The Process Map as an Instrument to Standardize Processes

Design and Application at a Financial Service Provider

Authors

Heinrich, Bernd
Henneberger, Matthias
Leist, Susanne
Zellner, Gregor

Abstract

The standardization of processes and the identification of shared business services in a service-oriented architecture (SOA) are currently widely discussed. Above all in practice, however, there still is a lack of appropriate instruments to support these tasks. In this paper an approach for a process map is introduced which allows for a systematic presentation – as complete as possible – of the processes in an enterprise (division). After a consistent refinement of the process has taken place by means of aggregation/disaggregation resp. generalization/specialization relations, it is possible to identify primarily functional similarities of the detailed sub-processes. The application of the process map at a financial service provider (FSP) highlights how these similarities can be taken as a basis to standardize processes and to identify shared services.

Keywords

Process map; business process management; financial service provider

1 Introduction and Problem Definition

Process models are widely used in practice: the analysis and the reorganization of processes are essential prerequisites to improving the efficiency of operational procedures. At present, the focus lies on reducing costs and on increasing flexibility in view of future requirements. Much discussed approaches to achieve these goals are process standardization or industrialization as well as the shared use of services within the framework of service-oriented architectures (Balgheim and Ollagnier 2005, p. 7, 21; Koch and Rill 2005, pp. 13-18; Bieberstein et al. 2005, p. 5; Kilian-Kehr et al. 2007). Concerning the first approach, financial service providers, for instance, expect to reduce the large number of process variants and to lower cost by completely or partially standardizing them. And, what is more, the communication of those involved in process modeling and execution is facilitated by using a standardized terminology. Alongside the latter the identification of similar or identical functions within an enterprise is the basis for developing standardized services. SOAs, in particular, undertake to split up consecutively the functionality of monolithic application systems (Papazoglou and Georgakopoulos 2003; Erl 2005; Winkler 2007, p. 258). If services are shared, it is generally assumed that not only the number of redundant services is reduced, but that the costs for their operation, administration and development go down as well. Apart from that service orientation is generally expected to provide higher flexibility in view of future requirements that can be implemented faster (since existing functionality can be accessed directly) (Koch and Rill 2005, pp. 13-18).

In practice, however, the said benefit potentials of SOAs can frequently not be actualized (as will be highlighted later on in the FSP case study), since the process models, the identification of business services is based on, do not exist in the required quality. Even though FSPs have modeled, analyzed and reorganized some of their processes in a couple of projects, this mostly happened without aligning them and, moreover, was done with different modeling languages and tools. Thus there is often lacking a consistent (and as complete as possible) representation of the enterprise's processes which would, however, be necessary to achieve the said goals.

Hence in this context important questions remain to be answered: Firstly there is the task of creating a systematic and complete (as can be) process documentation in an enterprise (division). Further on, the consistent refinement of the process models must be ensured – from the concise representation of the value chains via the detailed processes and sub-processes down to the very granular functions. And, in a third step, on the basis of the aforementioned postulations, the questions have to be answered how processes or sub-processes supposed to be standardized and how shared business services can be expediently identified. This paper takes on these questions and presents the concept of a process map.

To derive the answer to those questions, our research methodology follows the paradigm of design-science research as presented by Takeda et al. 1990 and Hevner et al. 2004. The goal of design-science research is to build and evaluate artifacts (e.g. constructs, models, methods), that are intended to meet identified business needs (see Hevner et al. 2004, p. 78-80). Following the design cycle (see Takeda et al. 1990, p. 43), the awareness of the problem is derived in section 2 and requirements for the suggested solution are defined. The development of the artifacts (structure of the process map) is explained in section 3. The evaluation of the suggested solution is described in section 4, where the application of a process map at a FSP is discussed and the benefits of the solution are outlined. Finally the results are summarized and critically reviewed in the conclusion.

2 Requirements for a Process Map

2.1 Definition of the Requirements

As described in the introduction the process map is destined both to standardize processes and to identify identical or similar functions as the basis to define shared business services.

The first point relates to the fact that at present, FSPs, for instance, have to cope with a large number of different process variants: i.e. depending on the access channel, the customer group, the executing organization unit, etc., very often different processes are implemented for the same or very comparable products. Even though, for instance in individual channels, a partly different customer interaction is required (e.g. for the customer legitimation), differences in the subsequent processing – apart from few exceptions – of product completions or contract alterations can scarcely be justified. Reasons for the large number of variants are historically grown channels and varying organizational responsibilities (e.g. for customer groups). Besides process design was often made without the knowledge or without the (conscious) harmonization of the existing procedures, since the harmonization would have tied many resources due to incomplete or abstract process documentations. This highlights the importance of providing consistent process documentation both for the standardization of existing processes and for the design of future processes. Processes – proven and tested as well as optimized – can then be used as so-called blueprints in projects in order to avoid to "produce" new variants again.

The second point refers to the development of a SOA, in particular, to the identification of business services. SOA is an architectural approach where applications are provided by self-describing, self-contained and network accessible services encapsulating (business) functionality. We refer e.g. to Erl (2005) or Papazoglou and Georgakopoulos (2003) for an in depth analysis. Currently many papers are published that advertise the benefits of SOAs. Often reusability of services is mentioned as an important advantage. Reusability in general refers to the possibility to share existing services and is merely seen as a side benefit enabled by standardized interfaces and enterprise-wide service registries (Erl 2005, p. 47). In this context it is interesting that "approx. 80% of the services are used only once or twice and only approx. 5% more than 20 times" (quoted in Lochmaier 2006, p. 71). If a SOA, as stated in the same paper, is assumed to form a "standardized interface between processes and IT functions", the question arises how a process map to identify shared services can be systematically made use of. The idea here is not primarily to identify services that can afterwards be reused but to identify similar business functions pointing at candidates for standardized, shared services. We speak of candidates, because a process map solely covers business aspects. Technological restrictions stemming for instance from the underlying applications of course have to be considered as well for the decision whether to realize a shared service or not. But for the purpose of this paper we focus on strategic and process-related considerations and not on technical restrictions or implications. Abstracting from the underlying technology can be helpful to focus on the selected perspective(s) of enterprise modeling. Enterprise modeling concepts like Business Engineering (Österle 1995), SOM (Ferstl and Sinz 1997) or Memo (Frank 2002) for example, offer different perspectives (strategy, organization/processes and information system) that are related and connected closely to each other, but nevertheless are examined separately (see Frank 2002, p. 1260, Ferstl and Sinz 1997, p. 351).

In order to achieve both goals the process map should support first and foremost the identification of similar processes, sub-processes or functions. Similarity in this context is to be understood as the result of a comparison of two processes, sub-processes or functions on the model level. This, in turn, implies that the models contemplated are comparable by means of at least one identical criterion. In accordance with the purpose of this paper, structural and functional similarities are differentiated (see Becker 1995, p. 133-150; Rosemann 1996, pp. 172-174, Loos 1996, Fettke and Loos 2005, p. 90). Structural similarities (structural analogies) refer to conformances of the model structure (internal view) (Rosemann 1996, p. 173). In process models the conformity is checked with respect to the utilization of identical model elements (regarding semantics and/or syntax) – e.g. specific functions, events, connectors (see

Becker et al. 2000, p. 51; Becker 1995, p. 141-142). In contrast, functional similarities are given, if two processes conform regarding their functions, even if they have different model elements (external view) (Rosemann 1996, p. 173). Hence processes are perceived as functionally similar, when they offer alternative procedures but deliver equivalent results.

In this examination both structural and functional similarities are relevant. Functional similarities need to be considered, because the equivalence of the result is essential for process standardization or shared use. Consequently the aforementioned requirement can be specified to the effect that functional similarities can be identified with the help of the process map. On the other hand, structural similarities have to be paid attention to when designing a process map – this will be highlighted later on.

