Case Study – Automating Direct Banking Customer Service Processes with Service Oriented Architecture

Andreas Eberhardt, Oliver Gausmann, Antonia Albani

1 b.telligent GmbH & Co.KG
85748 Garching/Munich, Germany
andreas.eberhardt@btelligent.de

2 University of Augsburg
Chair of Business Informatics and Systems Engineering
86159 Augsburg, Germany
oliver.gausmann@wiwi.uni-augsburg.de

3 Delft University of Technology
Information Systems Design
2628 CD Delft, The Netherlands
a.albani@tudelft.nl

Abstract. The direct banking business is characterized by integrated distribution channel politics and holistic sales approaches combined with multi-channel-management. Direct banks in Europe and especially in Germany are currently facing increasing market competition. The crucial factors for growth are product innovation, cost control and the flexibility to react individually to each customer in a rapidly changing business environment. In order to compete, direct banks are forced to undergo a drastic transformation of business processes as well as organizational and managerial structures. The application of new concepts in building information systems is therefore necessary in order to further support business needs and allow for the management and adaptation of systems that are dependent on the fast changing market requirements. This paper shows how the information technology (IT) landscape of one of the five leading direct banks in Germany could be optimized by means of a service-based orientation. The case outlined in this paper focuses on the customer service domain. The main goal is to concurrently reduce costs by automating business processes and to increase the quality of customer services. A reference model for these customer service processes is then introduced. Based on this model, this paper describes a business component-oriented system architecture according to identified business components, and their corresponding services.

1. Introduction

Foreign banks entering the German banking market and the development of new customer specific products are putting pressure on the whole market. This fact, as well as the increased transparency due to the high distribution rate of Internet access in Germany, has led to a structural change of the entire banking market. Decreasing customer loyalty and simultaneously decreasing margins characterize the new market.
situation. Affiliated and smaller banks in particular are not able to handle their cost structure. Hence, a trend towards more centralization and the reduction of the affiliate banking business can be observed. Since 1995 the number of credit institutes registered in Germany decreased by more than 30%. The number of affiliated banks decreased by almost the same percentage [1]. Against this trend, however, the market for direct banking is increasing. An important cornerstone of this success can be attributed to an underlying cost consciousness [2]. Due to these facts, the market for direct banking will attract more competitors and cost reduction; flexibility and high technical standards will be required to meet the challenges presented by this situation. In order to cope with these challenges, a drastic transformation of the business processes as well as organization and managerial structures are necessary. Additionally, the deployment of information and communication technology (ICT) as well as the reengineering of the available information systems becomes inevitable, supporting and automating not only of the inter- but also the intra-enterprise business processes. The use of business components for the (re) design and (re) engineering of information system provides considerable benefits since they “directly model and implement the business logic, rules and constraints that are typical, recurrent and comprehensive notions characterizing a domain or business area” [3, p. 5].

The idea of building individual software systems by combining pre-fabricated software components from different vendors to construct unique applications was introduced for the first time by McIlroy in 1968 [4]. The software components of an information system that support directly the activities in an enterprise are usually called business components. A business component provides a set of services out of a given business domain through well-defined interfaces and hides its implementation [5]. The principle of modular design that underlies component-based software systems is equally important for the discussion of the technological as well as the economic advantages of component-based software systems. The compositional plug-and-play-like reuse of components might enable software component markets, where different components can be individually combined according to the customers' need. The general advantage of such systems is widely discussed in literature, c.f. [6-10]. Modular systems have been described as the result of a functional decomposition [11], and the conception of modular systems has been thoroughly analyzed by system theory.

Fundamental for this notion of modular systems is an overall domain-engineering concept in order to gain a perfection of component orientation. For domain engineering [12, p. 19-59, 13, p. 159-169], different domain engineering processes are well-known and in use, among others [14-16]. The methods mentioned contribute to different aspects of domain engineering theory; for example, in identifying prominent or distinctive features of a class of systems or in defining characteristics of maintainability and understandability of a family of systems. A domain engineering method which, throughout all stages of development, addresses the domain in which the business component is used, is the Business Component Modeling (BCM) Process introduced by [17]. Due to being explicitly developed for the context of business components, BCM considers additional prerequisites such as reusability,
marketability and self-containment, which are required for different domains. This paper contributes to the validation of the BCM process and specifically to the validation of the BCI-3D method by applying them to the domain of direct banking customer services.

