



University of Augsburg  
Prof. Dr. Hans Ulrich Buhl  
Research Center  
Finance & Information Management  
Department of Information Systems  
Engineering & Financial Management

**UNIA**  
Universität  
Augsburg  
University

Discussion Paper WI-307

## Towards a Financially Optimal Design of IT Services

by

Kathrin Braunwarth, Bettina Friedl

in: Proceedings of the 31st International Conference on Information Systems  
(ICIS), Saint Louis, Missouri, December 2010



The international Journal of  
WIRTSCHAFTSINFORMATIK

# TOWARDS A FINANCIALLY OPTIMAL DESIGN OF IT SERVICES

*Completed Research Paper*

**Kathrin S. Braunwarth**

Allianz Group  
Koeniginstrasse 28  
80802 Munich, Germany  
kathrin.braunwarth@allianz.de

**Bettina Friedl**

FIM Research Centre  
University of Augsburg  
86135 Augsburg, Germany  
bettina.friedl@wiwi.uni-augsburg.de

## **Abstract**

*The current financial crisis forces companies to allocate IT budgets more effectively and thus increases the demand for suitable methods to evaluate the financial impact of IT investments. This especially applies to service-orientation, a design paradigm which facilitates the standardisation and flexibilisation of business processes and IT applications, topics that currently are very much in vogue in science and practice. This paper focuses on the realisation of a new functionality by IT services and presents a methodology to determine their financially optimal functional scope on the continuum between realising just one IT service providing the whole functionality and realising many IT services each providing only a small share of functionality. This approach allows for a multi-period financial valuation of an uncertain demand for the new functionality, as well as of an uncertain company-wide reuse of the corresponding IT services. Finally, the methodology is evaluated by an example from a financial services provider.*

**Keywords:** Service oriented architecture (SOA), IT services, Web services, IT investment evaluation

## **Introduction**

In recent years, the top business priorities for CIOs in many industries have been geared towards the improvement of business processes (Gartner 2010). This area has a distinct growing need for IT aligned and automated business processes, which is caused by a lack of quality in many processes, relatively high coordination efforts and unnecessarily high process costs (Loehe and Legner 2010). In this context, service-oriented architectures (SOA), where loosely coupled IT services encapsulate some reusable business aligned functionality to support business processes, have become well-established. Compared to large monolithic IT applications SOA provides, among other things, a higher level of agility for the company-wide IT landscape (Choi et al. 2010) as well as convenient and cost saving integration of new functionality (Hagel 2002). Therefore, IT executives in many companies focus on developing their IT architecture towards SOA: in 2008 SOA was already utilised by 20% of 260 companies that participated in a worldwide survey conducted by Accenture and another 38% were planning SOA implementations (accenture 2008). However, this boom in implementing SOA was stopped in its track by the worldwide financial crisis where many companies had to radically cut their IT budgets. In 2009 alone, 82% of CIOs in the private sector had to undertake cost reduction measures of 15% on average (Capgemini 2009b). As a result, many companies show a serious backlog of IT investments and are thus faced with the challenge to allocate still tight IT budgets effectively. Consequently, the development and application of adequate valuation methods for the financial impact of IT investments have attained extreme importance in order to achieve an effective use of information (Watson et al. 1996), which in turn is essential for the return of revenue that plunged dramatically during the crisis. This, among other things, especially applies to service-oriented design which must also be driven by its financial impact.

When deciding on how many and which IT services should be implemented to realise an additionally required range of functionality, companies have to trade-off financial advantages of IT services with varying functional scope: on the one hand, the reuse of IT services, a main SOA advantage, considerably increases with a decrease in functional scope as IT services with a small functional scope are more likely to fit into other business processes (Aier 2006). This reuse of IT services cuts down cost intensive and time-consuming future realisations of additional functionality. It also contributes to a higher degree of company-wide process standardisation which will have a positive impact on the costs for IT support in the long run. On the other hand, costs of realising and operating a new functionality rises considerably with the number of additionally launched IT services as this is accompanied by an increase in interfaces and a higher complexity of the company's IT service portfolio (Schelp and Winter 2007). As a result, companies today tend to implement relatively large IT services in order to minimise specification, implementation, and operational costs for the realisation of an additional required functionality. In doing so, they completely neglect the financial benefits of future reuse. One explanation for this may be the lack of suitable decision models.

The objective of this paper is to contribute to the closure of this research gap. Therefore we adopt the perspective of a single business unit (BU) and develop an optimisation approach to determine the financially optimal functional scope of IT services. During this process, we analyse and integrate the uncertain cash flows resulting from the specification, implementation, and operation of IT services by the BU itself, as well as the uncertain cash inflows resulting from a company-internal offset from the reuse of the IT services by other BUs.

The paper is structured as follows: in the next section we substantiate the research gap by discussing several factors of influence for the financially optimal functional scope of IT services and examining whether existing literature addresses these factors. Based on this the model is presented in the section thereafter. The practical use of this model is illustrated by an example taken from the financial services sector in the subsequent section. The paper concludes with a short summary, a discussion of the limitations, and perspectives for further research in the last section.

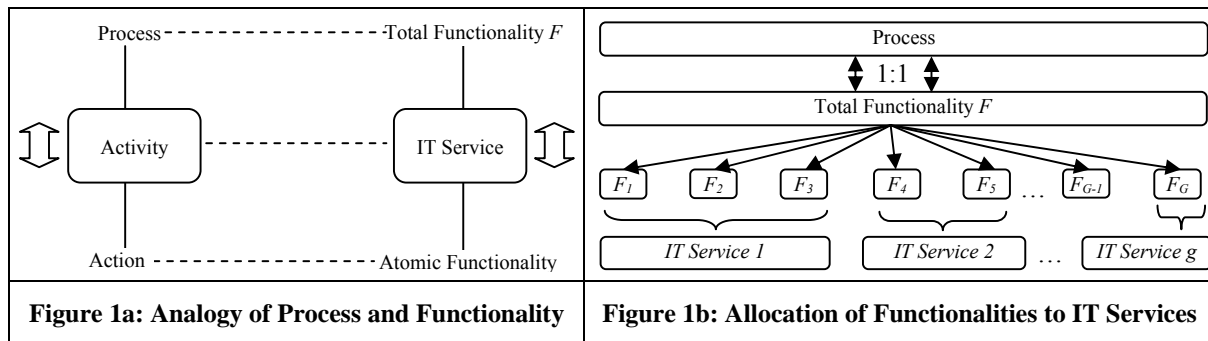
## **Substantiation of the Research Gap**

In recent years, service-orientation has become the major paradigm for the design of both enterprise architectures and IT applications (we refer to Erl 2008; Krafzig et al. 2007; Papazoglou and Georgakopoulos 2003 for an in-depth analysis). Current information systems research and practice therefore provide a variety of definitions and different interpretations of the service concept. However, most authors either adopt a business perspective on services (as in Klose et al. 2007; Schelp and Winter 2007) where a service is "a well-defined, encapsulated, reusable, business-aligned capability" (Arsanjani et al. 2008), or a technological one (as in Albani et al. 2003; Zhang et al. 2005) where a service is "a discoverable, invocable software resource that has a service description and interface [such as a Web service]" (Arsanjani et al. 2008). In view of this pluralism, we are required to specify the service perception this

research is based on. We aim to establish a connection between business and technological aspects by utilising the following definition of an IT service:

*In this article we think of an IT service as a software-technically realised artefact such as a Web service that encapsulates a certain range of business functions – which will now be referred to as functionality – and is used as a module in one or multiple business processes for the (partly) automated execution of selected activities.*

The design of such IT services requires “decisions [...] about how to divide a problem [e.g. a newly required functionality] into logical [IT] services” (Millard et al. 2009). According to the established definition of IT services, we confine our considerations to functionalities that require some kind of software support. When considering their business processes, companies have to decide which activities should be rendered by IT services. In order to avoid any confusion about the nature of an activity, for this paper we draw on the widespread standard UML 2.0, where an activity is defined as a major task that can be decomposed in a set of indivisible atomic actions (OMG 2005; OMG 2006). Analogously, a total functionality  $F$  also comprises multiple, indivisible *atomic functionalities*  $F_1, \dots, F_G$ . When designing IT services, companies thus have to decide, how to bundle these atomic functionalities in IT services or, in other words, where to place an IT service on the continuum between “the IT service provides just one atomic functionality  $F_i$ ” versus “the IT service provides the total functionality  $F$ ”. This analogy between business processes and the related total functionality  $F$ , which requires some kind of software support, is illustrated by Figure 1a and Figure 1b:



In order to reach a close alignment of technical possibilities and economic interests, a BU that requires an additional or renewed functionality  $F$  for one of its business processes has to decide on how to bundle the atomic functionalities in IT services, among other things, from a financial point of view. As illustrated by Figure 1a, the implementation of a single IT service providing the total functionality  $F$  and the implementation of a distinct IT service per atomic functionality  $F_i$  are the extreme solutions of the raised question. In practice, every intermediate scope where several atomic functionalities are bundled into one IT service can be chosen. In this paper, we will compare IT services by means of the amount of provided atomic functionalities, which is sometimes also called the *granularity* of IT services (Aier 2006). We distinguish between small IT services that provide one or only a few atomic functionalities  $F_i$  and large IT services that provide many atomic functionalities (see Aier (2006) or Papazoglou and van den Heuvel (2006) for a similar attribution of IT services). However, we have to admit that we do not distinguish exactly between the two attributions. Instead, they seamlessly blend into each other. As a consequence, we mostly use the term *relatively* to indicate this fuzziness. As this paper starts from a predefined demand for an additional or renewable total functionality  $F$ , the implementation of relatively small IT services with each providing only a few atomic functionalities stringently results in a high number of IT services for realising the total functionality  $F$ , whereas the implementation of relatively large IT services with each providing many atomic functionalities stringently results in a small number of IT services.

