Capability-based Service Identification in Service-Oriented Legacy Modernization

FRANK J. FREY, Münster
CARSTEN HENTRICH, PricewaterhouseCoopers AG WPG, Frankfurt/Main
UWE ZDUN, University of Vienna

Moving from a heterogeneous IT landscape with a set of legacy applications to a more flexible SOA-based architecture is a major strategic concern for many large organizations. Key drivers are to build a robust, easy-to-maintain and cost-efficient architecture. Service identification is a key task in this context, and it must cope with the inherent complexity of existing IT systems and their constant evolution. In this paper we describe the CAPABILITY-BASED SERVICE pattern that identifies services and defines the service model based on a model of business capabilities. Business capabilities describe what an enterprise does, not how something is done in the enterprise. Service identification based on business capabilities is an alternative to process-based service identification, especially useful in cases where process modeling is not feasible, for instance, because of budget or time limitations or where processes are not stable enough over time.

Categories and Subject Descriptors: D.2.11 [Software Architectures] Patterns

General Terms: Design

Additional Key Words and Phrases: SOA, business capabilities, patterns, software architecture, service-oriented legacy modernization

1 INTRODUCTION

A software pattern is a technology independent conceptual solution to a generic software design problem Large enterprises, grown over decades, organically or by mergers and acquisitions, have difficulties to be enabler for business innovation and cost-efficient at the same time. They often have outdated hard-to-maintain software technology and complex applications with overlapping functionality. However, the legacy applications are still essential for many enterprises, because they are vital for the execution of the business and contain valuable business logic (Khadka et al., 2012). Faced with a grown IT architecture with large (monolithic) applications and many point-to-point connections, both difficult and costly to maintain, many enterprises intend to untie and modernize their complex IT by adopting a service-oriented architecture (SOA) based on flexible, loosely coupled services (Maréchaux, 2008).

Therefore large enterprises initiate strategic IT programs for modernizing essential parts of their IT and transforming it into a SOA. For example, Figure 1 shows key drivers of a large IT modernization program for a logistics provider, aiming to standardize its application architecture by replacing outdated, complex applications with loosely-coupled services, re-used for business processes across all relevant regions of the world.

Author's address: Frank J. Frey, Vivaldistr. 6, 48147 Münster, Germany. Carsten Hentrich, PricewaterhouseCoopers AG WPG Friedrich-Ebert-Anlage 35-37, 60327 Frankfurt/Main, Germany, carsten.hentrich@de.pwc.com. Uwe Zdun, Software Architecture Group, University of Vienna, Vienna, Austria, uwe.zdun@univie.ac.at.

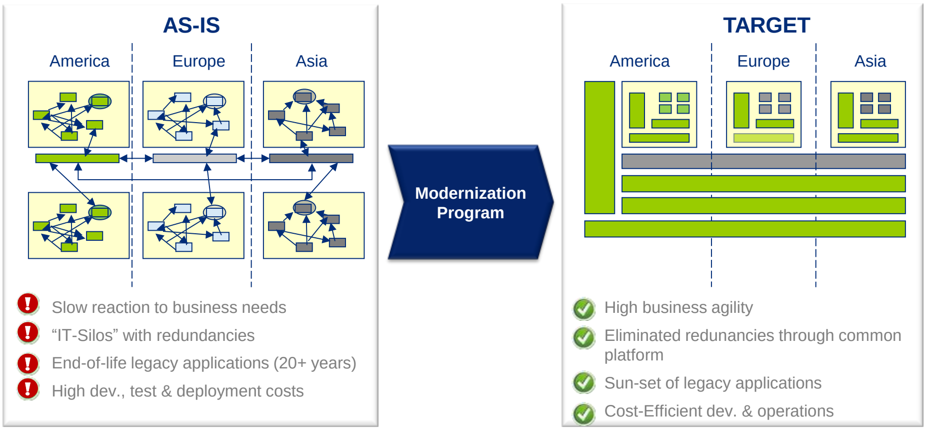
AS-IS
America
Europe
Asia
Slow reaction to business needs
"IT-Silos" with redundancies
End-of-life legacy applications (20+ years)
High dev., test & deployment costs
Modernization Program
TARGET
America
Europe
Asia
High business agility
Eliminated redunancies through common platform
Sun-set of legacy applications
Cost-Efficient dev. & operations

Figure 1: Drivers of IT Modernization - Example from Logistics Provider

Taking into account such key drivers for IT modernization programs, companies expect the following benefits of such programs:

- *cost-reduction* – for example, by replacing legacy applications based on outdated technology and difficult-to-change architecture that cause high maintenance costs;
- *business process optimization* – for example, referring to improving the efficiency of business processes by better IT support with improved user experience and a higher degree of automation;
- *business agility and faster time-to-market* – for example, referring to a flexible IT architecture, enabling businesses to introduce new products or engage with new partners or in new markets.