The outline of the rest of the paper is as follows. In section 765 the BCM process will be briefly explained. Since the identification of business components is still a crucial factor, the Business Component Identification (BCI) method [18] and its further development (BCI-3D) [19] is also shortly described. The BCI-3D method is used for the identification of reusable and marketable business components for the direct banking customer services domain. A detailed description of the customer service domain can be found in section 3. The biggest challenge concerning a specification of the requirements for service oriented IT architectures is an overall and detailed analysis of the considered business domain. The result of this domain analysis is either a defined reference model or a business domain specific domain model or a mix of both. In all cases, reference models as well as domain models have to be derived from the business processes of that specific domain and build the basis for the design of component oriented models and the development of the corresponding component-oriented application systems, as described in section 4. Since this paper contributes to the validation of the BCM process and specifically to the validation of the BCI-3D method, the evaluation results and conclusions are given in section 5.


A precondition to component-based development of application systems by using business components is a stable component model. In order to obtain stable business component models, a well-defined identification process is necessary. The basis for the identification of reusable, marketable and self-contained business components is an appropriate and high quality business domain model. Such a model not only serves to satisfy the requirements for a single application system but rather for a family of systems – and therefore for a certain domain. In order to achieve this, we used the Business Component Modeling (BCM) process [17, 18] as shown in Fig. 1.

![Fig. 1. Business Component Modeling Process [17]](image-url)

The Business Component Modeling Process can be divided into two major parts, the *Component Based Domain Analysis* and the *Component Based Domain Design*. Rectangles denote sub-phases of the process, and ellipses contain the resulting
diagrams and models of each sub-phase [17]. In this paper we concentrate on the Domain Scope and the Business Components Identification sub phases and will not describe the Refinement and Standard Specification sub-phase. For a detailed description of the whole process we refer to [17]. In the Domain Scope sub-phase, a model of the respective domain has to be derived. Therefore the domain of interest is identified, characterized and business processes with their functional tasks are defined. In addition, data is collected to analyze the information objects and their relationships. Possible sources of domain information include existing systems in the domain, domain experts, handbooks, requirements on future systems, market studies, and so on. As a result of the first sub-phase a functional-decomposition diagram and a domain based data model are generated. Several methodologies and languages can be used to derive the domain model as e.g., DEMO (Design & Engineering Methodology for Organizations) [20-22], ARIS [23], UML (Unified Modeling Language).

The business domain models are fundamental for the next sub-phase in the BCM process, namely the Business Components Identification phase. Since the identification of business components is strongly dependent on the quality of the underlying business model, the use of an adequate methodology is absolutely necessary. In order to optimize the process of identifying high quality, reusable and marketable business components the Three Dimensional Business Components Identification Method (BCI-3D) has been introduced by Albani et al. [19]. BCI-3D is an extension of the Business Components Identification (BCI) method [18], which has been applied in several case studies such as [24, 25], and which has been improved while considering the evaluation results of the case studies mentioned.

In BCI-3D one can distinguish between three types of relationships necessary for the identification of business components; the relationship between single process steps, the relationship between information objects and the relationship between process steps and information objects. A relationship type distinguishes between subtypes expressing the significance of a relationship. For example, the relationship between single process steps expresses – based on their cardinality constraints – how often a process step is executed within a process and therefore how close two process steps are related to each other in that business domain. The relationship between information objects defines how loosely or tightly the information objects are coupled. In addition, the relationship between process steps and information objects defines whether a corresponding information object is used or created while executing the respective process step. All types of relationship are of great relevance in order to define which information object and process steps belong to which component. The relationships are modeled in the BCI-3D method using a weighted graph. The nodes represent either information objects or process steps and the edges characterize the relationships between the nodes. Weights are used to define the different types and subtypes of relationships and build the basis for assigning nodes and information objects to components. In order to optimize its display, the graph is visualized in a three-dimensional representation having the process steps and information objects arranged in circles and without showing the corresponding weights (see Fig. 7). By satisfying defined metrics such as minimal communication between and maximal
compactness within business components, the BCI-3D method groups process steps and their corresponding information objects with the aim of obtaining an abstract component model in a top-down way. The constraint of providing optimal grouping while minimizing communication means that an optimization problem needs to be solved with a genetic algorithm. The algorithm starts with a predefined solution and improves it by incremental iteration [19]. The starting solution is generated using a greedy graph-partitioning algorithm [26]. For improving the initial solution, the Kernighan and Lin graph-partitioning algorithm [27] has been implemented. The result of applying the BCI-3D method to a defined domain results in a business component model, describing their relationships and provided and/or required services.