For some time now, the identification of structural similarities has been supported by approaches regarding schema integration (see e.g. Batini et al. 1986) or by designing joint meta-models (see e.g. Karagiannis and Kühn 2002; Kühn et al. 2003). The two latter approaches, in particular, concentrate on merging and standardizing (process) models which were created with different languages resp. with different meta-model elements. As a rule the identification of similarities does not take place on the basis of functional criteria, but by means of using the same structures. It is fairly difficult to find starting points for standardizing different (inefficient) process variants by exclusively considering structural similarities. Thus the approach subsequently introduced backs on the identification of functional similarities. Structural similarities on the other hand are particularly important for the design of the process map.

Another important requirement is the consistent and complete (as can be) process documentation. “Complete” in this context primarily means that all processes of the enterprise (division) are modeled from their initiation (e.g. by the end customer) to their completion (e.g. an interface to an external supplier). In the following this is referred to as an End-to-End (E2E) process representation. It offers the possibility to check the similarity of processes in the respective parts of the E2E representation (e.g. customer service).

Apart from the requirement to model all processes completely according to the E2E representation, there is as well the problem of a consistent refinement of the process representation, i.e. the creation of model levels consistent to each other with differing levels of detail. This is necessary, because an abstract modeling of processes, for instance in a value chain, provides only little (authentic) evidence for the standardization or the identification of shared services. To ensure the consistence of the process map it is therefore necessary to introduce unambiguous design principles which define the relations between the model levels:

- Aggregation/disaggregation relations: they describe the composition of a design result on one level derived from partial design results on a more detailed level (is-part-of-relation) (see vom Brocke 2003, p. 77).

- Generalization/specialization relations: the design result for the general attribute is created by abstracting the design results for the *instances* of the general attribute (is-a-relation) (see vom Brocke 2003, p. 77).

Hence an additional requirement is the specification of adequate aggregation/disaggregation relations as well as generalization/specialization relations in order to ensure the consistent refinement of the processes. The latter is necessary in order to avoid that processes, sub-processes or functions with a different level of detail are compared, since this impedes the finding of similarities considerably. In practice, this is problematic, because the level of detail is often not distinctly defined and frequently depends on the model designer. In summary the following requirements can be phrased:

1. The process map supports the *identification of functional similarities*; structural similarities are used for their structuring.

2. The map gives a largely complete documentation of the processes as an *E2E representation.*

3. The process map contains several model levels with *different levels of detail* (from abstract value chains to functions).

4. To ensure the consistency of the model levels *aggregation/disaggregation* as well as *generalization/specialization* relations have to be defined.

2.2 Related Work

Regarding process modeling the structural view (e.g. which functions does a process comprise?) and the behavioral view (in which order are the functions of a process executed?) have to be distinguished. Most of the papers published focus on the behavioral view which also becomes evident when looking at the numerous modeling languages that have been developed to that purpose (e.g. Event-Driven Process Chains). Regarding the problems that have been outlined in the beginning, the exclusive usage of these languages is not appropriate, since they normally do not define any principles for the realization of the four aforementioned requirements (for instance the aggregation/disaggregation relations). They consider predominantly the order in which functions are processed. The order, however, is not adequate when it comes to identifying functional similarities.

In the literature there are different terms which refer to modeling languages that focus on the structural view such as process architecture (Österle 1995, p. 61), process map (Braun and Winter 2005, p. 72), and process landscape (Gruhn and Wellen 2000, p. 297). The most important approaches with regard to the aim of this paper will be subsequently analyzed regarding the aforementioned requirements.

The emphasis of process modeling in Business Engineering according to Österle and Winter (2003) lies on specifying the functions and sub-processes which are indispensable in a value chain, and on representing the order in which they are performed. The main goal regarding the design is – according to Winter – the optimal organization of the added value (Winter 2003, p. 102). In this context the process architecture model describes the coactions of the value chains on the level of enterprises respectively divisions (processes and service relations). It aims at the most important enterprise processes (Österle 1995, p. 127) as well as at their interfaces to suppliers and customers (Winter 2003, p. 103). Even though the latter aim is related to the second requirement (E2E representation), neither aggregation/disaggregation and generalization/specialization relations have been defined for the development of the process architecture model nor have functional similarities or consistent refinements of the processes been dwelt upon.

Ferstl and Sinz (1997) have developed – within the framework of their SOM methodology – a language to model the process structure in the form of business objects (appropriate subsystems of the enterprise) that are linked by means of transactions (Ferstl and Sinz 1997, p. 13). Transactions represent communication channels to exchange services between business objects. In the so-called interaction schema (IAS) business objects and transactions between them and the enterprises' environment are specified and refined by decomposing them consecutively. Apart from that it is defined how the exchange of services is coordinated (by negotiation or instruction). Thus, in the SOM methodology the structural view on a process – according to Ferstl and Sinz – is constituted by the exchange of services between business objects and their coordination (Ferstl and Sinz 1997). In the methodology principles (the so-called decomposition rules) are laid down to recursively refine an IAS over several levels of detail. Decomposition rules in fact support disaggregation and specialization. However, only the coordination and the service exchange is refined by decomposing transactions (into initiating, contracting, enforcing transactions according to the negotiation principle or into control and feedback transactions according to the feedback control principle) and business objects (into sub-objects establishing a feedback control loop). Further criteria regarding the disaggregation and the specialization are not named. In principle, the SOM methodology does indeed allow an E2E representation; it is, however, not postulated. The systematic identification of functional similarities is not an explicit goal, either.

By using the term "process landscaping" Gruhn and Wellen introduce an approach that describes the relations between process models (Gruhn and Wellen 2000a; Gruhn and Wellen 2000b). In doing so the authors proceed in two steps: in a first step "clusters" of process models are identified and their interfaces are specified. In a second step the clusters are refined and individual process models with a higher level of detail are derived. A concrete criterion as to how the clusters are composed is not named. The authors phrase hints ("logically closely connected process models" (Gruhn and Wellen 2000a, p. 299) and propose to interview those who are responsible for the processes. Apart from that the authors formalize terms which are used for checking syntax and semantics. According to their aim, these terms predominantly refer to the checking of the interfaces during decomposition (Gruhn and Wellen 2000a, pp. 305-308). Due to the lack of information on aggregation/disaggregation and generalization/specialization relations, in particular, the approach of Gruhn and Wellen does not reach out far enough as regards this paper.

A framework tailored to organize and systemize the management of the existing variants in reference models is the so-called process selection matrix (IDS 2000, pp. 159-160). It is part of the ARIS toolset and is used in a similar way in the SAP standard software to give a survey of reference processes and to show how they are selected in specific enterprises. In a matrix, scenarios are defined as columns opposed to the main processes in the lines. If the functionality represented by a main process is used in a scenario, it will be allocated to the scenario in the corresponding line of the matrix as a scenario specific process. SAP closely resembles this concept. In so-called SAP Solution Maps industry-specific or product-specific reference processes are structured along two dimensions. The dimensions may vary e.g. from industry to industry or from product to product. Buck-Emden and Zencke (2004, pp. 58-59) provide an example for a SAP Solution Map that is structured along communication channels and business functions. Communication channels and business functions are considered as key capabilities in this context. However, it is not described in this paper, how key capabilities – and thus dimensions – are identified or how SAP Solution Maps can be constructed in general. Therefore, similar to process selection matrices, the concept of a SAP Solution Map is mainly constituted by its two-dimensional structure. Even though, in principle, the allocation of processes into a two-dimensional structure supports the identification of functional similarities, as we will see later on, neither concrete criteria for the differentiation of scenarios, main processes or key capabilities are provided nor is the purpose of an E2E representation focused. Several model levels are indeed provided but they are not generated by means of aggregation/disaggregation or generalization/specialization relations. In lieu thereof the scenario-specific processes in a process selection matrix are, in general, directly described more closely by means of Event driven Process Chains (IDS 2000, p. 159). In the case of a SAP Solution Map, in general, natural language is used to describe the more detailed levels (see SAP 2007).

