competitive and core differentiated competencies. However, even if a firm offers the right customer value to the right market segments and does so better than its rivals, it is still possible that the firm might not be profitable. That is, superior relative customer value offered to the right customer segments, although necessary, is not always a sufficient condition for profitability. Therefore, the cost of making money is a vital ingredient for succeeding. Offering the right value to the right customer segments and being positioned advantageously vis-à-vis suppliers, customers, rivals, potential new entrants, and substitute products may still not be enough for a firm to capture the revenues that its positions suggest it should. To keep the cost low a company should standardize its NCCs and thereby apply IT system Best Practices to all NCCs and the attached main and supporting processes.

1.8 Summary and Conclusion

We have discussed the important fact that a business model is not a strategy. The separation of model from strategy is the strength and weakness of the business model concept. Because the business model is the product of the strategy, a business model can only be as strong as you're your strategic business objectives (SBOs), critical success factors (CSFs), and key performance indicators (KPIs). If a company wants business-IT alignment, it needs to align the strategy to the business model, the business model to the process model, and the process model to its systems — and all of this with the right architecture and governance framework.

The primacy of how the strategy does and should interlink with the business model is apparent in Michael Porter's influential strategic framework and value chain framework. See the mention of Porter [1994] in the Bibliography for his

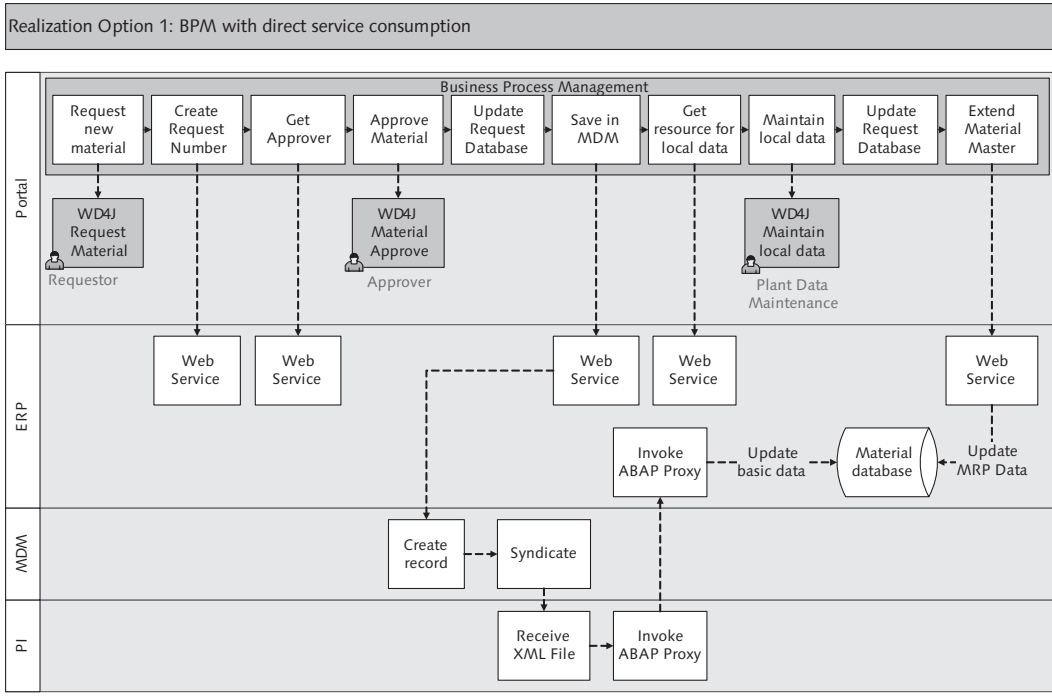


Figure 7.2 Sample Architecture Utilizing SAP NetWeaver BPM Platform

7.2 Ericsson: Using Business Rules to Enable Globalization of Supplier Master Data Governance

The many country-specific requirements make it difficult to consolidate supplier master data governance globally. By leveraging BPM and business rules to guide users through these requirements, the project has delivered a robust and user-friendly solution.
– Stacey Drinan, IT Platform Area Manager, Ericsson AB

Ericsson, a global supplier of telecommunications equipment, has been engaged in a multi-year program of continuous improvement of its master data processes. In its latest project, Ericsson sought to add an additional level of usability to its supplier master data processes. The company incorporated business rules into their process automation to guide correct user input of supplier data, which had specific and individual needs for the countries where the suppliers are located. This project

is expected to speed up the master data process by over 50% — and with a high business-user acceptance rate.

7.2.1 Background

Ericsson is a world-leading provider of telecommunications equipment and related global services to mobile and fixed network operators. Over 1,000 networks in more than 175 countries utilize their network equipment, and 40% of all mobile calls are made through Ericsson's systems. Ericsson is one of the few companies worldwide that can offer end-to-end solutions for all major mobile communication standards. For more information please see *www.ericsson.com*.

Business Case

Ericsson is a truly global company and, as such, the sourcing organization is global, with interaction and agreements made with suppliers in over 200 countries. The scope of the agreements varies from global agreements that are applicable for all of Ericsson, to agreements specific to only select local Ericsson operations.

From a strategic perspective, Ericsson was consolidating its business support systems into a few global systems. Related to the sourcing processes, Ericsson had:

- Two global SAP systems (that were being merged into one at the time) where all procurement was done
- A global SAP business warehouse for follow-up and reporting
- A global contract management system
- A global tool for strategic sourcing

In this environment, accurate and high-quality supplier master data was key to ensure that the business processes ran smoothly and that accurate reporting could be performed. It was also crucial that suppliers could be identified across the global sourcing systems to ensure this quality in processes and reporting.

Ericsson had worked with master data quality in the global systems for many years. At the time that Ericsson decided to implement BPM, a central global master data management group was responsible for the process of maintaining master data in the global business systems, on request from the business units.

The business case for this project included two goals: first, increasing master data quality to increase efficiency in overall business processes and accuracy of analytics and, second, increasing efficiency in the specific process of creating and maintain-

ing supplier master data. The Global Master Data Management group handled around 3,000 requests monthly for supplier master data updates. The business case calculation was based on a 50% reduction of handling time per request. The solution goal would be to capture the request handling time for each request for easy reporting on this important KPI.

As-Is Process

In the original process, a request for master data changes for supplier master data was made by sending an Excel form to the Global Master Data Management group. Any issues with the request were handled by emailing back and forth between the business user who was requesting the change and the responsible person at the Global Master Data Management group. Once the request was validated, the responsible person at the Global Master Data Management group had to enter the data manually in up to four different systems, depending on the type of request.

In typical business fashion, many of the business rules detailing what information had to be entered into the supplier master data were scattered in different instruction manuals and in the heads of staff members. The business rules state whether data is mandatory or optional, qualifies as default values, requires specific formatting of data, and so on. Many of these business rules dependent on in which country the supplier is located or in which country the supplier is to be used. Global and regional alignment cause these rules and requirements to change frequently. For supplier master data, such business rules are related to the following types of data:

- Payment methods
- Bank details
- Tax numbers
- Withholding tax
- Business partner roles in SAP
- DUNS numbers
- Vendor returns

7.2.2 BPM Solution

The vision for the project was to streamline (see Figure 7.3) and automate (see Figure 7.4) the process via the following solutions:

1. **Self-service creation**
   Business users access online request forms in the SAP NetWeaver Portal. The forms guide the users to enter data correctly, depending on the scenario and other key selection criteria (i.e., country).
2. **Intelligent guidance**
   Online forms guide the users to enter data correctly, with business rules adding "intelligence" to the forms to help business users with specific requirements, such as a country's tax information.
3. **Orchestrated request routing**
   The request is routed through a workflow, using SAP NetWeaver BPM, to the Global Master Data Management group for enrichment and validation.
4. **Automated data syndication**
   When the request is approved, the data is automatically posted in SAP NetWeaver Master Data Management (MDM) and automatically syndicated to all relevant consuming systems.

With this solution, data integrity and data quality are ensured with a minimum of manual intervention and without copying and pasting of data at any stage. Also, the solution guides the business users to enter the correct data in their requests from the beginning, which significantly reduces the overall lead time for processing the request and increases business user satisfaction with request handling.

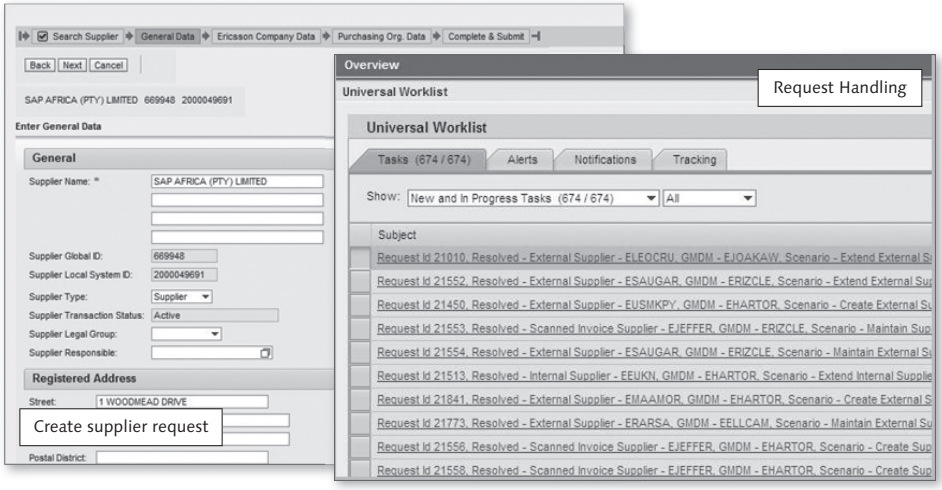


Figure 7.3 Streamlining the Request Process in the Core To-Be Process

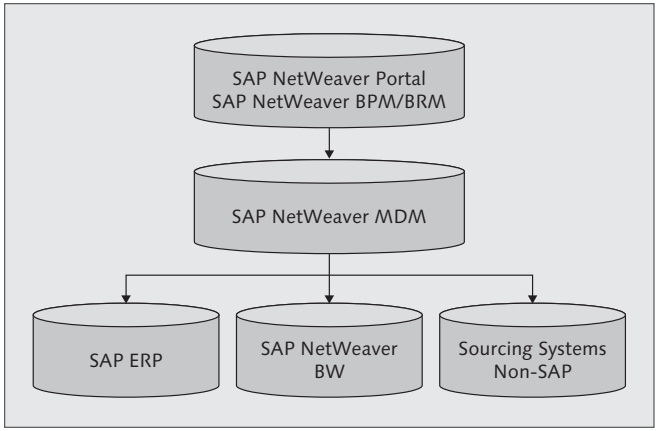


Figure 7.4 Automated Syndication of Data as Part of the To-Be Process

Implementing the Solution

The project ran for about 1.5 years with the following phases:

1. Prestudy/business justification: four months. The business case was developed simultaneously with Ericsson's beta testing of SAP NetWeaver BPM and SAP NetWeaver Business Rules Management (BRM) to evaluate the feasibility of the upcoming project. During the feasibility evaluation, parts of an existing guided procedures–based scenario were ported to BPM to understand how well this would support Ericsson's requirements.
2. Requirement gathering: five months. This stage focused on data model and process aspects. Meanwhile, the IT team was entering the ramp-up program for SAP NetWeaver BPM.
3. Iterative design-prototype-test cycles: six months
4. Final build: one month
5. Final test and go-live preparation: two months
6. Go-live and business rollout: one month

The architecture of SAP NetWeaver BPM and SAP NetWeaver BRM allowed for a very efficient project development model, with a lot of interaction between the lead developers and lead business users in rapid design-prototype-test cycles. This ensured that all key business requirements were captured early in the process and that key business users got an early feeling for the new process and solution.

A lot of effort was put into designing a streamlined process and an engaging user experience to speed up rollout and adoption of the new process. Usability specialists were engaged throughout the project. Additionally, a special usability test week was performed with 12 business users who were familiar with the existing business process but had no knowledge of the new solution. This activity captured critically useful information for usability improvements.

The rollout was done globally in a condensed three-wave schedule. This allowed the project to complete the full global rollout within six weeks. To support the rollout, online training material was produced alongside the final developments in the project.

The project was run as a joint business and IT delivery project, where the business was responsible for requirement gathering, business alignment, testing, and business rollout. The IT delivery team was responsible for the design, build, and deployment of the solution. Experts from SAP Partner Ecenta AG delivered the main design and build work around SAP NetWeaver BPM, SAP NetWeaver BRM, and SAP NetWeaver Master Data Management. This had been part of an overall IT project managed through Ericsson's outsourcing partner, IBM.

Results of the Project

During the assembly of information for this use case, the project had been live for a week and the first batch of live requests had been processed. Experience from the last week in production and the user training and business testing sessions clearly indicated that the business users appreciated the solution as intuitive and easy to use. Business users also appreciated the guided approach and the prompt validation of input data using the business rules. Now, correctly entered data is helping Ericsson reach their "zero errors" vision for their business processes. Measurements of request handling time also indicate that the project goal of reducing the request handling time more than 50% is clearly within reach.

Sample Solution Architecture for an SAP Landscape

The solution was built on the existing enterprise IT architecture at Ericsson that includes the following SAP systems (see also Figure 7.5):

- SAP NetWeaver MDM as the single-source-of-truth repository of supplier master data
- SAP NetWeaver Portal for all end-user interaction with the solution

- SAP NetWeaver Process Integration for integrating the different components of the solution
- SAP NetWeaver BPM as the primary workflow and orchestration engine for the new process
- SAP NetWeaver BRM for managing and maintaining business rules related to master data

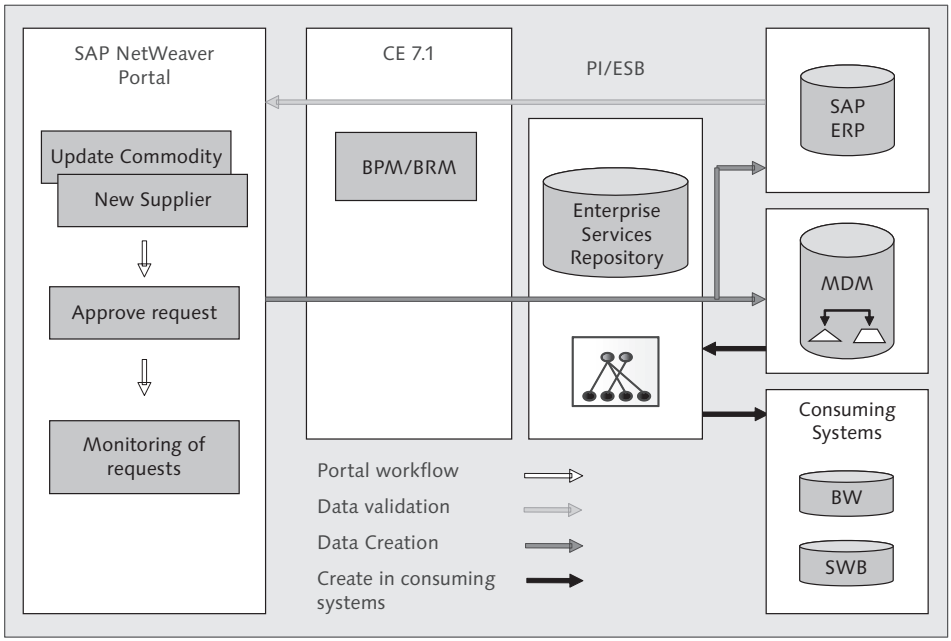


Figure 7.5 Overview of the Solution Architecture

The architecture of the solution consists of four layers with clearly defined responsibilities:

1. The portal layer is used for all end-user interactions with the solution. This layer manages access control to the solution and all screens/workflows, which the end users access through the enterprise portal interface.
2. The CE environment is the host for the SAP NetWeaver BPM and SAP NetWeaver BRM components. SAP NetWeaver BPM is the process engine where the business process logic and workflow are executed, as are the rules engine where the business rules are executed at runtime. The SAP NetWeaver Composition

Environment (CE) also hosts the Java-based enterprise services to manage the interaction between BPM and the MDM/ERP systems.

3. SAP NetWeaver PI is used as the communication middleware for the solution. The enterprise services are registered in the service registry of SAP NetWeaver PI, which manages their runtime execution.
4. Consuming systems are the last level, which consume new and updated master data from the master data management solution. Consuming systems are SAP systems (SAP ERP and SAP NetWeaver BW) and a non-SAP system — the Sourcing Workbench (SWB) — a supplier/procurement management solution.

The process is a request and approval workflow built with SAP NetWeaver BPM (see Figure 7.6). The associated process step implementations included standard and custom-built Web services.