### **Key Drivers for the Financially Optimal Design of IT Services**

From a financial point of view there are several factors, henceforth referred to as *key drivers*, which essentially influence the optimal functional scope of IT services (see Aier 2006; Boerner and Goeken 2009; Erradi et al. 2006; Millard et al. 2009). In this subsection, we will briefly discuss these key drivers and evaluate whether a key driver favours the realisation of relatively small and many or relatively large and few IT services.

**Specification, implementation, and operational costs:** for an additional or renewable total functionality  $F$  these costs are made up of two components: the specification, implementation, and operational costs for the new functionality itself (these costs can be thought of as those occurring if the company implements just one IT service for the whole total functionality  $F$ ) and the specification, implementation, and operational costs for the required interfaces. While the first component is independent from the number of additionally launched IT services, the second is highly dependent on this number, as specification, implementation, and operational costs considerably increase with the number of interfaces (Boerner and Goeken 2009). In addition, the complexity of a company's overall IT service portfolio also increases with the number of additionally launched IT services. This leads to increasing efforts for the composition and monitoring of IT services (Schelp and Winter 2007) and to high network traffic (Papazoglou and van den Heuvel 2006). Depending on a company's IT infrastructure capacity, a rising demand for network and server resources can result in poor performance of IT applications which directly impacts the level of user satisfaction. With regard to customer-oriented IT applications such as customer self services or the IT support of the consultation and sales process of a financial services provider (FSP), decreasing user satisfaction may be reflected in additional costs for complaint solutions and shrinking revenues from prospective customer contacts (Hirschman 1990; McCollough et al. 2000). In order to avoid such negative effects and increase network efficiency, companies sometimes have to undertake extensive IT infrastructure investments. Hence, from a financial point of view, few relatively large IT services are more beneficial with regard to specification, implementation, and operational costs of a new functionality and of the required interfaces.

**Reuse:** the possibility of subsequently reusing IT services once implemented is the main advantage of service-oriented design. This reuse of already existing IT services cuts down cost intensive and time-consuming future implementations of additional functionality and holds great potential for enormous cost savings. Besides, human and technical resources, which are scarce due to tight IT budgets, can be used elsewhere. This allows companies to conduct more IT projects per unit of time and resource and consequently to increase the IT support and standardisation of business processes since more business processes will use the same implementation of a specific functionality. As a result, cash inflows from these business processes tend to increase as errors due to manual execution can diminish significantly. In this context, the reuse frequency of an IT service tends to increase in correlation with a decrease in functional scope, as relatively small IT services are more likely to fit within other business processes (Aier 2006). However, implementing too small IT services may affect reuse adversely due to potentially tight coupling of data models or high dependency on a particular order of interaction (Millard et al. 2009). Furthermore, identifying appropriate IT services becomes more complex with an increasing complexity of the company's IT service portfolio (Aier 2006). In summary, from a financial standpoint many relatively small IT services have a beneficial tendency in connection with reuse. However, the hypothesis "the smaller the IT services, the higher their reuse and thus the resulting cost savings potential" does not apply universally.

**Modifications of existing IT services:** functional change requests for the total functionality  $F$  occur frequently in business practice due to dynamically changing requirements. However, these change requests usually affect just one or a few atomic functionalities. As a result, the costs of modifying the supporting IT services are usually lower for relatively small IT services as just one or a few IT services with a relatively clear code have to be modified and not a large and complex IT service where it is harder to find the affected atomic functionalities. In some cases it is even sufficient to simply re-composite already existing IT services to realise a change request (Schelp and Winter 2007). If companies moreover follow fundamental SOA design principles such as high internal cohesion and loosely coupling of IT services (Erl 2008) and if they use appropriate interface standards (such as the Web Services Description Language (WSDL)), relatively small IT services have a considerably lower interface complexity as only little information has to be interchanged (Schelp and Winter 2007). For that reason, interface modifications are also feasible with justifiable expenditure. Similar to reuse, there are, however, also drawbacks from too small IT services: extensive reuse of IT services, which is facilitated by relatively small IT services, may cause considerable dependencies between business processes. In this case, a rather simple modification of a single IT service may have a high impact on many other business processes and therefore induce additional cash outflows to manage unintended side effects (Schelp and Winter 2007). In summary, from a financial standpoint many relatively small IT services have a beneficial tendency in connection with modification costs. However too small IT services may cause higher modification costs due to unplanned dependencies.

**Error handling in the production system:** companies have to consider two conflicting effects when dealing with error handling in the production system. On the one hand, it is much easier to detect the source of an error in the production system when a total functionality  $F$  is bundled in only a few IT services as in this case, companies have a clear idea of the connection between business processes and the supporting IT services (Papazoglou and van den

Heuvel 2006). On the other hand, modifying existing IT services and fixing errors is a lot easier when dealing with relatively small IT services as each has a rather clear code (see previous paragraph). So, from a financial standpoint, there is no clear preference to the functional scope of IT services when addressing error handling in the production system as both relatively small and relatively large IT services present advantages that can be leveraged.

To summarise the financial aspect, there are various key drivers that essentially influence the optimal functional scope of IT services. If companies neglect the financial impact of reuse, they tend to realise few relatively large IT services as this minimises specification, implementation, and operational costs. However, this involves the risk of multiple realisations of the very same functionality and often leads to huge follow-up costs, which could have been avoidable. Consequently, companies should integrate the financial impact of future reuse in today's design of IT services. However, in this case, there is no clear indication whether implementing many relatively small or few relatively large IT services is the dominant strategy. Instead, both options bear specific financial advantages and disadvantages so that companies are faced with a trade-off situation. This should be considered during the determination of the functional scope of IT services in order to achieve a good business/IT alignment (Chan and Reich 2007). In the next subsection, we examine existing approaches to the identification and design of IT services and analyse to what extent they consider the financial advantages and disadvantages of different functional scopes of IT services.

### **Related Work**

When applying service-oriented design, companies are faced with the challenge of “identifying the right [IT] services, organizing them in a manageable hierarchy of composite [IT] services [...and], choreographing them together for supporting a business process” (Papazoglou and van den Heuvel 2006). To tackle this challenge as good as possible, companies can draw on a plethora of design principles for IT services (see Erl 2008), which are often grouped into the four categories *interoperability*, *interface orientation*, *autonomy & modularity* and *business orientation* (see Klose et al. 2007). These design principles can serve as basic guideline for the identification and design of IT services. Looking at them closely, *high cohesion* within an IT service, *weak logical coupling* between IT services, and the design of IT services for *reusability* are the most important design principles in the determination of the functional scope of IT services (see Erl 2008; Papazoglou and van den Heuvel 2006).

Based on these fundamental design principles there is a lot of current research within computer science and information systems which addresses the identification and design of IT services and provides frameworks, step-by-step approaches, and procedure models for the design of IT services. In this context, we can distinguish between approaches dealing with the whole service-oriented design and development lifecycle (see Arsanjani et al. 2008; Papazoglou and van den Heuvel 2006; Quartel et al. 2004) and research confining itself solely to the design of IT services (see Aier 2006; Erradi et al. 2006; Winkler 2007). Furthermore, we can identify research addressing the design of Web services as one specific technology for the realisation of IT services (see Albani et al. 2003; Millard et al. 2009; Zhang et al. 2005) and approaches mainly focusing on business aspects of IT services without implying a certain technology (see e.g. Klose et al. 2007; Kohlmann and Alt 2007; Schelp and Winter 2007).