To illustrate the domain scope and component identification sub-phases with their resulting diagrams and models, the BCM process is applied to the domain of direct banking customer services described in the next sections.

3. The domain of Direct Banking Customer Service

A sustainable economic growth of direct banks in the German banking sector particularly depends on efficient cost structures as well as cost-effective and quality-conscious customer service processes. Compared to traditional banks, the direct banking sector in Germany is characterized by high rates of customer growth. Due to this fact, new challenges regarding business processes and IT systems arise. The risk of growing internal costs due to inefficiencies of the implemented service processes and service tasks has to be managed at an early stage. Consequently, the business domain service, and particularly the domain customer services, is in the focus of interest of a direct bank. The business processes in the mentioned business domains are heavily dependent on guidelines and directives. In the example case, these directives are defined and modeled by a company-wide process department. Fig. 5 illustrates an example directive. A directive describes the business tasks, the corresponding responsible organizational units (cost center) and the related resources in a graphical based form. To each of these business process steps, guidelines focuses on legal restrictions, check criteria or documentation needs. The implementation and daily handling of these guidelines and directives are periodically audited and evaluated in an internal review. The sum of all guidelines and directives can be seen as a complete reference model for the domain direct banking customer service. The reference model of one of the leading direct banks in Germany has been used to analyze the domain of customer service.

The actual handling of the customer services is mostly characterized by manual business processes. Information is either entered directly in the supporting IT system or integrated via the web front-end of the customer portal. The main problems are system inflexibility concerning the automation of functionality and integration of new business requirements in the IT system on the one hand and difficulties concerning the full integration of the system landscape, especially the information exchange with the mother bank on the other hand. Due to this fact there are cost and time intensive
inefficiencies in the customer processes as well as supplementary costs of the error handling of the information integration in the system landscape.

In order to model the business functionality of the example domain, functional decomposition diagrams, as used in the ARIS methodology by [23], have been used for describing business tasks and business activities. A functional decomposition diagram splits the main tasks of a business process in four areas: Task area, Function area, Sub-function area and Elementary function area. A more detailed splitting of business process tasks is not efficient regarding a general management view. Fig. 2 illustrates for the customer service domain the task area Service in a functional decomposition diagram. Due to space limitations, we do not provide the process models which visualize the execution order of the single business tasks and activities.

Fig. 2. Functional decomposition diagram for customer services at a direct bank

The task area Services provides several functions such as Customer Services, Administration Services and Helpdesk Services. In Fig. 2 only the Customer Services function and an extract of its sub-functions are listed. The complete range of sub-functions within the customer services function can be restricted to 27. In the present case, ‘relevant’ sub-functions were limited to tasks considered in the daily contribution margin calculation. This limitation facilitates the focus on the most commonly used and most expensive tasks and it considers more than 90% of the whole business activities in the mentioned domain. To get a better understanding of customer services in the direct banking sector, selected sub-functions (highlighted grey in Fig. 2) are described. In this context, the description of tasks is always in the form ‘verb - subject – adverb’ in order to standardize their description and to avoid ambiguity of the modeling notation.

The sub-function Customer address change / Master data change provides tasks like “change name” or “change customer address” for example in case of marriage, divorce or removal. For all these tasks, the differentiation of the communication channel used is important, for example concerning business intelligence requirements, such as calculating the customer service process costs (contribution margin) and controlling the requirements of the different communication channels in the future. A
precondition to change the customer master data via mobile device, Internet or telephone is a valid customer legitimization. Therefore, that sub-function also describes typical direct banking tasks like legitimization of persons by means of the PostIdent proceeding. The German PostIdent proceeding is the standard person legitimization proceeding accomplished by the Deutsche Post AG.