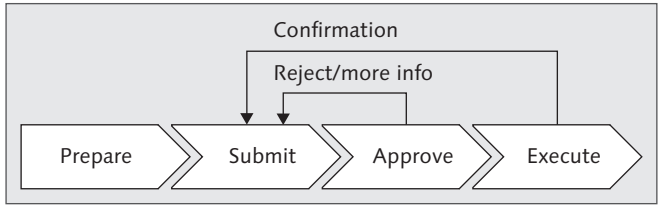


Figure 7.6 Request and Approval Workflow

The process covers various business scenarios:

- Create/maintain external supplier
- Create/maintain internal supplier
- Create/maintain supplier candidate
- Maintain payment block
- Create "other" request

All business scenarios share the same basic workflow structure and the same superset of available attributes for entry or modification. For each scenario, several other key attributes such as supplier, country, and buying organization/country and several business rules are defined. The business rules state which information and attributes should be captured, whether the information is mandatory or optional, and what default values and validation rules apply. All business rules are centrally defined, maintained, and executed in the SAP NetWeaver BRM component. Permitted value ranges are taken from SAP NetWeaver MDM or from the SAP ERP

system. Validation of ERP-related attributes such as payment and shipping details is done by calling services from the SAP ERP system.

Figure 7.7 shows how the business rules are defined in SAP NetWeaver BRM, based on which business scenario is being executed. For a certain scenario (for example, create external supplier), and for a certain attribute (for example, order currency), the rule defines whether the attribute is mandatory, optional, hidden, or read-only (present).

Another of Ericsson's requirements within the program has been to very accurately capture the legal tax and registration numbers for suppliers. The business rules for this information vary by country all over the world. The type of numbers, how many are needed, and format vary by country, but at the same time it is very important to capture this information accurately to ensure a smooth relationship with key suppliers. Ericsson used SAP NetWeaver BRM to define the rules, by country, for which fields should be entered and how they should be validated. Figure 7.7 shows a sample of these rule definitions.

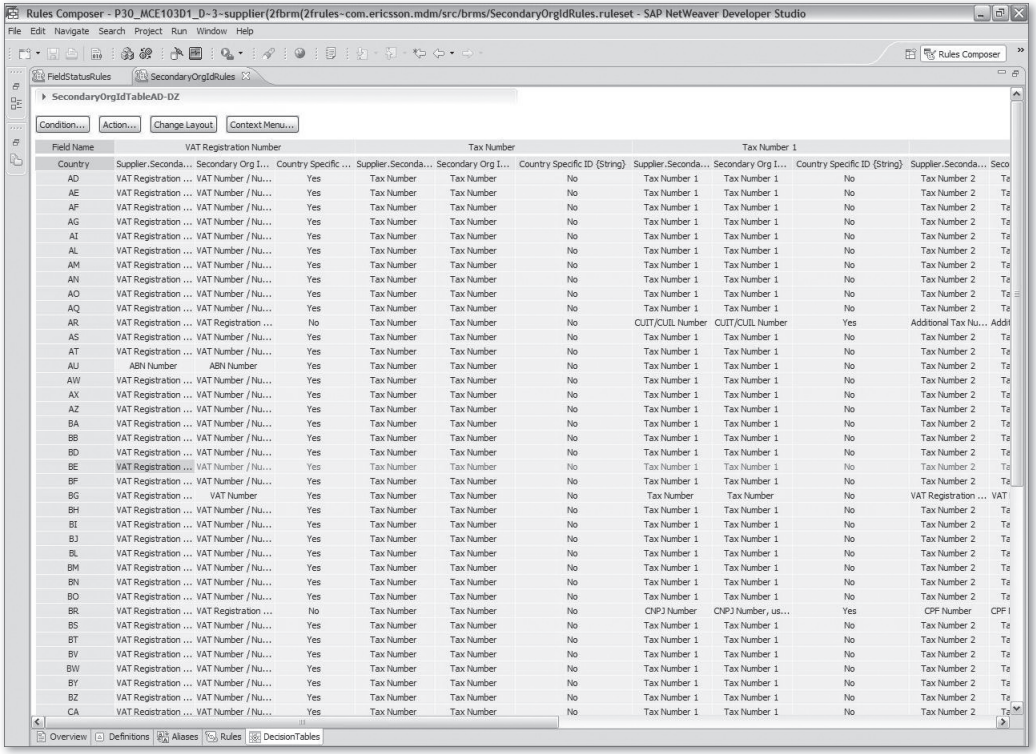


Figure 7.7 Rule Definitions

The rules for capturing tax and registration numbers are expected to change (and improve) frequently. With an integrated tool like SAP NetWeaver BRM, it is anticipated that the master data business experts at the Ericsson Global Master Data Management group will be able to maintain the rules directly, verify them, and then apply the updated rules for the productive application without a big IT project each time.

7.3 SAP IT: Accelerating Postmerger Data Enrichment and Migration

> *Our goal is to get faster at integrating acquisitions. A key component of this is migrating and enhancing the quality of data of the acquired company to our core systems in a rapid fashion. Using BPM allows us to define and continuously improve a best-practice data migration process and to easily adapt it for a project's unique requirements.*
> *– Oliver Bussman, CIO, SAP AG*

Shareholder return on any acquisition correlates with how fast the acquired company is integrated, so SAP AG is continuously looking for ways to improve its methods for postmerger integration. One area for improvement is how fast the acquired company's business processes and data can be migrated over to the core SAP systems. To meet a management mandate for improving the speed of this process by at least 50%, SAP Global IT chose to leverage a BPM approach to streamline its postmerger data migration processes and turned to SAP's value prototyping team to implement this process.

7.3.1 Background

SAP AG is the world's leading provider of business software solutions and the fourth-largest software company in the world. Headquartered in Walldorf, Germany, with regional offices around the world, SAP offers applications and services that enable companies of all sizes and in more than 25 industries to become best-run businesses.

Business Case

While SAP's strategy relies largely on organic growth, SAP also pursues growth opportunities through strategic acquisitions. A key factor in realizing value from a major acquisition is successfully integrating the acquired company as fast as

Sample Solution Architecture for an SAP Landscape

Figure 9.6 illustrates KAESER's modular architecture. With this service-oriented architecture, KAESER has developed a software infrastructure that is capable of building highly efficient and responsive processes throughout the value chain.

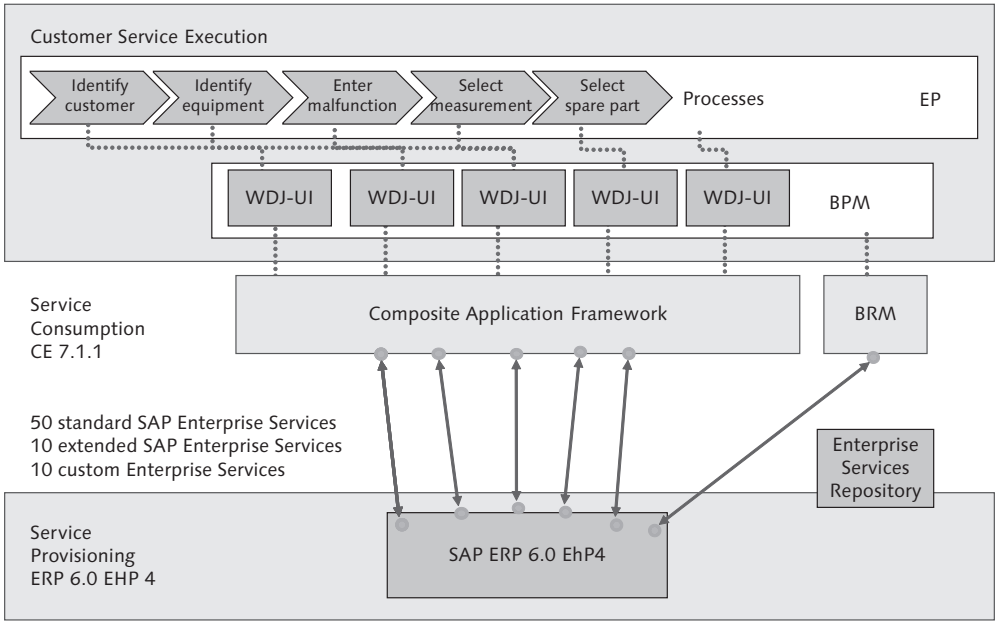


Figure 9.6 Solution Architecture

9.2 Braskem S.A.: Realizing the Value of Efficiency and Visibility in Supplier Processes

The value realization that is part of BPM methodology was what made this approach so compelling to Braskem. We were introduced to this by the SAP Value Academy and leveraged the SAP Value Lifecycle Manager as the value management tool for our BPM Center of Excellence.
– Marcos Antonio Miliani, Processes and Systems Manager, Braskem S.A.

Braskem S.A. is Latin America's leading manufacturer of thermoplastic resins, founded from the merger of several companies. To help support the integration and simplification of their operations, Braskem worked with SAP partner Firsteam to use the BPM methodology from SAP and leveraged the SAP NetWeaver Business

Process Management component to deploy more integrated and transparent business processes. Initial improvements from this large-scale project have already resulted in new operational efficiencies and cost reductions.

9.2.1 Background

Braskem S.A., headquartered in São Paulo, Brazil, is the largest petrochemical company in the Americas by production capacity and the seventh-largest in the world. As Latin America's leader in thermoplastic resins, Braskem understands the transformative power of technology. The plastics made by this innovative Brazilian company are used in thousands of products, from toothbrushes and baby bottles to automotive parts and computer components.

Market leadership, competitiveness, and technological autonomy aligned with the commitment to promote sustainable development are the basis for Braskem's strategy. Braskem uses leading-edge technology in its manufacturing operations and to support its business objectives. As part of this vision, Braskem initiated a company-wide program to simplify, integrate, and energize its business processes through the power of service-oriented architecture.

Business Case

Braskem S.A. was established in 2002 from the merger of several companies as part of a major consolidation of the Brazilian petrochemical industry. Since that time, Braskem has acquired additional companies to expand its product portfolio and area of operations. As a result, a chief concern of Braskem was the integration and simplification of operations and business processes that incorporate the business of the acquired companies.

Braskem's IT group was tasked to develop a programmatic approach for integrating business operations and strengthening the business management model. This would then lay a foundation for further growth and internationalization of Braskem.

First, Braskem needed to establish a means of common communication and goal-setting between business leaders and IT so that IT implementations were aligned with business operational needs with the strategic vision of the company. Additionally, Braskem's IT department needed to learn a methodology for business-value identification and a way to prioritize IT demands.

With a clear method for aligning business and IT in place, Braskem could then focus on integrating and optimizing end-to-end operations. Braskem wanted to

be able to proactively identify inefficiencies and opportunities for optimization by enabling measurement and monitoring of process operations. Also, Braskem wanted to simplify employee training and improve user adoption of IT systems by creating consistent user experiences for process automation. With these goals in mind, the company developed a solid business case for the project after attending the SAP Value Academy program and using the value lifecycle manager tool from SAP to identify the specific business benefits. The business case strategy followed the SAP value engineering methodology. With this tactic, they compared development costs using SAP NetWeaver BPM with a more traditional approach of custom development through writing code. For the first project, Braskem estimated the productivity they would gain by automating the first targeted process and centralizing operations within a shared services center.

Braskem decided to set up a business process competency center for process redesign and chose the SAP NetWeaver BPM component to help model and deploy the new process automation.

The company identified 13 key business processes for reengineering and selected the high-priority accounts payable process for services suppliers as a starting point — a logical choice because Braskem had the following business scenario:

- Over 3,000 services suppliers spread throughout Brazil
- 20,000 *notas fiscais* (Brazil-specific invoices filed with the government) received per month
- Approximately 1,200 contract managers
- A highly manual process with handling errors that caused rework and losses
- Payments that were not being received within required periods
- A process without traceability and control

Braskem's vision for the solution was to eliminate unnecessary payment delays and improve their visibility and control over costs and expenses. Specific business goals for this process redesign project included:

- Better visibility of spending per vendor and traceability of payments
- Improved efficiency when executing the payment process
- Improved cost optimization through on-time payment and ability to leverage early payment discounts
- Automated compliance with Brazilian requirements for registering legal invoicing documents

- Learning and realizing benefits of the BPM approach for continuous process optimization

Metrics chosen to map these goals included:

- Measurement of process cycle times
- Tracking of on-time payment rate
- Average design man-hours for process design

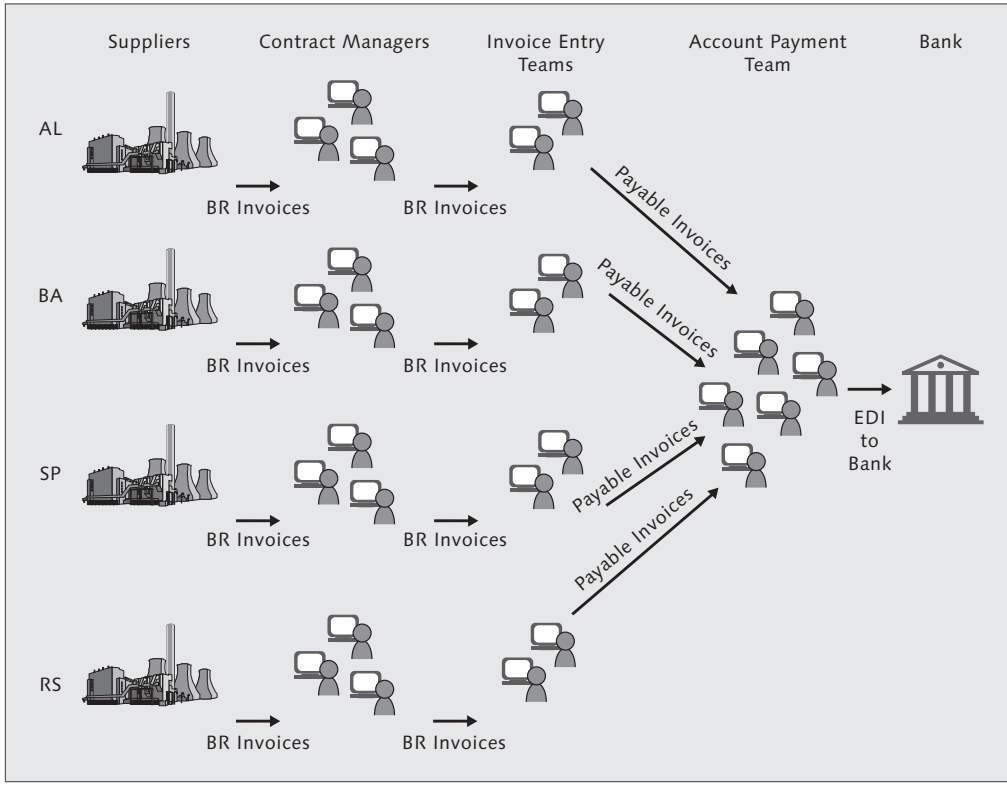


Figure 9.7 Braskem's As-Is Supplier Payment Process

As-Is Process

As depicted in Braskem's as-is supplier payment process in Figure 9.7 (decentralized and manual invoice receipt, acceptance at company plants and regional centers, and centralized accounts payable), Braskem's old payment process was a highly manual process that was spread across many teams and as a result lacked

visibility and control. Invoices were received by contract managers at each of Braskem's local plants and sent to one of several regional invoice entry teams to manually enter the invoice into SAP ERP. Checking invoices against data in the SAP systems and the many related tracking spreadsheets for mismatches to contracts, due dates, value, and quantities was a manual process. Every time an error (deviation) was found, the invoice had to be sent back to the contract manager to fix the error. This particular manual exception was the main reason for payment delays.

9.2.2 BPM Solution

As part of the process redesign, Braskem decided to concentrate the handling of invoices in one shared services center as depicted in Figure 9.8. This figure illustrates the to-be process of outsourced digitization of invoices and a shared services center for approving and paying invoices.