As SOA concepts and technologies are leaving the hype phase, SOA (Barry, 2003) is considered by many organizations as an option for modernizing their existing IT landscape. In a SOA functionality is provided by loosely coupled services, which are aggregated to higher-level services and re-used for various business processes.

Adopting SOA does not mean that all services must be built from the scratch. In fact, a big advantage of SOA is that legacy applications can be wrapped and re-used as service providers. These services can be used by other services or components in a loosely coupled manner using flexible integration mechanisms, such as an enterprise service bus (ESB) or a service registry. For example, various ESBs provide powerful transformation functions which support creating a standardized conceptual data model used for all services and the integration of various legacy applications with their proprietary data models.

Furthermore moving to SOA is not only a matter of IT, but rather an aspect of a transformation to service-oriented enterprise (SOE) (Masak, 2007). Such a transformation goes along with a focus on business processes both from a functional and from a technical perspective. This leads to the concept of a *process-driven SOA* in which services are invoked to perform business processes (Hentrich & Zdun, 2012). Hence, a *process-driven SOA* is expected to facilitate optimizing and changing business processes.

An architectural modernization program, which aims at transforming legacy IT architectures into a SOA, usually consists of various projects for enabling legacy applications to support services, for developing new services, and for migration. Overall such a program has a timeline of five years or even longer (Brahmanandam, 2011). Defining the scope of a modernization program and determining the target architecture, which is robust and flexible enough to fulfill the business requirements several years ahead, is a critical task, which is performed during an early stage of a modernization initiative. Transforming an application-centric architecture to a SOA means that services will provide the functionality that is currently provided by legacy applications. Furthermore, usually changed requirements need to be addressed by services.

Thus, identifying services that fulfill the requirements of the current and future business is a key concern for the definition of an IT modernization program based on a SOA (Börner & Goeken & Rabhi, 2012). This paper introduces the CAPABILITY-BASED SERVICE pattern, which helps to overcome challenges related to identifying the right services in the analysis phase of large IT modernization programs. This pattern was discovered and derived from practical experience.

It is based on a business capability model, and by this, differs from approaches which need a comprehensive business process model for identifying services. Simply put, a business capability describes *what* an enterprise does, and a business process describes *how* it is done (Rosen, 2010). Considering the example of a coffee shop, the “ability to brew coffee” is an essential business capability. The simplistic, illustrative example in Figure 2 shows that the description of a corresponding business process “brew coffee” entails a more details and describes the activities and process flow of the brewing process.

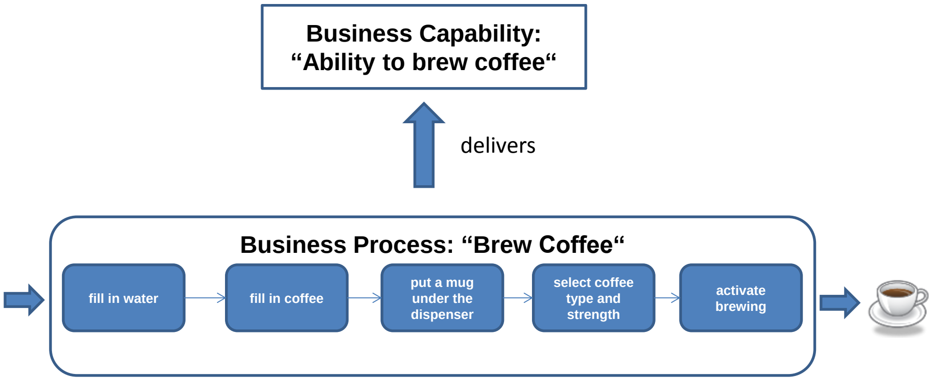

Figure 2: Illustration of the Difference between Business Capability and Process

This paper is structured as follows: Section 2 explains typical problems which arise during the service identification process. Furthermore, related patterns are discussed. With a focus on the analysis of the needs and challenges of these early phases, Section 3 explains the capability-based service pattern that aims to overcome the specific problems and challenges of business-/IT-alignment. Section 4 summarizes findings from a focus group about service identification which we have run with 10 experts to confirm our views on the pattern and further explore the realm of service identification. Finally, Section 5 concludes and gives some suggestions for future work.

2 SERVICE IDENTIFICATION FOR SOA ADOPTION

Although legacy modernization and SOA adoption have been extensively discussed in recent years, still there is no commonly accepted approach for identifying services. Instead, a great variety of methods and approaches exists. For instance, see a recent literature review on legacy to SOA transformation (Khadka et al., 2012).