With reference to the process selection matrix the so-called process object selection matrix was developed by Schütte (Schütte 1998, pp. 220-231). In this matrix, process and configuration objects of a configurable reference model are contrasted. Process objects contain knowledge of those functions which are valid irrespective of individual contexts (Schütte 1998, p. 226). Examples for such generic process objects are the sales order or the supplier's purchase order. Either of them can be specialized in different process variants. Configuration objects represent knowledge of the process object sequence within the process variants. The configuration objects convey how the process objects behave within an individual application context (Schütte 1998, p. 7). For Schütte, the most important argument in favor of the process object selection matrix (and to the disadvantage of the process selection matrix) is the object-related process definition. From his point of view this corresponds to the natural conceptualization in enterprises where processes are defined by means of objects (Schütte 1998, p. 231). The small number of object-related process definitions in practice is, however, contradictory to this argument. Moreover, it is hardly possible to ensure a largely complete process representation (second requirement) by means of the exclusive consideration of objects. And, what is more, concrete aggregation/disaggregation and generalization/specialization relations are not provided.

An approach which focuses the last-mentioned relations is the Process Handbook developed by Malone et al. (1999). The latter uses the so-called Process Compass which differentiates between two criteria ("horizontal vs. vertical navigation") for the representation of the processes. Processes are horizontally specialized by means of objects. Vertically, however, the processes are disaggregated into sub-processes (Malone et al. 1999, p. 428). With this compass which allows for the navigation in four directions (aggregation/disaggregation and generalization/specialization) the authors generate a database (Process Handbook). It is destined to simplify the access to existing processes and, in particular, to make apparent similarities between sub-processes. To that purpose Malone et al. identify ten generic activities (create, modify, use, move, preserve, destroy, manage, separate, combine, decide) which they see as the comprehensive basis of all processes (Malone et al. 1999, p. 436). The structure of the Process Handbook is then aligned on the highest level on the basis of the generic activities. Afterwards all activities are decomposed further down into the components "what" and "how". The Process Compass is, above all, meant to fulfill the aforementioned requirement to identify functional similarities. Sub-processes show functional similarities, if they can be allocated to identical generic activities reflecting the perceptible "function" of the sub-processes. Apart from that different modeling levels are included and, as a rule, generalization/specialization as well as aggregation/disaggregation relations are differentiated. A specialization does, however, merely take place by way of the question "how", i.e. concrete construction criteria are missing. Besides, this approach does not aim for a complete description according to the E2E representation (compare second requirement).

Besides the discussed process modeling approaches, there are a number of approaches that specifically aim at the identification and the design of services in a SOA (Papazoglou and van den Heuvel 2006; Erl 2005; Winkler 2007; Bieberstein et al. 2005, p. 96). In general they propose a top-down approach where process models constitute the starting point of service identification. Winkler (2007) for instance describes the usage of activity diagrams that are decomposed over several modeling levels until individual functions are revealed that can be realized as a service. However, it would be senseless to counter these approaches with our requirements. Their main artifacts are the services that should be identified or designed. In contrast in our approach the artifact is the process map which can then be used to identify functional similarities and thus to identify standardization potential and candidates for shared services. Accordingly, a process map can assist in service design and identification. It is not a service identification and design methodology in itself. Instead it can be regarded as complementary to the aforementioned approaches.

3 Development of a Process Map

On the basis of the aforementioned approaches the concept of a process map will be developed hereafter starting with the design of its first model level. As mentioned before, the technological perspective is not part of our approach.

Functional similarities of two processes, sub-processes or functions gear to the equivalence of their solutions, i.e. the sub-processes are supposed to show an identical or a similar input-output behavior (two sub-processes each generate an identical or a similar output on the basis of an identical or a similar input), even though alternative procedures may be used to transform the input into the output.

In conjunction with the requirement to model the processes as completely as possible this would mean to compare the results of E2E processes (for instance the results of *distribute and process credit product* vs. *distribute and process investment product*). As regards an analysis of the standardization potential, proceeding like that is, however, too abstract. Hence it does make more sense to subdivide (disaggregate) E2E processes and then compare the sub-processes which have been structured according to consistent criteria. In this context the generic activities (Malone et al. 1999) respectively, to a certain extent, the generic transactions (Ferstl and Sinz 1997) can have a supportive function. With their help the E2E representation can be subdivided into generic sub-processes (e.g. *enter credit application*). Regarding these generic sub-processes, functional similarities can then be examined a lot more precisely (e.g. comparing *enter credit application* vs. *enter investment product application*). This comprises, above all, the comparison of input and output objects (e.g. *credit application* vs. *investment product application*) of the sub-processes that have now been sufficiently detailed. The comparison can be carried out (manually) by domain experts. Moreover, in case descriptions of concrete sub-processes are available containing the specification of input and output objects resp. pre- and post-conditions, an automated comparison with an adequate tool support is conceivable (e.g. when UML activity diagrams are used for process modeling these prerequisites could be specified by means of the Object Constraint Language). In this context the application of semantic web standards and ontologies appears to be particularly interesting. They permit a semantically more comprehensive analysis, for instance by using equivalence or sub-class-of relations. Heinrich et al. (2007) describe their application for the analysis of process components in-depth, however, with a different aim than the one in this paper. Nonetheless the formalization introduced by Heinrich et al. can help to (automatically) identify functional similarities in a comprehensible way by means of the comparison of input and output objects.

To subdivide real E2E processes into sub-processes and to allocate them to the generic sub-processes structural similarities can be made use of: sub-processes which match regarding (individual) model elements or the relations between them can be allocated to a generic sub-process both quickly and consistently. As a result this helps to “fill” the process map with real processes. Hence the generic sub-processes of the E2E representation are used as the first dimension to structure the process map (see Fig. 1). If we assume that the process map’s graphic format is two-dimensional for reasons of readability and manageability, the question arises what the second dimension is like. As laid out before, the reason for standardizing efforts is the existence of a high number of process variants which offer equivalent, and hence redundant, solutions for e.g. different products, access channels etc. To specify precisely the differences it is necessary to specialize according to these process objects in the second dimension. In principle process objects could be both specialized (e.g. all products → groups of product → individual products) and disaggregated (e.g. products → service features, brand and price). However, we suggest a specialization as it is easier to trace functional similarities (provided that the generic sub-processes serve as the first dimension), if different types of the generic sub-processes (e.g. for individual products) are compared (see Fig. 1). Here the hitherto existing approaches – as already discussed in section 2.2 – cannot (or only on a small scale) be used, since they do not explicitly define (functional) generalization/specialization relations for the process objects. This is may be due to the fact that a certain number of possible process objects can be considered as the second dimension of the process map. Possible process objects are fields of demand/product groups, customers/groups of customers or access channels. A selection can only be accounted for with regard to the concrete purpose and context of application (as we will see in section 4).

Even though it is not possible to define a universal rule for the selection, the experience made for instance by FSPs highlights that the utilization of access channels or customer groups as the second dimension on the *first* model level does not lead to the intended aim. In general generic sub-processes which would be adequate to identify standardization potential cannot be generated. For instance the comparison of how *consulting* takes place in the *in-house* vs. the *mobile sales* (different access channels) without referring to individual products is of little significance for the first model level (and later on for the second level as well). Differences only become apparent when considering specific products or at least groups of products.