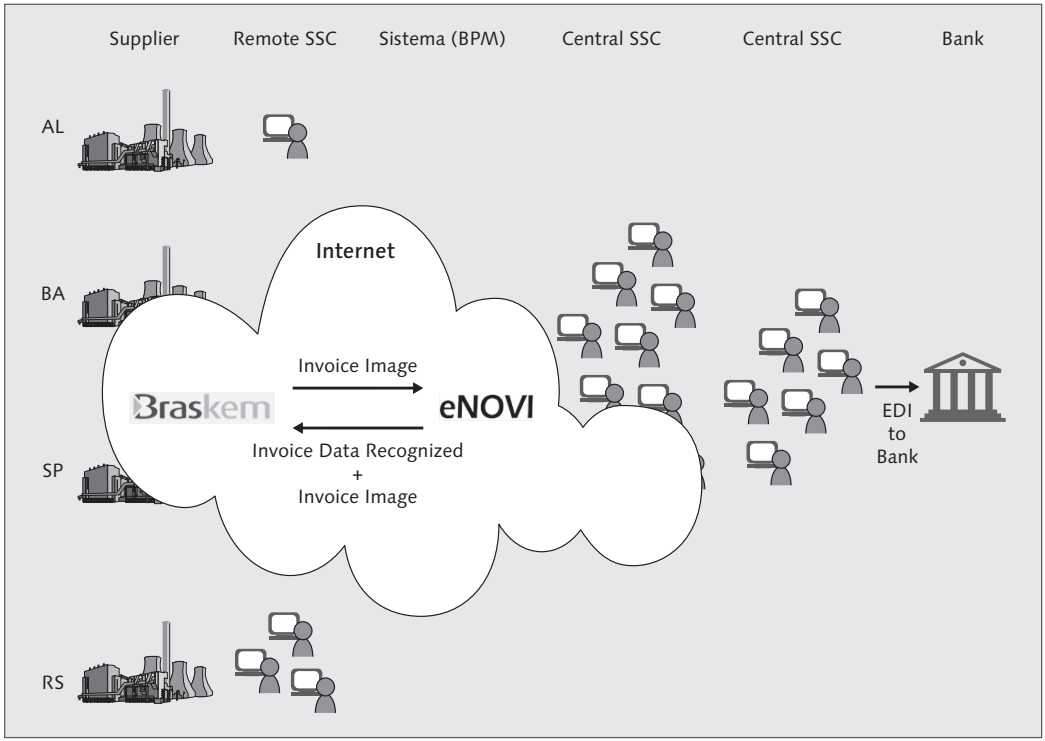


Figure 9.8 Braskem's To-Be Process for Supplier Payment

As depicted in Figure 9.8, *notas fiscais* (invoice) submitted by suppliers to Braskem are scanned and sent to the service provider, eNOVI, who digitizes the invoice and transmits the information electronically to Braskem. Invoices are automatically matched with purchase orders. When a discrepancy is noticed, or a match cannot be made, a contract manager manages the exception and work with the supplier to correct the invoice. When automation shows a match with no discrepancies, and for invoice problems that are resolved by the contract manager, the invoice is then sent to the accounts payable team.

After analysis and optimization, the process for automating validation and approval of invoices was designed in SAP NetWeaver BPM (see Figure 9.9).

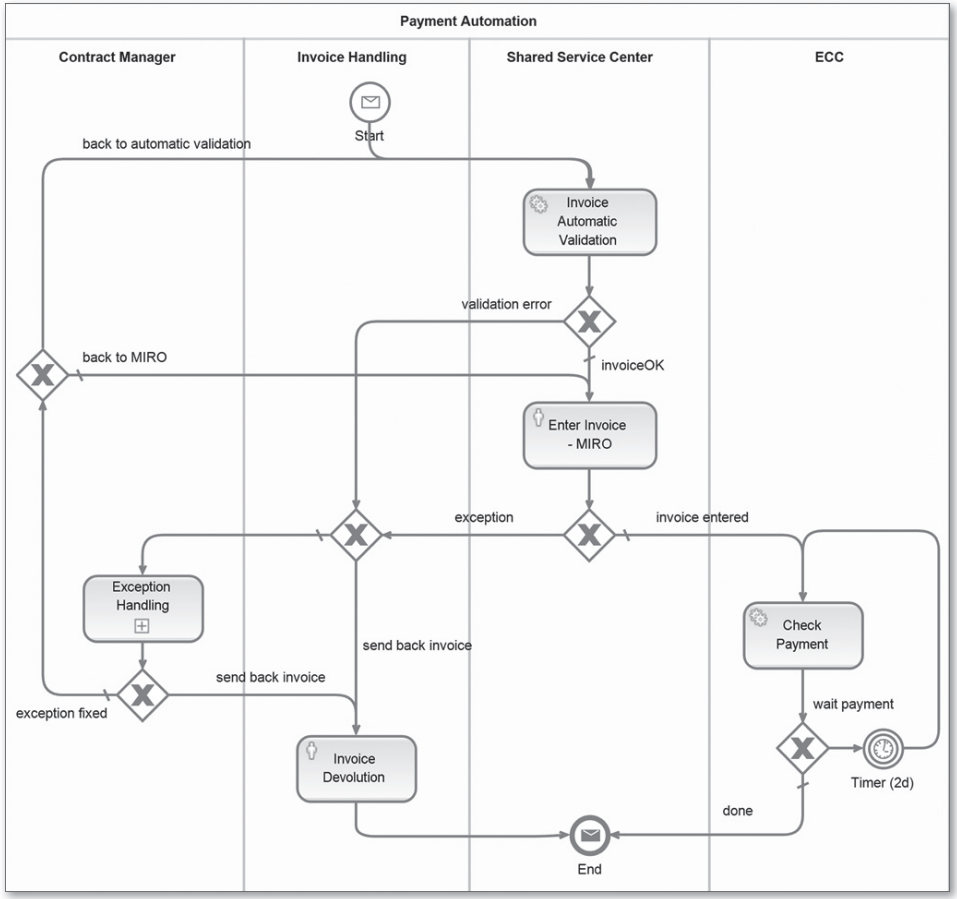


Figure 9.9 New Invoice Matching and Payment Process at Braskem

As designed by the IT team, Braskem's BPM payment process starts via a Web service that receives invoice data. The first step is an automated process step that verifies invoice data against data within Braskem's SAP system. Several validations take place, such as validation of invoice value, entry sheet, due date, and tax value. Validations in this step depend on invoice type and are performed using decision tables that are implemented using business rules within BRM decision tables and custom-developed enterprise services provisioned within the ERP system. If there is any exception during this process, BPM routes the invoice to a contract manager to handle the exception. If the data passes validation, the next step is to enter the invoice using a MIRO transaction.

During the MIRO step, it is now possible to see the invoice image, data entered during the BPM process, and any resolutions made by contract managers. At any time during the process, it is possible to cause BPM to enter a devolution step, where the devolution is registered and process is completed.

Once the invoice has been entered into the system, another automated step checks payment data within ERP to update the process status for KPI tracking and completes the process, keeping a record of the payment execution.

To help Braskem's business managers monitor the status and progress of specific status instances, the dashboard depicted in Figure 9.10 was developed.

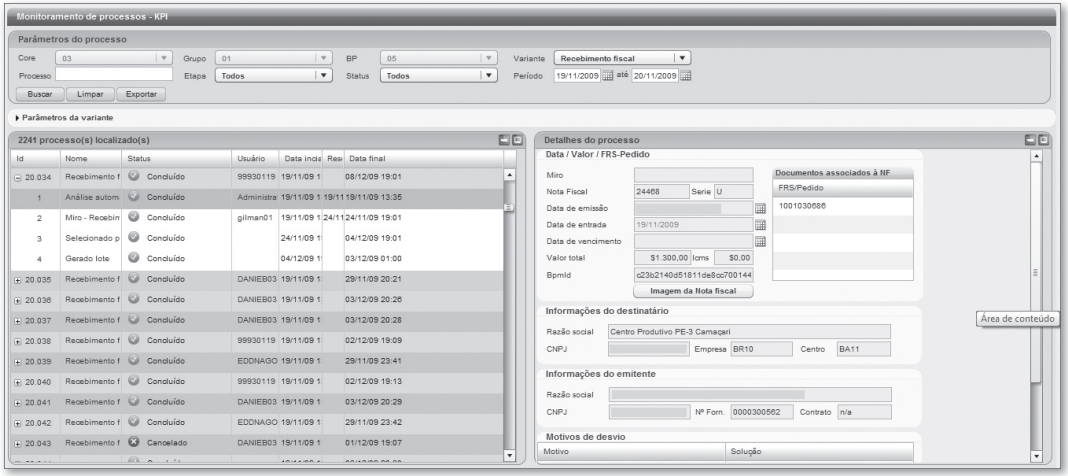


Figure 9.10 Braskem's Dashboard for Viewing Process Status and Context

Implementing the Solution

For the business blueprint, the methodology chosen was the SAP SOA Implementation Roadmap. This was an obvious choice, because the version of ASAP methodology available at the time did not include SOA and BPM accelerators. (Editor's note: the new ASAP methodology 7.0 as described in this book has incorporated the SAP SOA Implementation Roadmap and now supports BPM and SOA implementations.)

One complicating factor was the need for Braskem to upgrade their SAP for Oil and Gas solution portfolio to leverage preexisting functionality to implement their BPM processes and to take advantage of SAP-supported enterprise services. The upgrade cycle for this solution caused the BPM implementation to take longer than it ordinarily would while the enhancement pack was tested, and SAP worked to fix some errors in Brazilian localization.

Another issue solved during the realization phase was the requirement to send a process instance to a different number of approvers depending on the owner of the object. This feature was not available out-of-the-box in the first version of SAP NetWeaver BPM 7.11, so a workaround was developed using a mix of BPM objects and Web Dynpro programming. The workaround developed is depicted in Figure 9.11.

An additional complicating factor was that each contract manager needed to receive a specific task alert if a particular invoice processing exception was related to a contract that the contract person managed. This was complicated by the following process information relationships:

- An invoice is related to *n* entry sheets
- Entry sheets are related to a purchase order
- Purchase orders are related to a service contract
- A service contract has *n* contract managers

To manage a task involving this many levels, the process needed the ability to dynamically branch synchronous parallel tasks for the variable number of contract managers that may be related to a particular invoice.

The solution subprocess dynamically creates each necessary task with specific information for each manager and controls the execution of the tasks. The subpro-

cess ends only when all processing exceptions are handled and then synchronizes information from multiple steps to send back to the main process.

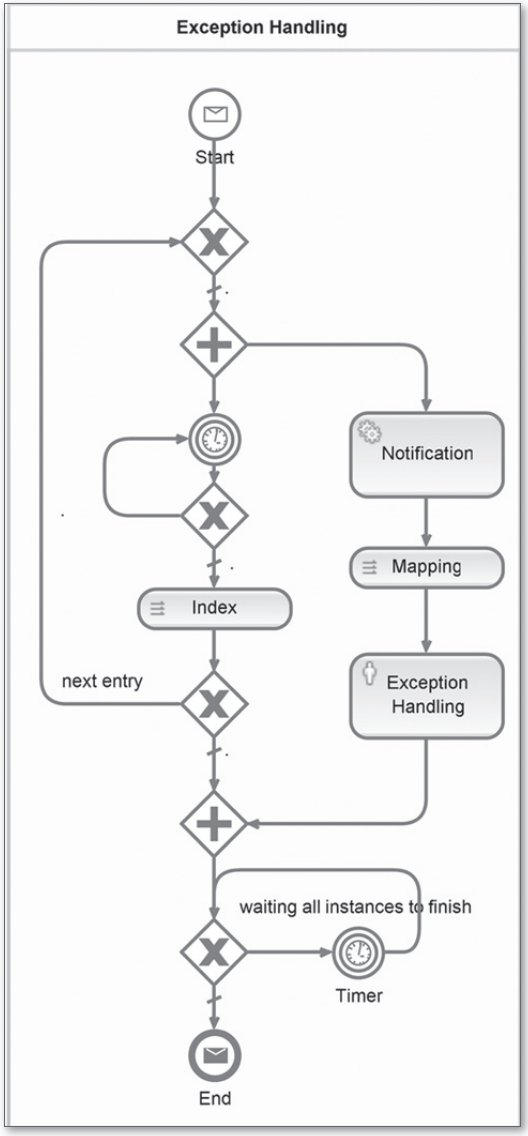


Figure 9.11 Workaround Subprocess for Variable Number of Approvers

Results of the Project

With the initial phase of the project completed, Braskem has deployed an integrated payment process that addresses the complete workflow from invoice receipt and approval through payment processing. As a result, invoice registration and processing costs have dropped 30%, and 95% of supplier payments are now made on schedule, which helps Braskem take advantage of discounts and avoid penalties. The ability to process electronic invoices more efficiently is also making individual transactions easier to track (see Table 9.3).

Key Performance Indicator	Impact
Time needed to redesign processes	-20% to -30%
Training costs	-20%
Invoice registration and processing costs	-30%
On-time payment rate improvement	60% to 95%

Table 9.3 KPI Tracking

Braskem's process transformation program is just getting started. The company is now turning its attention to the remaining 12 processes. Braskem's goals include enhancing traceability in contracting, streamlining sales order entry, and simplifying the company's critical export processes. Such improvements will play an important role in shaping the future success of this growing chemical company.

Sample Solution Architecture for an SAP Landscape

With Braskem's broad SAP software landscape, SAP NetWeaver BPM was a particularly attractive choice, offering native integration and a less costly and complex implementation than alternative technologies.

As shown in Figure 9.12, the payment process initiates with digitalization of Brazilian invoices. The digital invoice image is sent to a service provider who, through an OCR process, recognizes fields from the invoice image and sends it back to Braskem, integrated via an interface with SAP NetWeaver PI.

The service provider sends the invoice picture as a PDF file and the fields recognized from the invoice image as xml data. SAP NetWeaver PI receives this information. The PDF file is then stored in the SAP content server, and the invoice data is sent to a custom Web service that starts the BPM process in SAP NetWeaver CE.

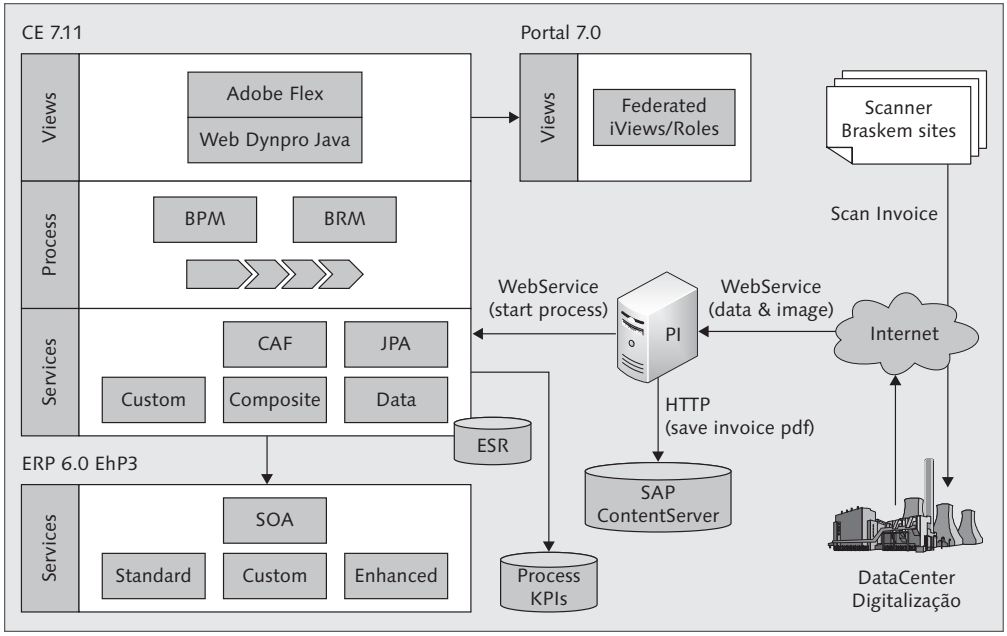


Figure 9.12 System Architecture of Braskem's Service Supplier Invoice Checking and Payment Process

More Information

For questions and in-depth information about how to conduct business modeling, process harmonization, the ICASIO pattern, the use of the pattern within your business cases, business process improvements, service-level agreements, and modeling of business processes, process steps, and activities for process harmonization, please visit *www.openroundtable.org*.

13.3 ASAP Methodology 7 Core

In 2009, behind-the-scenes work was undertaken to harmonize the way we project-manage SAP implementations. The result is the new ASAP Methodology for Implementation 7, which was launched in February 2010. The new ASAP methodology brings together the previous ASAP methodology, Business Intelligence Solution Accelerator (BISA) methodology, value delivery principles, business process management methodology, and service-oriented architecture methodology.

The ASAP implementation methodology is a phased, deliverable-oriented methodology that streamlines implementation projects, minimizes risk, and reduces the total cost of implementation. ASAP takes a disciplined approach to project management, organization change management, solution management, business process management, value management, and other disciplines applied in the implementation of SAP solutions. There are two highly visible components of the new ASAP methodology.

The ASAP Roadmap 7 core, which covers the entire project lifecycle — from evaluation through delivery to postproject solution management and operations — and the value, process, and application lifecycle illustrated in Figure 13.16. The new ASAP Roadmap 7 core has been made leaner, increasing its practicality, and provides transparency of value delivery through consistent business case reflection and ensures efficient guidance for service-oriented architecture (SOA), business process management (BPM), and traditional implementation projects.

