SOA in large-scale SAP templates

Dipl. Wirtsch.-Inf. Michael Sambeth
Dr. Peter M. Heinckiens

SAP Deutschland AG & Co. KG
Hasso-Plattner-Ring 7
69190 Walldorf
michael.sambeth@sap.com
peter.heinckiens@sap.com

Abstract: SAP templates are a frequently used form of SAP software and business process deployments. This paper outlines common challenges when dealing with SOA and interfaces in large-scale SAP templates, such as global templates. It focuses on the architecture principles around the traditional tension point of integration: balancing standardization with local innovation. This includes an exemplary view on the components to be applied and their lifecycle management throughout the release management of the SAP environment. A customer example is used to illustrate the challenges and the applied architecture.

1 SAP templates

Template development takes place on process, application and technology level. The result is a re-usable reference system. Template management allows customers with multi-site SAP installations to efficiently manage their business processes across geographical distances, such as part of a global rollout approach: from initial template definition to template implementation, and template optimization.[1]

Figure 1: General template approach

[1] See also http://www.sdn.sap.com/irj/sdn/alm-template-management

A combination of standardization of processes/master data and choice of rollout approach results in the implementation approach. Integration and interface requirements need to follow the requirements of the implementation approach.

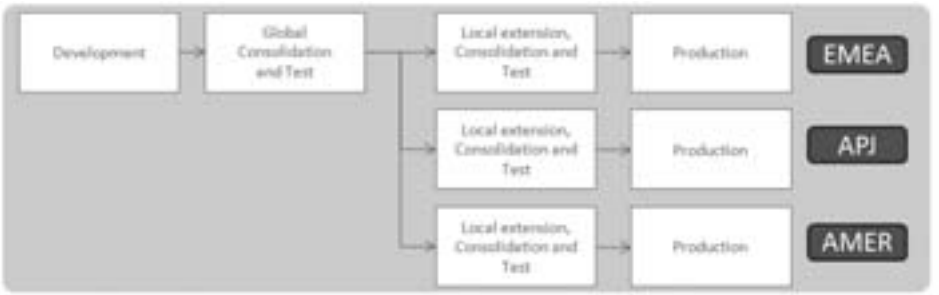

Figure 2: Multi-site SAP template landscape

The physical architecture of templates is designed in a three-tier deployment architecture which typically consists of a development, a consolidation and a production system SAP instance. Depending on the scope and requirements the production system instance can be a single instance (e.g. a single global system) or being several production instances.

The template exposes and integrates interfaces for connectivity with other systems. The interfaces make the business processes of the template accessible (provisioning), which represents a common use case of SAP integration. SAP lists standard interfaces of the SAP Business Suite on the Enterprise Services Workplace documentation platform.[2]

The infrastructure used in order to design, manage and maintain interfaces is provided by the SAP NetWeaver technology platform, mainly with the component SAP NetWeaver Process Integration (NetWeaver PI).[3]

2 Integration architecture use case

2.1 Determination of SOA candidates

The operating model defines the level of business process standardization and business process integration [Ro06]. Enterprise architecture provides the organizing logic for business process and IT capabilities reflecting the customers' operating model. General SOA and interface requirements can be derived from the operating model as well as from the customer roadmaps as they evolve between different maturity states of enterprise architecture.

[2] Please see http://esworkplace.sap.com
[3] Please see http://www.sdn.sap.com/irj/sdn/nw-soa

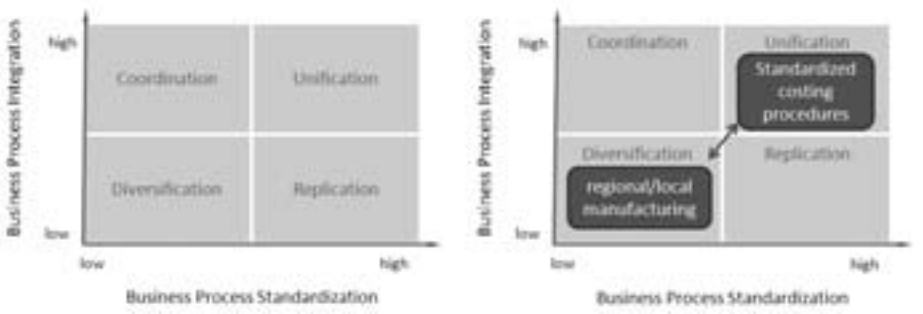

Figure 3: Operating model with Cross-quadrant integration demand example

Integration scenarios that connect elements located in different quadrants represent potential SOA candidates. Such scenarios potentially benefit from the provisioning capabilities of the systems located in the "unification" quadrant as well as from the integration capabilities of SOA middleware, e.g. the discovery, routing and mapping functionality when connecting the systems located in the "diversification" quadrant. Moreover different cardinalities, e.g. multiple local/regional manufacturing plants and a potentially single costing system also suggest the consideration of a de-coupled architectural approach such as SOA.

