A Design Method for Inter-Organizational Service Processes

Rainer Schmidt

Department of Computer Science
University of Applied Sciences
Beethovenstraße 1
73430 Aalen

Abstract. Service processes play a more and more important part in modern economies. However, their design does not achieve the flexibility and efficiency known from ordinary business processes. Furthermore, their double identity of being process and product at the same time is not properly represented in present design methods. Therefore, a new method for the design and the support of inter-organizational service processes is introduced. It is based on so called perspectives for separating independently evolving parts of the service processes. Based on it, a component-oriented approach for process design is developed.

1 Introduction

The providing of services plays a more and more important role in modern and internationally networked economies. Service processes are often provided by coordinating a multitude of human activities as part of a service process. For example to run a computer system, many different activities such as user administration, software installation, network administration etc. have to be coordinated. These activities are often not provided by a single company, but by a multitude of companies. Therefore, service processes are often inter-organizational and allow different service providers to combine their core competencies. Service processes must be very flexibly adaptable to the customer's requirements, they are also products. However, this flexibility is in conflict with another requirement for service processes: they must be provided efficiently. If a service process is individually tailored to the customer's requirements, its standardization is low. As a consequence, there are no or only low scaling effects. On the other hand, to achieve a high efficiency, a high standardization is needed to gain scaling effects. This standardization however, contradicts with the requirement to flexibly fulfill individual customer requirements. Therefore, the goal is to create a design method, which offers both individually customized service products and efficient provisioning of the services at the same time.

The paper will proceed as follows. In section 2, the properties of service processes and especially the differences to ordinary business processes are analyzed. In section 3, a method for designing service processes is defined. It uses process components for

Schmidt R. (2006).
A Design Method for Inter-Organizational Service Processes.
In *Proceedings of the 3rd International Workshop on Computer Supported Activity Coordination*, pages 3-12
DOI: 10.5220/0002486700030012

enhancing the reusability and efficiency of the process. The structure of the service components is described in section 4. In section 5, the structure of the component repository is defined. Adaptation and composition of the components is covered in section 6. Related research is discussed in section 7. A summary and outlook on further work is given in section 8.

2 Service Processes

To clarify the characteristics of service processes, a case study is used which is based on the ITIL module incident management / service desk [7], [13]. The service process is a three level IT-support for problems of a computer system. The IT-support process is operated by a service provider in the building of the customer. The three level user support is composed of a service desk at level 1, a team of specialists at level 2 and third-party specialists at level 3. All support levels interact with the customer to analyze the problem and they access the customer's computer system to configure it. Furthermore, each level has a defined reaction time for requests of the customer. The service desk at level 1 is the primary point of contact for the customer's staff. All problems and requests are collected by the service desk. The service desk has to react to incidents within 10 minutes. Many incidents can be solved by the service desk, in this case the service desks repairs the computer system directly. Only if a incident cannot be solved by the service desk, it is forwarded to the second level support. Thus the specialists are not bothered with problems below their qualification such as resetting forgotten passwords. The second level support has to react within two hours. But there are also problems, which can not be solved by the second level support. These problems are forwarded to specialists of external service providers who are the third level support. They have to react within one day.

3 A Component-oriented Method for Designing Service Processes

To create a method to design service processes, a dilemma between flexibility and efficiency has to be resolved. On one hand a process has to be provided, which is individually tailored to the customer's requirements. However, individually tailored processes are unique and therefore offer no possibility to reuse parts of the process. Thus the efficiency of an individually tailored process is low. On the other hand, one can use a highly standardized process which can be executed efficiently, because there are reuse and scaling effects. However, standardization also implies that individual customer requirements cannot be taken into account. This dilemma has existed since the beginning of industrialized mass production of goods. A common way to escape from this dilemma is the use of components to design products. Mass produced products are composed of standardized components individually specified by the customer. Products built from components are very flexible, because they can be adapted to individual customer requirements by selection of appropriate components. At the same time, components also provide a high efficiency, because they are standardized and can profit from scaling effects. Both the efficiency from mass production and the individual solution to the customer can be achieved.

The component oriented design of service processes starts with the elicitation and analysis of the (informal) customer requirements as shown in Fig. 1. The formalized requirements are used to retrieve service components from a component repository. This repository is not static but can easily be extended by additional components, which may be independently developed, for example by a subcontractor. In a gap analysis, the potential solution using the retrieved components is compared to the original requirements definition. Gaps can be handled in three ways. First, the original requirements definition can be renegotiated with the customer (1). This may lead to tradeoffs as discussed in [1]. Second, the creation of new components can be initiated to fill the gaps (2). Third, the gaps can be closed by adaptation, as described later on. The next step is the selection of components for the final solution. These components are adapted and finally composed to a composite service process. The adaptation is done by using specialization. Specialization is done by adapting the component to individual requirements without changing their external interfaces and behavior. The encapsulation of the component by its interfaces impedes the visibility of internal changes.