The next logical step after creation and legitimization of a customer is the account opening. Therefore, the corresponding sub-function Account opening provides tasks such as “check residential status of customer” in case of an abroad customer address for dispatch. If the customer wants to trade options or financial futures, a clarification of trading risks has to be collected and a risk group has to be calculated. The differentiation of the used communication channels in this sub-function is also an important step.

The last steps in the customer lifecycle are the customer inheritance business task, listed as sub-function Inheritance management in the functional decomposition diagram. The first task is the request of an official inheritance confirmation. Afterwards, the bank has to check and delete standing orders for exactly defined cases and criteria. If the customer has a maestro or credit card, these cards have to be locked, and the credit card service provider has to be informed. Additionally, the tax authorities have to be informed about the account balance of the last day before the inheritance. In case of clarified inheritance, the last task is the disclosure of the customer accounts.

The last sub-function we describe is Returned consignment. As described above, each of the illustrated sub-functions listed in figure 2 includes several tasks, which are listed in the functional decomposition diagram by means of elementary functions. Due to space limitations we only list and describe the elementary functions of the Returned consignment sub-function in more detail.

At first, a returned consignment has to be checked and classified. In special cases like PIN numbers or TAN lists, the returned consignment has to be destroyed immediately. In all other cases, it has to be verified if the address of the returned consignment matches the most recent customer address for dispatch in the service system (check customer address in own system). If the customer address is corresponding and no lock reason is set in the system (check lock reason returned consignment), the cover of the new correspondence is marked with “2”, and it is then sent to the same customer address for dispatch. If this correspondence returns a second time, the customer address for dispatch is compared to the customer tax address in the host system. In case of a difference between these two customer addresses, the returned consignment is sent a third time to the customer address of tax. At the same time, the customer address for dispatch is deleted in the existing system. In case of a third returned consignment, the customer account is locked with the lock reason returned consignment, and the returned consignment is collected in the customer records. If no customer record exists, a new one is to be created. The next task is to check the customer’s resident status. If the resident status is “non-resident”, the internal audit is to be informed. If the resident status is “resident”, an address match is made with forward orders of the Deutsche Post AG. If there is still no new customer address identified, an address research is started. Therefore, the mortality database is scanned. If new information about the customer address cannot be found, the so-called EMA requests are scanned. An EMA request is a request from
other companies like shippers who had similar problems with this customer. If there is no address match in the EMA files, a new EMA entry is made. After 6 months, the customer account is checked for discharge. Therefore, a supervision charge is accounted. If a custody account also exists, it will be disposed at the same time. Finally, the account balance is written off with respect to the defined detailed instructions. The customer account is deleted. Finally, all the documents, protocols and official forms have to be archived.

Fig. 3. Functional decomposition diagram of the returned consignment sub-function

A functional decomposition diagram (see Fig. 4) shows all the necessary business tasks with their sub tasks, but does not refer to any information objects, such as customer address, which are used or created while executing those functions. Furthermore there is no chronological order between different tasks. Therefore, the relation between single tasks and information objects has to be modeled in an alternative manner as shown, for example, in Fig. 5.

As mentioned in section 2, there are several methods concerning the modeling of business processes. In the actual case the considered direct bank uses directives and guidelines for the definition of the main tasks, their orders and their dependences. Fig. 5 illustrates such a directive. The graphical modeling is done in the Adonis tool and the provided notation possibilities. Each of these directives and guidelines are provided verbally. The guidelines determine check criteria, supervision periods and
documentation necessities in detail. Every directive additionally indicates the triggering events. In the example under consideration, the triggering event is the physical entry of a returned consignment. The above mentioned destroy returned consignment task in case of PIN numbers or TAN lists is not clearly mentioned in the directive, but in the corresponding guidelines. However, the directives give hints about the relation between tasks, depending on the business environment. All available business process models are provided in the local intranet network and show the flag ‘valid’ or ‘invalid’. As consequence it can be assured, that all important business processes are companywide communicated and known.