However, determining the functional scope of IT services is an issue that is usually not addressed explicitly, but solved implicitly by the design of the single approaches. There are for example various bottom-up approaches which focus on legacy systems and derive IT services by wrapping existing IT applications (see Zhang et al. 2005). In this context, the functional scope of IT services is determined in a natural way by the functional scope of the already existing IT applications. So-called top-down approaches (see Patig and Wesenberg 2009) in contrast draw on a range of tools such as business process modelling, process decomposition, domain decomposition, asset analysis and portfolio management to systematically break down a certain piece of business information such as business goals or processes to gain potential IT service candidates (see Kohlmann and Alt 2007). In this context, the functional scope of IT services normally results from the chosen decomposition level. We can observe approaches that provide a rather coarse decomposition and consequently generate relatively large IT service (see Arsanjani et al. 2008; Boehmann and Kremer 2005), and approaches that provide a very fine decomposition and consequently generate relatively small IT services (see Winkler 2007). In general, there are only a few authors that explicitly address the determination of the functional scope of IT services (see Aier 2006; Boerner and Goeken 2009; Erradi et al. 2006; Millard et al. 2009; Papazoglou und van den Heuvel 2006). Based on a rough classification of IT services according to their granularity, these existing papers normally discuss the advantages and disadvantages of relatively large or small IT services qualitatively. Although, most authors agree on the thesis that relatively small IT services are accompanied by a higher reusability, they often attach more weight to the drawbacks of a high number of

IT services and recommend a rather coarse-granular design of IT services with each IT service providing a relatively large amount of functionality. However, they do not formalise and offset the financial consequences of the varying reusability of different functional scopes and the financial drawbacks of a high number of IT services.

The first beginnings of the valuation of costs and benefits of the varying reusability of software artefacts with different functional scopes can be found in the area of component identification. Wang et al. (2006a, 2006b) for example integrate reuse costs and reuse efficiency in their stability based component identification method. They propose a top-down domain analysis and create a tree structure that provides a step-by-step decomposition of functionality. Although they formalise the financial impact of different decompositions, they, however, do not conduct a financial optimisation of the functional scope of components.

To sum up, we achieve the following findings: although there is lot of research dealing with the identification and design of IT service, only a small part of this research explicitly addresses the problem of determining the functional scope of IT services. And when doing so, authors usually balance advantages and disadvantages of different functional scopes of IT services qualitatively and deduce on this basis general recommendations for the optimal functional scope of IT services, whereas a quantitative formalisation and trade-off has been missing until now. Furthermore, reuse is most often addressed as an abstract design principle, but often fails to occur in business practice (Heinrich et al. 2009). One explanation for this may be the lack of methods for the evaluation of IT service candidates with regard to their reuse potential. The quantitative analysis of the financial consequences of different possibilities for the functional scope of IT services, and in particular, the integration of the financial impact of future reuse would however be necessary in order to achieve an effective allocation of tight IT budgets and to reduce the current backlog of IT investments.

The objective of this paper is to contribute to closure of this research gap. However, like most of the previously mentioned approaches to the design of IT services, our model does not consider the problem in its entirety. Instead, the focus is sharpened in three ways: first of all, we take a company-internal viewpoint and restrict the analysis to IT services which are implemented and operated by a large company's own IT department. This is justifiable insofar as there are only a few large companies that can satisfy their demand for IT services by commercial supply while the vast majority of large companies have to develop its IT services proprietarily to a large extent (Cappemini 2009a). Secondly, the optimisation model is derived from the perspective of a single BU that requires an additional or renewed functionality for one of its business processes. This can be justified as due to tight IT budgets the realisation of new functionality is mostly initiated by a single BU itself instead of a central IT department. However, within a decentralised decision-making and paying structure the charging BU has per se no financial incentive to integrate future reuse in today's determination of the functional scope of IT services. While it would have to bear the extra costs of realising many IT services each with a small functional scope, other BUs would draw the benefits of cost savings due to the reuse of already existing IT services. If companies wish to support reuse within decentralised decision-making and paying structures, they have to implement internal offset mechanisms. Therefore, we thirdly assume that there are company-internal offset mechanisms for the reuse of existing IT services so that a BU can refinance part of its costs for the specification, implementation, and operation of IT services. In summary, we will address the following research question:

*Which design of IT services should a BU that requires additional or renewed IT support for one of its business processes choose if it takes into consideration both, the uncertain cash flows for the specification, implementation, and operation of the IT services by the BU itself, as well as the cash inflows from the company-internal offset derived from the reuse of these IT services?*

## **An Optimisation Model for the Financially Optimal Design of IT Services**

To clarify our model, we first will introduce the general setting and formalize the different components of the total cash flow that accompany the specification, implementation, and operation of IT services. Subsequently, we will present a two-stage approach for solving the optimisation problem.

### **General Setting**

The starting point of the model is that a single BU requires an additional or renewed total functionality  $F$  for one of its business processes. Using the analogy between business processes and functionalities established in the last section (see Figures 1a and 1b), this total functionality  $F$  is decomposable into  $G \in IN$  atomic functionalities

$F_1, \dots, F_G$  with each  $F_i$  ( $i = 1, \dots, G$ ) representing one atomic action of the business process. The BU has to decide how many and which IT services should be realised to bundle these atomic functionalities. According to the software lifecycle, the BU usually has a planning horizon of several periods  $t = 0, \dots, T$ . First of all, we state the following assumption with regard to the allocation of atomic functionalities to IT services:

**Assumption 1:** *The allocation of atomic functionalities to IT services is determined at the beginning of the planning horizon ( $t = 0$ ) and will not be modified during the rest of the planning horizon ( $t = 1, \dots, T$ ).*

As a consequence of assumption 1, a specific realisation of the total functionality  $F$  makes itself out to be a set  $\{S_1, \dots, S_g\}$  of IT services ( $\mathbf{g} \in \{1, \dots, G\}$ ), with each  $S_j$  ( $j = 1, \dots, g$ ) encapsulating some atomic functionalities (see Figure 1b for an example). We shortly refer to a specific implementation  $\{S_1, \dots, S_g\}$  of IT services also as a *cut of IT services*. In reality, changes to IT services frequently occur. This makes assumption 1 seem rather restrictive at the first instance. However, only the allocation of atomic functionalities to IT services is fixed, whereas modifications of atomic functionalities are possible at any time, but are not further considered in this paper. If the BU even wants to change the allocation of atomic functionalities to IT services, this can also be modelled by simulating a new implementation of IT services for the realisation of the total functionality  $F$ .

SOA is a greater design paradigm for enterprise architectures as a whole (see Erl 2008), whereas this paper focuses on the implementation of a single total functionality  $F$  on BU level. That is why we start from the premise that a SOA platform and its necessary IT infrastructure (e.g. the enterprise service bus) are already implemented and available within the company. Following common service-oriented design principles, a successful IT service design requires business and technical standardisation, the use of open, widely applied industry standards and comprehensive, uniform service specifications as well as stably managed service contracts (Erl 2008; Legner and Heutschi 2007). In this context various standards and meta languages were developed in recent years. WSDL for example provides a standard for the configuration, communication and design of interfaces. The automated detection of syntactic and semantic marked IT services (e.g. using standards such as Semantic Annotations for WSDL (SAWSDL), Web Ontology Language for Web Services (OWL-S) or Lightweight Semantic Descriptions for IT services on the Web (WSMO-Lite)) can furthermore be supported via the standard UDDI (Universal Description, Discovery and Integration), which is used in many intranets. Standardised communication can be assured using SOAP (Simple Object Access Protocol) or REST (Representational State Transfer). Based on this, we make the following assumption on the preparation of the company for service-oriented design:

**Assumption 2:** *The groundwork for service-oriented design and development is already in place within the company. In particular, there is a SOA platform and the necessary supporting IT infrastructure as well as a central repository for the company's IT services. Furthermore, the company applies standards for the specification of IT services and the corresponding interfaces.*

Assumption 2 assures the existence of a company-internal integration platform for IT services (see Widjaja and Buxmann 2009) and the compatibility of different IT services using comprehensive service specifications and uniform interface descriptions. This is fundamental to guarantee a smooth interaction of different IT services and to form the basis for their reusability (Albani et al. 2003; Kohlmann and Alt 2007). Otherwise, the retrieval of IT services for reuse is more complex and the costs of embedding new IT services are needlessly high.

