Model Synchronization in Sociotechnical Service Systems

Michael Thieme, Lars-Peter Meyer, Kyrill Meyer and Christian Zinke

Business Information Systems
University of Leipzig
Leipzig, Germany

ABSTRACT

Integrated product-service systems become complex sociotechnical services consisting of human and technological components. Existing management approaches are limited as they neglect changing resources and processes during the service lifecycle. The lack of proper procedures and methods going beyond the initial service engineering leads towards unsystematic and low-quality services. This contribution addresses this gap between the real service processes and the underlying service model. The paper will describe a 4-steps procedure model, which defines a framework for synchronization routines in sociotechnical service systems. The methodic approach focuses on the definition and integration of synchronization procedures in addition to standard service development. Besides the theoretical basis, we demonstrate the practicability with a case study.

Keywords: Service Systems, Service Engineering, Roundtrip Engineering, Sociotechnical Systems, Synchronization Routines, Procedures, Reverse Engineering, Active Knowledge Management

AGILE ENGINERING FOR SERVICES

Today, all spheres of the society undergo a continuing progress of technization, digitalization and servitization. This process induces to more integrated product-service systems including goods, services and ICT (Rust and Miu, 2006). These integrated products become complex sociotechnical (service) systems consisting of human and technological components (Meyer and Thieme, 2012). Existing management approaches are limited to the development of such complex service systems. There is no consideration of changing resources and processes during service delivery which is caused by missing procedures that enable an efficient and systematic change management. However, these changes are not included into the original service model due to lack of time, costs, missing awareness or missing procedures, which can lead towards unsystematic and low-quality service delivery (Thieme et al., 2013).There is a lack of proper procedures and methods going beyond the initial engineering of services that address changes of surrounding conditions in the operative phase of the service lifecycle. Missing documentation and implementation of changes in the service model significantly complicates the management of complex services and in addition leads to tacit knowledge mostly concentrated to one person on how the service is produced and delivered in reality.

This contribution addresses the presented gap between the real service processes and the underlying service model. The paper will describe an agile development method, which defines synchronization routines for the development of sociotechnical service systems. In so doing, we make use of the concept of roundtrip-engineering (RTE) and introduce it into the field of service engineering. The proposed methodic approach focuses on the definition and integration of synchronization procedures into the service concept using reverse engineering. The goal is to enable an efficient modification and update of the service model. Filling the gap of missing reverse engineering procedures enables a complete RTE-system for services and increases efficiency in service delivery and innovation. Besides the theoretical basis of the procedure model, we demonstrate the practicability and benefit of our approach with a case study in the area of ICT and knowledge management.

SERVICES AND ROUNDTRIP ENGINEERING

Engineering Service Systems

The definition of services is highly generic and needs to be clarified. Our definition follows the understanding of Böttcher (Böttcher 2009), who defines service science as follows:

> *"Service Science is the scientific discipline that is concerned with the description, explanation, design and configuration of socio-technical systems. These connect different partners through activities and the use and (ex-) change of resources in accordance to agreements between these partners with the goal of creating value for one or more partners or the whole system."*

Such an understanding defines services as socio-technical systems, where the different elements along with the subjects and objects and their resources frame the view (Chesbrough and Spohrer, 2006) (Edvardsson, 2006). In organizational development, the term sociotechnical systems describes an approach to complex organizational work design that recognizes the interaction between people and technology in workplaces. The term also refers to the interaction between society's complex infrastructures and human behavior. In our understanding, services are complex socio-technical systems consisting of human and technological components that need to be developed and engineered following a systematical approach.