In some projects the process of service identification is conducted rather intuitively, based on the experience of the involved individual people, without a defined method or heuristic (Börner & Goeken & Rabhi, 2012). However, various structured approaches and methods for service identification have been proposed. We distinguish between top-down and bottom-up strategies. Top-down service identification means that services are identified based on business requirements. Bottom-up approaches start with the analysis of existing applications. Furthermore there are ‘meet-in-the-middle’ strategies, which combine top-down and bottom-up approaches (Fareghzadeh, 2008). Figure 3 illustrates the different strategies for service identification.

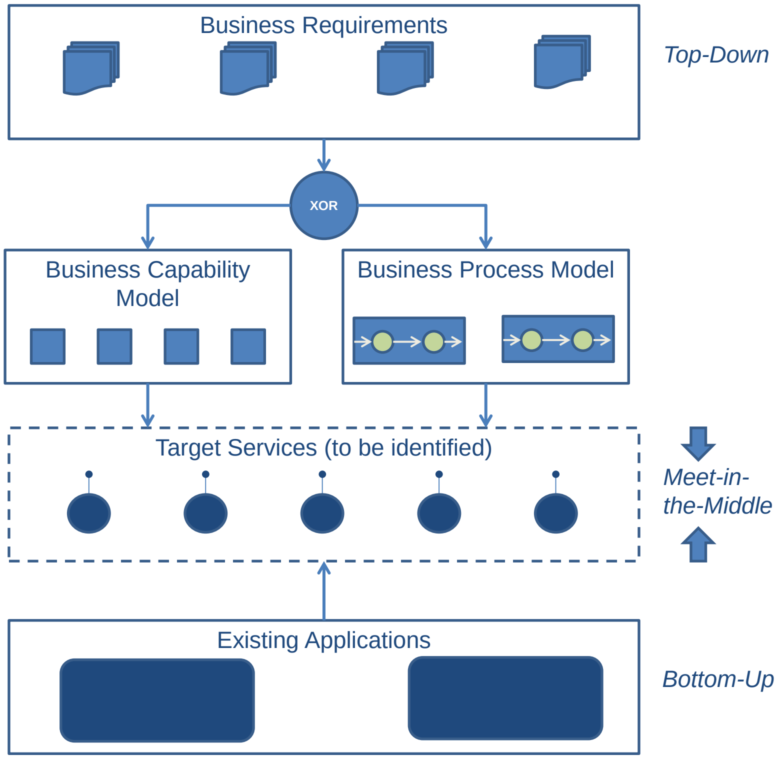

Figure 3: Service Identification Strategies

Different methods for service identification aim at different analysis objectives. For example, during the execution of a project, aiming at building a service-wrapper for a legacy application, the code of this legacy application is analyzed in detail to re-engineer, define service interfaces, and implement the service wrappers for that application. In this example, a bottom-up strategy providing a detailed technical view seems more suitable than a top-down strategy.

In contrast, at the very beginning of a strategic IT modernization program, a critical task is to define how the desired target architecture should look like several years ahead. Understanding business vision and requirements is an essential task to ensure that the envisioned target architecture is aligned with the business needs. At this early stage, a high-level view on the targeted services is needed to evaluate and compare different options of the target architecture, outline and plan the modernization program from an enterprise perspective. Thus, a top-down strategy or meet-in-the-middle strategy seems adequate because these strategies provide the required input from the business for strategic alignment and for defining the target architecture of the modernization program. Figure 3 distinguishes between top-down approaches for service identification based on business capability models and top-down approaches based on business process models. The first one relates to the CAPABILITY-BASED SERVICE introduced in this paper while the latter, for example, relates to the BUSINESS-DRIVEN SERVICE pattern for service identification from Hentrich and Zdun (Hentrich & Zdun, 2006).

The BUSINESS-DRIVEN SERVICE pattern provides a top-down solution for typical challenges of alignment between business and IT in a SOA. An essential part of its top-down solution is the mapping of high-level strategic business goals to to-be business processes, for which so-called macroflow[1] models are designed, which are broken down further until, finally, activities of microflows[2] are mapped to business-driven services[3].

[1] Using the term macroflow we refer to long-running, rather business-oriented processes.

[2] We use the term microflow to refer to the short running, rather technical processes.

[3] Please refer to Hentrich and Zdun for further patterns of process-driven SOAs (Hentrich & Zdun, 2006; Hentrich & Zdun, 2012).

This pattern is very well applicable, if business process models already exist or activities for business process modeling are planned as part of an early analysis phase of a modernization program. However, in real-life cases, it occurs that business process models are not existing or incomplete and comprehensive business process modeling is neither funded nor fits to the tight timeline of the analysis phase. In contrast, the analysis phase aims at program definition, including a target service model and other aspects of a target architecture (i.e. information architecture, technology and security architecture), in a rather short period of time. In such cases it might not be possible or feasible to apply service identification as suggested by the BUSINESS-DRIVEN SERVICE pattern.