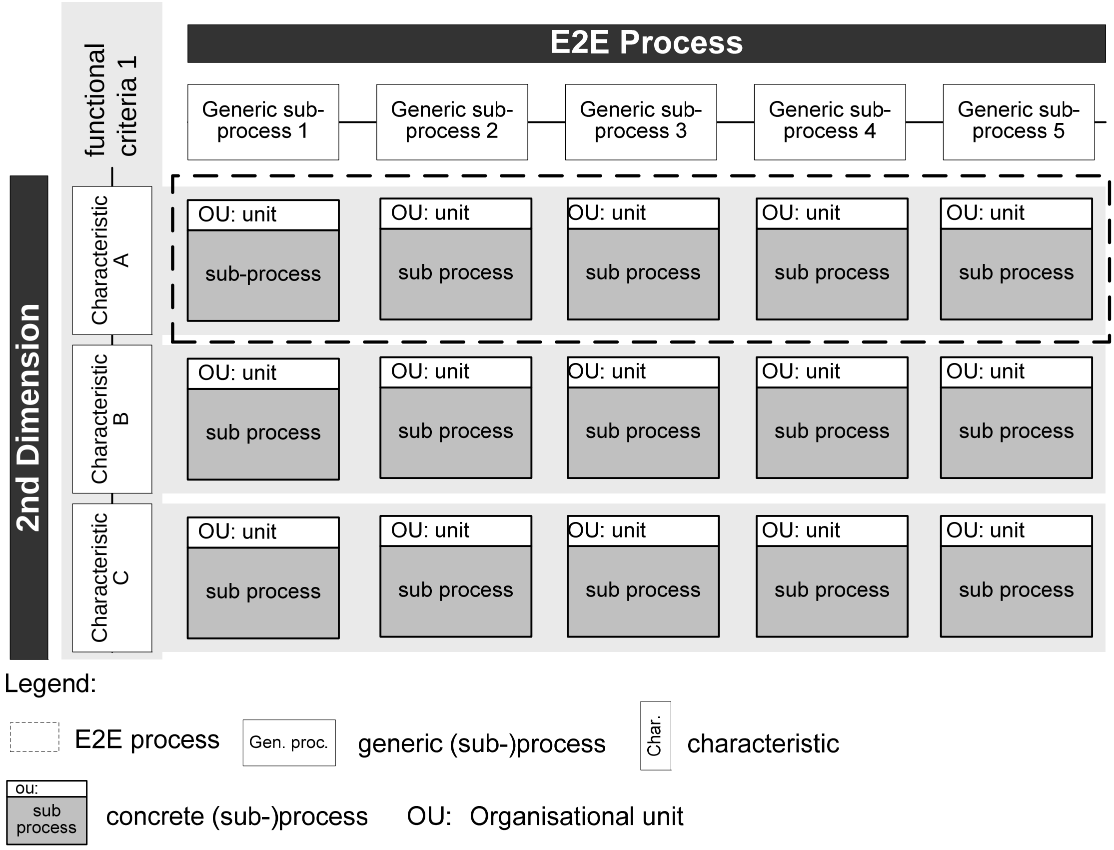

Fig. 1: Simplified framework for the first model level of the process map

After developing the first model level the question arises how to refine consistently processes on further levels (third requirement). As already mentioned, the aggregation/disaggregation resp. the generalization/specialization relations can be utilized. This means, that the first dimension of the generic sub-processes of the E2E representation is disaggregated further down. Likewise the second dimension is specialized, i.e. if, for instance, product groups are used here, a specialization into the individual products takes place on the next level (see Fig. 2). By means of this refinement differences and common features in the processes are made more transparent. In doing so it is fundamental to ensure that the disaggregation and the specialization respectively are carried out both disjunctively and completely. This alone will guarantee that the processes are represented as completely as possible in the first place and that they can secondly be unambiguously allocated to the generic sub-processes.

A further disaggregation and specialization of the dimensions, and thus of the processes, on further levels can make perfect sense depending on the application context (see section 4). Ultimately this is an appreciation of the time and effort for the modeling and the aim that is pursued. Especially, if similarities or differences are supposed to be unambiguously identifiable in order to create a suitable basis for deciding about standardization, a high level of detail appears to be necessary.

If functionally similar sub-processes and functions are identified, a colored marking is made in the process map (see Fig. 2). On the one hand functionally similar sub-processes can point to standardization potential. This is the case, if there are differences regarding the way to proceed which are, however, not factually necessary. Functional similarities in sub-processes can be, on the other hand, a starting point for shared business services in order to avoid redundant implementations. Again candidates for business services are especially functions that can be delivered in a standardized way, for instance because they are regarded as "commodities" within an enterprise or within an industry. These issues will be dwelt on in section 4.

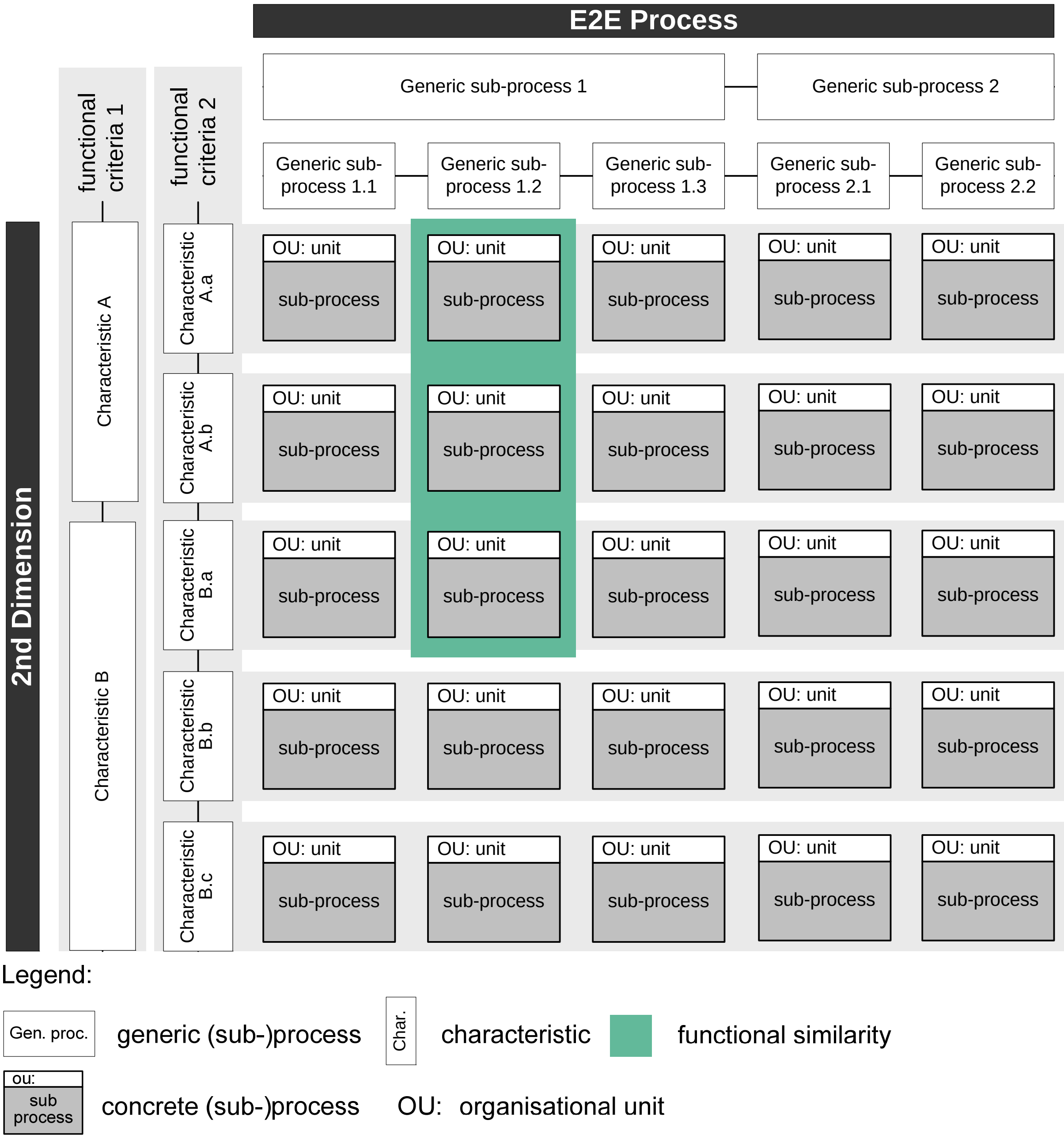

Fig. 2 Simplified framework for the second model level of the process map

4 Practical Application at a Financial Service Provider

4.1 Goals of the project

The described procedure how to design a process map was applied during a project at a large FSP. The intention was to systemize the processes in retail banking in a process map in order to standardize on the one hand cost-intensive process variants (as far as economically reasonable) and to identify on the other hand shared business services.

The FSP – as it is generally the case in the banking sector – has a huge number of different process variants and a highly complex and historically grown IT landscape. A multitude of diverse products and product variants are mainly distributed via a widely ramified network of branch offices, but also via other access channels as for instance call centers, Internet or mobile sales activities.