2.2 Customer business environment

The SAP template is designed to replace several existing systems and thereby creating the new standard for business processes and methods (finance, compliance) [Ds10]. It is positioned at the shared services and assets layer. Significant integration takes place with local manufacturing or supply chain systems – the "logistics domain". In addition to providing a common IT platform offering best practices the template is also used to ensure mandatory processes.

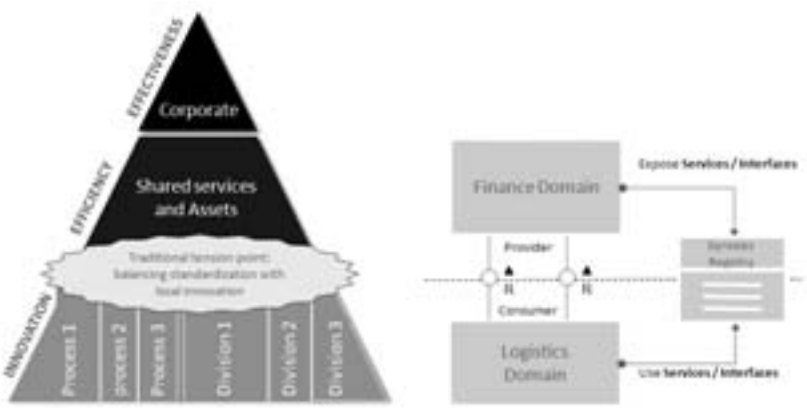

Figure 4: General approach

Services are offered at the process domain border.[4] As a result, the capabilities of the interfaces and processing "contract" between those two domains impact the overall architecture, complexity and cost of the template.

2.3 Customer example functionality "Production Control Monitor"

The functionality "Production Control Monitor" (PCM) implements product costing for automotive vehicles. This includes produced vehicles as well as refitting and vehicle acquisition (e.g. used vehicles or leasing buy-back). The exposition of PCM in the finance domain is driven by the standardization of the costing and calculation processes. This leads to increased preciseness of valuation on stock and allows different manufacturing plants to be potentially compared with each other. Further optimization of such as time and effort for regularly reporting and others are supported by this strategy.

The costing process is initiated by either a newly manufactured finished vehicle or by an event that affects an existing vehicle. The PCM process itself is a complex process consisting of multiple steps that are executed inside the SAP ERP system. Core process steps are bill-of-material explosion, costing of material and labour, the valuation on stock and an optional integration of the controlling modules in e.g. case of refitting.

Dimension	Challenge	Decision
Heterogeneous system integration	The data structure and data types should not be SAP proprietary	The interfaces are designed outside-in leveraging the principles of SOA and the SAP Global data types.[5] This allows evolving into open and technically increased interoperable interfaces from and into the participating SAP systems.
Standardization and localization	Foster the transition towards the new standard with minimized effort	Leave legacy systems untouched where possible. Apply SOA middleware for mapping of legacy data or local interfaces towards the new global standard interface structure. New interfaces should be integrated directly to global standard interfaces. Minimize exceptions (e.g. when a local interface cannot be mapped against the new global standard and hence a local message type and interface is needed in the ERP system).
SOA middleware	Scope of the SOA middleware during runtime	Use SOA middleware primarily for technical processing (e.g. XML schema validation, mapping, routing, service and event management), not for stateful business process related actions.

Table 1: Sample SOA challenges in global SAP template

The provisioning of the PCM functionality as a service yielded several challenges. The decisions shown in table 1 illustrate a dependency between SOA-related decisions and the SAP template strategy. The technical design of the template relates to the design and requirements of the interfaces and that relates to the processing inside the SAP system which is following the template functional scope and roll-out strategy.

[4] In this case the logistics domain consists of a high number of systems, mainly legacy systems, whereas the new SAP template to be introduced is a central component of the finance domain.
[5] Please see http://wiki.sdn.sap.com/wiki/display/GDT and the SAP Enterprise Services Repository.

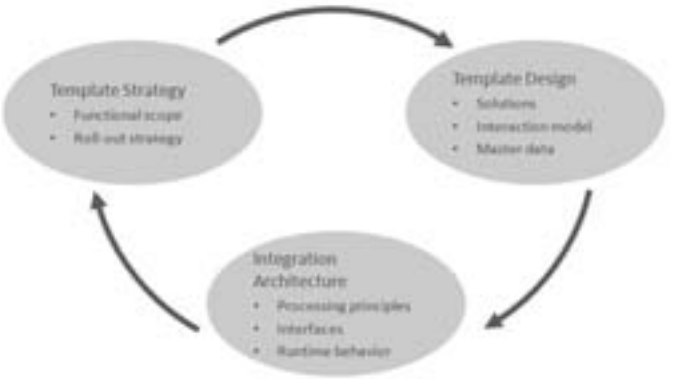

Figure 5: General influencing factors

Additional aspects of the cost dimensions with relation to SOA and interfaces in the project are covered in the next chapter.