Having defined all business tasks and the corresponding information objects, the relationships between business tasks, between information objects, and between business tasks and information objects need to be defined in order to use them in the business components identification method. For the purpose of domain analysis, the two task relations standard and optional were defined. A standard relationship between tasks arises from the business process structure. In Fig. 5 the arrows reflect this relationship. The stringent following of each arrow with regard to the different tracking possibilities describes a standard relationship apart. The starting point of all standard relationships is always the triggering event. Not taking into account the final process course all standard relationships end at the directive-ending node. The second relationship type between different tasks is the optional relationship. In this context, an optional relationship refers to a possible relation between two tasks, clearly depending on the predetermined guidelines. For instance the creation of minutes during a customer conference call is not always mandatory.

The authors decided to distinguish between a standard and an optional relationship to have the possibility of different weightings during the business component identification. With these relationships already defined, the communication effort of

Fig. 5. Business process directive: handling of returned consignment
the identified business components is minimized due to the above-mentioned algorithms of the BCI-3D method.

In addition to the relationships between tasks, the relationships between task and information objects have to be considered. As already stated, a task can create information objects (outgoing information object) or use (incoming) information objects. For example, a customer correspondence can only be sent if the customer address for dispatch is known. To execute the task Sent customer correspondence the incoming information objects customer address for dispatch, stamp, envelope and prepared customer correspondence are required.

Finally, the relationships between information objects have to be provided for efficient business component identification. Three different types of relationships are defined in this case: related-to, part-of and state-of. In general, a related-to relationship is defined between entities like customer and contract. A customer has at least one or more contracts whereas a contract is attributed exactly to one customer. The relationship part-of signals that an information object is part of another information object. For example, the information objects customer addresses for dispatch and customer tax address are part of the information object customer master address data. Finally the relationship state-of signals a change in the condition of an information object. For instance, a customer contract runs through the conditions customer request and offered contract before becoming the status customer contract.

All the mentioned relationships belong to the set of relationships provided and recommended by the BCI-3D method. In order to gain all relevant information for modeling the supporting information system by means of business components, ten representative customer service processes were modeled in the described proceeding, including business tasks, information objects and all their relationships as mentioned above. All the considered services processes are highlighted in Fig. 6. Applying the BCI-3D to the services mentioned resulted in a business component model as illustrated in section 4.


The modeling of the above mentioned customer service processes leads to 288 different information objects and 159 business process steps (BPS). Concerning the information objects it was not distinguished between paper based and digital information objects. In order to use the business component identification method for identifying reusable and marketable business components, all relationships need to be weighted. The reason therefore is to define how strong objects (process step or information objects) are related in order to ensure their grouping within one and the same business component. With this approach, individual business process structures of the direct bank case are analyzed in detail. The weights result from feedback received while using the BCI-3D method and from the detailed analysis of the actual business domain. In the actual case, the weighted relationships were defined as follows:
Weights for relationships between information objects:
  i. related-to: 100
  ii. part-of: 1000
  iii. state-of: 1000

Weights for relationships between business tasks:
  i. standard: 1000
  ii. optional: 100

Weights for relationships between business tasks and information objects:
  i. create: 1000
  ii. use: 100

As a result, the main focus of the business component identification will be on minimal communication efforts between different *standard* tasks, ensuring loosely coupling of business components, and on tight cohesion, ensuring optimal grouping of related business functionality. In addition, each information object, which is created by a specific business task, is placed within the same business component as the business tasks itself.

Fig. 7 illustrates the identified component model.

As a result, the main focus of the business component identification will be on minimal communication efforts between different *standard* tasks, ensuring loosely coupling of business components, and on tight cohesion, ensuring optimal grouping of related business functionality. In addition, each information object, which is created by a specific business task, is placed within the same business component as the business tasks itself.

Fig. 7 illustrates the identified component model.