More Information

You can view and display the ASAP Roadmap 7 core via different tools. You can download it as an HTML extract via SAP Service Marketplace *http://service.sap.com/asap*, SAP BPX Community *http://www.sdn.sap.com/irj/bpx/asap*, and SAP Solution Manager, where the ASAP Roadmap 7 core can be assigned to your project in project administration, Transaction SOLAR_PROJECT_ADMIN, as illustrated in Figure 13.17 and deployed in Transaction RMMAIN as shown in Figure 13.18.

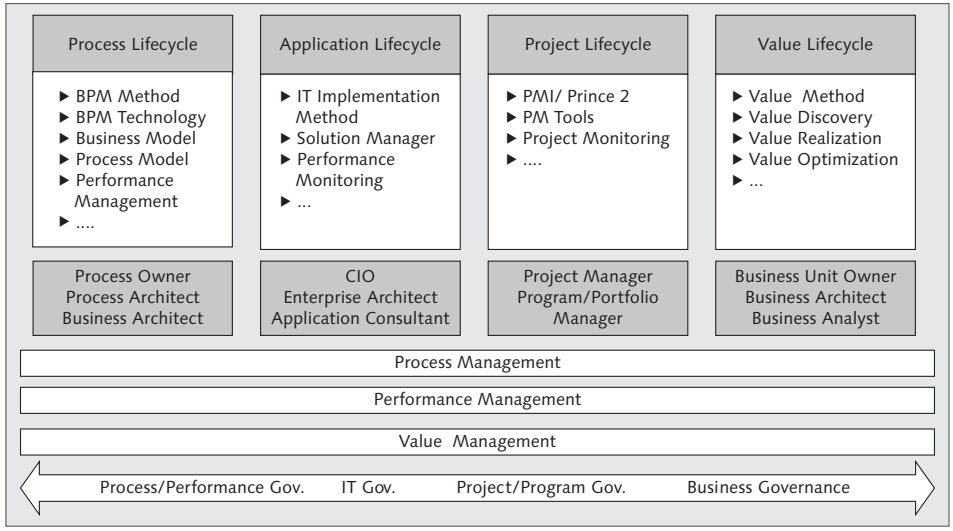


Figure 13.16 The New ASAP Methodology Supports the Four Lifecycles: Process, Application, Project, and Value

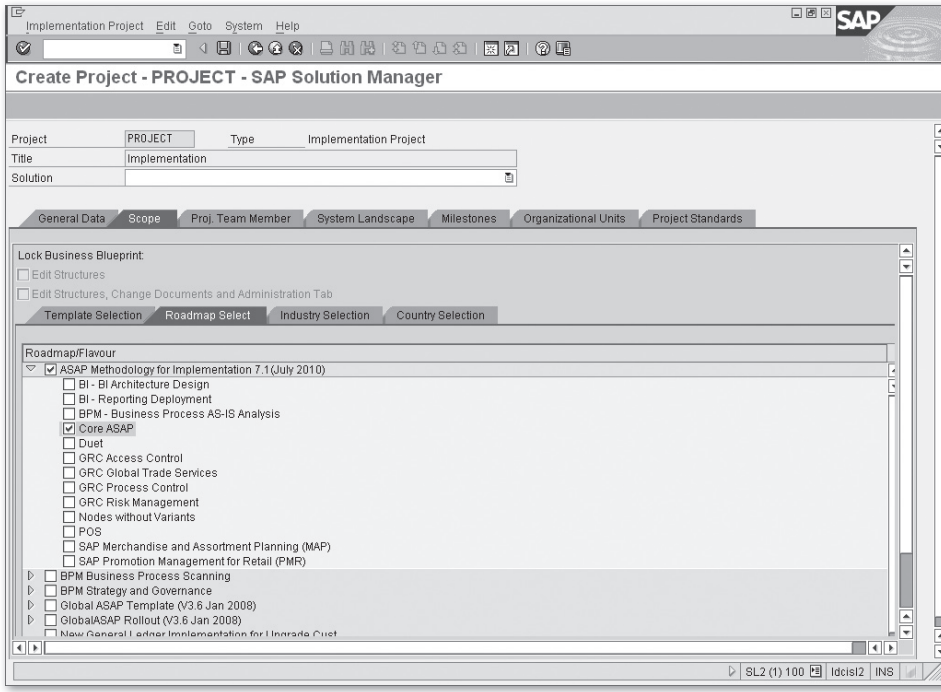


Figure 13.17 Assign ASAP 7 Roadmap in SAP Solution Manager, Project Administration (SOLAR_PROJECT_ADMIN)

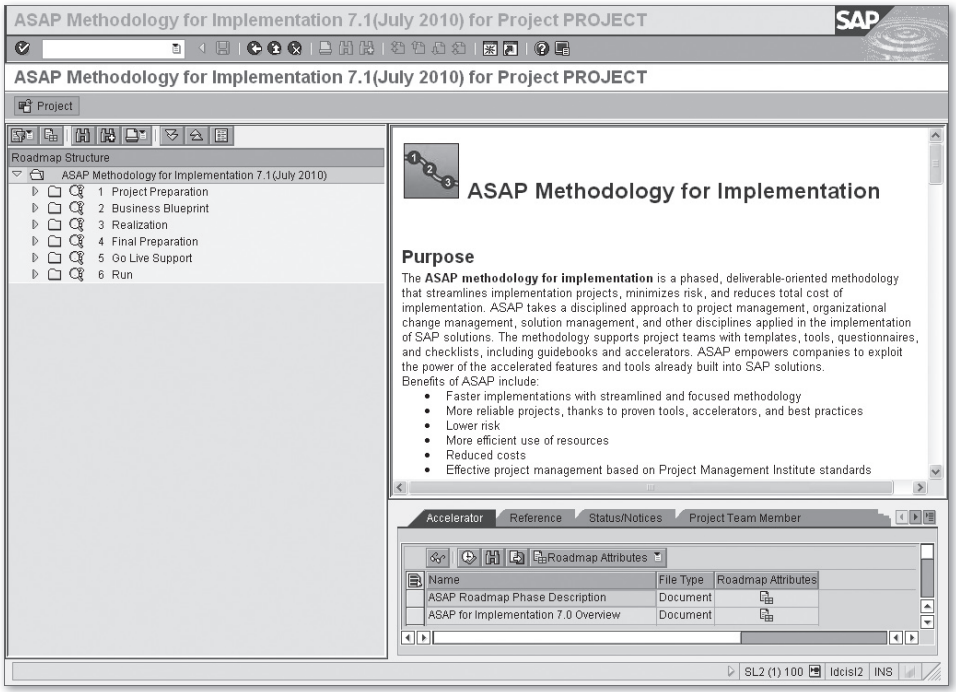


Figure 13.18 ASAP 7 Roadmap in SAP Solution Manager (RMMAIN)

The second set of visible components of the ASAP methodology is the business add-ons to ASAP that extend the ASAP Roadmap with modular business implementation content. The business add-ons provide proven implementation content for implementation of various industry solutions, solutions packages, and other related areas such as agile methodology, BPM, SOA, and enterprise architecture (EA) governance and strategy frameworks. We will describe in detail the business add-ons to ASAP in Section 13.4.

In the following sections we will introduce each of the ASAP phases in the ASAP Roadmap 7 core, as illustrated in Figure 13.19 and describe how value delivery, business process management, and service-oriented architecture are reflected in the new methodology and how to apply it when you implement an SAP solution where you need to take into consideration both enablement of Best Practices and enablement of own practices, also referred to as composite applications. You can build the composite applications on top of SAP Business Suite's Best Practices with the application core processes and on arbitrary backend systems. Composite applications follow the SOA paradigm of "non-intrusiveness," which means these

applications are bound to provide modification-free process extensions to the core business applications. This section will also describe which skills enablement are required for the project team members to practice the new ASAP methodology, which now covers the value, process, application, and project lifecycle.

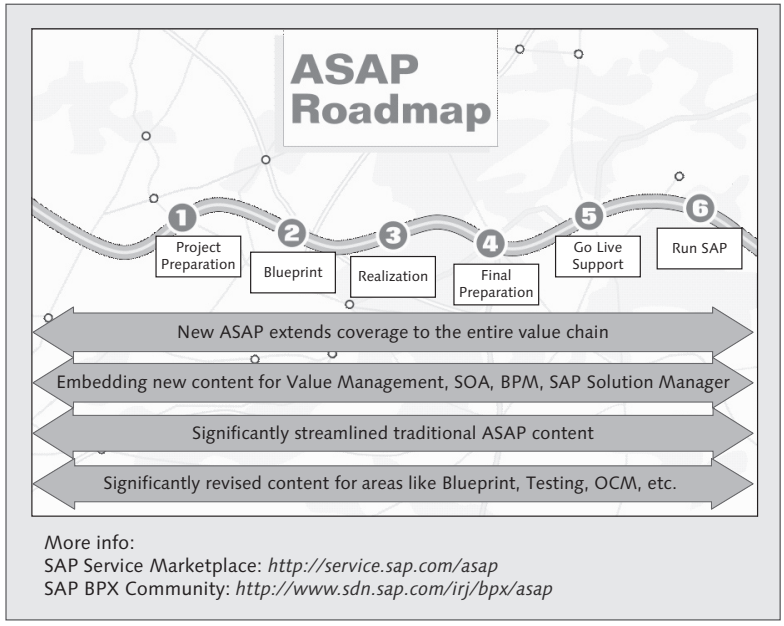


Figure 13.19 ASAP 7 Includes Six Phases

We will start with phase 1, project preparation, and describe how value delivery, business process management, and service-oriented architecture are reflected in this phase, including the skills enablement requirements for the project team members.

13.3.1 Project Preparation

Project preparation is the first phase of the implementation project, where preplanning of all relevant project management disciplines is conducted and documented in the project management plan, for example, procedures for integrated change control, management of issues, scope, time, cost, quality, project staff, communication, risk, and contracted resources and services. Defining these procedures enables structured project execution, monitoring, and controlling in subsequent project

phases and contributes to ensuring project success. As shown in Figure 13.20, project preparation includes seven work streams:

- 1.1 Project management
- 1.2 Organizational change management
- 1.3 Training
- 1.4 Data management
- 1.5 Business process management
- 1.6 Technical solution management
- 1.7 Integration solution management

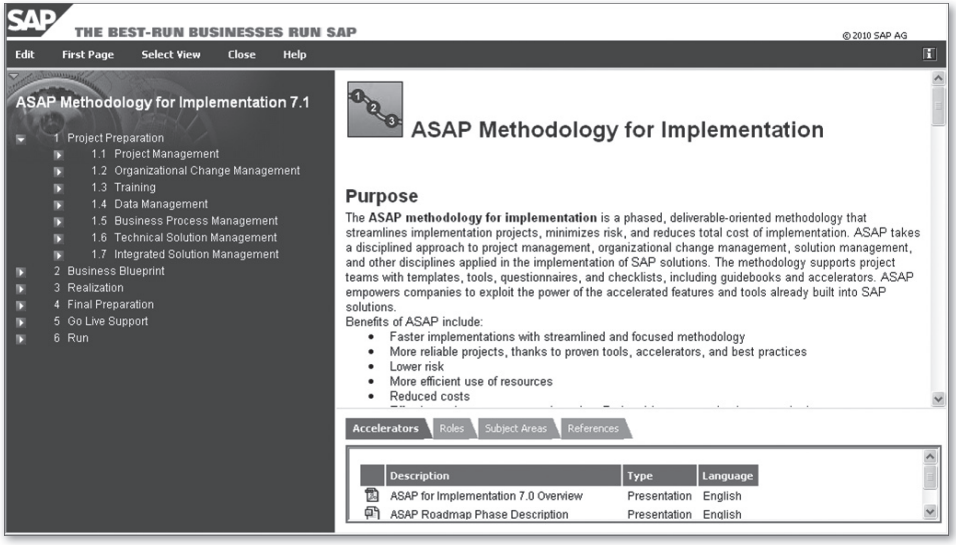


Figure 13.20 Project Preparation – Seven Work Streams

The project management work stream is completely aligned with Project Management Institute (PMI) project management standards as defined in the PMI Project Management Body of Knowledge (PMBOK). Project standards for various activities throughout all phases are defined or start to be defined in this phase:

- 1.1.5.1 SAP Solution Manager usage guidelines
- 1.1.5.2 Business process modeling standards (new)
- 1.1.5.3 Initial development management standards

- 1.1.5.4 SAP services deployment plan
- 1.1.5.5 Software system configuration standards
- 1.1.5.6 Enhancement and modification standards
- 1.1.5.7 Support package and upgrade standards
- 1.1.5.8 Change request and transport management standards
- 1.1.5.9 Test management standards
- 1.1.5.10 Postimplementation service and support standards
- 1.1.5.11 Enterprise service design standards (new)
- 1.1.5.12 Composite application design and development standards (new)

We recommend having these standards in place when implementing an SAP solution. Note that a number of new standards have been added. You can find more details on these standards later in this chapter.

The project preparation phase includes several deliverables, milestones, and key decisions, as illustrated in Table 13.7. For each deliverable, the ASAP Roadmap explains in detail the purpose, inputs, and outputs and where it's applicable and gives further details and information about the expected result.

Purpose	Deliverables	Milestones & Key Decisions
Initial planning and preparation	Project scope defined	Corporate review completed
Define the project goals, scope, and objectives	Implementation plan and rollout strategy	Scope defined
Identify, on-board, and train team members	Detailed scope document	Project team staffed and trained
	Costs and benefits validation	Project team organization, responsibilities and location
	Project standards	Roll-out plan mandates/ constraints
	Project infrastructure	Policies for to be project organization
	Knowledge transfer approach	System retirement objectives/ mandates/ constraints

Table 13.7 Deliverables in Project Preparation

Purpose	Deliverables	Milestones & Key Decisions
	Implementation work plan	Training budget and approach
	Master data design	Key stakeholders for communications identified
	Interface list	Implementation plan in place
	Testing strategy	Corporate review completed
	Data cleansing strategy	

Table 13.7 Deliverables in Project Preparation (Cont.)

After this short intro to the project preparation phase, we will take a closer look at the value delivery considerations in this phase.

Value Delivery Considerations in Project Preparation

Value determination is part of the business process management work stream as shown in Figure 13.21. The purpose of the value determination deliverable is to create a value-based solution design to determine value drivers and key process changes for the implementation project to ensure value delivery.

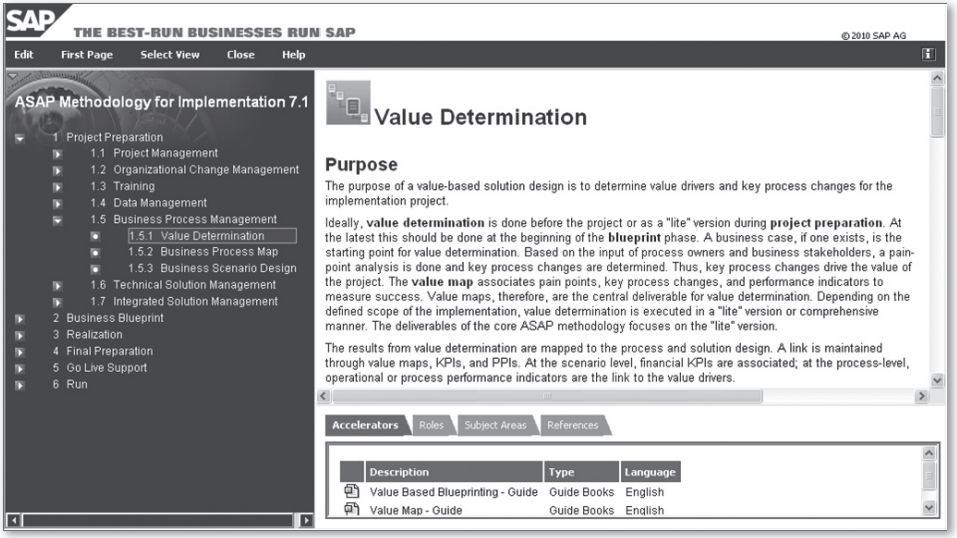


Figure 13.21 Project Preparation – Value Determination

The objective of value delivery is to ensure that the project lives up to the value expectations according to the targets that are stated in the initial business case. This value-based approach serves the following purposes in project preparation phase:

1. Execute the project according to the business case or value map targets.
2. Monitor and track the project value delivery based on the initial business case or value map and report the status of value delivery at an early stage of quality gates (Q-Gates).

Value-based solution design plays a critical role in determining value drivers and key process changes for the implementation project. To realize the intended business value for this initiative, it is essential to address key success factors and establish a clear, shared set of expectations for program value creation; achieve a rapid program launch with effective value-based governance; make the business case actionable and measureable by defining design imperatives, key performance indicators (KPIs) and process performance indicators (PPIs); establish ongoing value management discipline to ensure that the business blueprint phase (following the project preparation phase) and implementation reflect design imperatives.