Even though there existed detailed process representations at the FSP – check lists as well as Event driven Process Chains – that served as operating instructions, they normally only mapped the very excerpt of the entire process that was relevant to the organizational unit. By means of a sequential alignment of the process representations it was possible to create an E2E representation for some of the processes. Since, however, the individual representations had been compiled and revised by different members of the staff over time, the level of detail, the terms used as well as the focus of the documentations' contents varied. And, what was more: it showed clearly that the originators lacked the overall knowledge and thus the understanding of the entire process.

4.2 Development of the Process Map

The first task when designing the process map was to develop an E2E representation for the processes in retail-banking that had so far not existed. To this end it was necessary to identify generic sub-processes that were suited to structure the processes – irrespective of the FSP's products, access channels, customer groups, etc. On the basis of an evaluation of the so far existing documentation (operating instructions) propositions were devised. These were discussed and finalized in workshops with software project coordinators as well as with the staff responsible for the operating instructions. In this context the following generic sub-processes on the first level of the process map were defined: *analyze customer demand, product-specific consultation, close contract/order, handle contract/transaction* and *book transaction* (for explanations see further down). As turned out later on, 90% of the FSPs retail banking processes could be decomposed according to these generic sub-processes. When doing this decomposition it was irrelevant in which organizational units these sub-processes were (currently) executed. In the remaining 10% of the retail banking processes some generic sub-processes were missing, since for instance a product-specific consultation did not take place.

Concerning the second dimension of the process map several different process objects turned out to be eligible for the first model level:

- *Product groups:* For each product there usually exist one or several E2E processes (for instance for different access channels). Thus product groups can be used to differentiate and classify the processes on the first level of the process map. A basic raster to group products on the first model level constitutes *fields of demand*. On the other levels then a specialization into individual products (level 2) and further on into product variants (level 3) can take place.
- *Customer groups:* Another possibility is to classify the E2E processes according to customer groups. Presently in retail banking customer groups are classified above all according to their asset. Process variants for different customer groups primarily exist in the area of investment products where specificities, for instance regarding the consultation had to be paid attention to.
- *Access channels:* In sales, in particular, processes for the same products partly differ considerably depending on the type of access channel or the organizational assignment.
- *Business transactions of the product life cycle:* A classification of the processes according to the product life cycle is possible, too. In almost all of the cases it is possible to structure life cycles logically starting with the closing of the contract (e.g. the opening of a deposit) via contract modifications (e.g. modification of the deposit conditions) up to the notice of cancellation (e.g. the cancellation of the deposit). Each of these business transactions can be represented as an E2E process. However, it is only reasonable to classify according to the business transactions, if a differentiation into product groups has been made before. Hence the usage of this dimension for the first level of the process map was out of the question.

The process object *product group* was defined as the second dimension of the process map. On the one hand this appeared to make sense, since in the workshops the process differences between products were rated to be enormous and an analysis of the processes for different products (above all on the levels two and three) was considered to be promising in view of the identification of sub-processes supposed to be standardized. Besides it turned out in discussions with the business departments that the intuitive comprehension of the process object *product group* was much better than for instance that of access channels. Apart from that the other process objects appeared to be inadequate, because a representation abstracting from the product (group) on the first level only led to minor process differences in the case study.

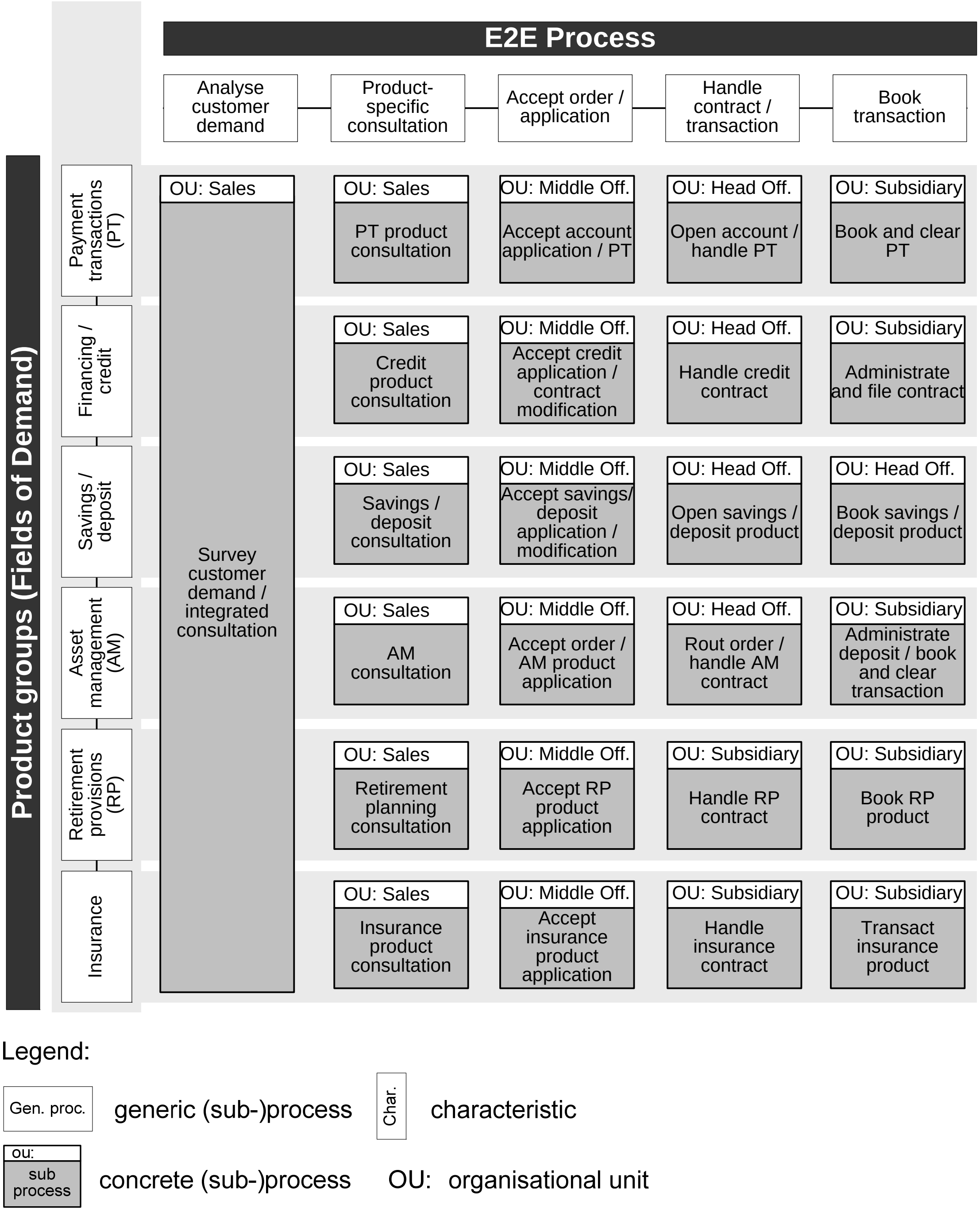

Fig. 3 Structure of the process map on the first level in the case study of the FSP

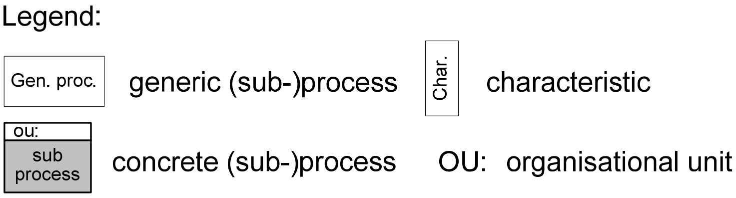

Fig. 3 highlights the first level of the developed process map. The generic sub-processes of the E2E representation in the horizontal and the product groups in the vertical set the raster for the allocation. For instance when specifying the generic sub-process *accept order/application* subject to the product group, this leads to the sub-processes *accept credit application/contract modification (*for the field *financing/credit*) and *accept order/AM product application* (for the field *asset management*). The sub-process *analyze customer demand* is the only one for which no specialization by means of the product group has been made. Here, in particular, it is intended to provide a consultation that is solution-oriented and spans the different product groups. Only at the end of the consultation the identification of adequate products for the customer is supposed to take place. The independence from the product group in fact serves as a criterion to delimitate the generic sub-process *analyze customer demand* from the generic sub-process *product-specific consultation*. Additionally, there is customer contact (as another criterion of delimitation) up to the sub-process *accept order/application,* whereas there is none as from the sub-process *handle contract/transaction*.