This systematic development with a defined demand on quality and productivity is the central task of service engineering (Meyer and Böttcher, 2011). As a research field of service science (Schneider et al., 2006), it deals with procedures, methods and tools for the development of services. The overall aim of service engineering is the optimization of structures and processes within service systems, where e.g. productivity, quality, costs, and repeatability can be criteria. Consequently, it is strongly connected to the design of development processes for services and the search for links to other relevant research disciplines, e.g. software engineering or information engineering (Fähnrich, 1998). In service engineering, procedures and modeling frameworks are a well explored research area. Based on the work of Böttcher (Böttcher et al., 2009) (Bullinger et al., 2003) a complete services description consists of three basic models: product model, resource model, and process model. The scientific literature provides numerous procedures, methods and models covering the area of new service development as well as the management of services (Meyer, 2011). There has been no work conducted dedicated to reverse engineering procedures that cover the synchronization of the generated service model with changes over time of the implemented service. So far, the use of service models is restricted to the innovation stage of new service development projects and in many cases disregarded after implementation. As our previous research has shown, this a big obstacle in practice (Thieme et al., 2013). The lack of documentation of changes over time makes the management of processes and tasks almost impossible. Missing synchronization of service model and reality results in managing chaos, as services, in contrast to products, quickly change over time.

The Concept of Roundtrip Engineering

As presented in previous publications (Meyer et al, 2013), RTE is mainly used in the domain of software development and engineering in order to synchronize source code and UML (Knublauch and Rose, 2000) (Demeyer et al., 1999). In short, RTE is a concept to utilize the benefits of two different abstraction layers (or perspectives / views / modelling languages) and keep them synchronized.

Initially, Aßmann describes RTE as *"an instance of the method of domain transformations"* (Aßmann, 2003). Henriksson and Larsson (Henriksson and Larsson 2003) provide a general definition for the fundamental understanding of RTE. This definition describes RTE as a system of product, design and corresponding transformation procedures. Hereby, forward engineering characterizes a procedure intended for the creation of a product by a given design. Respectively, reverse engineering characterizes a procedure intended for the extraction of a design given a product. If it is possible to extract a design and based on this create an identical product (compared to the initial product used for extraction), then product, design and the transformation procedures form a RTE system (see figure 1).

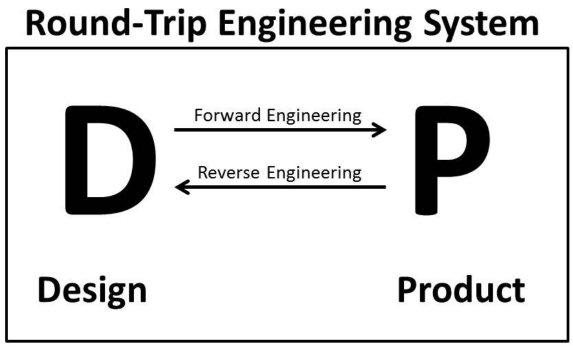

Figure 1: RTE-System (Meyer et al., 2013)

Although this definition seems intuitive, it is too strict to be useful in practice as it assumes that the source and the target are isomorphic to each other. As stated by Hettel, in most application scenarios it is not necessary and not applicable to provide transformation procedures which create the opposite view from scratch. Most of the time the synchronization of changes between the two views is sufficient or even to be preferred (Hettel et al., 2008). Hettels view is more of a practical value and fits our thoughts of the following section.

The underlying idea of RTE can be applied in other application fields. The goal of the roundtrip concept is to enable a preferably automatic synchronization of different views or levels of abstraction. In this work, we adopt the core idea of forward and reverse engineering procedures in order to update changes during service exploitation.

Application of RTE for Sociotechnical Services

Although almost all work conducted on RTE is concentrated in specific on software model transformations, the underlying idea can be applied in other application fields. Besides software modeling, first work has been conducted in the area of business process management (Barros et. al, 2007). Salvendy and Karwowski (Salvendy and Karwowski, 2010) briefly discuss the application of RTE in service engineering as a method for determining the correctness of implementation. Up till now, no further work is known to us that addresses the concept of RTE in the context of service engineering. In this chapter, we will briefly outline the application of RTE in Service Engineering based on the definition of Hettel.