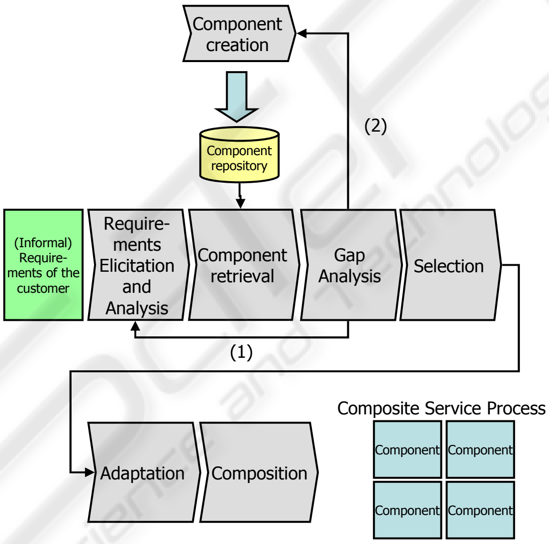

Fig. 1. Component-oriented design and modeling of service processes.

To realize a component oriented design-process for service processes three steps have to be made:

1. The structure of the components has to be specified. Particularly, the component granularity has to be defined. Components may be small or big, containing a few or a lot of functionality. This decision on component granularity is crucial to achieve flexibility and efficiency. It is also important to reach the goals of low coupling and high cohesion [4].

2. The structure of the component repository has to be defined. Only in a properly designed component repository it is possible to find appropriate components and fulfill the customer's requirements.
3. Adaptation and composition mechanisms have to be defined which fit with the component granularity

4 Service Components

Choosing the component granularity is an important step to create components for the component-oriented design process. To find the appropriate component granularity, the analysis of the service processes has to be done: It is necessary to identify the dimensions of change of the service processes. Knowing them, it is possible to maximize the cohesion of the components and to minimize the coupling of the components. These dimensions of change in service processes can be captured by so-called perspectives. Perspectives are disjoint sets of model elements, which describe independently evolvable parts of the process. For example, the organizational structure of service processes can be changed completely while the operational perspective remains unchanged. Different approaches for defining perspectives are compared in [3].

Basic Perspectives. There are five basic perspectives. The functional perspective describes what the process has to do; particularly it defines the process goal. The operational perspective specifies activities executed during the process. The control perspective defines, when and under which preconditions activities are performed. The informational perspective specifies the information which shall be exchanged between activities. The organizational perspective associates roles with activities.

Additional Perspectives. There are some characteristics which differentiate service processes from ordinary business processes and therefore have to be analyzed. First, there are a lot of interactions, especially with the customer. Second, external resources are needed during the process and third, a predefined service level has to be maintained. These characteristics require additional perspectives which will be described below.

Interaction Perspective. A characteristic of service processes is their high degree of division of labor with a high involvement of external participants. In traditional production processes the customer is only interested in the outcome of the process but not the process itself. In service processes, there are many interactions between the service provider and the customer and third party service providers. Both have to be integrated during the whole process and not only at the beginning and the end of the process: In the example above, the customer has to be interrogated for further details of his incident report. Advice is sought from the third level support. As service processes contain many interactions, it is necessary to provide flexibility in changing and integrating new interactions into the process. Interactions have to be adapted to changed customer requirements and new interactions have to be integrated due to new customer requirements. To achieve this, a new perspective has to be created when defining the metamodel for service processes. This perspective is called interaction perspective. It abstracts from different types of interactions comparable to the patterns defined in [2].

Resource Perspective. Service processes differ from traditional business processes also because they extensively use external resources both from the customer and third party service providers. Resources have to be appropriately obtained, integrated and administered [21]. For example, before configuring the customer's computer system, one has to have administrative privileges to do so. In addition, if external resources are needed for service providing but no longer available but, a procedure has to be started. Finally resources of the customer have to be given back at the end of the service providing. To properly represent changes in the resource perspective, it must be easily possible to add, change and remove resources.