Altogether, 21 business components were identified. Fig. 7 illustrates the single component “service component” on the left side and the complete component model on the right side. All BPS are placed ‘on the ground’ of the component model, whereas the information objects are situated above, ‘on top’ of the component model. The mentioned relationships between BPS and information objects are represented by connectors. It can be seen, that the main communication of the single component is inside the component. The complete component model on the right side strengthens this statement. The communication efforts between the single components are minimized.

The main components identified are: archive component, service component and master data component. The biggest component on the right side obviously shows more external rather than internal communication. This indicates that the information objects of this component are created by business tasks located in other components. This observation could be seen as a contradiction of the main focus, which is less communication between the business components within the aforementioned relationships. However, a closer examination of the functionality provided by this business component (i.e. the archive component), provides us with
a clearer understanding of this interaction. This is to say, when protocols, forms and customer requests are demanded, they are stored and offered as service in the archive component. That way, the identified communication structure is efficient. Information objects, which are needed for executing a task within a component, are created in the service component, not in the archiving component. The other two sizeable components provide the main data concerning customer master data or the account master data with the main tasks fitting customer creation and account opening.

With the structure of this component model, the characteristics of the customer services domain become obvious. There are independent customer requests for services. In general, a customer requests banking products and services step by step. According to this perspective, the identified business components generally contain one service with all the relevant tasks to fulfill each of these steps. Furthermore, where possible these components are defined as discreet and independent. Fig. 8 illustrates an extract of the component model for the customer process customer attracts customer, following the UML 2.0 notation [28] presenting the components with their required and provided services.

In reality, the customer service process customer attracts customer is mainly manually implemented. As a result, this process provides a lot of effort and produces costs and errors. Based on the domain analysis, the BCI-3D tool suggests four components for handling this process. The business component customer attracts customer provides all tasks for recording the service process. To check the correctness of a request to this component, the service get sales prospect information is necessary to import all related sales prospect information about the prospective customer and the new customer. If the request fits the business rules, the service offer confirmation customer attracts customer is provided in the component model. At the same time, a letter of thanks to the attracting customer is prepared. Therefore, the actual customer address for dispatch is provided in the component model by means of the service get actual customer address.

The second business component deals with customer requests to change the customer gift, which will be sent to new customers. Therefore the actual shipping list of the customer gift has to be requested in the component model via get shipping list presents. If the customer gift is not yet foreseen in the actual shipping list, a change of the customer gift is still possible and a confirmation of the new gift is provided in the component model by the service offer confirmation customer attracts customer. At the same time, this component provides the general service of checking a customer gift in the actual shipping list via offer present not in actual shipping list. This service is necessary to fulfill the tasks of the third component. This component prepares customer claims.

With the mentioned service of the second component, it is possible to check if an unsent customer gift is foreseen in the actual shipping list. The business process determines the validity of a shipping list up-to one week. Therefore, the following shipping list should also be checked. The shipping list is demanded by the forth component via get shipping list present. If the customer gift is not included in any shipping list, the service offer error present is provided in the component model to business component 1 to restart the recoding process. If there is a
Fig. 9. Component model – business components customer attracts customer

problem with the quality of the customer present, the service offer error handling service provider sends the necessary information directly to the service provider.

The forth business component presents the main interface to the service provider. This component maintains the shipping lists. If a customer present is chosen in any of the other components, this information is included via the service get confirmation customer attracts customer. Consequently, this component also offers the actual shipping lists to the component model via offer shipping list presents. Each week the customer gifts are demanded by the service provider via the service offer interface service provider. If a customer wishes to get a bonus instead of a gift, this component also prepares the accounting of the bonus. Therefore, the actual customer account number has to be demanded in the component model via get account number.

The extract of the business components customer attracts customer gives an initial insight into automating business processes. All tasks except the customer claim can be implemented and automated, at least for customer requests via the Internet. The identified components provide not only services to fulfill the customer service process, but these services easily can be reused by other components in other business domains. With such an IT-architecture, a focus on the important customer request – the claiming requests – is possible without losing quality for the other task. In contrast, the automation of the mentioned tasks provides faulty insertion and correction in the actual systems. A customer can check immediately and independently whether a change of his gift is still possible. Additionally he could track the proceeding of this service process.
5. Evaluation and Conclusions

The initial purpose for writing this paper was a six-month project at one of the five leading direct banks in Germany. The challenge was to reduce costs by automating data warehouse processes and to increase the quality of customer services at the same time. Therefore the domain customer services needed to be analyzed in detail. In order to use a methodological approach, the business components modeling process (BCM) was applied parallel to the normal project work, focusing mainly on the first two sub-phases, the domain scope and the business component identification. The methodology was used for 10 of the mentioned 27 customer services with the objective of developing a service-oriented approach for an alternative implementation of the supporting IT-landscape.