In this application scenario the source model is the service model and the target model represents the executed service as an instance of the source model applied in reality. Thus the diagram from figure 1 needs to be updated as shown in figure 2. Following the understanding of RTE in software development, we define RTE-Systems for sociotechnical services as follows:

> *“RTE-systems for sociotechnical services consist of a service model, service instance and corresponding transformation procedures. Hereby, forward engineering characterizes a procedure intended for the realization of a service instance by a given service model. Respectively, reverse engineering characterizes a procedure intended for the synchronization of service model given a change in the service instance.”*

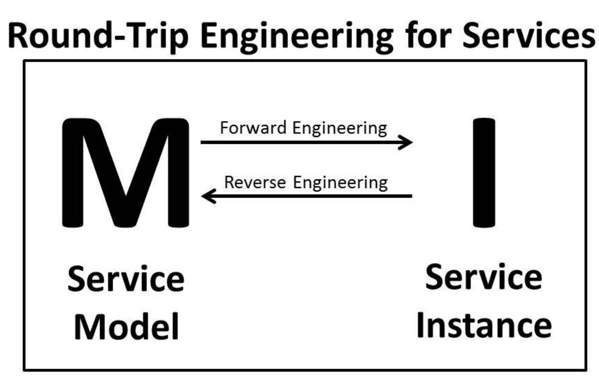

Figure 2: RTE-System for Services (Meyer et al., 2013)

In service science, service modeling, forward engineering procedures and managing services is a well explored research area (Bullinger et al., 2003) (Maglio 2006). The literature provides numerous procedures, methods and models covering the area of new service development (see (Meyer, 2011) for an overview), forward engineering procedures as well as the management of services and service instances, e.g. software-service co-design (Meyer, 2009) or service level management (Sturm et al. , 2000). So far, there has been no work conducted dedicated to reverse transformation procedures. This brief outline shows that service research provides a solid basis for the creation of RTE-systems. Nevertheless, the gap in the literature on reverse engineering procedures needs to be filled in order to be able to create complete RTE-systems for services as defined in the earlier section.

A FRAMEWORK FOR RTE SERVICE SYSTEMS

Methodical Approach

The proposed framework for the engineering of roundtrip service systems is based on the theoretical concepts of service orientation, RTE and agile development methods. We developed the framework following a 4-tasks procedure, which is described as follows:

- Task 1: Literature Review and State of the Art

The first task for the development of this framework was a literature review on service development procedures (Meyer, 2011) and RTE (briefly presented in section 2). The output of this task was used as input for step 2. The review has shown that in service science, synchronization of changes has been neglected. The concept of RTE has been successfully applied in software development in order to address this task, but so far has been not deployed in other application areas.

- Task 2: Theoretical Basis

In a second step, we integrated the existing work on service science and software development by applying the RTE concept in the area of service development and engineering. By doing so, we developed the theoretical basis for RTE Service Systems that has been presented in section 2 of this paper. The main idea of our approach is, to combine the levels of service model and service instance using forward and reverse engineering in order to enable synchronization of both levels.

- Task 3: Application in Practice

The next step was the application in a real life scenario. The aim of the case study was to test the practicability of the idea and to identify possible challenges and obstacles (presented in chapter 5). The practical experience in creating a RTE-System enabled the development of a procedure framework.

- Task 4: Definition of the Framework

Based on the developed theory (presented in section 2) and the application in a case study (presented in section 4), we defined a 4-stage procedure model as a generic framework for the development of RTE-Service Systems. This framework is presented in the next section.

RTE Procedure Framework

The aim of the presented framework is to offer a guideline for the development of RTE-Systems for services. The proposed framework consists of a four-stage procedure model.