The inputs required for value determination in project preparation are a value-based opportunity storyline created by clearly identifying the value built into the business case including benefit objectives, relevant processes and key process changes, financial operational KPIs and PPIs for measurement, and expected values and costs. We also recommend establishing project management and value tracking with these methods: including a value expert, integrating a value schedule, and reporting for value delivery. Another recommended activity is to set up a project value framework that includes key inputs such as benefit objectives, relevant processes and key process changes, KPIs, value potentials, and costs. By correctly completing the recommended activities for a value-based approach, the project will gain an overall value-based solution proposal.

Let's now take a closer look at the business process management considerations in the project preparation phase.

Business Process Management Considerations in Project Preparation

Business process management is one of the seven work streams in project preparation. Figure 13.22 shows the BPM work stream.

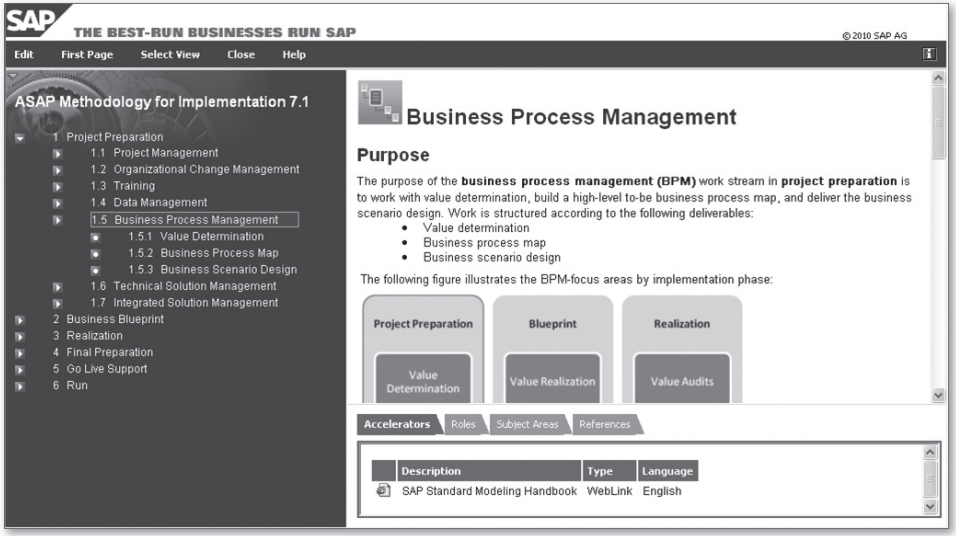


Figure 13.22 Business Process Management Work Stream

The purpose of the business process management work stream in project preparation is to work with value determination, build a high-level to-be business process map, and deliver the business scenario design.

The work-stream deliverables are enhanced to expand on the business case and ensure that the value drivers are incorporated into the solution design. In addition to identifying the value drivers, key process changes are also identified for input into the solution transformation design deliverable that is part of the business blueprint BPM work stream. The creation of business process maps helps the project team verify the agreed upon scope of the project and provide inputs for the business blueprint workshop content. Business process maps provide the framework for business process modeling and therefore help control the scope of the project. Decomposition of the business scenarios during project preparation is the starting point and acts as the foundation for the detailed business process decomposition that takes place during the business blueprint stage. The primary changes for business process management during project preparation therefore involve the new work packages:

- **Value determination**
  Covered in Section 13.3.1.

- **Business process map**
  Builds the foundation for the process hierarchy and process scope of the implementation.
- **Business scenario design**
  Provides an understanding of the essential processes at the scenario level and builds the foundation for further process decomposition that will take place in the business blueprint phase.

The inputs required for the business process management work stream are project scope as specified in the statement of work catalog of as-is business process documentation. If the catalog of as-is business process documentation does not exist, we recommend executing an as-is analysis before starting the to-be design. The as-is analysis is not included in ASAP 7 core but can be added via the business add-on to ASAP that delivers business process as-is analysis methodology as illustrated in Figure 13.23, where the add-on to ASAP: Business Process As-Is Analysis is activated.

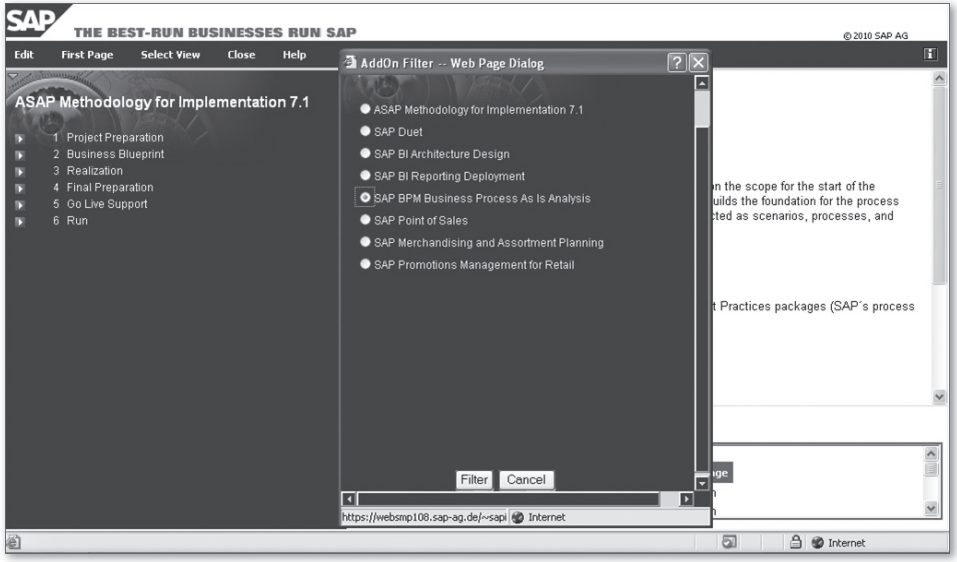


Figure 13.23 Activate Business As-Is Analysis Methodology Add-On

In Figure 13.24 you can see how the additional as-is analysis methodology has been merged into ASAP core.

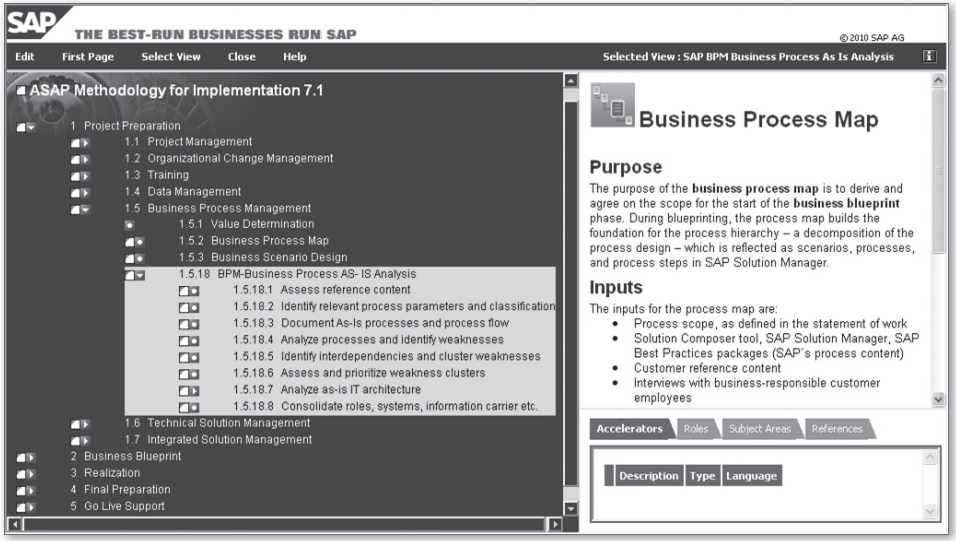


Figure 13.24 Business Add-On that Delivers Business As-Is Analysis Methodology Merged into ASAP Core

You can also get help from the business add-on to ASAP that delivers redocumentation using SAP Solution Manager and SAP Enterprise Modeling by IDS Scheer to identify and analyze your automated as-is business processes. Figure 13.25 illustrates this add-on. For more information please go to SAP EcoHub at *http://ecohub.sdn.sap.com/*.

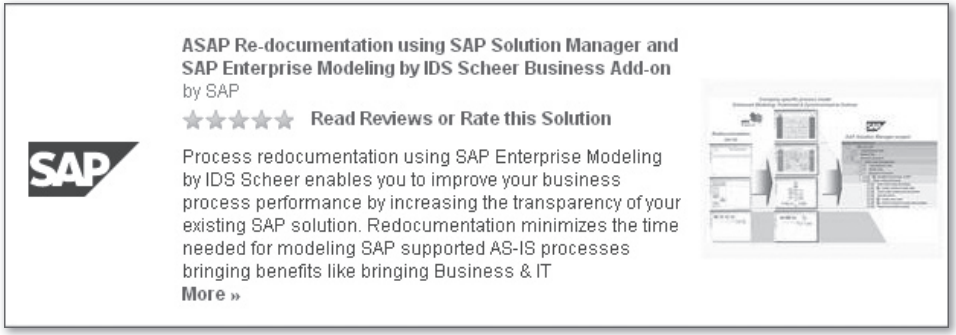


Figure 13.25 Business Add-On that Delivers Re-Documentation Using SAP Solution Manager and SAP Enterprise Modeling

The last input is to define business process modeling standards and business to modeling tools. For this activity we get help from project standards (1.1.5) and business process modeling standards (1.1.5.2).

The purpose of the business process modeling standards is to have a standard approach for executing process modeling. SAP provides a standard modeling handbook that is linked as an accelerator to this deliverable in ASAP.

More Information

The SAP Standard Modeling Handbook is available on the BPX Community as a wiki. For more information please go to *http://wiki.sdn.sap.com/wiki/display/ModHandbook/SAP+Modeling+Handbook+-+Modeling+Standards*.

To support the test management standards (1.1.5.9) and the different test activities during the implementation, the following business add-ons to ASAP are available for delivering content for testing:

- Testing Strategy
- SAP Quality Center by HP
- TAO for SAP
- TDMS
- SAP LoadRunner by HP

More Information

For more information please go to SAP EcoHub at *http://ecohub.sdn.sap.com/*.

Now that we have taken a closer look at value delivery and business process management considerations, we will take a deeper look at the service-oriented architecture considerations in the project preparation phase.

SOA Considerations in Project Preparation

The decision to implement SOA usually represents an important architectural paradigm shift for a company — within both the business and IT organizations. The burden of SOA implementation typically falls most heavily on the organizational side of the enterprise, where new skills and responsibilities have to be introduced along with focused attention to the business requirements including new IT capabilities and to the tighter relationship between IT and business. The ASAP

7 methodology implies key SOA considerations and activities within the following work streams within the project preparation phase:

- Work stream: project management (1.1), project management standards (1.1.5). New development standards need to be defined for enterprise service design standards (1.1.5.11) and composite application design and development standards (1.1.5.12).

  For more details about composite development architecture guidelines and standards, which is one of the key accelerators, follow the link to composite application design and development standards (1.5.12) (see Section 14.1).

- Work stream: technical solution management (1.6). The purpose of the technical solution management work stream is to outline essential technical and infrastructure deliverables that are appropriate to the initial project planning of an SAP implementation project. When defining the technical and infrastructure deliverables, you also need to include the deliverables for a composite application and enterprise services development environment, as illustrated in Figure 13.26.

Figure 13.26 Project Preparation – Technical Solution Management

We recommend establishing an enterprise service-oriented architecture strategy and governance to ensure the success of your project. Effective enterprise SOA strategy and governance calls for a holistic management approach that integrates

and aligns the corporate business strategy, the IT strategy, and the planning and operational activities associated with enterprise SOA solutions. This approach encompasses people, processes, and technologies. In most companies, some elements of enterprise SOA governance already exist. For instance, you can leverage IT governance as part of the foundation for enterprise SOA governance. But enterprise SOA governance is much more; it involves organizational structures, skills, and procedures aligned with business needs. To establish the enterprise service-oriented architecture strategy and governance you can get help from the business add-on to ASAP that delivers an enterprise service-oriented architecture strategy and governance framework. For more details about this business add-on to ASAP, please go to Section 13.4.3.

Because this is the last part of this section, we will take a look at the skills that project team members need to have to practice the new ASAP methodology.

Consultants, Business Process Experts, Project Managers and Team Leads – Considerations in Project Preparation

Consultants and business process experts who join the project during the project preparation phase play a larger role in the project than before. Not only do they assist the project managers in validating high-level scope, but they have to think in terms of a value- and process-based implementation by assisting in the preparation of to-be process measurement and prepare for value delivery. Team leads are engaged during project preparation to initiate their own work stream. No longer is it sufficient to rely on only the project manager and technical architect to create the deliverables during project preparation. The foundational deliverables in project preparation set the scope, strategy, and value focus for the remainder of the project. In today's global economy and tough economic climate, consultant team leads and project managers need to be business oriented. Meeting the constraints of time, cost, and quality while delivering a project is not enough. Projects must be viewed strategically within the context of the business and provide measureable value. Implementing projects that may not deliver the intended value until a few years down the road is very challenging and will require team leads to gain the knowledge to work with the new paradigm.

We will now go to the next phase — the business blueprint — and describe how value delivery, business process management, and service-oriented architecture are reflected in this phase, including the skills enablement requirements for the project team members.

14.1 Composite Development Architecture Guidelines

The Composition Development Architecture Guidelines have been created based on feedback and experience from the first customers using SAP NetWeaver Composition Environment. The guidelines include recommendations that will help you in implementing applications following the SOA principles.

14.1.1 Value Proposition of SAP NetWeaver CE

SAP NetWeaver Composition Environment (CE) targets two distinct areas. First, SAP NetWeaver CE enables model-driven development of own practices, also referred to as composite applications. Secondly, customers are enabled to design, deploy, and run Java applications with SAP NetWeaver Composition Environment following the JEE standards.

14.1.2 Platform Overview

SAP NetWeaver Composition Environment is designed and implemented as a usage type of the SAP NetWeaver Java stack that integrates with different components of the full SAP NetWeaver stack on various levels (see Figure 14.1). Therefore, SAP NetWeaver Composition Environment, once it is installed in the customer landscape, leverages already existing components:

- **SAP NetWeaver Portal**
  Composite applications can be incorporated into a customer portal via a federated portal network (FPN).
- **Knowledge management (KM)**
  SAP NetWeaver CE frameworks can connect to a remote KM and its content.
- **SAP NetWeaver Business Warehouse (BW)**
  Data can be retrieved from remote servers and used in composite applications.
- **SAP NetWeaver Development Infrastructure (NWDI)**
  SAP NetWeaver CE could host its own NWDI, but it is also possible to configure SAP NetWeaver CE to use a remote NWDI.

Besides a lean runtime, SAP NetWeaver CE offers a standards-based design time, the Eclipse-based SAP NetWeaver Developer Studio (NWDS). The goal of SAP NetWeaver CE's design time is to reduce the total cost of understanding and expedite time to value by:

- Embracing community standards and Best Practices
- Providing good tool support for leveraging the SAP application through Web services

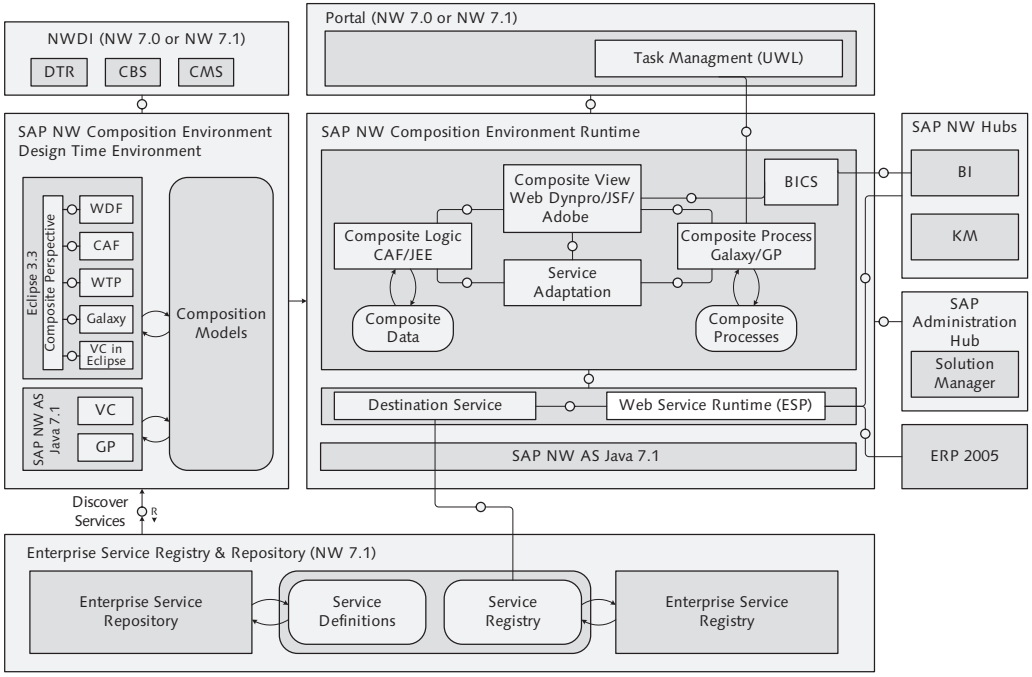


Figure 14.1 SAP NetWeaver CE Structure Overview

Overview of Layers

A composite application is structured in such a way that it contains content for specific purposes. Some parts are UI related, whereas some parts define the process flow, and other parts are specific to the business logic. The common functionality that all frameworks provide is the consumption of enterprise services.