### **Formalisation of the Different Cash Flow Components**

As this paper aims to determine the financially optimal functional scope of IT services from a single BU's point of view, we have to analyse the financial consequences that are associated by the specification, implementation, and operation of a specific cut  $\{S_1, \dots, S_g\}$  of IT services. For their formalisation, we utilise the total cash flow as this provides a summary of all cash flows resulting from the realisation of a specific cut  $\{S_1, \dots, S_g\}$  of IT services and of their distribution over time. This procedure facilitates a multi-period planning and ensures the independency from accounting or tax interests. When deciding on the cut  $\{S_1, \dots, S_g\}$  of IT services in  $t = 0$ , it is hardly possible to estimate future cash flows for the following periods  $t = 1, \dots, T$  exactly as they usually depend on various company-internal and external factors such as the development of the market and, consequently, are often highly volatile. We have to take into account this uncertainty in order to reach a realistic estimation of the financial consequences of different cuts  $\{S_1, \dots, S_g\}$  of IT services. Considering the key drivers for a financially optimal functional scope of IT services discussed in the last section, we restrict our considerations to the specification, implementation, and operational costs and the company-internal reuse of IT services as these key drivers usually have a significantly higher impact in comparison to modification and error handling. Our model thus takes into account:



- 1) the specification and implementation costs for IT services in period  $t = 0$  as these considerably depend on the number of IT services that have to be specified and implemented,
- 2) the cash inflows and cash outflows of the “internal” use of IT services by the BU itself in the periods  $t = 1, \dots, T$ , as these also considerably depend on the number of IT services that have to be operated, and
- 3) the cash inflows of the company-internal offset of the “external” reuse of IT services by other BUs in the periods  $t = 1, \dots, T$ , as the frequency of this external reuse (FEU) considerably depends on the functional scope of the single IT services within a specific cut  $\{S_1, \dots, S_g\}$ .

### Specification and Implementation of IT Services

The specification and implementation costs for the new functionality  $F$ , which occur in period  $t = 0$ , usually contain two components: first of all specifying and implementing the total functionality  $F$  itself generates cash outflows  $O_{basic}$ , which are independent from the realised cut of IT services. These costs can be thought of as those occurring if the BU implements just one IT service for the whole total functionality  $F$ . In addition, the specification and implementation of each interface generates additional cash outflows  $O_{int}$ . As each IT service  $S_j$  requires a corresponding interface, we obtain the following cash outflows  $O_0$  in period  $t = 0$  for the specification and implementation of a specific cut  $\{S_1, \dots, S_g\}$  with  $g$  IT services:

$$O_0 = O_{basic} + g \cdot O_{int}$$

Based on its experience with software development projects, the BU usually provides a fixed budget for the specification and implementation of the required IT services. Therefore, we can start with the premise that the cash outflows from the specification and implementation of IT services are deterministic.

### Internal Use of IT Services by the BU Itself

Regarding the internal use of IT services by the BU itself, the total functionality  $F$  is potentially required for every interaction with the BU as this constitutes a potential demand for the execution of the corresponding business process. This frequency of internal use (FIU) of the total functionality  $F$  during the periods  $t = 1, \dots, T$  is uncertain and can be modelled as a discrete random variable  $\tilde{N}_t$ . It is accompanied by an uncertain cash inflow  $\tilde{I}_t^I$  which results from the fact that the BU ascribes part of the total cash inflow of the business processes to the total functionality  $F$ . This cash inflow is independent of the specific cut of IT services as with one process execution all IT services have to be executed in order to provide the total functionality  $F$ . If we denote by  $R_I$  the mean expected cash inflow per execution of the business process, which is ascribed to the total functionality  $F$  and can be determined by means of a business case, we can formalise  $\tilde{I}_t^I$  as

$$\tilde{I}_t^I = R_I \cdot \tilde{N}_t$$

The internal use of the total functionality  $F$  also generates cash outflows  $\tilde{O}_t^I$  for the operation and maintenance of the corresponding IT services as well as for the use of central resources (e.g. network and server capacity) during the periods  $t = 1, \dots, T$ . These cash outflows are uncertain due to indeterminate FIUs and depend on the specific cut  $\{S_1, \dots, S_g\}$  of IT services as some of the incorporated costs depend on the number of IT services. If we denote by  $C$  the mean expected cash outflow per use of an IT service, which can be determined by means of the company’s internal experience with the operation and maintenance of IT services, the expected cash outflow per process execution for a cut  $\{S_1, \dots, S_g\}$  with  $g$  IT services will at least add up to  $g \cdot C$ . As the complexity of the company-wide IT service portfolio considerably increases with an increasing number of IT services (Aier 2006; Papazoglou and van den Heuvel 2006; Schelp and Winter 2007), we precautionary provide the opportunity to model even disproportionately increasing operational and maintenance costs, integrate a complexity parameter  $k \in [1; \infty[$ , and formalise the cash outflows  $\tilde{O}_t^I$  for each period  $t = 1, \dots, T$  as

$$\tilde{O}_t^I = g^k \cdot C \cdot \tilde{N}_t$$

### External Reuse of IT Services by Other BUs

The BU is supposed to offer its IT services at the company's internal repository for IT services. Since the underpinning assumption of this paper is, that there are company-wide offset mechanisms for the reuse of IT services, the BU can refinance parts of its specification, implementation, and operational costs for the realisation of the total functionality  $F$ . Conversely, consumers of IT services can save costs of a new specification and implementation of a similar functionality by reusing already existing IT services from another BU. Naturally, BUs will only have an incentive to reuse already existing IT services if the costs for reusing IT services are lower than those for a new implementation of the required functionalities. In accordance to the *pay-per-use principle* we take the premise that there is a company-wide mean expected fee  $R_2$  per reuse of an IT service which can be extrapolated from historical data. The FEUs of an already existing IT service  $S_j$  during the periods  $t = 1, \dots, T$  are also uncertain and can thus be modelled as a discrete random variable  $\tilde{N}_t^j$ . Attention should be paid because in contrast to the FIU we normally have a different FEU for each IT service  $S_j$  of a specific cut  $\{S_1, \dots, S_g\}$  and each period  $t = 1, \dots, T$ . This results from the fact that there is usually a different need for the single atomic functionalities that are encapsulated in a specific IT service  $S_j$ . The offset from the reuse of IT services by other BUs generates additional cash inflows  $\tilde{I}_t^E$ , which are uncertain due to indeterminate FEUs, and can be formalised as

$$\tilde{I}_t^E = R_2 \cdot \sum_{j=1}^g \tilde{N}_t^j$$

The additional cash outflows for the operation and maintenance of IT services as well as for the use of central resources (e.g. network and server capacity) are normally attributed to the BU reusing IT services and thus, will not be considered any further.

To sum up, implementing a new functionality  $F$ , the charging BU obtains a total cash flow  $\tilde{C} = (\tilde{C}_0, \dots, \tilde{C}_T)$  that reflects both cash inflows and cash outflows of the specification, implementation, and internal use of the corresponding IT services, as well as cash inflows of the external reuse. Assembling the different components, we obtain the following cash flows per period  $t = 0, \dots, T$  and per cut  $\{S_1, \dots, S_g\}$  of IT services:

$$\tilde{C}_0 = -O_{basic} - g \cdot O_{int} \quad (t=0) \text{ and } \tilde{C}_t = \tilde{I}_t^I - \tilde{O}_t^I + \tilde{I}_t^E = (R_1 - g^k \cdot C) \cdot \tilde{N}_t + R_2 \cdot \sum_{j=1}^g \tilde{N}_t^j \quad \text{otherwise } (t \geq 1)$$

### Determination of the Financially Optimal Cut of IT Services

The objective of this paper is to determine the financially optimal cut of IT services. Therefore, we are faced with the challenge to value the uncertain total cash flow resulting from a specific cut of IT services in order to enable the comparison of different possible cuts. Following common methods of investment analysis and consolidating all cash flows to the point of decision making  $t = 0$  by determining their net present value (NPV), we obtain

$$NPV(\tilde{C}) = -O_{basic} - g \cdot O_{int} + (R_1 - g^k \cdot C) \cdot \sum_{t=1}^T \frac{\tilde{N}_t}{(1+r)^t} + R_2 \cdot \sum_{t=1}^T \frac{\sum_{j=1}^g \tilde{N}_t^j}{(1+r)^t}$$

This NPV is uncertain since the adopted FIUs  $\tilde{N}_t$  and the FEUs  $\tilde{N}_t^j$  of the IT services are uncertain. A design decision solely based on the NPV's expectation may have serious negative consequences as the cut  $\{S_1, \dots, S_g\}$  with the highest expected NPV can simultaneously be characterised by heavily fluctuating FIUs and FEUs. We therefore also have to integrate the corresponding risk of the NPV and propose to use the following risk adjusted NPV (raNPV) to value the different cuts:

**Assumption 3:** *The raNPV of a cut  $\{S_1, \dots, S_g\}$  of IT services is determined using the well-known preference function  $\mu - \alpha \cdot \sigma^2$  with  $\mu$  denoting the cut's expected NPV,  $\sigma$  denoting the NPV's standard deviation and  $\alpha > 0$  denoting the BU's risk aversion.*

This raNPV corresponds to a preference function that is developed according to established methods of decision theory and is especially consistent with the Bernoulli-principle (see Bernoulli 1954) which requires decision-makers to choose the alternative with the highest expected utility. A similar preference function has been developed by

Freund (1956) and was applied to other IT investment decisions over the last decades for instance by Hanink (1985) and Zimmermann et al. (2008). The higher the value of  $\alpha$ , the more risk-averse is the BU.