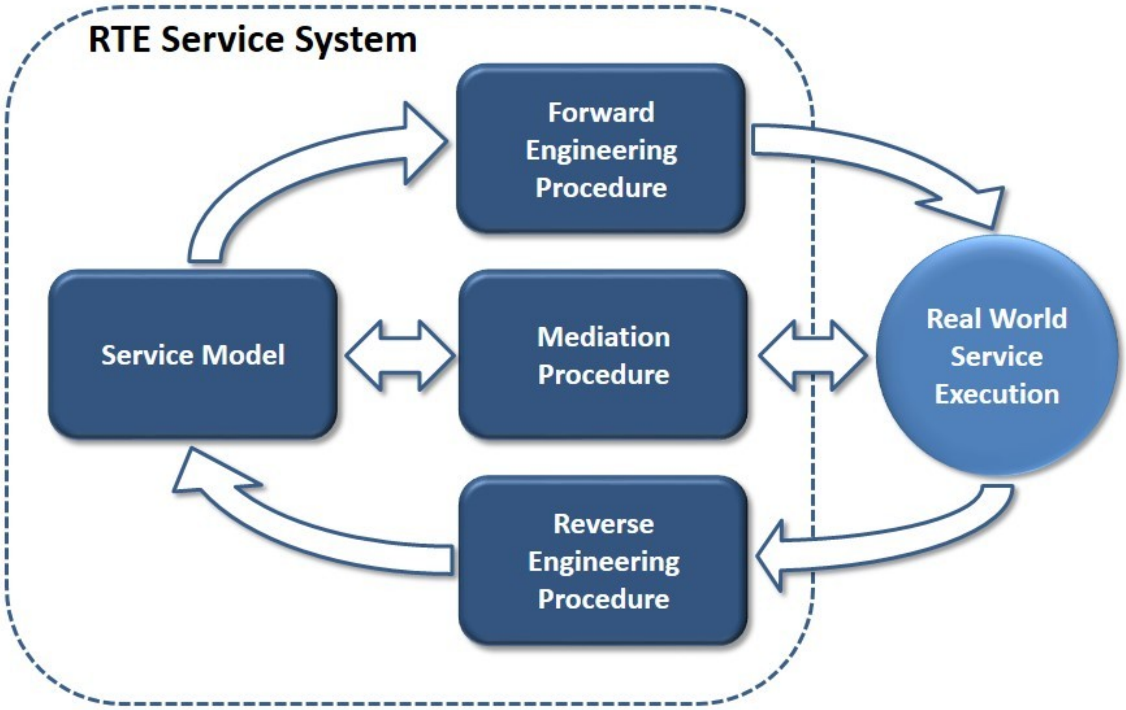

Figure 3: Visualization of the 4-steps procedure model

Although the visualization in Fig. 3 may suggest a linear process model, the procedure is intended to be run iteratively over the entire service lifecycle. Similarly to the agile software development methods like SCRUM, the aim of the proposed iterative approach is to enable an incremental and evolutionary development of the RTE System by self-organized teams and allow a regular, rapid and flexible response to change. The goal of this approach is the enabling of efficient feedback loops and adaption cycles. The stages are described as follows:

- Step 1: Generate Service Model

The initial starting point is the generation of a formal service model. Based on the work of Böttcher (Böttcher, 2009) a complete service description consists of three basic models:

1. product model – covers modularization of service components, configuration information and constraints.
2. resource model – describes all resources available and needed for the services
3. process model – contains process descriptions for the execution of the services

The generated service model contains all information regarding the realization of the service provision. It serves as a guide for the stakeholders and enables to reconstruct and comprehend the modeled service. In the beginning, it is not essential to generate a perfect model that describes the service system in every detail. As the intention is to evolve the model during several iteration loops, the model has not to be necessarily complete at the end of this step. At the first iteration loop, even a blank page of paper acting as a first resource may be sufficient in order to move on to the next step.

- Step 2: Engineer Forward Procedures

Forward engineering defines procedures intended for the realization of a service instance by a given service model. These procedures provide information generated in step 1 to the user. Users include all stakeholders involved in the service process, e.g. service costumer or staff of the service provider. The access to the information from the service model has to be presented according to the needs of each user type. Mostly, the information is provided using ICT tools. The degree of application of dedicated ICT needs to be determined by cost benefit analysis. At the end of this step, a sociotechnical system is created which allows access to all information related to the service.

- Step 3: Engineer Reverse Procedures

Reverse engineering defines procedures intended for the synchronization of the service model given a change in the service instance. The aim of the procedures is to gather variations in the real life application and enable a backflow of the identified changes and/or improvements into the service model. The service is analyzed regarding likely changes and corresponding routines to address these changes. Therefore, each service component and resource has to be assigned a responsible person that is in charge of updates and the target of corresponding information flow. The extent to which these procedure can be realized as automatic, semi-automatic or manual procedures depends on the level of ICT implementation defined in step 2.