The main SAP NetWeaver Composition Environment frameworks (see Figure 14.2) are:

- **Java EE frameworks (EJB, JSP/JSF)**
  This is the basic framework for Java Enterprise Edition (Java EE) applications. SAP NetWeaver CE supports all applications that are Java EE 5 compliant.

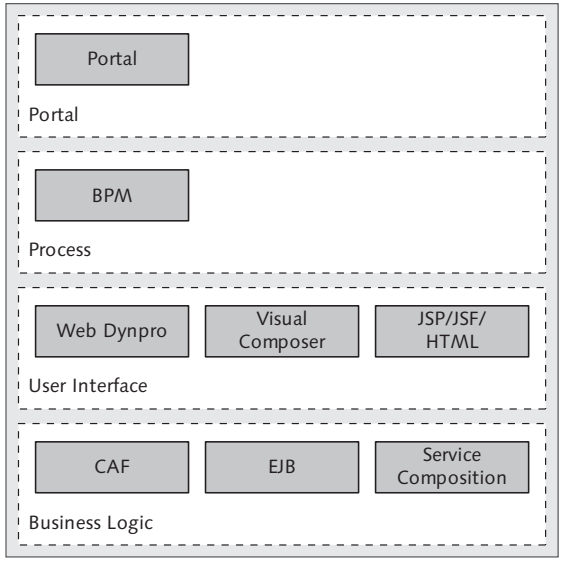


Figure 14.2 Layering of SAP NetWeaver CE Frameworks

- **SAP Composite Application Framework (CAF)**
  On top of the Enterprise JavaBean (EJB) framework, SAP provides the functionality to define business objects (BOs) and services in a model-driven way. The logic is modeled, and EJBs are generated.
- **Web Dynpro foundation**
  Most SAP UI applications run with the Web Dynpro runtime. The Web Dynpro runtime is a framework that runs inside the Java EE web container. It provides its own programming model including components, controllers, views, and so on. It follows the model-view-controller (MVC) pattern.
- **SAP NetWeaver Visual Composer**
  Model-driven UI-applications can be developed efficiently with SAP NetWeaver Visual Composer in a completely model-driven way. The design time is available as a browser-based application running on the server or as a tool in SAP NetWeaver Developer Studio (NWDS).
- **SAP NetWeaver BPM**
  Model-driven process definitions and execution are supported via SAP NetWeaver BPM, a component that uses Business Process Modeling Notation (BPMN) as the standard notation from model directly to execution.
- **SAP NetWeaver BRM**
  SAP NetWeaver BRM supports model-driven rule definition and execution.

Server Architecture

Though it seems obvious at first sight, SAP NetWeaver CE cannot simply be divided into a runtime stack running on a server and a design time running in Eclipse. The Eclipse parts are all related to design-time purposes; however, not all components on the server are relevant just for runtime. Complete design time solutions run on the server, mainly with SAP NetWeaver Visual Composer.

The applications still run on the Java EE 5 stack and utilize the standards, which means the runtimes on top of the Java EE 5 standard use the concepts of the Java server. The most important frameworks running on top of the Java EE runtime are:

- The Web Dynpro runtime is integrated with the web container, and every Web Dynpro application runs in the Web Dynpro servlet.
- The BPM runtime runs on the Java EE infrastructure. Especially for execution of a process, the cluster capabilities of the server are used to scale execution of many process instances.

Design-Time Architecture in SAP NetWeaver Developer Studio

Composite applications are mainly developed in the Eclipse environment. Guided procedure and SAP NetWeaver Visual Composer models are developed on the server. The specific SAP NetWeaver Composition Environment frameworks have tools that are best suited for their use cases to reduce the development time of an application.

As explained before, there are several design times on the server, but the goal is to bring these toolsets to Eclipse. Lately, there have been some improvements in SAP NetWeaver Visual Composer in Eclipse. SAP NetWeaver Visual Composer runs locally in Eclipse, and it is no longer necessary to connect to a server to model a UI or portal content.

The toolset for the domain-specific models are bundled within the composite designer. This tool provides a consistent overview of a composite application, showing the dependencies and checking if the contracts between the various objects in the various domains are violated.

SAP NetWeaver CE Programming Model

SAP NetWeaver CE provides a programming model like all platforms do. There are specific frameworks for the domains (user interface, process, business logic, etc.),

but the entities of the domains are connected in defined ways. Figure 14.3 shows how they are connected.

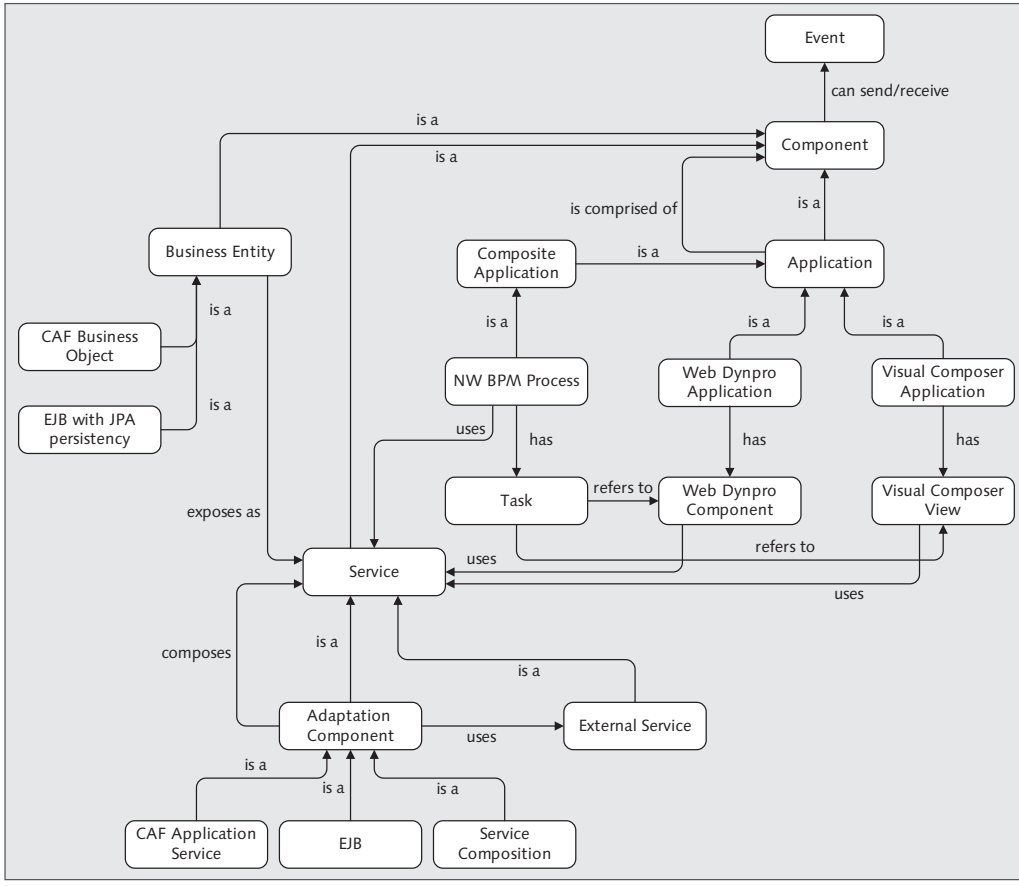


Figure 14.3 SAP NetWeaver CE Programming Model

The domains contribute the following entities to the overall programming model:

- **BPM**
  Provides the process and task definition. A process can use services and tasks. The task itself can use user interfaces.
- **Web Dynpro**
  Provides a Web Dynpro application that can run stand-alone and provides the

Web Dynpro component as reusable entity. Web Dynpro components can use services.

- **SAP NetWeaver Visual Composer**
  Provides the application and the views in SAP NetWeaver Visual Composer models. SAP NetWeaver Visual Composer models can use services.
- **SAP CAF**
  Provides the SAP CAF application service as an adaptation component to adapt data and the SAP CAF business object to locally store data. The SAP CAF application services are exposed as Web services and can compose other services.
- **Service composition**
  Provides service composition models that are exposed as services and can compose other services.
- **Java EE**
  Provides EJB technology to implement business logic and the Java Persistence API (JPA) to persist data locally. Every EJB can be exposed as a service.

The service notion is very important in the composition environment. In the local case, for example, SAP CAF application services and service composition, are exposed as Web services or EJBs. In the remote case, SAP NetWeaver CE can consume services that are available as Web services or remote function calls (RFCs).

14.1.3 Structure of Composites

A composite application usually contains a lot of components (see definition of the programming model above), assembled in development components and software components. The power of the frameworks allows powerful application, but unfortunately the composite application can be structured such that developer productivity, performance, or maintainability is not achieved. The following guidelines therefore explain basic principles for structuring a composite application.

Business Data

A composite application contains a lot of artifacts from different domains. Many of these artifacts can be used to store information. We describe the appropriate type of storage in the following.

The programming model of SAP NetWeaver CE provides various domain frameworks for specific purposes:

- Definition and execution of business processes: SAP NetWeaver BPM
- Definition and execution of user interfaces: SAP NetWeaver Portal, Web Dynpro Java, Java EE (JSP, JSF), and SAP NetWeaver Visual Composer
- Definition and usage of business entities: SAP CAF, Service Composer, and Java EE (EJB, JPA)

A business entity is not (yet) a concrete modeling artifact in the CE landscape but abstracts (from a consumption perspective) a certain business concern. In SAP NetWeaver CE, we perceive the business entities mainly as data-centric artifacts and as mediators when accessing services that operate on the associated business data.

Alternate Approach

If you opted for the loosely coupled approach, utilize the business entity concept to transform business data from the type system of the service providers (e.g. SAP backend BO) to your canonical data type system in the composite context and vice versa.

Depending on the chosen implementation technique, business entities have various characteristics. Very prominent implementation/usage patterns are:

- Provide intermediate data storage for the local execution context (SAP CAF, Java EE); see Figure 14.4.

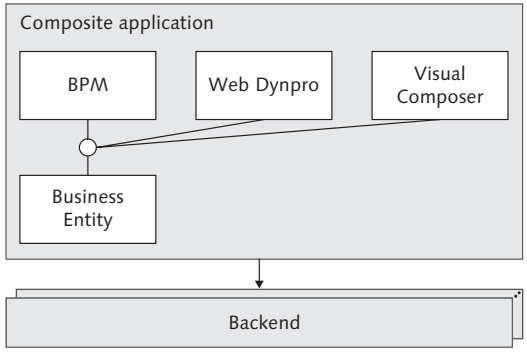


Figure 14.4 Intermediate Storage

- Provide facades to access a single or multiple more complex entities in a simplified manner, such as enterprise services or business objects (Service Composer); see Figure 14.5.

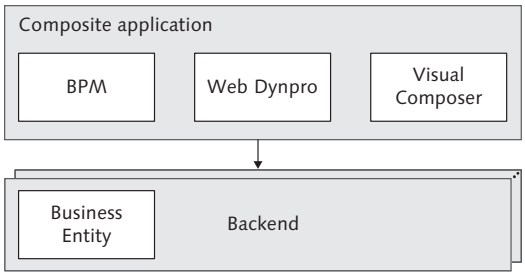


Figure 14.5 Storage in the Backend

A design decision is made at the beginning of the composite application development concerning where to store the data. Both options are possible, and which one is appropriate depends on the business logic. The local persistency of data is the preferred option if it is acceptable to have a copy of the data in the composite application. Keeping the data in the composite would mean the data in the composite application has to be synchronized later with the remote systems later. If the data in the backend systems always has to be consistent, and a local copy is not acceptable, then the data can only be persisted in the remote systems.

SAP NetWeaver BPM and the UI technologies also provide their own method for managing data in their context, which can be used to store/transport data from the business context. There are therefore various options for dealing with the data and the characteristics of the storage concept:

- **Storing data in the BPM process context**
  A data object in the workflow holds the data that has to be available for all human activities and automated activities. The data is only transferred to the activities in a mapping step. After an activity is executed, the data object is updated.

  Scope: Data in a process context is persistent and available during the lifetime of the workflow, that is, only in the scope of the process instance.

- **Storing data in the UI component context**
  If the composite is a UI application without a process surrounding it, it is a valid option to store the data in the context of the Web Dynpro application. This option is only possible if the data should not be persisted on the server side, that is, to survive a server shut down.

  Scope: Data in a Web Dynpro context is transient — available during the lifetime of the UI component, that is, a user session.

- **Storing data in a business entity (here SAP CAF)**
  Business objects in SAP CAF can be used as persistent storage of data that has to be available in processes and UI applications. Here, the lifecycle of the content is not bound to a UI session or process instance. It has a lifecycle of its own.

  Scope: Data in an SAP CAF business object has its own lifetime, is persistable, and is not coupled to anything else.

Based on the concept of separation of concerns, but also based on performance considerations, we recommend that you keep the amount of business-related information stored in the process context small. Try isolating the business data and maintaining it in a business entity instead.

Leading Artifact

A composite application contains a lot of artifacts from various domains. Many of these artifacts can be used to store information.

During the definition process of the composite, it is important to think about the leading artifacts of the composites. A leading artifact is an entity in a composite application that is accessible during the whole lifecycle of a composite and can hold the data so that other entities can access the data and do not have to replicate all content.

Therefore, the composite application designer has to decide which would be the leading artifact of the particular composite or part of it. This is essentially a question of the scope of the data and the lifecycle. There are different options based on some criteria:

- If the process definition is the most important entity, the natural choice is for the data to follow. As a consequence there would be no local persistency (after execution of the process instance).
- If the data is important and should have its own lifecycle, an SAP CAF business object is a good choice.
- If the data is only relevant in one UI task, the data can be stored in the Web Dynpro context.

As usual in application development there could be a mixture of these approaches, and this is not a contradiction. The only decision to make is how important a specific part of the application is, for example if the data simply follows the process definition, if the data should only be available in a user interface, or if the data should be persisted in the database.

Nevertheless, there are some possible overlaps. It would be possible, for example, for the data to be persisted in an SAP CAF business object or the process context. Both solutions therefore have advantages and disadvantages (see Table 14.1).

Arguments for Storing Data in the Process Context	Arguments for Storing Data in the SAP CAF BO and Referencing It in the Process via ID
Data that is required in the whole process is connected with the process itself by defining it in the process.	▶ The data can be accessed independently of the process instance. Data has its own lifecycle. ▶ A large amount of data is not transferred between activities. Only the ID is passed between the process steps.

Table 14.1 Storage of Data

Complexity

Complexity is one of the key problems in designing a composite application. It can exist at different places:

- In the control flow
- In the data flow

Avoid Complexity in Control Flow

SAP NetWeaver BPM offers a graphical design tool. The process graph shows control flow and data objects and their usage in process steps. This allows for a quick overview of the process flow in general *if* the size of the process remains manageable. *Manageable* means the process designer does not have to look at and simultaneously understand the interaction between more than 25 to 50 graphical objects at a time. Our first recommendation is therefore:

Recommendation: Keep process model complexity at 25 to 50 process steps.
If your model grows beyond 50 steps, there will be clear candidates for reuse, that is, portions of your process that encapsulate a meaningful (set of) feature(s) worth putting into a separate process. Use subprocesses in SAP NetWeaver CE 7.20 in the form of *referenced subprocesses* wherever possible to encapsulate reusable functionality. In SAP NetWeaver CE 7.11, use an automated activity to invoke another process. You can also leverage *embedded subprocesses* in SAP NetWeaver CE 7.20 to manage the complexity of your process model even if your subprocess is not intended for reuse.

Be aware that there is also a runtime aspect to this recommendation. The process visualization (graphical log) will look as complicated as your design time model but in addition has instance data attributed to it. The business log (textual representation of your process instances) needs to convey this information in textual form; there it will be even harder to see complex relationships between steps.