Service Level Perspective. Not only the execution but also the potential to execute the service process is important to the customer. In the example above, it is important for the customer that his staff may call the service and start the service within a predefined reaction time. Therefore service providers have to make available a predefined potential to perform a service process. This potential is measured as service level. In the example above, a service level defines the maximum reaction time. To reach a certain service level, resources have to be kept ready, as services cannot be kept in store as material products. In the example, one has to keep ready properly trained staff available in the service desk, regardless whether there are calls or not. The service level perspective is needed to define the potential to perform activities. It describes the rights and duties for the customer and the service provider, the service performance indicators (SPIs), the measurement of the service performance indicators and change procedures. Service levels have to be easily adapted to changing business requirements.

Analyzing Processes Using Perspectives. Applying these considerations to the case study, we get the representation as shown in Fig. 2. Here the service process is split up into perspectives and perspective elements. Each perspective is shown as separate layer. (Not all perspectives are shown for the clarity of the drawing. The informational and the functional perspectives are not shown).

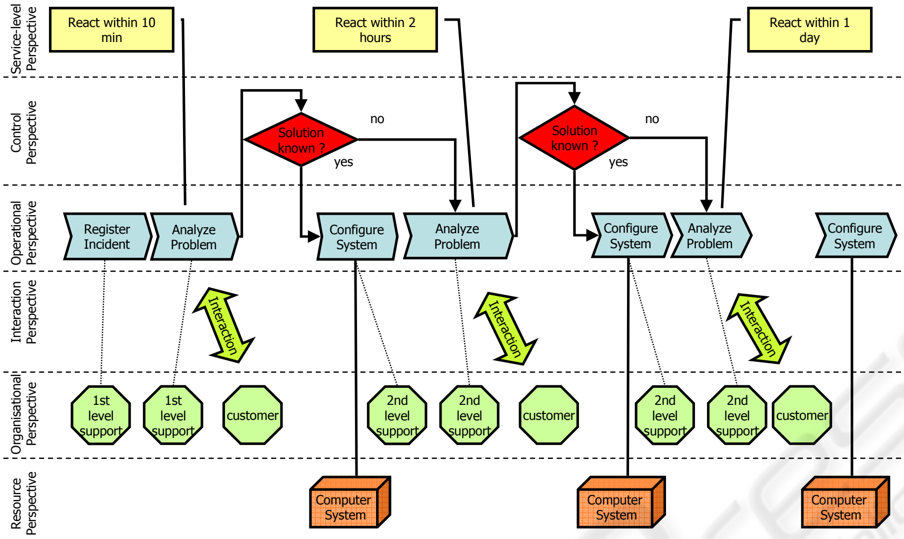

Fig. 2. Additional perspectives for service processes.

Component Structure. Based on the analysis of service processes using perspectives it becomes clear, that no component should contain functionality which belongs to different perspectives. A component should contain only functionality which belongs to exactly one perspective to achieve a high degree of cohesion [4]. However putting the complete functionality of one perspective into a component would not deliver the optimal solution as processes do not contain all elements of a perspective. Therefore, elements of the perspectives should be used as granularity, for example, the interaction type a or a single control flow construct such as fork or a single data type. A component implements the functionality of exactly one perspective element. It has a component interface which is composed of ingoing, outgoing and bidirectional interfaces. By using the functionality of only one perspective element, only one "dimension" of change is implemented within a component. Thus, the component can be flexibly replaced by another component. The adaptation of components to individual requirements and the connection to other components is done by specialization: the component is parameterized and connected to other components. Therefore, the specialization mechanism contains both parameterisation and connection mechanisms. The parameterisation mechanism adapts the component to the individual needs of the composite process. It uses the parameterisation information. The connection mechanism creates the connections of the component with other components. It uses the connection information.

5 Defining the Repository Structure

The repository is structured according to the perspectives. Therefore there are different branches for the functional, operational, organizational, informational, control, service level, resource and interaction perspective. Some of the perspectives are further split up. The functional perspective is split up into complex and simple processes. The organizational perspective differentiates the perspective elements role, person and organizational unit. Schema, schema elements and relations are the elements of the informational perspective. The control perspective is composed of elements to represent the control flow, exception handling and timing. Resources may be complex or simple. Interactions may be a simple one-way communication, bidirectional or follow complex protocols. Interactions can be further described by the following properties: First, the start of the interaction may be automatic or on user initiation. Second, the participants can be predetermined or have to be decided in an ad hoc manner. Also the participation of mediators, which are not member of the participating organizations, may be necessary, for example to settle a dispute. Third, the interaction may have a definitive structure and the end of the interaction is determined during the interaction. Finally, interactions can be differentiated if they have a defined outcome or not.

6 Adaptation and Composition of Service Components

A composite process is created by a set of interconnected and specialized process components. The connections and parameterisations are made by the parameterization and connection mechanisms described above. For different composite processes there is different specialization information. It is discriminated by the so-called global context identifier. The so-called local context identifier differentiates multiple uses of the same component in a composite process.