At this stage, it is clear that not all changes can be foreseen or respectively cannot be transferred into the service model automatically. Especially semi-automatic and manual procedures may need further adjustments and consultation of upper management. This arising gap is addressed with the next step.

- Step 4: Define Mediation Procedures

The definition of mediation procedures intends to address the task of the information transfer into the service model and to enable a systematic approach to unforeseen changes and innovation ideas. The mediation model defines roles, responsibilities and procedures regarding the change management of the service. The routines should include periodic sessions, where feedback of the stakeholders is gathered and evaluated in order to access tacit knowledge that may be transferred into innovative changes.

USE CASE SCENARIO: STOCK MANAGEMENT

The following case study intends to visualize the application of the developed framework and demonstrate the practicability of the theoretical approach. In order to do so, we selected the organizational service of stock management of an institution consisting of several departments with approximately 70 employees. The service covers all activities within the institution concerning the lifecycle management of acquired goods starting with the acquisition and ending by disposal.

Generate Service Model

In this case study, we have oriented ourselves to the design paradigm of service orientation with the aim to describe the stock management services following the view of a service oriented organization structure. Hereby, we defined service components in accordance to the SOA-principles in order to ensure reusability, loose coupling and autonomy. As space in this contribution is limited, we will not present the service in every detail, but give a short overview on the service components, service instances and resources.

In our example, we have defined three service components for our service: procurement, inventory and delivery. The component procurement covers all operations concerning the acquisition of goods. It is divided into the activities purchase, withdrawal from storage and delivered by employee. This means that the good can either be newly purchased, taken from storage or taken from an employee that has the good in use. The component inventory covers the area of managing the inventory and correspondingly consists of the activities add to inventory list, update inventory list and remove from inventory list. The component delivery covers the area of handling the goods. It consists of the activities handover (to employee), storing, sell and dispose (see fig. 4).

In our use case, the service has three possible purposes: acquisition of goods, transfer of goods between departments, employees and stock or disposal of goods. In each of this scenarios activities of each service component are combined in order to generate a service instance.

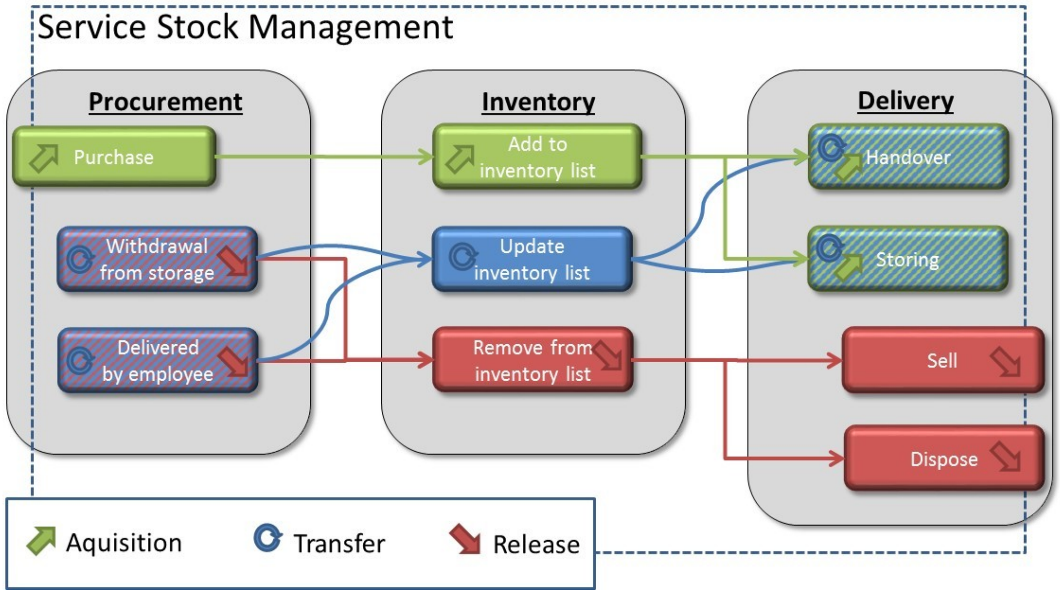

Figure 4: Visualization of the component model including service scenarios

A typical application scenario would be the purchase of a laptop by an employee. In our case, this would involve the activities purchase, add to inventory and handover to employee. Each of this activities is documented with a description and a process model including corresponding resources. Fig. 5 gives an overview on activities and resources that are needed during service delivery.