Closely related to the model complexity in terms of size (number of steps) is the shape of the model in terms of its graph. A nice compact graph attracts a second look, whereas a large, chaotic graph easily distracts from what it wants to convey: The idea of your business process could be in a single picture. BPMN is a very powerful graphical notation that you should use with care. One feature in particular tends to let people lose the overview of a diagram if used too excessively — hence our next recommendation:

Recommendation: Model block-oriented wherever possible.
Taking advantage of BPMN means you utilize the *flow-oriented* modeling technique, which is well suited to describing real business processes. Nevertheless, the *block-oriented* modeling concepts can come in handy sometimes: A block is a piece of a diagram with one entry line and one exit line. This makes it a portion of a diagram that easily lends itself to refactoring and that can, for example, be easily converted into a subprocess and be easily copied from one part of the diagram to another. Block-oriented diagrams, with their typical nesting and lack of overcrossing lines, tend to look simpler and convey the existing structure more easily. BPMN allows you to connect any step in your diagram to any other step: this feature tends to clutter the diagram and, if the source and target are not on the same screen, easily lets you lose track of the control flow. It should therefore be used with care.

Avoid Complexity in Data Flow

The BPMN diagram clearly shows the control flow for your process. The data flow is only visualized for top-level data elements, however. To find out where a certain attribute of a large structured data object is populated, where it is manipulated, and which steps it is actually consumed in requires navigation to each step and inspection of the input and output mappings.

Therefore, it wise to carefully plan the data context of the process (*global data*) and the data context of each step. A process designer must consider which data is actually required to "drive" the process (e.g., status variables that are needed to make decisions in the process, deadlines that need to be monitored, relevant user information): We call this *primary data*. In contrast to this is secondary data, which is only "carried" by the process because the process acts as a data mediator

between steps (e.g., details read via a service call that is then passed to a UI) or as a convenient generic persistence layer.

Recommendation: Name your primary data as such.
Use meaningful names, comments, or your own conventions to achieve this. Primary data, in particular status variables, must be understood by anybody who attempts to use, manipulate, or refactor the process. Overlooking or misinterpreting primary data results in difficult-to-catch application errors and misuse of the process.

Recommendation: Minimize the amount of secondary data in your process.
If you reuse services or UIs, you have to work with the given set of interfaces that you have to populate or interpret, meaning your choice of data you have to provide or accept is limited. You should, however, be able to use reduced (thinned out) data structures within your process that only store the attributes you actually need. In SAP NetWeaver Composition Environment [7.20], you can use CE service adaptation to thin out vast interfaces to the data you really require. See the performance and data volume sections to fully appreciate the meaning of this recommendation. See the concept/notion of business entities as well.

If you determine the interfaces of services and UIs yourself, for example, because you design them as part of your project, you should do everything you can to adhere to this guideline. In particular:

Recommendation: Design new interfaces with small signatures.
Let these interfaces expose less than 25 parameters and favor copy-by-reference to copy-by-value. Whereas the first part of the recommendation is self-explanatory (smaller interfaces = fewer dependencies, less complexity, and less data overhead), the second part merits an example because it is valid both for services and for UIs. Passing references (e.g., to business objects) typically works well if the receiving application can interpret the references and can access the referenced object's data effectively. Passing references typically means you transport only the ID/key of a business entity between different participants in the flow. For example, if you pass the reference to a service contract, the contract number, for example, to a UI, and the UI application can use this key to read the required details of the contract, such as the issue date, the contract value, and so on without a performance penalty. As a consequence, the UI's interface contains only the contract number instead of the 75 attributes that a fully fledged contract business object might have.

Copying data (as opposed to referencing it) has several other unwanted implications, such as:

- The copied data in your process can run out of sync with the original object's data. This can cause the process to make incorrect decisions, because it is based on outdated information.
- The copied data can pose a security risk if it is sensitive. In releases 7.11 to 7.20, SAP NetWeaver BPM does not offer fine granular access rights to process data. If you are allowed to see the process instance, you are allowed to see all of its data. Not every contract value or every business decision is suitable for viewing by an administrator. A service or a UI always implements dedicated security policies that are more finely granular (because they are specific to the object at hand) than a generic business process infrastructure.
- The amount of data that is generically stored with your process grows with the size of your data context. See the performance section for further impact.

Recommendation: Keep the number of attributes to be mapped at any one interface below 50.

Finally, a large data context or a large activity interface requires large data-mapping definitions. BPM supports a graphical mapping tool that can visualize even complex mappings efficiently. In practice, however, there are limits to the efficiency of any graphical tool.

If you find that more than 50 attributes are to be mapped, apply the recommendation "Design new interfaces with small signatures" guidelines above. A mapping designer will easily lose track if the entire mapping does not fit onto one screen. Refactoring and maintenance of mappings like this are problematic: Did I really need attribute X? How does the underlying UI interpret attribute Y? What is the correct mapping function for attribute Z? In some BPM projects, the mapping consumes up to 50 percent of the time needed to design a process. This is a clear indicator that simplifying the data structure will reduce project costs in at least the mid to long term.

Another aspect must also be mentioned here. SAP NetWeaver CE allows you to design a composite application very easily using BPM. One reason for this is that the BPM tool provides a generic persistence; at no extra design cost, all attributes of a business process are saved automatically whenever the corresponding process instance reaches an automatic save point. The process engine takes care of rollbacks, transaction handling, and so on. As with any automatism, there are limits.

Performance Considerations

Performance must be considered in two aspects:

- Design time
- Runtime

Design Time

In many ways, design-time performance is directly related to the size of your models. If you use all features of SAP NetWeaver CE at once, meaning the process editor, UI designers, composite designer, and so on, many elaborate components will be competing for CPU and memory resources. Significant improvements in terms of memory consumption in SAP NWDS have been implemented in version 7.20. Focusing on BPM, it is worth mentioning a few simple hints that can help you significantly reduce the footprint of the process composer.

Recommendation: Use "move-corresponding" wherever possible.

This is especially true for large structures. A mapping definition is represented as a model in the design-time repository. The complexity of this mapping model can be quite significant when many individual attributes are mapped from one BPMN artifact to another, for example from data objects to activities or from events to data objects. `MOVE-CORRESPONDING` is a convenient way to move data between structurally equivalent (or similar) deep data structures, based on (sub)attribute names and relative positions. In version 7.20, `MOVE-CORRESPONDING` has been condensed to one mapping command, regardless of the size of the structure to be mapped. In addition, if you design a canonical type of system for your composite and you either deal with high load or have many mapping definitions, make this feature part of your considerations.

In version 7.11, `MOVE-CORRESPONDING` is expanded at design time, meaning a set of mapping statements is recursively generated to map the entire structure. The memory footprint of mapping a larger structure (>50 attributes) in the input and output mapping of a human activity when using `MOVE-CORRESPONDING` can go down by a factor of 10 when using the version 7.20 design time. Memory consumption improves accordingly.

Recommendation: Reuse human tasks wherever appropriate.

To achieve good performance, the number of tasks should be reduced. A task in BPM is a reusable object. Because BPM implements the full web service human task compliant status model, a task should be imagined as a rather complex object (which it is). Reusing tasks as opposed to copying them has a significant (positive) impact on memory consumption in a BPM project.

Recommendation: Housekeeping at the IDE
It is advisable to constantly clean up your projects in the SAP NetWeaver Developer Studio to keep the performance of SAP NWDS on a good level. Close or remove artifacts in the IDE that active processes no longer reference in your runtime systems. Such artifacts could be:

- WD/VC UIs
- Service endpoint definition
- Individual task definition
- Rules

Please note that removing artifacts from a processes model (or referenced development component (DC) is an operation that you should execute carefully. If you deploy such reworked artifacts to a server that has still active instances based on the preceding definitions, it is likely that the deployment will invalidate these instances. This is considered to be an incompatible change. Be especially careful with the service endpoint (e.g., event trigger).

For Web Dynpro Java UI components a re-import feature is provided that allows the user to reread its definition for UI to update the metadata. Again, ensure that you do not perform incompatible changes (e.g., remove a parameter at the I/O of the UI); otherwise, running instances on a server might break when you deploy the new definition. If only compatible changes are made to the UI component, the system will preserve the existing mapping at the I/O interfaces from task to task UI.

There is currently no support from the system to detect incompatible changes automatically.

Runtime
Runtime performance here means factors that influence the CPU and main memory consumption of the process engine that runs on your Java server and executes the instances of your process model. There are several factors like this that you can influence with the design of your process model. A few theoretical remarks about the BPM process engine are required to elucidate the following guidelines.

The BPM engine uses an in-memory algorithm to efficiently share resources (memory and CPU) between all process instances currently in execution. On a clustered installation, one engine instance will run per node. Elaborate load balancing, com-

munication, and fail-over mechanisms are in place to ensure efficient use of the full cluster resource.

The engine executes process instances in a transactional and fail-safe manner. This means the state of a process is stored in the database (DB) at *save points* whenever the process logic, technical constraints, or general monitoring requirements demand it. Because the engine executes an arbitrary number of different process models, all with different data context definitions, it cannot use dedicated (transparent) DB tables to store this data. Instead, at every save point, the data context of a process is serialized to XML and stored as one "blob." When the data needs to be read back, it is fetched from the DB and parsed to re-instantiate the data objects in the memory. This engine-persistent storage is required to enable fail-over of process instances in case of a Java engine — or a hardware failure. If this occurs, the engine will simply reload its previous state and continue process execution from the previous save point.

Whenever a context switch occurs for a process instance, the entire instance must be serialized on the node where it is currently executed and later de-serialized on the node where execution is to continue. A context switch like this can occur when an incoming request is issued on a node in the cluster where the process instance is not currently running. Requests of this kind are service requests, a BPMN event (via correlation), or a human interaction leading to a status change of a task.

For auditing reasons, note that there are at least two save points for every activity in your process: one after creation when the input data is available and another one after completion, when the output data is available. Human tasks can have significantly more save points than automated activities, typically one after each status change (e.g., task created, task started, task claimed, task failed, task completed, etc.). Also note that all major changes to a process instance and its activities (including changed process data) are written to the business log, where they are the basis for providing information about the history and the current state of the process (in graphical or textual form).

In memory, data objects are represented using a standard compliant Service Data Objects (SDO) implementation.

The size and complexity of the data context of a process therefore directly impact engine performance in three ways:

- The data context of every process instance currently in execution must be kept in the main memory. Very large data contexts (>1MB) can significantly impact the number of process instances that can be simultaneously executed.

- When a save point is reached, the process instance must be serialized and written to the database. Serialization consumes application server CPU time, and the serialized data stream consumes DB server CPU time and DB space.
- When a process instance is reloaded from the database, its data is de-serialized, and the Java runtime objects are reconstructed.

Please consider that each attribute in the global data context is serialized and stored in the business log whenever it is changed. Each activity in the process has at least two save points, where the full input and output data, respectively, are serialized and stored.

Sizing

Note that a dedicated sizing procedure document for SAP NetWeaver BPM starting with release 7.20 is available on SAP Service Marketplace.

Structure

The artifacts of a composite application can be structured in many different ways according to the SAP component model. This is an explanation of how to structure a composite application.

During development of a composite application, one of the most important questions is how the application will be structured in projects, in particular, how many software components (SCs) and development components (DCs) will be used. Also, considering build-time procedures and the possibility of clustering content will help resolve structuring issues.

Software Component Granularity

An SC is usually created if the parts of the composite applications should be executed on different servers because an SC is defining the deployment granularity or if the parts of a composite application can be deployed independently, for example, if some of the functionality is optional.

Every software component archive (SCA) defines a distribution archive. The argument for SC granularity is therefore deployment. We definitely do not recommend creating different SCs only to bundle the business-relevant parts together (or separate them). If the parts of a composite application should be executed on one server, and there is absolutely no intention of deploying them on different servers, the content should be put into the same SC.

If the application design is done in such a way that different parts can be distributed independently, we recommend putting the parts of the application in different SCs, because the smallest distribution granularity in software logistics is the software component. The SC is also the entity that is versioned, so the SC is the entity that is meant to structure the deployment.

There are other reasons to define SCs, such as project organization and semantic reasons.

Development Component Granularity

The different domains provide separate DC types. It is therefore a good choice (and sometimes the only option) to put the content for different domains in separate DCs. Therefore, all process content is located in a process composer DC, all Web Dynpro content is part of a Web Dynpro DC, and so on.

A DC is the atom of reuse, that is, the smallest undividable functional piece. The composite application developer therefore has to plan for reuse and proper segregation of reusable pieces into a minimal number of DCs. For generic functions, individual DCs should be created to facilitate their reusability and to collect all required elements of the function in this DC.

As a result, parts that are meant for reuse should be put into separate DCs including sufficient definition time for planning the API and separate API and implementation. You can add the API to a public part of your DC in these ways:

- Choose the DC structure according to its function and not according to the organizational structure of the responsible developers.
- Choose the DCs in such a way that the involved developers work in the same team and at the same location. Only if these conditions are met can you use the inactive state of objects, which is mandatory for distributed responsibilities.

These two guidelines seem contradictory, but you should use both guidelines together, so that together they define the minimal structure of DCs.

After a change, you must rebuild the entire DC. You should therefore choose the size carefully so that it does not contain too many objects. This is a very broad statement, so some more details are required.

Build Time

This section provides more details regarding the general recommendations about how to structure content and what these recommendations look like in the light

of build times. To illustrate how the structure influences the build times, we chose Web Dynpro as an example.

Web Dynpro recommendations to structure the content are as follows:

Recommendation: Optimize the development performance with the best deployment granularity.
Web Dynpro development components (Web Dynpro DCs) should be as slim as possible to accelerate build/deploy/run turnaround cycles. If the Web Dynpro DCs become bigger, the build time will increase, so a small change in coding will result in builds that take a very long time.

Recommendation: Optimize the application architecture with the best Web Dynpro component granularity.
One business task should be implemented in one Web Dynpro component, but the Web Dynpro components should not be too large. If the components become too large, then it is not as easy to distribute the work, because the team of developers works on the same component. This would lead to blocking operations, and reuse potential is low. On the other hand, if the component granularity is too small, the application might not perform very well, because every component comes with a system management overhead. All of the components have to be handled by the system.

Recommendation: Apply Web Dynpro component separation principles.
Web Dynpro components have to be defined for a specific purpose, and a component should not be defined for two purposes at the same time. If a component is responsible for displaying the UI, then it should not do model import. So there are at least visual Web Dynpro components for the UI and faceless components for the model handling (connectivity). Generally, the components are designed for one purpose.

This last sentence in particular indicates that there is more behind the separation than just the theoretical guidelines. A very interesting point in this area is that the build times are to some degree related to the number of DCs. Table 14.2 summarizes the situation for the same functionality, located either in one DC or in four DCs. The measurement is performed on one PC, so the numbers are comparable to each other, but the numbers are not a guarantee that every PC with the same performance will achieve the same performance.

Merged Web Dynpro DC	Separate DCs
\all (a WD DC consisting of four WD components)	31.609 seconds
\change	7 seconds
\create	6.328 seconds
\result	11.141 seconds
\search	25.578 seconds
Sum	50.47 seconds

Table 14.2 Comparison of Build Times

What becomes very obvious is the fact that the build of a DC has some overhead. Opting for too many DCs will therefore create problems during build time.

The structure of the DCs of a composite application is therefore always a compromise between the guidelines for business separation of logic in DCs (as small as possible according to the business needs) and the overhead of the infrastructure for each DC. There is normally a conflict between optimizing either the number of DCs for the fastest overall build (as few DCs as possible) or the fastest build for developers (as many DCs as possible). The numbers in the table should provide some guidance about how much the additional overhead for development would be. If the decision is to use the more coarse-grained DC design, the build times for developers will increase. If the decision is to use fine-grained DCs, the overhead of the infrastructure (build time and maintenance of DC dependencies between the DCs) will increase. It is therefore important to make a balanced decision.

The actual guidelines are highly dependent on the application use case, but you still have to take into account the build time and infrastructure aspects during the decision process of the composite application structure.

Clustering of Content

Usually a composite application contains a lot of entities of one programming model, such as Web Dynpro components and EJBs. The recommendation is to put entities that are highly related into one DC. This does not mean all components are put into one DC. In principle, it is a question of how to cluster the components in DCs. A cluster belongs in one DC. If the connection between two potential clusters is very weak (only one or two dependencies), the components are put into two DCs. This recommendation can be defined as follows:

- If a set of components uses the same set of external components, the set of components should be put into one DC (see Figure 14.6). They belong to a cluster of components.