When applying this raNPV, we have to estimate both the expected value and the standard deviation of the FIUs  $\tilde{N}_t$  and the FEUs  $\tilde{N}_t^j$ , to form the basis for the valuation of different possible cuts  $\{S_1, \dots, S_g\}$  of IT services. In doing so, we first of all have to determine the underlying distribution of these random variables. Drawing on common stochastic we can model each interaction with the BU during period  $t$  as a Bernoulli distributed event with either executing the business process and therefore requiring the total functionality  $F$  or not (= relevant for the determination of the distribution of  $\tilde{N}_t$ ) or with either requiring a certain already existing IT service for reuse or not (= relevant for the determination of the distribution of  $\tilde{N}_t^j$ ). Starting with the premise that there is a high number of independent interactions with the BU per period  $t$  we benefit from the fact that the number of successful events i.e. the number of internal process executions  $\tilde{N}_t$  or external reuses  $\tilde{N}_t^j$  of an IT service during period  $t$  is approximately Poisson-distributed (see Chao and Scott 2000 and Elhedhli 2006 for a similar distribution assumption in the field of service systems or queuing systems design). We therefore can state the following assumption:

**Assumption 4:** *The FIU  $\tilde{N}_t$  of the total functionality  $F$  during period  $t$  is approximately Poisson-distributed with expected value  $\lambda_t$ . Analogous, the FEU  $\tilde{N}_t^j$  of an IT service  $S_j$  during period  $t$  is approximately Poisson-distributed with expected value  $\lambda_t^j$ .*

The Poisson-distribution is frequently applied for similar problems such as the number of customer requests or order transactions. As its variance coincides with its expected value, we can reduce the problem of determining the expected value and the standard deviation of the FIUs and the FEUs to the problem of just determining their expected values. This is a comparatively easy task for the FIUs as the BU normally justifies the requirement of an additional or renewable functionality  $F$  by a business case and thus usually has a valid estimation of the expected FIUs. In contrast, the estimation of the expected FEUs is rather difficult in practice and, if at all, only possible with extremely high time and resource effort as the BU has to estimate how often its IT services will be reused by other BUs. This reuse of already existing IT services depends on both the reuse potential of each atomic functionality  $F_1, \dots, F_G$  and the specific cut of IT services as this defines the bundling of the atomic functionalities. The BU thus has to estimate how far a certain bundling matches the demand of other BUs. As a BU is hardly able to estimate the demand of other BUs for a specific IT service  $S_j$  for several future periods accurately, we instead suggest to estimate for every IT service  $S_j$  solely a mean expected FEU  $\lambda^j$  over all periods  $t = 1, \dots, T$  and state the following assumption:

**Assumption 5:** *The BU applies for every IT service  $S_j$  a mean expected FEU  $\lambda^j$ , i.e. we have  $\lambda_1^j = \dots = \lambda_T^j = \lambda^j$ .*

We suggest the following procedure to estimate this mean expected FEU  $\lambda^j$ : starting with the company-wide process map (see Heinrich et al. 2009), which provides an overview of a company's business processes as a whole and the actions used in it, we analyse which other business processes require the atomic functionalities  $F_1, \dots, F_G$ . Based on this, we group atomic functionalities with a similar reuse potential into a cluster  $A_k$  and extrapolate in this way  $K \in \{1, \dots, G\}$  clusters  $A_1, \dots, A_K$  of atomic functionalities (for example this can be a classification as is "low FEU", "middle FEU", and "high FEU"). Next, we estimate a mean expected FEU  $\mu_k$  for each of these clusters  $A_k$ ,  $k = 1, \dots, K$ . We make a fairly conservative estimation as we would like to avoid an overestimation of the reuse potential. Finally, we use this mean expected FEU  $\mu_k$  for the estimation of the mean expected FEU  $\lambda^j$  of an IT service  $S_j$  and state the following assumption:

**Assumption 6:** *The mean expected FEU  $\lambda^j$  of an IT service  $S_j$  is determined by means of the company-wide process map as the minimum of the mean expected FEUs of the atomic functionalities  $F_i$  which are bundled in the IT service  $S_j$ . Thereby, the mean expected FEU of a single atomic functionality  $F_i$  is given by the mean expected FEU  $\mu_k$  of the cluster which the atomic functionality is contained in.*

Regarding assumption 5 we can determine the mean expected FEU  $\lambda^j$  of an IT service  $S_j$  by means of the following formula:

$$\lambda^j = \min_{i \text{ mit } F_i \in S_j} \{ \mu_k : F_i \text{ is contained in the cluster } A_k \}$$

We now have determined the expected values of all the FIUs  $\tilde{N}_t$  and FEUs  $\tilde{N}_t^j$ . This provides the basis for the determination of the raNPV of every possible cut  $\{S_1, \dots, S_g\}$  of IT services. For that we require the expected value of a cut's NPV as well as the NPV's variance (see assumption 3). While the expected NPV of a cut  $\{S_1, \dots, S_g\}$  of IT services is the sum of the weighted expected FIUs and FEUs (as the expected value in general has the feature of linearity) and thus easily determinable, such an aggregation does not hold true for the corresponding variance. Instead, in addition to the stand-alone variance of the single FIUs and FEUs, we require also their pairwise correlations. Starting from the premise that the company's IT infrastructure capacity is big enough, experience, however, shows that the FEUs of IT services usually are far more dependent on the specific cut of IT services and the underlying bundling of the atomic functionalities as well as its matching with other business processes than on the FIUs. Besides, both the FIU  $\tilde{N}_t$  and the FEU  $\tilde{N}_t^j$  at a period  $t$  essentially depend on a (now and then heavily) fluctuating customer demand and a market environment that is usually characterised by tight competition and global integration of markets. In comparison, there is, if any, mostly only a small dependency on the FIUs and the FEUs at other periods. In the financial services sector for instance, the demand for consultation is typically many times higher in the last quarter of the year than in the preceding three quarters so that correlations between the FIUs and the FEUs of different quarters would hardly be determinable. Hence, pairwise correlations between the different FIUs and FEUs as well as between FIUs or FEUs at different periods are most often negligible and we therefore assume:

**Assumption 7:** No correlations exist between the different FIUs  $\tilde{N}_t$  and the FEUs  $\tilde{N}_t^j$  at all.

Summarising all these deliberations, determining the financially optimal cut  $\{S_1, \dots, S_g\}$  of IT services is then equivalent to maximising the following raNPV:

$$raNPV(S_1, \dots, S_g) = -O_{basic} - g \cdot O_{int} + (R_1 - g^k \cdot C) \sum_{t=1}^T \frac{\lambda_t}{(1+r)^t} + R_2 \sum_{t=1}^T \frac{\sum_{k=1}^K g_k \cdot \mu_k}{(1+r)^t} - \alpha \cdot \left[ (R_1 - g^k \cdot C)^2 \sum_{t=1}^T \frac{\lambda_t}{(1+r)^{2t}} + R_2^2 \sum_{t=1}^T \frac{\sum_{k=1}^K g_k \cdot \mu_k}{(1+r)^{2t}} \right],$$

where  $g_k \in \{0, \dots, g\}$  denotes the number of IT services which are used with mean expected frequency  $\mu_k$ .

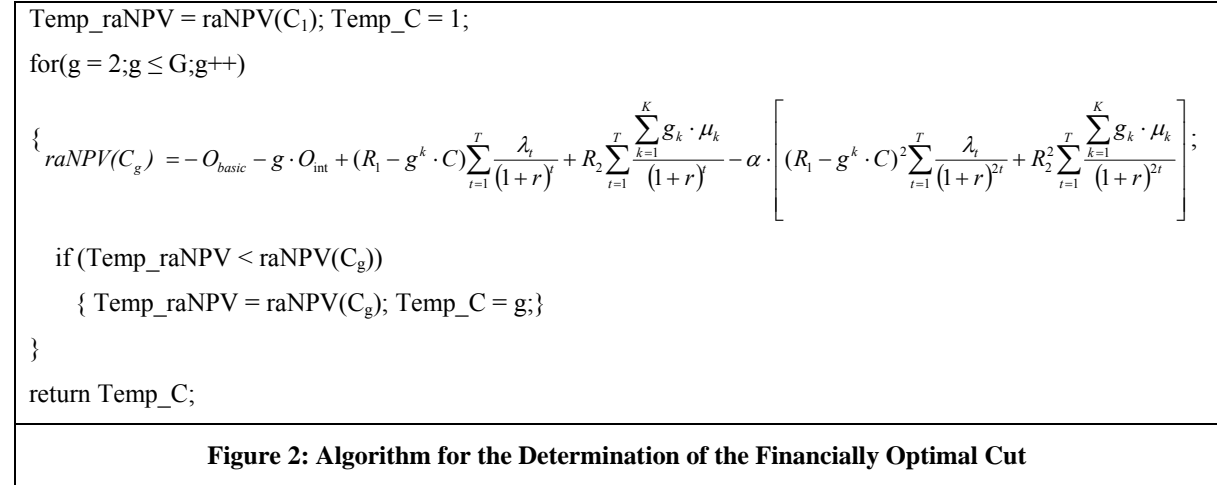
This raNPV is a function of the number  $\mathbf{g}$  of IT services of a certain cut  $\{S_1, \dots, S_g\}$ . Hence, the determination of the financially optimal cut of IT services constitutes a one dimensional, non linear optimisation problem within this variable  $\mathbf{g}$ . A possible solution is to calculate the raNPV of all possible cuts  $\{S_1, \dots, S_g\}$  of IT services and to select afterwards the cut with the highest raNPV. However, this approach is very elaborate for a huge number of atomic functionalities  $F_1, \dots, F_G$ . To reduce this complexity, we suggest a two-stage approach for solving the optimisation problem. This approach requires a significantly smaller expense and can be executed partly or completely automated. In the following, both stages will be explicitly explained:

### 1<sup>st</sup> Stage: Determination of Feasible Cuts of IT Services

Considering the NPV and raNPV defined above, we can observe that most of the summands, which contribute to the raNPV, only depend on the number  $\mathbf{g}$  of IT services of a regarded cut  $\{S_1, \dots, S_g\}$  and not on the specific allocation of atomic functionalities to IT services. Only the cash inflows from the external reuse of IT services can differ depending on the mean expected FEU of the single IT services  $S_j$ . So we can first focus on this part of the raNPV and thereby reduce the possibly huge number of cuts  $\{S_1, \dots, S_g\}$  with  $\mathbf{g}$  IT services by applying the following heuristic for each  $\mathbf{g} = 1, \dots, G$ : we first determine for every possible cut  $\{S_1, \dots, S_g\}$  with  $\mathbf{g}$  IT services a vector  $(g_1, \dots, g_K)$  where the  $k^{\text{th}}$  entry  $g_k$  indicates the number of IT services within the regarded cut  $\{S_1, \dots, S_g\}$  that are reused with mean expected FEU  $\mu_k$ . Consequently, one has  $g_k \geq 0$  for all  $k = 1, \dots, K$  and  $g_1 + \dots + g_K = \mathbf{g}$ . Then, we conduct a vector multiplication of this vector  $(g_1, \dots, g_K)$  and the vector  $(\mu_1, \dots, \mu_K)$  that contains all possible mean expected FEU  $\mu_k$  to obtain the overall expected FEU  $g_1 \cdot \mu_1 + \dots + g_K \cdot \mu_K$  for this cut  $\{S_1, \dots, S_g\}$ . Finally, we choose the cut  $\{S_1, \dots, S_g\}$  with the highest value of this overall expected FEU and call this cut  $\{S_1, \dots, S_g\}$  the *feasible cut* with  $\mathbf{g}$  IT services. Compared with a combinatorial determination of the raNPV for all possible cuts, this step generates only a small expense, which furthermore would arise anyway during a combinatorial procedure.

## 2<sup>nd</sup> Stage: Determination of the raNPV for All Feasible Cuts of IT Services

As a result of the first stage, we get a feasible cut  $\{S_1, \dots, S_g\}$  of IT services for each  $g = 1, \dots, G$ . This feasible cut generates the highest raNPV of all cuts containing  $g$  IT services as it achieves the highest overall expected FEU. To be able to proceed algorithmically, we arrange all feasible cuts lexicographically in the form  $C_1, \dots, C_G$ , where the cut  $C_g$  denotes the feasible cut containing  $g$  IT services. Using the vector  $(g_1, \dots, g_k)$ , with its  $k^{\text{th}}$  entry indicating the number of IT services within  $C_g$ , which are reused with mean expected FEU  $\mu_k$ , once more, the financially optimal cut of IT services can be determined by means of the algorithm that is illustrated in Figure 2:



This algorithm determines the cut with the highest overall raNPV and therefore constitutes the financially optimal solution of the design problem addressed in this paper. It has a complexity of  $O(G)$  which is significantly lower than the complexity of a fully combinatorial solution. Hence, the BU can determine the financially optimal cut with a justifiable expense and come to a better decision about the design of their IT services from a financial point of view. We will demonstrate this by applying our approach to an exemplary context in the next section.

## Operationalisation of the Approach

Financially sound IT service design has not yet made its way into business practice. Instead IT service design decisions are mostly affected by functional and technical aspects. If reuse of already existing IT services is considered at all, companies usually made a basic estimation of the frequency of company-internal reuse without analysing its financial impact. To mitigate this shortcoming, we will now illustrate how the model presented in the previous section may be operationalized by taking an example from the financial services sector. In this context, we will focus on the IT support of the consultation and sales process of the retirement arrangements department (RAD) of a German financial services provider (FSP). The RAD requires reengineering of the IT support of its consultation and sales process. As the company is currently developing its IT architecture towards SOA, this new IT support should be realised by IT services. Therefore, the RAD has to decide how many and which IT services should be implemented for realising the new IT support. As both the SOA platform and the required IT infrastructure are already available the RAD can focus on the implementation of the IT services for the process in question. Furthermore, the company has already established a central repository for IT services as well as company-wide offset mechanisms in order to facilitate the reuse of once implemented IT services. As financial budgets are relatively tight, the RAD considers the provision of its IT services within this repository to reach the best possible refinancing of the implementation and maintenance cost of its new IT support. In order to apply the model presented in the previous section the RAD has to divide its consultation and sales process into suitable atomic actions, which can be mapped to atomic functionalities forming the required total functionality  $F$ . For this purpose, it draws on company-internal and external expertise and obtains the twelve actions that are shown in Figure 3. When estimating the model's input parameters, the RAD is faced with the following major challenges:

1. How can the cash outflows and cash inflows of the internal use of the new IT support by the RAD itself be estimated with high precision?
2. How can the mean expected FEU of the new IT services by other BUs and the resulting cash inflows be estimated with high precision?
3. How can all other input parameters such as the complexity parameter  $k$  or the risk aversion parameter  $\alpha$  be determined?

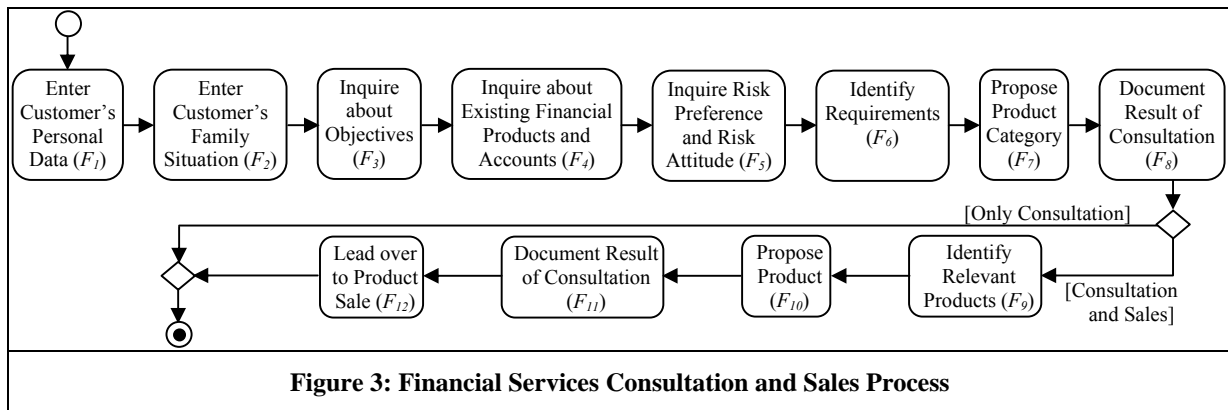


Figure 3: Financial Services Consultation and Sales Process

To estimate the cash outflows and cash inflows of the internal use of the new IT support by the RAD itself, the RAD has made up a business case that is grounded on the RAD's long experience in retirement arrangements and different scenarios relating to the future development of both the market for retirement arrangements as a whole and the RAD's market share. Concerning cost estimations, the RAD also includes the company's internal experience with IT development projects in the context of SOA and external benchmarks from similar projects in the financial services sector. The planning horizon runs up to ten years. Altogether, the RAD gets the following results for the cash outflows and cash inflows resulting from the department's internal utilisation of the new IT support:

The cash outflows for the specification and implementation of the new functionality which are independent from the cut of the corresponding IT services comes to a total of 1,000,000 €. Additionally, the RAD estimates cash outflows of 1,000 € for each implemented IT service that cover the costs of specifying, implementing, and testing an additional interface. Regarding the operation of once implemented IT services, the RAD estimates average cash outflows of 3 € per IT service for the IT support of a consultation, which leads to the recommendation of a specific product group but not to a product closure. The recommendation of specific products, in contrast, occasions twice as high average cash outflows of 6 € per IT service for the IT support of the whole consultation and sales process. Concerning the cash inflows, the RAD receives a fee of 120 € per hour for a consultation in retirement arrangements that leads to the recommendation of a specific product group but not to a product closure. As 22 € of this hourly fee can be attributed to the IT support and as consultation in retirement arrangements on average takes 1.5 hours, the RAD attributes 33 € to the IT support of a consultation. If the customer additionally closes a contract on a financial product after the consultation process, the consultation fee is dropped. Instead, a proportion of 20% of the surpluses generated by the product closure is attributed to the IT support of the consultation process. Since the average revenue for all retirement arrangement products is 250 €, the RAD in this case attributes 50 € to the IT support. Finally, for the first period, which equates one year in this example, the retirement arrangements department expects 1.3 million consultation sessions. This number is estimated to increase by 5% p.a. in each of the consecutive periods. On average, there will be a product closure in 30% of all consultation sessions.