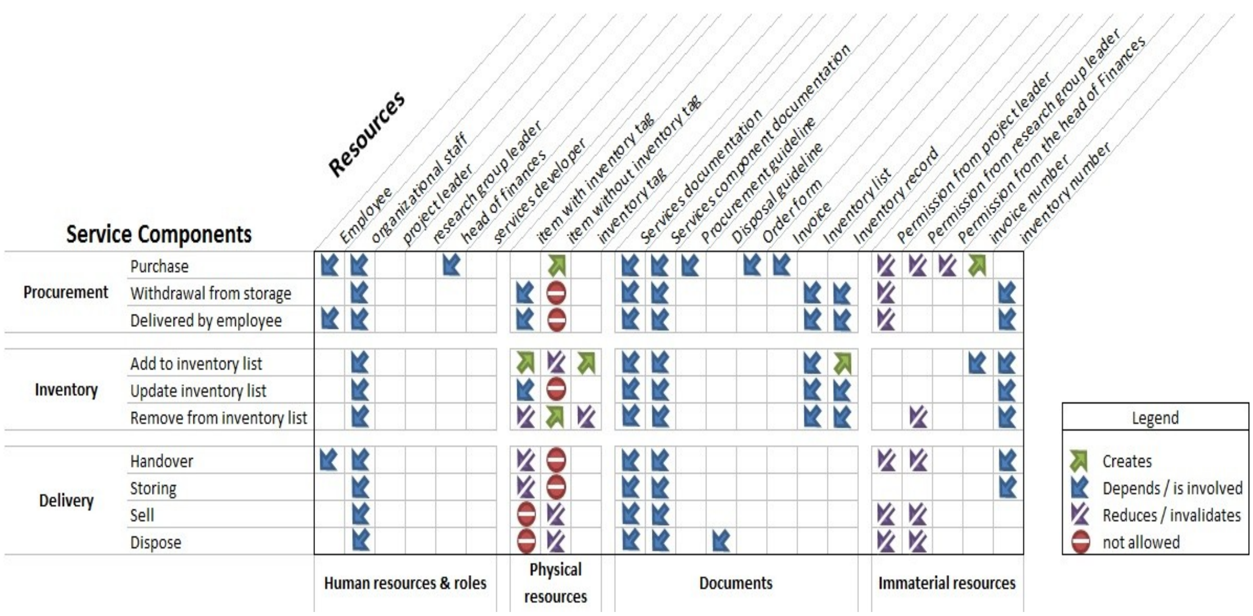

Figure 5: Visualization of the resource model with mapping to components

At the end of this step, we have generated a full service model consisting of a service description, a component model describing possible service instances, process models of all components and activates as well as a corresponding resource model. In other words a lot of information on the current state of the service that are written down as text, graphics and sheets.

Engineer Forward Procedures

The purpose of this step is to ensure, that the service model created in step 1 can be accessed by the stakeholders and that the needed information can be found as fast as possible. In our test scenario, we have decided to transfer all the informations, guidelines and hints into a knowledge management system. Involved stakeholder can enter this system via a web service and hereby get access the information about the service (see screenshot in fig. 6).

Figure 6: Representation of the service model using a knowledge management system

In this system, it is possible to generate different views on documents according to role of the user, add Q&A lists that guide the user to the right information and use a semantic search engine in order to find the right information. At the end of this step, we have a full service model that can be accessed via a standard internet browser. At this point, these informations are static and cannot be modified or updated.