Furthermore identifying services based on business process models may be risky, if it is applied in the analysis phase for the first outline of a target architecture. Business processes, as models of how the business is executed, may change within a few years. This means that at the end of a strategic modernization program, running for several years, completely different services are needed for changed or new business processes. In this and similar contexts and problem spaces, the CAPABILITY-BASED SERVICE pattern can be applied that is described in Section 3.

3 CAPABILITY-BASED SERVICE PATTERN

3.1 Context

Moving from a heterogeneous IT landscape with a set of (monolithic) legacy applications to a more flexible SOA-based architecture requires a large-scale IT modernization program which often runs over several years. The target state of the IT architecture needs to be defined at the very beginning of such a modernization program in order to set the direction for the program, to define its business case and to set-up a roadmap for its execution. Identifying and defining the right services is a critical aspect of the definition of the target architecture of a SOA transformation program.

3.2 Problem

How can services be identified for defining a SOA transformation program under the following conditions:

- upfront service identification is required for the program definition and for setting-up a high-level roadmap,
- service identification must be conducted within a short timeframe (in order to keep the momentum for initiating the SOA transformation program),
- business requirements several years ahead must be considered in order to provide the right services as a result of the SOA transformation program, and
- a comprehensive business process model is missing?

3.3 Problem Details

- Considering a heterogeneous as-is application architecture the service identification for a large SOA modernization program bears a high complexity that must be mastered. Furthermore key business-related drivers must be considered, for example, referring to business process optimization, business agility and reduced maintenance costs.
- There is a time lag between identification of service candidates for defining the program and the implementation of services during the program execution. On the one hand, relatively stable service candidates need to be identified so that only a limited amount of changes is required, and on the other hand, the service model should be flexible enough to deal with required changes.
- The business processes required in the future (or maybe even the business processes used today) are not known or at least many details are not known or they may evolve over time. There is no time or budget to fully model the business processes and use service identification strategies based on business processes like the BUSINESS-DRIVEN SERVICE pattern (Hentrich & Zdun, 2006).
- There is a risk of misalignment between business and IT during program execution. The identified services should be as stable as possible even considering this risk, meaning that service identification based on existing business process models, as in the BUSINESS-DRIVEN SERVICE pattern from Hentrich & Zdun (2006), might be infeasible.

- Overall a stable definition of the target architecture with a high-level definition of core business-related services is required to define and plan the modernization program.

3.4 Solution

Identify service candidates by applying a top-down approach based on a business capability model. A business capability defines which abilities are required for the execution of one or more specific business functions in the next years. Its definition is less detailed and more stable than that of a business process, leaving out details about how the capabilities are provided.

Each business capability is mapped to a service type, like business process service, composite service or atomic service (defined in detail below). After that, concrete service candidates are determined and mapped to each business capability. A de-composition analysis helps to identify more than one service candidate per business capability, if required. The focus lies on identifying high- and medium-level services for defining the SOA program. Identifying atomic services and detailed business process modeling are regarded as follow-up activities.

3.5 Solution Details

A CAPABILITY-BASED SERVICE is based on current and future business needs which are gathered in a business capability model. A business capability is a fundamental abstraction to describe the business requirements. Homann defines it as follows (Homann, 2006): “A *business capability* is a particular ability or capacity that a business may possess or exchange to achieve a specific purpose or outcome. A capability describes what the business does (outcomes and service levels) that creates value for customers; for example, pay employee or ship product. A business capability abstracts and encapsulates the people, process/procedures, technology, and information into the essential building blocks needed to facilitate performance improvement and redesign analysis.”

Figure 4 shows a top-down approach for determining a service model based on business capabilities. The business strategy is taken as input for an analysis of required business capabilities. As a result a business capability model is defined, which may consist of different hierarchies, starting with top-level business functions at the top and fine-grained business capabilities at the lowest level.

Taking into account that the granularity of the lowest business capabilities may not be completely harmonized, service types of different granularities are considered and mapped to each fine-grained business capability, for example, distinguishing between a high-level, a medium-level and an atomic service type.

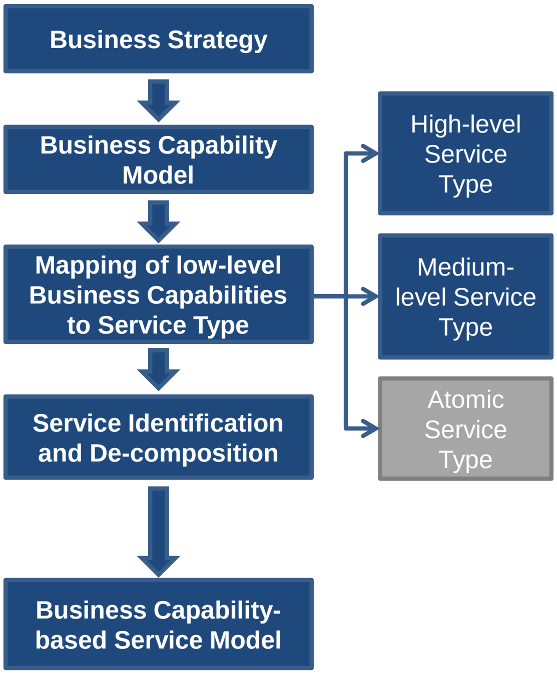
Business Strategy
Business Capability Model
Mapping of low-level Business Capabilities to Service Type
Service Identification and De-composition
Business Capability-based Service Model
High-level Service Type
Medium-level Service Type
Atomic Service Type