As can be derived from the diagram, the first level – due to its high degree of abstraction – naturally only serves as a means of orientation and as a “navigation structure” for the other levels. Apart from that, as regards the FSP, the organizational units which are responsible for the execution of a sub-process have been listed.

Level two could be developed through the disaggregation of the generic sub-processes and the specialization of the fields of demand. For instance, the sub-process *handle contract/transaction* was decomposed into *check contract/transaction*, *process contract/transaction*, *handout to customer* (see

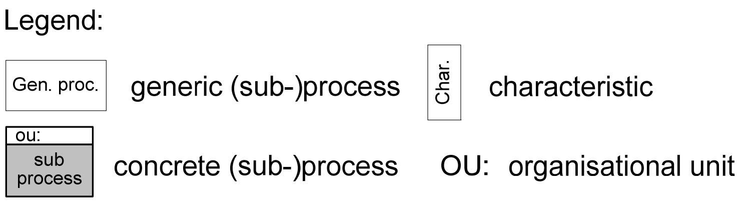

Fig. 4). Through this the first differences regarding the administration of the portfolio, i.e. of certain credit products, were made transparent. Besides, on the second level, the fields of demand regarding the individual products of the FSP were specialized (as an example the field of demand *financing/credit* and the product *overdraft loan* have been illustrated in

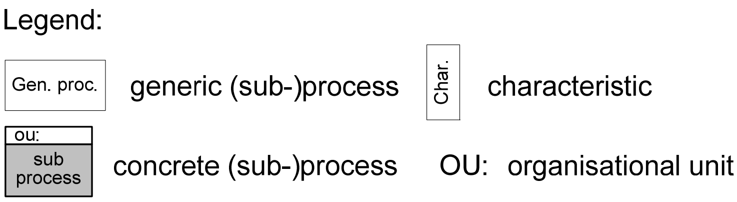

Fig. 4).

Although the generic sub-processes on level two still were quite abstract, partly very fundamental differences for the E2E processes could be derived. In order to work out these differences more clearly, it was agreed with the specialist staff to include as well the process object *business transaction of the product life cycle* on level two. Therefore for each product the business transactions *product opening*, *product modification* and product *cancellation/liquidation* were as well differentiated. Moreover the process costs (not part of the diagram for reasons of confidentiality) – where known – were added to allow for a quantitative analysis.

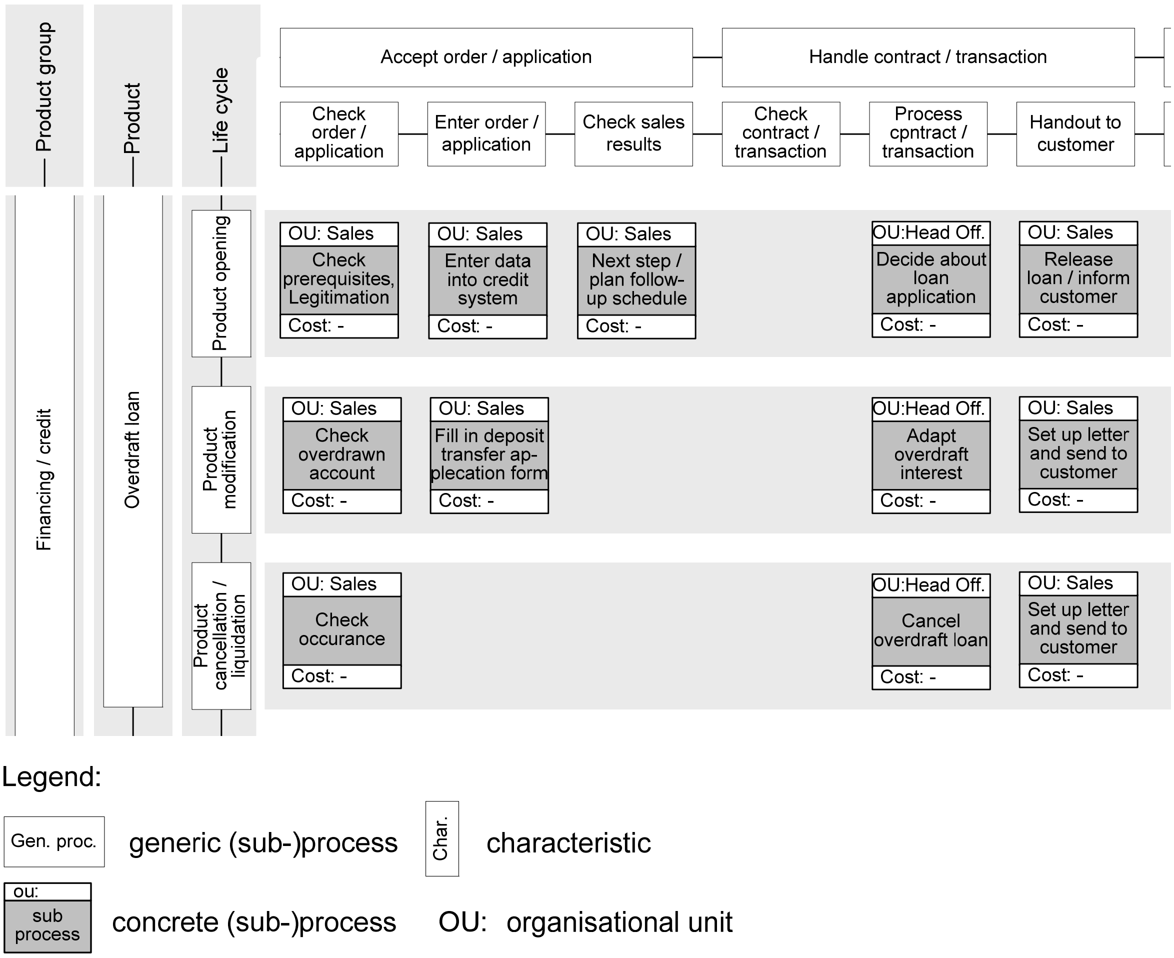

Fig. 4: Excerpt of the model level two of the process map in the example

The identification and the allocation of processes on the second and the third levels were carried out in workshops with the FSP's staff and by means of evaluating the existing operating instructions, In doing so, approx. 600 E2E processes could be worked out for the products on level two of the process map.

Level three was constructed on the lines of level two, i.e. the generic sub-processes were further disaggregated. Since the individual products could not be further specialized, the process object *access channel* was taken into focus. For each business transaction of a product it was examined in which access channel it can be carried out (e.g. a deposit can be opened via Internet, whereas this is not possible for an overdraft loan) and how the E2E process goes on. Together with the specialist staff the aforementioned differences could thus be discussed in-depth in order to sort out which processes could be simplified and standardized.

4.3 Practical Benefit of the Process map

Before the background of the set aims the benefit of the process map will be subsequently highlighted:

1. Standardization and simplification of process variants

The reasons for the different process variants at the FSP were, above all, the isolated development of processes, optimization measures per division, intransparent interfaces and a lack of possibilities to analyze and compare comprehensive E2E processes. In this context, the process map was particularly helpful in two respects: On the one hand it offered a consistent basis for communication and an orientation guide for the exchange of knowledge "about processes". On the other hand it allowed a systematic analysis of process variants. In particular, the following questions could be answered: Which process differences are there for one product regarding for instance different access channels and why do these differences exist? Which functional similarities can be observed in the processes of different products?

Whereas understandable reasons existed for certain process differences for identical or similar products (e.g. brochures have to be handed out when equity funds are sold, while this is not necessary - according to stock corporation law - when stocks are purchased), in other cases the reasons for the existing differences could just not be identified. For instance, there were different order entry systems for a stock order, an order of equity funds at capital investment companies, and for the application for shares. This offered the possibility of standardization. Consequently, the different order entry systems were harmonized by the FSP (after the profitability had been calculated).