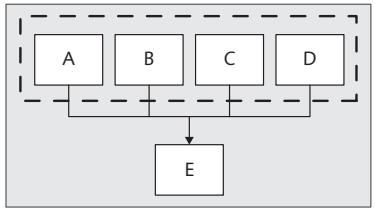


Figure 14.6 Bundling of Consumers

- If a set of components is grouped together and only one component is accessible externally (defining a public API), the components should be put into one DC (see Figure 14.7). They belong to a cluster of components.

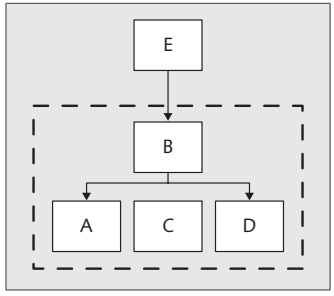


Figure 14.7 Bundling of Used Components

At the start of the composite application design process, you therefore need to think about the components and how they are connected.

Using Services

A composite application usually uses services from various other systems. This section describes how you can use these services.

Usually when the application design is done for the business functionality and the user interfaces that are required, the next question is how the services that are needed can be consumed. Sometimes the available services are not sufficient to fulfill the business requirements of the composite application, and therefore services have to be defined and implemented in a remote system. This design step is not described here and neither is the definition and implementation of these

services. After this is clarified, and the set of services to be used by the composite is identified, the obvious next question is where the data of the service calls is used. And here the first architecture decisions will be made, because SAP NetWeaver CE frameworks such as Web Dynpro, BPM, and SAP CAF provide the capability to consume services directly. The first decision now is whether a service should be used directly or not. This section will focus on this question and provide some guidance about how this can be done and what the advantages and disadvantages are.

The term *loosely coupling* gives you an indication of the design principle for an application regarding service consumption.

Very often the discussion is reduced to the question of whether an application talks synchronously or asynchronously with its peers. Although this is certainly one aspect, it doesn't cover all relevant dimensions of loose coupling. The goal of loose coupling is to reduce dependencies between systems. Therefore, to provide a definite answer about how tightly an application is coupled to other systems, you can be guided by a second simple question: What are the consequences for system A (the calling system) if you make changes in system B (the called system)? Probably, this question reveals a large number of dependencies between your application and others that go beyond a classification of the communication style between them. There are a few more assumptions about application design. Which assumptions can be made for coupling systems?

- **Assumption 1: location of the called system (its physical address)**
  Does your application use direct URLs for accessing systems, or is your application decoupled via an abstraction layer that is responsible for maintaining connections between systems? The service group paradigm used in SAP NetWeaver CE is a perfect example of what such an abstraction might look like. Using an Enterprise Service Bus is another example.
- **Assumption 2: number of receivers**
  Does your application take care of the receivers of a service call? Or does it simply drop a message "somewhere," and other mechanisms take care of transporting the message to the receiving systems? Don't underestimate the important of this assumption. If you take the loosely coupled approach, it means that you are not making any assumptions about the systems you are talking to. This implies a completely different architecture compared to tightly coupled applications. You never know when or if a service call returns due to the number of involved systems, and your application must be prepared for that.
- **Assumption 3: availability of systems**
  Does your application require that all of the systems you are connecting to

are up and running all of the time? This is obviously a very hard requirement, especially if you want to connect to external systems that are not under your control. If your answer is "Yes, all of the systems must be running all of the time," you are obviously tightly coupled in this regard.

- **Assumption 4: data format**
  Does your application reuse data formats as they are provided by the back end systems, or do you use a canonical data type system that is independent from the types of systems used in the called applications? If you reuse the data types from the backends, you probably have to struggle with data-type conversions in your application. This is not a very loosely coupled approach.
- **Assumption 5: response time**
  Does your application require the called systems to respond within a certain (acceptable) time frame, or is it acceptable for your application to receive an answer minutes, hours, or even days later?

There are even more dependencies, but the message should be clear: loose coupling is not one-dimensional. For each of the aforementioned aspects of loose coupling, you have to make decisions. And they are not easy, because moving toward loose coupling has serious implications for the architecture of your application. So loose coupling comes at a price, especially in terms of complexity, and you have to decide whether you want to pay the price for it.

The benefit of loose coupling is flexibility and agility. If you are aiming for a loosely coupled approach, you will get unparalleled flexibility for adaptations to changing landscapes. Because you aren't making any assumptions about the landscape your application is running against, you can easily adapt it as needed (provided your frameworks and tools support you like SAP NetWeaver Composition Environment does). This is especially important for partners and independent software vendors (ISVs) who can develop applications once and easily install and configure them at their customers' side. The application itself stays untouched.

It isn't just partners and ISVs who will benefit from this approach. It is useful within companies as well: Once you've established a successful new application, you will most likely want to reuse it within your company in other locations or regions. Very often, the IT landscape in the new locations differs from the one the application was originally designed for. If you take the loosely coupled approach right from the beginning, this undertaking will not frighten you. Another aspect you should consider is the probability of landscape changes during the lifetime of your new application. Due to mergers and acquisitions, or due to system con-

solidations, the landscape underneath your application is constantly changing. If you are not prepared for loose coupling, you'll be forced to adapt your application again and again.

Concrete Implementation of Loose Coupling in Service Consumption

The concrete implementation of the loose coupling principle in a composite application can be performed as follows:

- Enterprise services are used only via WS proxies in EJBs or SAP CAF external services.
- There is an intermediate layer that abstracts from the underlying enterprise service so that only the used data is exposed to the composite application. The intermediate layer defines a service contract to the upper layers, and the implementation of these interfaces is called a service contract implementation layer.
- The upper layers of a composite application (Web Dynpro, SAP NetWeaver Visual Composer, or SAP NetWeaver BPM) only use the service contracts.

Figure 14.8 shows the architecture of a concrete implementation of the loose coupling principle.

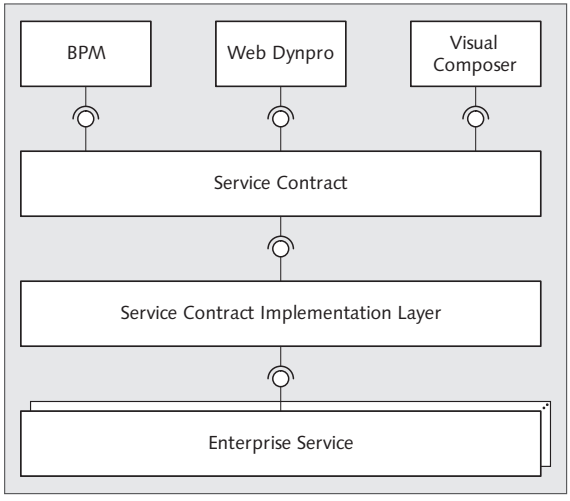


Figure 14.8 Loose Coupling Architecture

With the service contract layer, it is possible to achieve decoupling from a concrete landscape, data-type structure, communication protocol, and so on. The service

contract implementation layer can be implemented in different ways. First, the composite application design has to be clear if a complete decoupling has to be achieved. In that case SAP provides the following solutions:

- **SAP NetWeaver BPM**
  The service contract implementation layer implements an interface by using BPM modeling capabilities to allow execution of service calls asynchronously (see Asynchronous Write in Section 14.1.5). If an error happens during invocation of the Enterprise Service, a full error handling including user steps can be implemented.
- **SAP NetWeaver Process Integration (PI)**
  The implementation of the service contract implementation layer is not done via SAP NetWeaver CE; it is done via the process integration hub of SAP. The benefit is the same as the case of SAP NetWeaver CE; the only difference is that the capabilities of SAP NetWeaver Process Integration can be used (e.g., routing, ccBPM, etc.).
- **Java Message Service (JMS)**
  The service contract implementation layer contains a very small implementation of the service contract, and only an event via JMS is raised that is processed on the Java server itself, so the message queue is used to decouple the functionality. The benefit is that only Java EE technologies are used.

The above options decouple the execution of the enterprise service call from the basic composite application. If the asynchronous behavior is not a requirement, then the other technologies of SAP NetWeaver CE (SAP CAF, service composition, EJB) that transform the delivered data of the enterprise services to the composite can be used. This solution has the drawback that a complete decoupling of the UI parts of the composite application is not achieved; it goes in the direction of tight coupling with all of the disadvantages.

Structure of the Service Contract Layer in a Composite Application

The service contract and the service contract implementation layer have to be structured in a way that real decoupling is achieved. Figure 14.9 shows the structure of a decoupled composite application.

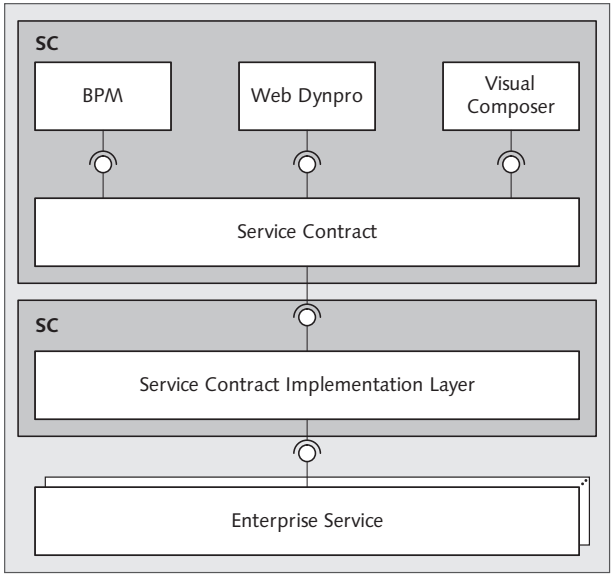


Figure 14.9 Structure of a Decoupled Composite Application

The entities of the composite application and the service contract definition are contained in one software component (SC), and the service contract implementation is contained in another SC. The contract and the implementation are in different SCs, because this allows switching of the implementations. These software components do not have any dependency, because they are loosely coupled and the communication between the composite application and the service contract implementation is only done via interfaces and configuration of the used services in the composite application.

Concrete Implementation of Tight Coupling in Terms of Service Consumption

Loose coupling is not always the best architecture. If you don't require flexibility, performance indicates that an intermediate layer is not allowed, or dependencies to the backend systems are not an issue, SAP NetWeaver CE allows all frameworks (Web Dynpro, SAP NetWeaver BPM, SAP NetWeaver Visual Composer, etc.) to call services directly. Figure 14.10 shows the architecture.

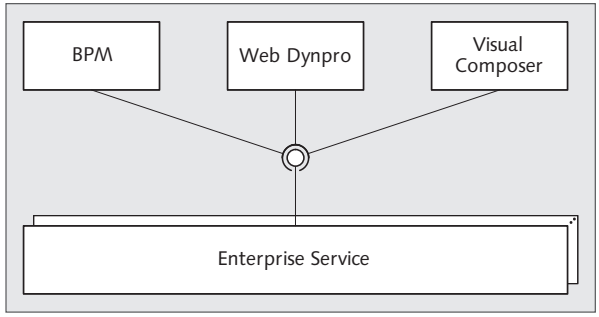


Figure 14.10 Tight Coupling

Example

The most commonly used implementation pattern in SAP NetWeaver CE is to implement an application service in SAP CAF that consumes one or many external services. The Java coding inside the application service is then responsible for mapping the data to the correct output data.

The intermediate SAP CAF application service is used if one of the following abstractions is required for the composite application:

- **Concrete landscape**
  The application services that are exposed by SAP CAF can be deployed with the composite application. If the landscape changes at the customer's side, however, other application services can be implemented using the same signature. Their implementation differs in such a way that the data is now retrieved from another system (or systems) within the customer landscape. Only a small configuration step is needed to activate the new application service for the composite.
- **Data type structure**
  The enterprise services provide a specific data type structure and require these data structures. If the application is not interested in all data, it is all right to implement an SAP CAF application service, define your own data types in SAP CAF, and only expose the data structures of CAF.
- **Communication protocol**
  SAP CAF provides the functionality to call Web services and RFCs. If the application is loosely coupled in the sense that it should be able to run with Web services and get the data via similar RFCs later on, the recommended way is to implement an SAP CAF application service and implement the switch inside the

application service. Whether the data comes from an RFC or a Web service is transparent for the upper layers of the application.

Web Dynpro and SAP NetWeaver Visual Composer can follow either loose coupling or tight coupling architecture styles depending on the specific requirements defined in the general considerations to use tight coupling or loose coupling.

Task Handling in a Process

A composite application usually contains process definitions and corresponding tasks. This section describes how to define the tasks.

As of Release 7, enhancement package 1 and enhancement package 2, the lifecycle of a task instance is bound to a surrounding process in SAP NetWeaver BPM, even though it is possible to define a task as a stand-alone artifact in its own DC.

Under very specific conditions, you can reduce the number of DCs. Imagine the following situation:

- Multiple SAP NetWeaver BPM processes are modeled that contain a human activity as their single artifact (ignoring start/end events).
- The tasks that back the human activity all point to the same task UI and differ only in the occurrence of the process context.
- The task description can be constructed by using/interpreting the process context.

In a situation like this, implement only a single, generic process, and create tasks and their work item description based on the process context, which needs to be fed accordingly by the start event message.

We therefore also recommend investigating whether existing processes, tasks, and task UIs can be refactored to make use of this approach.

As an example for a generic approach, you have a set of different processes running in the backend (say process A and process B). For each of the processes, at a certain point a confirmation should be requested from responsible users. Instead of modeling the processes "confirmation for step x (process A)" and "confirmation for step x (process B)," model a single process, "generic confirmation." Your start event message should carry all data that you require to make the necessary recipient determination and fill your fields in the confirmation UI. Add a human activity to the process. Implement your notification UI in Web Dynpro, and fill all task-specific information from the context. Connect your Web Dynpro UI as the

task UI to the activity in your process. Pass all your task-specific details from the start message to the human activity/task (thus the Web Dynpro context) via data mapping.

Alternatively, if the confirmation task is still trivial and based on the same attributes of the different processes but requests different UIs, think of a gateway in the process to select the different task and thus the target UIs.

Extensibility

The extensibility configuration framework was introduced in SAP NetWeaver Composition Environment 7.2, and the technology requires a specific structure. This section describes how the composite application has to be structured.

Extensibility is an important issue during the development of a composite application. Extensibility is normally understood as the ability of SAP NetWeaver CE to change an application behavior without design modification (only certain implementations are replaced). The support of the composite application therefore stays, but new functionality can be plugged in.

There are many ways to design an application for extensibility purposes, but SAP NetWeaver CE provides a specific framework for extensibility. The extensibility configuration framework demonstrates an innovative conceptual approach, which allows different interface implementations of the application parts to be exchanged during the runtime of an application.

To do this, you first need to define extension points in your application. An extension point is a reference to a development object interface of a redirectable technology. The extension point allows exchanging or actually redirecting to different interface implementations during the runtime of an application.

You can create extension points for the interfaces of the following development objects:

- Enterprise JavaBeans (EJBs)
- SAP Composite Application Framework (CAF)
- Web Dynpro. In the (individual) Adobe Interactive Forms (AIF) use case, the extension point is not an interface, but a single form marked as extensible.

To create an extension point, there must be at least two development objects from one of the above types, because there needs to be a relation between them.

The extensibility configuration framework requires a specific structure of the application. The application has to be prepared to be extensible.

SAP CAF and EJBs have to follow the same principle. Even if the EJB is only a place holder, the redirect from an existing implementation in an EJB can only happen in an EJB container. Therefore, an EJB that has to be redirected has to be invoked from another EJB.

The consequence of this limitation is that an EJB that is consumed by a web application (Web Dynpro, servlet, etc.) has to be wrapped by another EJB, because invocation of the first EJB is performed by the web container, not the EJB container of the Java EE engine.

The extension components (components that replace the functionality of the composite application) have to be put in a customer *product*. The original product does not contain the extensions; the extensions are put in the customer product that has a dependency to the original product.

14.1.4 Separation of Functionality

SAP NetWeaver CE provides frameworks for different purposes. There are frameworks for business logic implementation, user interface modeling, and process modeling. The functionality of these frameworks sometimes overlaps, and you need to decide when to use which framework. The following explanations will provide some guidance.

UI Flow (SAP NetWeaver BPM and Web Dynpro)

The issue: The various steps of a user interface can be implemented in different ways. In particular, Web Dynpro and SAP NetWeaver BPM can be used to model the user interface steps. This section describes how to decide how to model the steps.

In general, both Web Dynpro and SAP NetWeaver BPM work well together for the realization of business processes or composites. How to wire them together or to use them exclusively depends on the concrete requirements of a business process. As already described, business processes can come in a plethora of occurrences. In this chapter, we will focus just on human interaction, because this is where SAP NetWeaver BPM and Web Dynpro are complementary.

The two technical offerings are best described by the domains that they belong to. SAP NetWeaver BPM allows you to model and execute workflows (generally it is a

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z