Advantages of the use of the BCM methodology:

Applying the domain scope sub-phase to the example domain meant to analyze all business tasks and the relevant information objects in detail, allowing a more precise definition of the process costs. New service processes, which led to further margin calculation costs, were identified. Additionally, an overall service process documentation was created which led to a deep acceptance and support of the general proceeding of the BCM methodology in the business unit.

With the business components identification method, based on the resulted domain models, an automated approach was provided for the identification of business components. The metrics defined in this paper – being minimal communication between and maximum compactness of business components – are the basic metrics for the component-based development of supporting information systems. Based on the well-defined business task dependencies, business functions could be clustered within business components. The identified components have a potential of reuse due to the provided services. In consequence, redundant activities in the business process as well as technical redundancy in the data warehouse functionalities were identified. This fact led to direct cost reductions in the services processes and to indirect cost reductions concerning a reduced error rate and reduced maintaining costs in the data warehouse.

Furthermore, a huge potential, especially for the market of direct banks, arises on the basis of process automation. Thereby a concentration on core competencies can be achieved without losing quality in the existing customer services. In the example service process customer attracts customer 80% of the process costs occurred for activities with automating potential, only 20% for activities concerning the customer claim. The result of the BCI methodology is therefore used to calculate the benefit of a service oriented architecture on the base of the customer service processes due to process automation potential. In consequence, an individual business case for the implementation of a service-oriented architecture can be calculated based on a methodical approach. In case of a positive business case, the identified component model represents the complete conception at the same time.

Difficulties by the usage of the BCM methodology:

Since the identification of business components is strongly dependent on the quality of the underlying business domain models, a strong focus was set on modeling the example domain. During that phase, difficulties occurred due to the way of modeling,
especially with regard to the allocation of information objects. In most cases, the information objects could only be gained from the guidelines that were delivered with the specific processes. Therefore, depending on individual policies one process-guide could be described more precisely than another. Due to these reasons, a consistent level of abstraction for modeling meaningful business processes was not given. By contrast, the relationships between the functions were documented adequately within the project case. Accordingly, when analyzing the component model, some components contain only highly abstract information objects, as for example *scoring product* that represents the score value of an individual product. Polishing these information objects, for example to include scoring per product, would improve the correlation of these highly aggregated information objects.

The examination of ten customer services showed, that guidelines and directives are not sufficient to exploit the complete power of the BCM method. In consequence, use of the BCI-3D tool only makes sense to a certain level of abstraction. Additionally, by analyzing more than only ten service processes even better reusable business components would result.

*Hints for further developments for the BCI tool:*

Concerning direct banking requirements, security aspects on data have to be considered. Actually it is not possible to integrate new, sector specific dimensions like *data security* to the BCI-3D method. But these aspects influence in an important way the identification of business components. If a role concept could be considered during the component identification a big manual task of post processing could be avoided.

Another actual weakness of the BCI-3D tool implementing the BCI-3D method is the lack of interfaces for example concerning data dictionary or metadata tools. In most direct banks, the data models are maintained. If the reuse of individual direct bank data domains would be possible, the quality of the identified business components would be higher. Another interesting point is the integration of automatic business logic checks via the data load in BCI-3D. If the tool would provide a meta model to rebuild business dependencies, logical errors in the domain analysis could be detected.

Problems regarding the use of the BCI-3D tool arise from filling the flat files that constitute the basis for the optimization method, which results in enormous manual work while mapping the information gained in the domain model to the representation of the BCI-3D tool. Certainly, the most comfortable way of optimizing that step would be the integration of the tools for modeling the business domain and the BCI-3D tool.

6. References