By documenting the executing units the organizational interfaces of a process could be identified. Hence it was, for instance, possible to reduce the number of manual interfaces in the *deposit transfer* E2E process. The sub-process *enter order* and parts of the sub-process *handle transaction* were adapted according to the stock order processes. In the past, paper forms were filled in for deposit transfers in the branch office and were then passed on via manual interfaces like interoffice mail or by fax to the responsible central division for being registered there (see Fig. 5). After the system adaptation the orders could be – in line with the stock orders – put into the order entry system to be processed automatically. Thus the lead time and the process costs could be reduced considerably. This example highlights an advantage of the process map, since it is not only possible to compare the processes for product variants, but also to compare the processes of quite different products and services.

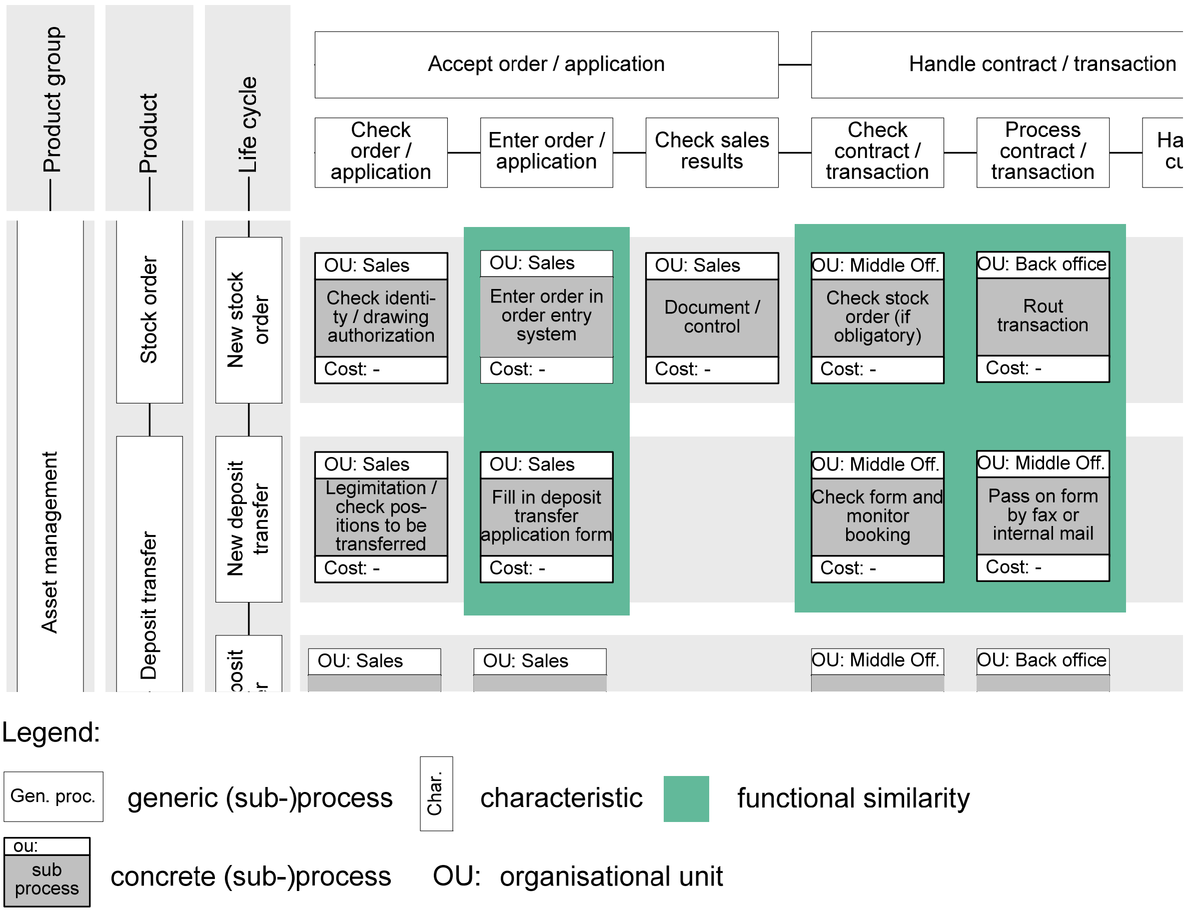

Fig. 5 Comparison of the processes stock order and deposit transfer

Another reason for the existence of different process variants are access channels which were regarded on the third level of the process map. For instance, certain routine checks were carried out either in the middle office or in the branch office depending on whether an order (e.g. the cancellation of a stock deposit) had been handed in personally or by mail by the customer. This practice resulted in cost-intensive variants with individual interfaces. Moreover, the development of the internet channel during the late 1990ies led to an increase in process and system complexity, since the internet processes often only converged with the processes in the branch offices at order execution or even not until orders were booked (e.g. in the corresponding trade systems).

In order to be able to contain the process diversity in future, the processes in the map are supposed to represent so-called blue prints for new projects. Due to the strict organizational separation it had until then often been common practice to design new processes for many product groups and divisions "in isolation", though similar processes partly already existed in other divisions. The map made it possible to accelerate the identification of existing processes. The advantage does not only lie with a faster, often better process design. At the same time the number of (sometimes more expensive) process variants is frequently reduced: thus a renewed "uncontrolled growth" of processes is obviated. There was, for instance, the requirement that, in future, it should be possible for customers to pay cash into their account at the office branches outside of the banking hours. In this case, however, the account had to be credited on the same day – in contrast to conventional night safe containers. With the help of the process map it was possible to identify three processes in retail banking by means of which the requirements could essentially be realized. They could be stored simultaneously with cost and time quantities from retail banking in order to facilitate the discussion and the selection of the adequate process without having to design another new complex "variant".

2. Identification of shared business services

In order to reduce the complexity of the application systems and to render possible shared functionalities, several pilot projects to introduce SOA had already been initiated at the FSP. A challenge lay in the identification of the business services. The wide-spread distribution of organizational responsibilities for the systems, which – from the historical point of view and due to the size of the IT landscape – made, of course, sense, presented an additional impediment to the efforts made. Here the process map can support the identification on the third model level. Candidates for shared services can be identified by checking the different processes allocated to a generic sub-process for identical or similar functionalities (for individual product variants and access channels). This was, for instance, carried out for the archiving of contract documents, which is regulated by law. Another example are procedures for mathematical optimization or numerical solutions which are required in several application systems for financing, credit and insurance products. They all are candidates for the definition of business services. Of course also technical aspects have to be considered. For instance individual functions may be part of a standard application system and thus it may be not reasonable to consolidate them in a shared service. Moreover cost/benefit-aspects of realizing shared services have to be verified. The documentation of the costs facilitates the calculation of the savings generated by the service. As regards the example the costs of a digital archiving service destined for the usage in multiple processes could be compared with costs for the present, partly manual archiving system.

5 Summary of the Results

In this paper the development of a process map has been presented. The aim was to detect possibilities to standardize processes as well as to identify shared business services. The practical application of the process map was exemplified in a case study. The essential results are:

- By consistently structuring the processes in an enterprise (division) with the help of criteria – generic activities of the E2E representation as well as product groups – the different processes can be compared to each other. It is thus not only possible to identify, in particular on the model levels with a higher degree of detail, functional similarities of the processes or functions, but also to determine differences regarding organizational responsibilities, IT support or quantities like costs or processing times. Once these differences have been made transparent, they can be put to test. In many cases processes can then be simplified, standardized in parts (also product-overlapping) and thus be "brought together".
- Apart from the standardization of existing processes the process map can also be employed in projects, e.g. if new products are introduced and thus "new processes" have to be designed. Frequently it turns out that new products can be sold, handled and booked by means of processes which are in parts identical with or similar to already existing ones. As regards the functional requirements and the handling of the product it is thus possible to detect adequate processes by means of the generic activities and to discuss them with the specialized staff.
- Last but not least the process map can support the identification of shared services, a subject which is presently discussed intensively before the background of SOAs. If functions in different processes are similar or identical, but are executed by different applications, it is necessary to check the possibility of standardizing the functionality. In this context two aspects are of importance: On the one hand services can be, above all, adequately identified, if the generic activities in the E2E representation on the third modeling level have been refined by disaggregation. On the other hand, the identification of services should be preceded by a process optimization, since otherwise it will come to an "electronification" of inefficient processes.