Figure 4: Top-down Approach for CAPABILITY-BASED SERVICE

In the next step for each low-grained business capability a concrete service is identified, considering the determined service types. For fulfilling some business capabilities more than one service may be required. Hence, a de-composition into several services may be required. However, not all business capabilities require new services. Some existing services may provide the required functional abilities for several business capabilities. Before a new service candidate is defined for a business capability, it must be checked if one of the existing service candidates may also provide the required ability (or if it can be extended to provide similar functional abilities required for more than one business capability).

Although Figure 4 shows a "waterfall"-like sequence of steps, the top-down approach is usually conducted in an iterative way[4]. It may begin with defining the business capability model and the service model for the first domain (or sub-domain). After that, when the analysis is repeated for the next domain, the existing services model may also be re-considered and changed, for example, refining a service, which is mapped to both domains.

The focus of the CAPABILITY-BASED SERVICE pattern lies on top-and medium-level services, identifying business process services and composite services for defining the high-level outline of the target architecture and for determining a roadmap for the whole modernization program. Determining required atomic services is considered as follow-up activity conducted during the initiation or execution of the program. Furthermore, detailed business process modeling of underlying flows and activities is also required during the program execution (as part of the design phase for each business process service[5]). Apart from atomic services some additional medium-level services may also be identified during this follow-up activity.

[4] Please note, that the execution of development projects for services and migration are not part of the upfront program definition phase. They follow during the execution phase. Although they need to be linked to the (upfront) service model, the development process can be determined based on skills and standards of the organization and its development team. For example, agile, iterative or other development approaches can be applied.

[5] Additionally, interfaces between business processes need to be analyzed and described.

3.6 Example

The example refers to CAPABILITY-BASED SERVICE in the logistics domain. As part of its express business the logistics provider offers time-definite delivery in a short period of time (i.e., within 24 hours around the world). The shipment process is the core operational process. Initiated by a customer's order, it begins with planning the shipment route, pick-up the shipment from the sender, and moving it across several locations with the help of various vehicles (i.e. planes, trucks, cars). Finally, the shipment is delivered to its final destination. Boxing and unboxing shipments into/from appropriate containers – also known as *containerisation* – is required before and after each movement from one location to another.

In this example the following service types are distinguished:

- *Business Process Service (BPS)*: A BPS is a stateful service supporting complex (non-atomic) business processes, which consist of several steps and include human interaction. It is composed out of other business process services, composite, or atomic services.
- *Composite Business Service (CBS)*: A CBS is an automated service providing business logic without human interaction. It is composed out of atomic business or data services. An example is a credit limit validation service based on complex business logic that receives user input and gets additional data and business rules from atomic data and business services.
- *Composite Data Service (CDS)*: A CDS is an automated service providing data without human interaction, which is composed out of atomic business or data services. An example is a data aggregation service, which gets data from various atomic data services and combines or aggregates the data.
- *Atomic Business Service (ABS)*: An ABS is an automated service without human interaction, which provides atomic business logic functionality. An example is a service for generating a business-related identifier (e.g., a tracking ID).
- *Atomic Data Service (ADS)*: An ADS is an automated service without human interaction, which provides atomic data retrieval or data manipulation functionality (e.g., atomic "CRUD" – create, read, update, delete services). An example is a service reading or changing address data.

Figure 5 shows an extract of a two-level business capability model. On the top there is the *Containerisation* high-level business capability. Apart from that, other business capabilities may be identified as high-level business capabilities. The second level illustrates some examples of fine-grained business capabilities for *Containerisation*. Some of them may also be relevant for other high-level business capabilities.