To clarify the idea of a composite process, a part of the case study shall be represented as composite process in Fig. 3. The decision “Solution known” is represented by component c1. The appropriate specialization information is identified by the context identifier i_e. The parameterization information contains the specialization information about the test “Solution known”. The information which allows to decide, whether the test “Solution known” is true or false is contained in component c6. Depending on the result, the connection information either continues the process with component c2 (“yes”) or component c5 (“no”). Component c2 is supplied with the connection information c3 and c4 as it uses them to do its task. If component c5 continues, the component c3 is used for representing the 2nd level support. However, c3 is used in two different contexts. Therefore, we find two sets of specialization information for component c3, differentiated by the context identifiers i_{e1} and i_{e2} respectively. The activities “Configure System” and “Analyze Problem” are represented by the components c2 and c5. The Computer System is symbolized by the component c4.

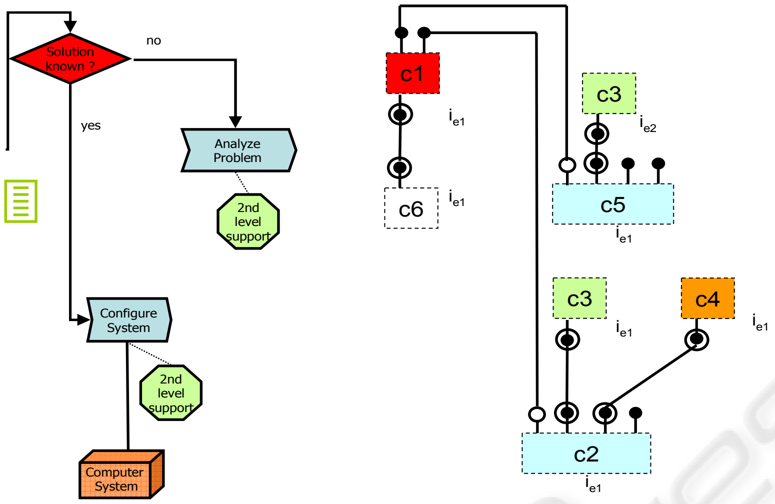

Fig. 3. Composite Process from example.

7 Related Research

To evaluate the approach presented here, a discussion with comparisons to related research shall be done. Related research can be found in a number of areas, however there are only few approaches which directly address the subject of the approach presented here. The most important one, is the Service Flow approach described in [KlWe01], [19], [18]. It describes how to model service processes and to execute them. However, the approach neither identifies the dilemma between flexibility and efficiency nor does it provide a component-oriented approach for modeling. Furthermore, the approach only covers interactions in detail. Further perspectives such as the service level or the resource perspective are not covered. The support of service processes is done by using a coarse-grained architecture.

The problem addressed by the approach presented shows some relationship with the problem of reusing reference models. Reference models give a recommendation for designing processes for a defined purpose. By capturing best practices, they are a reuse mechanism for processes. The problem is, that they both have to be adapted to specific requirements and have to reuse as much process knowledge as possible. Thus, the problem or reusing reference models can be compared with the reuse of service processes. Six classes of methods for the reuse of reference models are differentiated according to their reuse mechanisms. Two classes, the compositional-monolithic and the generic-monolithic approach ([6], [17], and [16]) use monolithic reference models. The composition of a service process out of predefined component is thus not possible. Pattern-language oriented approaches such as [20] define a multitude of dependencies between different models. However, the definition of an appropriate component granularity and suitable interfaces is omitted. Catalog-based methods [5] and knowledge based methods ([9], [15]) are aimed at the retrieval of already defined reference models as a whole. No method how to appropriately design components for reuse is presented. The library based approaches ([11], [10]) store (parts of) reference models in a library using a classification or type system. However, no recommendations for an appropriate granularity of the library entries are given. Instead the formalization of the library elements is focused as in [10].

There are also approaches, which do not show a relationship although similar terms are used. For example, service-oriented architectures (SOA) also use the term service, for example [14] but target services provided by computers. Therefore one has to differentiate between service processes, which provide services, and processes which are executed using services, better called service-based or oriented processes.

8 Summary and Outlook

Service processes are an important group of inter-organizational business processes. They have special properties which require new methods for their design and execution. They are at the same time processes and products and therefore have to be flexibly adaptable to the customer's requirements while being offered at a competitive price. Furthermore service processes show a high degree of interaction with external participants such as the customer and subcontractors. Another difference to standard business processes is the integration of external resources, for example the customer's computer system into the process. Finally, service processes not only have to produce a defined process output but they have also to provide a defined potential to provide the process output called service level.