Two points should be watched critically which call, amongst others, for future research. On the one hand, it is necessary to improve the so far existing tool support for the maintenance and the further development of the processes in the map. An adequate tool support considerably influences the acceptance of the process map in an enterprise. Hence from a scientific point of view it should be taken into consideration to describe the terms used in the process map semantically and to store them in an ontology. On the other hand, the question must be answered, how the structure as discussed and defined in the example (especially as regards the generic activities) can be conferred to other FSPs and which adaptations are necessary to create a "reference process map". The theoretical conception that has been worked out does offer an adequate basis in this respect.

Literature

Balgheim, T., Ollagnier, J.-M. (2005) Redefining Core Banking – Worldwide Survey, Accenture – SAP 2005, Bankenstudie, Walldorf.

Batini, C., Lenzerini, M., Navathe, S.B. (1986). A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys, Vol. 18, Nr. 4, 323-364.

Becker, J. (1995) Strukturanalogien in Informationsmodellen - Ihre Definition, ihr Nutzen und ihr Einfluss auf die Bildung von Grundsätzen ordnungsmässiger Modellierung (GoM). In: König, W. (ed.): Wirtschaftsinformatik '95 - Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit, Physica-Verlag, Heidelberg, pp. 133-150.

Becker, J., Rosemann, M., von Uthmann, C. (2000) Guidelines of Business Process Modeling, in Aalst, W., Desel, J., Oberweis, A., (eds): Business Process Management, Models, Techniques, and Empirical Studies. Volume 1806 of Lecture Notes In Computer Science, Springer, pp. 30-49.

Bieberstein, N., Bose, S., Fiammante, M. (2005) Service-Oriented Architecture Compass - Business Value, Planning and Enterprise Roadmap, IBM Press.

Braun, C., Winter, R. (2005) A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform, Desel, J., Frank, U. (eds.): Enterprise Modeling and Information Systems Architectures, Proc. of the Workshop in Klagenfurt, GI-Edition Lecture Notes (LNI), Klagenfurt, 24.10.2005, Gesellschaft für Informatik, Bonn, p. 75.

Buck-Emden, R., Zencke, P. (2004) mySAP CRM – Kundenbezogene Geschäftsprozesse mit SAP CRM 4.0, SAP Press, Bonn.

Erl, Thomas (2005) Service-Oriented Architecture – Concepts, Technology, and Design, Prentice Hall, Boston.

Fettke, P., Loos, P. (2005) Zur Identifikation von Strukturanalogien in Datenmodellen - Ein Verfahren und seine Anwendung am Beispiel des Y-CIM-Referenzmodells von Scheer. WIRTSCHAFTSINFORMATIK 47 (2), pp. 89-100.

Ferstl, O.K., Sinz, E.J. (1997) Modeling of Business Systems Using the Semantic Object Model (SOM) - A Methodological Framework, in: Bernus, P., Blazewicz, J., Schmidt, G., and Shaw, M. (eds.) Handbook on Architectures of Information Systems. International Handbooks on Information Systems, Volume I, Springer, pp. 339-358.

Frank, U. (2002) Multi-Perspective Enterprise Modeling (MEMO): Conceptual Framework and Modeling Languages, in: Proceedings of the Hawaii International Conference on System Sciences (HICSS-35): Honolulu, pp. 1258-1267.

Gruhn V., Wellen, U. (2000a) Process Landscaping eine Methode zur Geschäftsprozessmodellierung. WIRTSCHAFTSINFORMATIK 42 (4), pp. 297 – 309.

Gruhn V., Wellen, U. (2000b) Structuring complex software processes by Process Landscaping, in: Reidar, C. (ed.), 7th European Workshop on Software Process Technology, EWSPT 2000. Kaprun, Austria, February 2000 (appeared as Lecture Notes in Computer Science No.1780), pp. 138–149.

Heinrich, B., Bewernik, M., Bolsinger, M., Henneberger, M., Krammer, A., Lautenbacher, F. (2007) Conditional Planning of Process Activities – Towards a Semantic-Based Approach. Working Paper, University of Augsburg.

Hevner, A., March, S., Park, J., Ram, S. (2004) Design Science in Information Systems Research, in: MIS Quaterly, 28 (1), pp. 75-105.

IDS Scheer AG (2000) ARIS Method, Version 5.0, 2000.

Karagiannis, D., Kühn, H. (2002) Metamodeling Platforms, in: Bauknecht, K.; Min Tjoa, A.; Quirchmayer, G. (eds.): Proceedings of the Third International Conference EC-Web 2002 – Dexa 2002, Aix-en-Provence, Frankreich, September 2002, pp. 182-196.

Koch, M., Rill, M. (2005) Serviceorientierte Architekturen bei Finanzdienstleistern, ibi research an der Universität Regensburg GmbH, 2005.

Kilian-Kehr, R., Terzidis, O., Voelz, D. (2007) Industrialization of the Software Sector, in: WIRTSCHAFTSINFORMATIK 49 Sonderheft, pp. 62-67.

Kühn, H.; Bayer, F.; Junginger, S.; Karagiannis, D. (2003) Enterprise Model Integration, in: Bauknecht, K.; Min Tjoa, A.; Quirchmayer, G. (eds.): Proceedings of the Fourth International Conference EC-Web 2003 – Dexa 2003, Prag, Tschechische Republik, September, pp. 379-392.

Lochmaier, L. (2006) Den großen Schritt wagen, die Bank 2, pp. 67-71.

Loos, P. (1996) Geschäftsprozeßadäquate Informationssystemadaption durch generische Strukturen. in: Vossen, G., Becker, J. (eds.): Geschäftsprozeßmodellierung und Workflow-Management - Modelle, Methoden, Werkzeuge, Bonn, pp. 163-175.

Malone, T. W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C. S., Bernstein, A., Herman, G., Klein, M., O'Donnell, E. (1999) Tools for Inventing Organizations: Toward a Handbook of Organizational Processes, Management Science 45 (3), pp. 425-443.

Österle, H. (1995) Business Engineering: Prozess- und Systementwicklung, Band 1 Entwicklungstechniken, 2nd edn, Springer, Berlin.

Österle, H., Winter, R. (2003) Business Engineering - Auf dem Weg zum Unternehmen des Informationszeitalters, Springer, Berlin.

Papazoglou, M. P., Georgakopoulos, D. (2003) Service-Oriented Computing, in: Communications of the ACM 46 (10), pp. 25-28.

Papazoglou, M. P., van den Heuvel, W.-J. (2006) Service-oriented design and development methodology, in: International Journal of Web Engineering and Technology 2 (4), pp. 412-442.

Rosemann, M. (1996) Komplexitätsmanagement in Prozessmodellen: methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden.

SAP (2007) SAP Business Maps: Solution Maps. Available online http://www.sap.com/solutions/businessmaps/solutionmaps/index.epx, accessed 2007-09-20.

Schütte, R. (1998) Grundsätze ordnungsmässiger Referenzmodellierung : Konstruktion konfigurations- und anpassungsorientierter Modell, Wiesbaden, Gabler.

Takeda, H., Veerkamp, P., Tomiyama, T., Yoshikawam, H. (1990) Modeling Design Processes, in: AI Magazine, 11 (4), pp. 37-48.

vom Brocke, J. (2003) Referenzmodellierung – Gestaltung und Verteilung von Konstruktionsprozessen, Logos Verlag, Berlin.

Winkler, V. (2007) Identifikation und Gestaltung von Services – Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich, in: WIRTSCHAFTSINFORMATIK 49 (4), pp. 257-266.

Winter, R. (2003) Modelle, Techniken und Werkzeuge im Business Engineering, in: Österle, H., Winter, R. (eds.): Business Engineering - Auf dem Weg zum Unternehmen des Informationszeitalters, 2nd edn., Springer, Berlin, pp. 87-118.

Woods, D. (2003) Enterprise Services Architecture, O'Reilly, Sebastopol.