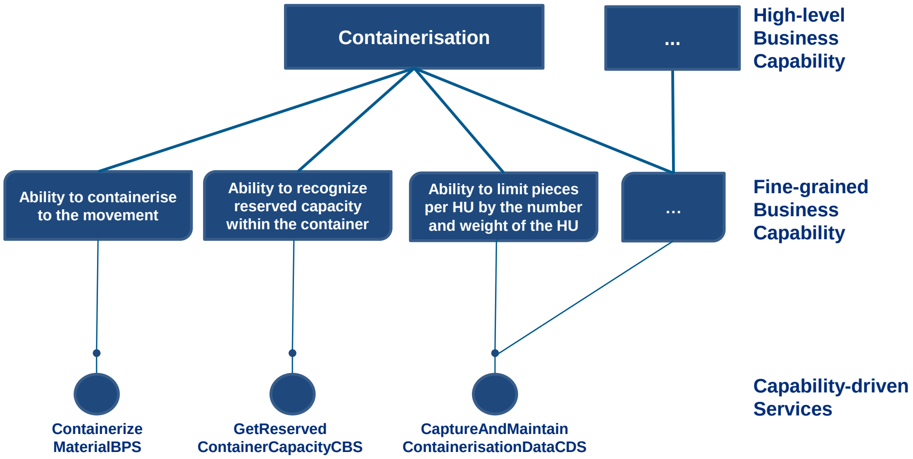
Containerisation
...
High-level Business Capability
Ability to containerise to the movement
Ability to recognize reserved capacity within the container
Ability to limit pieces per HU by the number and weight of the HU
...
Fine-grained Business Capability
Capability-driven Services
Containerize MaterialBPS
GetReserved ContainerCapacityCBS
CaptureAndMaintain ContainerisationDataCDS

Figure 5: Example for Capability-based Services

Applying the top-down approach for service identification described above, there are services mapped to each of the fine-grained business capabilities. Figure 5 indicates that different types of services have been identified for the fine-grained business capabilities. For example, a *Business Process Service* is mapped to the “Ability to containerise to the movement”, because it relates to a business process, and requires several more fine-grained capabilities. The other business capabilities “Ability to recognize reserved capacity within the container” and “Ability to limit pieces per HU [handling unit] by the number and weight of HU”, which are depicted in Figure 5, refer to more fine-grained service types. However, they still entail a complexity, which cannot be fulfilled by an atomic service. So they require a composite service type and are mapped to a *Composite Business Service* (GetReservedContainerCapacityCBS) and to *Composite Data Service* (CaptureAndMaintainContainerisationDataCDS). Furthermore, it is indicated that the *Composite Data Service* also provides functionality that is required for other fine-grained business capabilities.

3.7 Known Uses

- Infosys applied the capability-based service pattern in a real-world project regarding a strategic IT modernization program to identify a high-level service model in a preliminary analysis project for a large logistics provider.
- Oracle and Boehringer Ingelheim showed how a business-capability-driven approach has been applied in a successful project for a process-driven SOA (Deb & Utschig, 2012).
- Taking into account his experience from various industry projects, Keller describes a capability-based management pattern for Business-/IT-Alignment in his book about IT-related enterprise architecture (Keller, 2012).
- The approach proposed by Homann suggests business capabilities as a foundation to ensure that the implementation of a SOA is stable during evolution of the system and the chosen implementation architecture relates to the actual or desired state of business (Homann, 2006).
- The SoaML approach for service identification proposed by Amsden has a business process focus, but the service identification is based on capabilities for identifying candidate services (Amsden, 2010).
- Kohlborn et al. suggest a scientific approach for the identification and analysis of business and software services in which business capabilities and capability decompositions are key elements of service identification (Kohlborn et al., 2009).

3.8 Consequences

The top-down solution of the CAPABILITY-BASED SERVICE pattern is business-driven and facilitates the alignment between business and IT. However, focusing on high-level business capabilities (i.e., answering the question: what abilities does the business need?) rather than detailed business processes (details of how is the business executed) helps to speed up the process of service identification. The required business capability model can be defined in a short timeframe, compared to a comprehensive business process model. This can be a critical aspect for the definition of the IT modernization program, because in real-life the early analysis phase is usually rather short for keeping the momentum from the definition to the actual execution of the program. If an enterprise lacks a comprehensive business process model and does not have the time for a comprehensive business process modeling upfront, the capability-based services pattern may be better applicable for service identification than patterns which require a detailed business process model.

Furthermore business capabilities tend to be more stable over time than business processes. Both, rapid definition of the business capability model and its robustness over time, are critical concerns for identifying (the right) services at the beginning of a long-running SOA program. Hence, the CAPABILITY-BASED SERVICE pattern is especially beneficial for setting-up a long-running SOA modernization program.

The CAPABILITY-BASED SERVICE pattern is focused on identifying high-level services for defining the high-level target architecture, for determining required projects, and for setting up a program roadmap. Applying this pattern comes along with some limitations. First, skilled people experienced in business analysis, SOA, and enterprise architecture definition are required to conduct the top-down approach for service identification. Second, follow-up activities are required during the program execution, for example, detailed business process models must be defined to identify additional services, which are needed on middle- and lower-level. Third, interfaces between business processes need to be analyzed in detail and for each service a detailed requirements analysis and design is required. These follow-up activities also come along with detailing the planning of related projects. Furthermore feedback loops are required in order to incorporate the additional knowledge gained during the program execution. The service model and, eventually, the business capability model may be refined and updated. Thus, sophisticated (enterprise architecture) tool-support is required for administrating and updating the models and their dependencies.