As a basis for estimating the mean expected FEU of the new IT services, the RAD first has to estimate the mean expected FEU of any atomic functionality. Following the approach presented in the previous section, the RAD first determines the potential reuse of each of the twelve actions within its consultation and sales process by using the company's process map. Second, the RAD verifies these basic estimations through employing expert interviews. Third, for reasons of simplification, the RAD maps these verified estimations to three different classes of potential reuses, namely 600,000 reuses, 1 Mio reuses and 1.3 Mio reuses p.a. This classification has been determined by clustering the verified estimations. The result of this classification is summarised in Table 1:

Atomic Functionality	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$
Expected FEU	1.0	1.3	1.3	1.0	1.0	0.6	0.6	1.3	1.0	0.6	1.3	1.0

Finally, before applying the optimisation model the RAD also has to determine the company's risk-aversion  $\alpha$ , its risk-free interest rate  $r$  and the complexity parameter  $k$  as well as the mean expected income per reuse of an IT service. While the first two parameters are predefined for the whole company by central controlling, the RAD has to analyse the company's internal experience with IT service development projects and the price structure on the company's central repository to gain the second two parameters. Altogether, the RAD obtains  $\alpha = 0.02$ ,  $r = 4\%$  and  $k = 1$  as well as an average price for reusing an IT service of 1 €.

Table 2 provides an overview that summarises the required data and where it can be retrieved within the company.

Necessary Parameter	Value of the Parameter	Exemplary Data Origin
Cash outflows for the specification and implementation of the new functionality	1,000,000 €	Business case
Average cash outflow for the operation of a single new IT service	3 € for consultation only 6 € for consultation and sales	Experience from other SOA projects within the company
Average cash inflow of the use of the new functionality by the RAD itself	33 € for consultation, 50 € for consultation and sales	Experience from the existing consultation and sales process
Average cash inflow of the reuse of a single IT service within the company	1 € per reuse	Experience from the internal repository for IT services
Risk-free interest rate $r$	4%	Predefined for the whole company by central controlling
Parameter of risk aversion $\alpha$	0.02	Predefined for the whole company by central controlling
Complexity parameter $k$	1	Experience from other SOA projects within the company
Planning horizon $T$	10	Business case

Now, we are ready to apply the proposed two-stage optimisation approach presented in the previous section. In doing so, we restrict our considerations to cuts where the corresponding IT services encapsulate a sequence of successive atomic functionalities and thus have to analyse  $2^{11} = 2048$  different possible cuts of IT services. We first determine for each possible number  $\mathbf{g} = 1, \dots, 12$  of IT services the cut with the maximum total expected FEU by other BUs. Of course this is easy with  $\mathbf{g} = 1$  or  $\mathbf{g} = 12$  as there is only one cut of IT services. However, it can be rather elaborate with other values of  $\mathbf{g}$  as – even with our limited basic population – there are many potential cuts of IT services. As a result, we get for each  $\mathbf{g} = 1, \dots, 12$  one cut of IT services maximising the expected FEU. Secondly, we calculated the raNPV for these twelve feasible cuts of IT services. Table 3 provides the maximum total expected FEU and the corresponding raNPV for each possible number  $\mathbf{g} = 1, \dots, 12$  of IT services. The results are quite different: on the one hand, a cut with one IT service per atomic functionality ( $\mathbf{g} = 12$ ) results in approximately 20 times the reuse of a cut with one IT service that encapsulates the whole functionality ( $\mathbf{g} = 1$ ). On the other hand, however, the raNPV initially increases with the number of IT services, but decreases from four IT services onward. The raNPV ranges from 219.6 Mio. € achieved when implementing three IT services to -34.8 Mio. € achieved when implementing twelve IT services. If the RAD decided intuitively on the cut of its IT services on the basis of the maximum expected FEU it would implement twelve IT services and end up with the financially worst solution with a negative raNPV. If the RAD makes a decision based on the raNPV as proposed by the approach described in this

paper, it will select a cut that consists of three IT services with  $F_1$  and  $F_2$  each implemented as a single IT service and  $F_3$  to  $F_{12}$  implemented in a third IT service. Realizing this cut the RAD implements significantly less IT services and achieves only 25% of the maximum expected FEU, but implements the financially best solution.

<b>g</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Maximum Total Expected FEU	0.6	1.6	2.9	4.2	5.2	6.5
raNPV [Mio. €]	203.3 €	213.5 €	<b>219.6 €</b>	219.4 €	210.3 €	197.1 €
<b>G</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
Maximum Total Expected FEU	7.5	8.5	9.4	10.4	11.4	12
raNPV [Mio. €]	175.2 €	146.8 €	111.2 €	70.0 €	22.4 €	<b>-34.8 €</b>

Therefore, we can finally state that the decision on the design of IT services based on the expected FEU, as it is intuitively done today in the majority of cases, frequently leads to false investment decisions and a suboptimal overall IT service portfolio. That is why it is not sufficient to focus on the pure number of reuses, but essential to take into account the financial consequences of different possibilities to design IT services as this can make an important difference when implementing IT services in the context of SOA. However, this would mean a tremendous change in today's practice.

## Practical Implications, Limitations, and Conclusion

Service-oriented design and development are intensively discussed in research and practice. However, so far there is still little research available regarding the valuation of the financial consequences of different functional scopes of IT services. The objective of this paper is the formalisation and valuation of the total cash flows that accompany different possibilities to realise a new functionality by means of IT services. We therefore focus on the perspective of a single BU and develop an optimisation model for the determination of the financially optimal functional scope of IT services. This approach simultaneously integrates the cash flows from the implementation and operation of the IT services by the BU itself as well as cash inflows resulting from the offset of the company-internal reuse of once implemented IT service by other BUs. The underpinning assumption is that the company supports the reuse of IT services by providing a central repository, where BUs can offer their IT services and thus refinance parts of their specification, implementation and operational costs for an additional or renewable functionality. Finally, the applicability of the approach is illustrated by an example showing a typical decision situation of a FSP.

Although designing IT services for reusability is seen as one of the major advantages of SOA in science and practice, appropriate methods for evaluating the financial value of a certain design of IT services for reusability has often been absent until now. Based on a number of assumptions, we found that the consideration of the financial consequences of different design possibilities for IT services leads to a better decision with regard to the corresponding risk adjusted net present value. Thus, companies are faced with the challenge to establish decision models (such as this presented within this paper) to support the financially optimal design of IT services in order to achieve a close business/IT alignment. Companies with a decentralised decision-making and paying structure additionally require suitable incentives for a single BU, such as internal marketplaces for IT services or central offset mechanisms, to support the design of IT services considering their company-wide reuse potential.

Certainly, the interpretation of the results of the model presented in this paper is restricted due to some limitations: first of all, this paper focuses exclusively on the financial impact of IT service design and neglects technical or functional aspects. This restriction seems, however, justifiable as a number of related articles already addressed these aspects. This paper, hence, complements existing research with an in-depth, quantitative analysis of the financial impact of IT services' design and can be applied as an additional step to improve existing service-oriented design and development methods. Secondly, we made some fairly restrictive assumptions with regard to the allocation of the atomic functionalities to IT services and the distribution of the FIUs and FEUs of IT services as well as the non-existence of correlations between these frequencies. These assumptions provide a starting point for



further research in this area which should address these assumptions separately and analyse them more profoundly. Thirdly, the issue of determining the fee for the reuse of IT services by other processes or BUs within the same company is not broached, although this deeply affects the reuse potential of once implemented IT services. Additionally, the expected FEUs were estimated roughly by forming clusters of atomic functionalities and applying mean expected FEUs over different periods. This seems, however, justifiable as the objective of the paper is to develop a first sound quantitative methodology for the determination of the financially optimal design of IT services realising a certain amount of additional or renewable functionality. Appropriate approaches for the determination of the required input data would be necessary and should be addressed by further research.