2.4 Customer project cost dimensions related to SOA

In addition to the PCM example the template included ~30 processes in procurement, payroll processing and customer ordering. Due to the expected messaging volume it was important to take additional considerations of the SOA-related aspects of the template:

- Count of interfaces: The granularity of the services cut impacts the resulting number of interfaces and hence the maintenance cost. The project established their derivation rules. Moreover this has lead to the necessity for an additional investment to extend the processing of the ABAP proxies. [6]

- Sourcing: A dual sourcing strategy for service implementation was established: on one hand side using highly skilled people for the design and derivation of services and on the other hand side allowing a factory-like implementation of the coding or mappings thereby guaranteeing maintainability of the code.

- Portability: The IT environment of large organizations multiple systems share interfaces with those central / global systems. The interfaces and related coding of this template were designed with the target to being portable to other SAP templates in the customer environment.

- Structural communication errors: Clustering of errors and relation back to the sender in order to give the key user directions when searching for the root cause of the error. Moreover, the errors can be corrected in a mass-handling fashion. [7]

[6] This resulted in a much smaller number but functionally broader interfaces compared to having leveraged the interfaces of the SAP standard. However, additional components were being applied in the SAP template to extend the ABAP proxy implementation.
[7] Additional components were being applied in the SAP template to optimize the error handling for key users.

- Empowerment: Key users don't have access the SOA middleware. Instead, business process errors should be resolved on the SAP template system.[8]

About 80% of the interfaces are implemented as services, ~20% are SAP iDOC interfaces. By treating all service interfaces the same way a strong cohesiveness on interface granularity and implementation was achieved to lower maintenance cost.

2.5 Lifecycle management

Application lifecycle management (ALM) provides processes, tools, services, and an organizational model to manage solutions throughout the complete application life cycle.

Figure 6: Tuple of participating components of a production instance

ALM controls the propagation of changes from development via consolidation into the production landscape thereby managing the dependencies of the artefacts of the different systems, e.g. an interface which is exposed through SAP NetWeaver PI is dependent of the corresponding ABAP proxy implementation and coding. Architectural principles ensured that the NetWeaver PI development system is aware of all interfaces. At runtime, the NetWeaver PI system belonging to the ERP production system is applied.

Figure 7: SAP template lifecycle management

All software artefacts can be distributed and managed through SAP Solution Manager. This includes ABAP coding, Java coding as well as SOA content belonging to the Enterprise Services Repository such as data types, message types, interfaces, mappings and others. Quality gate management is used to define "synchronization points" for multiple or heterogeneous transports.

[8] On one hand side some legacy logistics systems were technically unable to efficiently process the responding error messages. On the other hand in case of finance systems most of the interfaces are SAP-inbound.

3 Summary

Service provisioning and consumption in an SAP environment is tightly connected to the business processes of the SAP system. The design and application in a large environment should be based on a few but central guiding principles. Examples for such principles are the derivation rules for the exposition of SAP functionality, the granularity of interfaces or the runtime error handling procedures. This typically affects the blueprint phase of an implementation project. It seems hardly possible to introduce SOA to a large-scale SAP template in later project phases. Tools and components necessary to model, code and maintain SOA related artefacts in a SAP environment are available today.

The featured use case gave a successful example on how to manage SOA-related requirements in the scope of a SAP template design and deployment. Key questions such as opening up business processes or questions around cost-efficient implementation work are addressed through architectural principles. These principles especially ease communication and collaboration between the teams having different professional backgrounds, e.g. the middleware team and the ERP team. Major integration challenges could be solved through embedding of SOA thinking in the SAP template architecture. Moreover, relying on principles instead of tools allowed the project to scale the effects throughout the majority of business processes of the SAP template.

SOA can support business or IT strategies such as process domain modelling, system consolidation or the introduction of shared services. Balancing the different dimensions of template design, integration and roll-out approach plays a key role in order to meet business process integration, standardization and project budget goals. In order to succeed, the template core team should be included in enterprise architecture related activities.

The management of standardization and re-use of interfaces represents a central dimension of SOA governance. Other SAP and non-SAP systems in the customer landscape may benefit from the achievements of the designed interfaces and SOA artefacts. This supports evolving from independent silo applications into increased business application modularity.

Literature

[Ds10] Deutschsprachige SAP Anwendergruppe – Vorträge der Technologietage 2010: Standardization of Financial Processes for international locations of Daimler https://www.dsag.de/uploads/tx_dsagbase/documents/02_G_Basica-Sambeth.pdf (download 24.04.2010)

[Ro06] Ross, J.W.; Weill, P.; Robertson, D.C.: Enterprise Architecture as Strategy: Creating a Foundation for Business Execution. Harvard Business School Press, 2006.