4 RESULTS FROM FOCUS GROUP ON SERVICE IDENTIFICATION

At the EuroPLoP 2013 conference a focus group with ten experts (SOA architecture & software development) was conducted for discussing the topic of service identification, with special focus on the BUSINESS-DRIVEN SERVICE and the CAPABILITY-BASED SERVICE patterns. The goal was to confirm our views on and experiences with the patterns and identify and explore the key characteristics, purposes, goals, benefits, and limitations of service identification approaches. In the focus group we first explained the concepts presented in this paper in detail. Then we conducted group discussions in two groups, each moderated by one of the authors. Next, the discussion results were recorded and analyzed by the authors.

Key findings of our focus group are:

- There are different purposes for service identification (i.e. transforming an IT application landscape vs. implementing one service-based system). The purpose of the service identification determines how many details of the identified services are required and which degree of uncertainty is acceptable.
- For a first high-level estimation, for example, a rough sketch with a lot of uncertainty is acceptable, while a detailed service model with a small degree of expected variance and risk (of changes or unknown services) is required for planning, architectural design and implementation.
- Experience and knowledge as well as existing models (business process model or business capability model) play a significant role for choosing a pattern, like the BUSINESS-DRIVEN SERVICE pattern based on a business process model or the CAPABILITY-BASED SERVICE pattern based on capability-based model.
- It is difficult to estimate the stability and the quality of business alignment of the different approaches for the long duration of an IT modernization program.
- Apart from both the BUSINESS-DRIVEN SERVICE and the CAPABILITY-BASED SERVICE patterns, there exist various other approaches, i.e., based on reference models, participatory design, prototyping, and organizational analysis.
- It is difficult to estimate the overall effort of different approaches. For example, a rough sketch of services upfront requires more effort for detailed modeling and required changes at a later point of time.

5 CONCLUSION

In this article we have discussed the process of service identification in the context of a preliminary analysis phase of an IT modernization program, which is focused on a transformation to a SOA. Considering this context, we introduced the CAPABILITY-BASED SERVICE pattern to provide a solution for typical challenges of identifying the right services, which are aligned with current and future business needs several years ahead.

The BUSINESS-DRIVEN SERVICE is an alternative pattern for service identification. It is commonly used in the context of process-driven SOAs. Using the CAPABILITY-BASED SERVICE pattern does not reject the concept of a *process-driven SOA* and related patterns (see Hentrich & Zdun, 2012), but simply there are different challenges and solutions to conduct service identification. Which way is the best depends on the specific pre-requisites and challenges of an organization. For example, if the organization works with a comprehensive, stable business process model which has been established in the organization, the BUSINESS-DRIVEN SERVICE pattern is likely a good choice to identify the high-level services at the beginning of a modernization program.

Figure 6 illustrates, first, the detailed and comprehensive modeling of business processes and services upfront and further refinement during the program execution. Second, it indicates that a big part of the required business process models already exist from the beginning (the grey boxes). So, there is only additional effort required for detailing existing business process models and deriving service models.

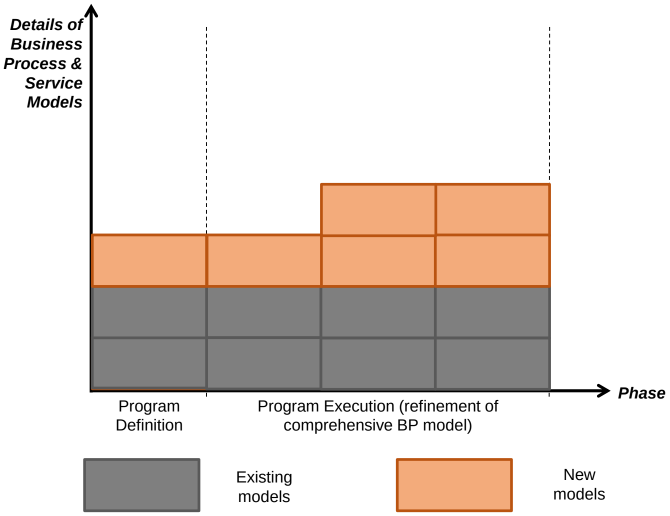

Figure 6: Upfront Service and Business Process Modeling of BUSINESS-DRIVEN SERVICE pattern

The CAPABILITY-BASED SERVICE pattern provides a solution to identify service candidates particularly during a preliminary analysis phase of a modernization program. It is particularly preferable over a process-driven approach, if a comprehensive business capability model is already available and applied to manage the application portfolio, like in the case study of a leading logistics provider, described above. Nevertheless it requires detailed modeling of services and business processes during the execution of the program, which is indicated in the following illustrative figure.