To cope with these requirements a component-oriented design and modeling method has been developed. It uses perspectives to identify the independently evolvable parts of the processes. By using the perspectives, components can be defined with a maximum of cohesion and a minimum of coupling. Based on the method, composite processes are built from components in a way individually specified by the customer. A high degree of individualization can be achieved, by choosing from a multitude of components. By using a component oriented approach both efficiency and the individual solution to the customer can be achieved.

Further work will apply the concepts developed to ITIL in the German chapter of the ITSMF (Information Technology Service Management Forum). Further research will cover the following topics. First, in a distributed environment, components may use different model representations such as event oriented process chains versus Petri nets. Therefore an integration mechanism has to be developed. The distributed environment may also require the distribution of the component repository. Second, a mechanism for revising and creating versions of the components has to be developed. Third, market mechanisms have to be created to exchange components.

References

1. Alves, C., Filho, J. B. P., Castro, J., Analysing the Tradeoffs Among Requirements, Architectures and COTS Components *IV Workshop on Requirements Engineering*, Buenos Aires, Argentina, November 2001.
2. M. Aksit, K. Wakita, J. Bosch, L. Bergmans, A. Yonezawa: Abstracting Object Interactions Using Composition Filters. In R. Guerraoui, O. Nierstrasz, M. Riveill (Eds.): Object-Based Distributed Programming, ECOOP '93 Workshop, Kaiserslautern, Germany, July 26-27, 1993. LNCS Vol. 791, Springer Verlag, Berlin, 1994.
3. M. Bernauer, G. Kappel, G. Kramler, W. Retschitzegger, Specification of Interorganizational Workflows - A Comparison of Approaches, Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003),
4. J Eder, G Kappel, M Schrefl: Coupling and Cohesion in Object-Oriented Systems. - Proc. Conference on Information and Knowledge Management, …, 1992 –
5. Fernandez, E. B.; Yuan, X.: Semantic Analysis Patterns. In: A. H. F. Laender, S. W. Liddle, V. C. Storey (Hrsg.): Conceptual Modeling – ER 2000 – 19th International Conference on Conceptual Modeling, Salt Lake City, Utah, USA, October 9-12, 2000 Proceedings. Berlin et al. 2000, S. 183-195.
6. Hars, A.: Referenzdatenmodelle – Grundlagen effizienter Datenmodellierung. Wiesbaden 1994, Saarbrücken 1993
7. www.itil.co.uk
8. Klischewski, R., Wetzel, I.:, Serviceflow Management: Caring for the Citizen's Concern in Designing E-Government Transaction Processes, Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35). IEEE, 2002 Etegv03.pdf
9. Krampe, D.: Wiederverwendung von Informationssystementwürfen – Ein fallbasiertes werkzeuggestütztes Ablaufmodell. Wiesbaden 1999.
10. Lang/Bodendorf: Gestaltung von Geschäftsprozessen auf der Basis von Prozeßbausteinbibliotheken. In: HMD 1997, 198, S. 83-93
11. K- Lang: Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen, DUV-Verlag, Gabler, Wiesbaden, 1997
12. Regev, G., Wegmann, A.: A Regulation-Based View on Business Process and Supporting Proceedings of the CAiSE'05 Workshop, p. 91-98.
13. Schmidt, R.: IT-Service-Management – State and Perspectives. 4. itSMF Kongress Hamburg 2004
14. Schmidt, R.: Flexible Support of Inter-Organizational Business Processes Using Web Services. Proceedings of the CAiSE'05 Workshop, p. 51 - 58.
15. Schulze, D.: Grundlagen der wissensbasierten Konstruktion von Modellen betrieblicher Systeme. Aachen 2001
16. Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung – Konstruktion konfigurations- und anpassungsorientierter Modelle, Gabler, Wiesbaden, 1998.
17. Schwegmann, A.: Objektorientierte Referenzmodellierung – Theoretische Grundlagen und praktische Anwendung. Wiesbaden 1999
18. I. Wetzel, R. Klischewski: Serviceflow Beyond Workflow? Concepts and Architectures for Supporting Inter-organizational Service Processes. CAiSE 2002: 500-515
19. I. Wetzel, R. Klischewski: Serviceflow beyond workflow? IT support for managing inter-organizational service processes. Inf. Syst. 29(2): 127-145 (2004)
20. Wolf, S.: Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme. Aachen, 2001
21. Zdravkovic, J.; Henkel, M..: Enabling Flexible Integration of Business and Technology in Service-based Processes. Proceedings of the CAiSE'05 Workshop, p. 107 – 114.