Summing up, despite the potential for improvement the optimisation model presented in this paper constitutes a valuable extension of existing approaches to the identification and design of IT services by providing a first basis for BUs to realise IT services with a financially optimal functional scope.

## References

- accenture. 2008. *High Performance IT 2008: There's No Substitute for Substitution*, accenture.
- Aier, S. 2006. "How Clustering Enterprise Architectures helps to Design Service Oriented Architectures," in *Proceedings of the 2006 IEEE International Conference on Services Computing*, Chicago (USA), pp. 269-272.
- Albani, A., Keiblinger, A., Turowski, K., and Winnewisser, C. 2003. "Identification and Modelling of Web Services for Inter-enterprise Collaboration Exemplified for the Domain of Strategic Supply Chain Development," in *Lecture Notes in Computer Science: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, R. Meersman, Z. Tari, and D.C. Schmidt (eds.), Berlin: Springer-Verlag, pp. 74-92.
- Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., and Holley, K. 2008. "SOMA: A method for developing service-oriented solutions," *IBM Systems Journal* (47:3), pp. 377-396.
- Bernoulli, D. 1954. "Exposition of a New Theory on the Measurement of Risk," *Econometrica* (22:1), pp. 23-36.
- Boehmann, T. and Krcmar, H. 2005. "Modularisierung: Grundlagen und Anwendungen bei IT Dienstleistungen," in *Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement*, T. Herrmann, U. Kleinbeck, and H. Krcmar (eds.), Heidelberg: Physica-Verlag, pp. 45-84.
- Boerner, R. and Goeken, M. 2009. "Identification of Business Services – Literature Review and Lessons Learned," in *Proceedings of the 15th Americas Conference on Information Systems (AMCIS)*, San Francisco (USA).
- Capgemini. 2009a. *Studie IT-Trends 2009*, Capgemini.
- Capgemini. 2009b. *Global CIO Report - Harnessing Information Value: Could you be a digital winner?* Capgemini.
- Chan, Y.E. and Reich, B.H. 2007. "IT alignment: what have we learned?," *Journal of Information Technology* (22:4), pp. 297-315.
- Chao, X. and Scott, C. 2000. "Several results on the design of queueing systems," *Operations Research* (48:6), pp. 965-970.
- Choi, J., Nazareth, D.L., and Jain, H.K. 2010. "Implementing Service-Oriented Architecture in Organizations," *Journal of Management Information Systems* (26:4), pp. 253-286.
- Elhedhli, S. 2006. "Service System Design with Immobile Servers, Stochastic Demand, and Congestion," *Manufacturing & Service Operations Management* (8:1), pp. 92-97
- Erl, T. 2008. *SOA Principles of Service Design*, Upper Saddle River, NJ: Prentice-Hall.
- Erradi, A., Anand, S., and Kulkarni, N. 2006. "SOAF: An Architectural Framework for Service Definition and Realization," in *Proceedings of the 2006 IEEE International Conference on Services Computing*, Chicago (USA), pp. 151-158.
- Freund, R. J. 1956. "The Introduction of Risk into a Programming Model," *Econometrica* (24:3), pp. 253-263.
- Gartner. 2010. "Leading in Times of Transition: The 2010 CIO Agenda," in *EXP Premier Report January*, Stamford, CT: Gartner, Inc.
- Hagel, J. 2002. "Edging into Web Services," *McKinsey Quarterly* (4:4), pp. 29-37.
- Hanink, D. M. 1985. "A Mean-Variance Model of MNF Location Strategy," *Journal of International Business Studies* (16:1), pp. 165-170.
- Heinrich, B., Henneberger, M., Leist, S., and Zellner, G. 2009. "The Process Map as an Instrument to Standardise Processes – Design and Application at a Financial Service Provider," *Information Systems and e-Business Management* (7:1), pp. 81-102.
- Hirschman, A.O. 1990. *Exit, Voice and Loyalty: Responses to Decline in Firms, Organizations and States*, Cambridge, MA: Harvard University Press.

- Klose, K., Knackstedt, R., and Beverungen, D. 2007. "Identification of Services – A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning," in *Proceedings of the 15th European Conference on Information Systems (ECIS)*, St. Gallen (Switzerland).
- Kohlmann, F. and Alt, R. 2007. "Business-Driven Service Modeling – A Methodological Approach from the Finance Industry," in *Business Process & Service Computing: First International Working Conference on Business Process and Services Computing*, W. Abramowicz, L. Maciaszek (eds.), Leipzig, pp. 180-193.
- Krafzig, D., Banke, K., and Slama, D. 2007. *Enterprise SOA – Service-Oriented Architecture Best Practices*, Upper Saddle River, NJ: Prentice-Hall.
- Legner, C. and Heutschi, R. 2007. "SOA Adoption in Practice – Findings from Early SOA Implementations," in *Proceedings of the 15th European Conference on Information Systems (ECIS)*, St. Gallen (Switzerland).
- Loeche, J. and Legner, C. 2010. "SOA Adoption in Business Networks – Does SOA Live Up to High Expectations?" in *Proceedings of the 18th European Conference on Information Systems (ECIS)*, Pretoria (South Africa).
- Millard, D.E., Howard, Y., Abbas, N., Davis, H.C., Gilbert, L., Wills, G.B., and Walters, R.J. 2009. "Pragmatic web service design: An agile approach with the service responsibility and interaction design method," *Computer Science - Research and Development* (24:4), pp. 173-184.
- McCullough, M.A., Berry L.L., and Yadav, M.S. 2000. "An Empirical Investigation of Customer Satisfaction After Service Failure and Recovery," *Journal of Service Research* (3:2), pp. 121-137.
- Object Management Group (OMG). 2005. *Unified Modeling Language: Superstructure*, OMG.
- Object Management Group (OMG). 2006. *Unified Modeling Language: Infrastructure*, OMG.
- Papazoglou, M.P. and Georgakopoulos, D. 2003. "Service-oriented Computing," *Communications of the ACM* (46:10), pp. 25-28.
- Papazoglou, M.P. and van den Heuvel, W.-J. 2006. "Service-Oriented Design and Development Methodology," *International Journal of Web Engineering and Technology* (2:4), pp. 412-442.
- Patig, S. and Wesenberg, H. 2009. "Role of Process Modeling in Software Service Design," in *Lecture Notes in Computer Science: Service-Oriented Computing: Agents, Semantics, and Engineering*, R. Kowalczyk, Q.B. Vo, Z. Maamar, and M. Huhns (eds.), Berlin, Heidelberg: Springer-Verlag, pp. 420-428.
- Quartel, D., Dijkman, R., and van Sinderen, M. 2004. "Methodological Support for Service-oriented Design with ISDL," in *Proceedings of 2nd International Conference on Service Oriented Computing*, New York (USA).
- Schelp, J. and Winter, R. 2007. "Towards a Methodology for Service Construction," in *Proceedings of 40<sup>th</sup> Hawaii International Conference on System Sciences (HICSS)*, Hawaii (USA).
- Wang, Z., Xu, X., Zhan, D. 2006a. "A Survey of Business Component Identification Methods and Related Techniques," *International Journal of Information Technology* (2:4), pp. 229-238.
- Wang, Z.-J.; Zhan D.-C., Xu, X.-F. 2006b. "STCIM: A Dynamic Granularity Oriented and Stability Based Component Identification Method," *ACM SIGSOFT Software Engineering Notes* (31:3), pp. 1-14.
- Watson, H.J., Aronson, J.E., Hamilton, R.H., Iyer, L., Nagasundarma, M., Nemati, H., and Suleiman, J. 1996. "Assessing EIS benefits: A survey of current practices", *Journal of Information Technology Management* (7:1/2), pp. 23-30.
- Widjaja, T. and Buxmann, P. 2009. "Service-oriented architectures: modeling the selection of services and platforms," in *Proceedings of the 17th European Conference on Information Systems (ECIS)*, Verona (Italy).
- Winkler, V. 2007. "Identifikation und Gestaltung von Services. Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich," *Wirtschaftsinformatik* (49:4), pp. 257-266.
- Zhang, Z., Liu, R., and Yang, H. 2005. "Service Identification and Packaging in Service Oriented Reengineering," in *Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, Taipei, (Taiwan, Republic of China).
- Zimmermann, S., Katzmarzik, A., and Kundisch, D. 2008. "IT Sourcing Portfolio Management for IT Service Providers – A Risk/Cost Perspective," in *Proceedings of the 29th International Conference on Information Systems (ICIS)*, Paris (France).