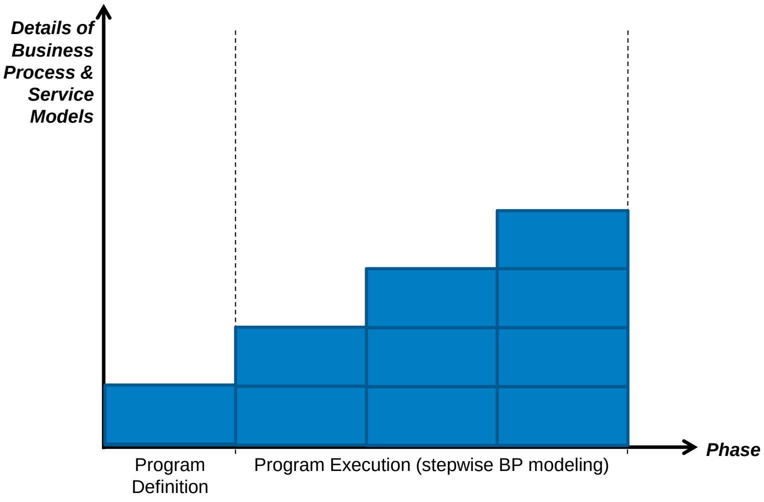

Figure 7: Step-wise Service and Business Process Modeling for a CAPABILITY-BASED SERVICE

A key benefit of the capability-based service pattern is that it facilitates a durable alignment between business and IT. A business capability model, which is a key part of its solution, tends to be more stable over years than, for example, the control flow and other details of a business process. For this reason, eventually, capability models are better suited as a basis for a high-level program definition than business process models.

Taking into account the context of SOA modernization programs as well as other purposes for service identification further research is required to get more insights about the overall effectiveness and efficiency of different approaches (business capability-driven, business process-driven and other approaches).

Defining a SOA modernization program does not only require the definition of to-be services, but also determining which existing applications can be re-used in a SOA and which new components need to be developed. Thus, further research is also required to clarify how capability-based services can be leveraged for defining the target architecture, assessing legacy applications as well as determining a program roadmap (i.e., determining and planning required projects for service-enabling applications, for developing new services, and for migration).

ACKNOWLEDGEMENTS

We would like to thank our shepherd Ernst Oberortner for his strong support and constructive feedback for improving the paper.

REFERENCES

Amsden, J. (2010). Modeling with SoaML, the Service-Oriented Architecture Modeling Language: Part 1. Service identification. IBM Developer Works. http://www.ibm.com/developerworks/rational/library/09/modelingwithsoaml-1/.

Barry, D.K. (2003). Web Services and Service-oriented Architectures. Morgan Kaufmann Publishers. San Francisco, CA.

Börner, R. & Goeken, M. & Rabhi, F. (2012). SOA Development and Service Identification A Case Study on Method Use, Context and Success Factors. In Frankfurt School – Working Paper Series No. 189.

Brahmanandam, M. (2011). Legacy Application Modernization. Database Trends and Applications. http://www.dbta.com/.

Deb, M. & Utschig, C. (2012). SOA & BPM Success Using Business Capabilities – Concepts and Examples. 5th International SOA, Cloud, Service Technology Symposium. http://www.servicetechsymposium.com. London.

Fareghzadeh, N. (2008). Service Identification Approach to SOA Development. World Academy of Science, Engineering and Technology, Vol. 2 No. 9, pp. 243-252.

Hentrich, C. & Zdun, U. (2006). Patterns for process-oriented integration in service-oriented architectures. In Proceedings of 11th European Conference on Pattern Languages of Programs (EuroPlop 06). Irsee, Germany, pp. 141–189.

Hentrich, C. & Zdun, U. (2012). Process-Driven SOA: Patterns for Aligning Business and IT. CRC Press.

Homann, U. (2006). A Business-Oriented Foundation for Service Orientation. Microsoft MSDN. http://msdn.microsoft.com/en-us/library/aa479368.aspx.

Keller, W. (2012). IT-Unternehmensarchitektur – Von der Geschäftsstrategie zur optimalen Umsetzung. dpunkt.verlag. Heidelberg, Germany.

Khadka, R. et al. (2012). Legacy to SOA Evolution: A Systematic Literature Review. In A. Ionita, M. Ltoiu, G. Lewis. Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments. IGI Global, pp. 40-70.

Kohlborn, T. et al. (2009). Identification and Analysis of Business and Software Services - A Consolidated Approach. IEEE Trans. Serv. Comput. 2, 1, pp. 50-64.

Maréchaux, J.L. (2008). Modernize legacy systems using an SOA approach. IBM Developer Works. http://www.ibm.com/developerworks/webservices/library/ws-soa-legacymod/index.html.

Masak, D. (2007). SOA? – Serviceorientierung in Business und Software, Springer-Verlag. Berlin.

Rosen, M. (2010). Business Processes Start with Capabilities. BPTrends. http://www.bptrends.com/.