Engineer Reverse Procedures

In order to ensure that the information are up to date, we make use of the feedback functionality of the system (see screenshot fig. 7) and devote it as a reverse engineering procedure. Therefore, each resource that is embedded in the knowledge management system has been assigned a responsible person. This person will receive the feedback that is send via the feedback function and is responsible to process this information according to the mediation routines defined in the next step. The feedback is sent by users that identify possible misunderstandings, missing or outdated information.

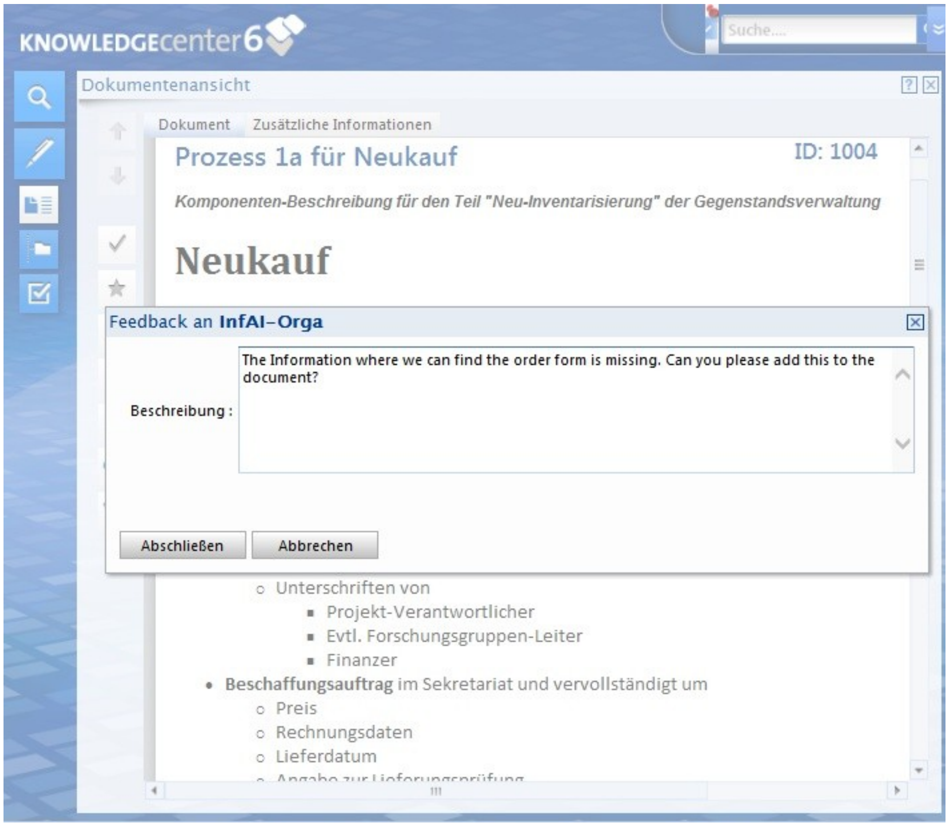

Figure 7: Feedback function used as reverse engineering procedure

Define Mediation Procedures

The mediation model defines procedures that allow to process information that is generated in step 3 and to access information that cannot be addressed with reverse engineering procedures. In our use case scenario, the model defines that an innovation workshop is carried out in regular intervals in order to address changes that cannot be foreseen and mapped with reverse engineering procedures. The workshop is organized by the person who is in charge of the service as a whole. Furthermore, the model defines that minor changes regarding service components or resources that are indicated via the feedback procedure are carried out by the assigned person. In case changes are of great moment, the person is advised to contact the person responsible for the service who in a second step can call in an innovation workshop.

CONCLUSIONS AND FUTURE WORK

The research presented in this paper can be qualified as work in progress. So far, we have developed a theoretical framework for the development of RTE service systems that will need adjustments and polishing based on further practical experience. The developed procedure model can be considered a prototype where each step will need more refinement and deeper analysis, especially the area of reverse and mediation procedures. The future will concentrate on the development of reverse engineering procedures and the display of user specific information (as part of forward engineering), what we call active knowledge management. The next step will be to analyze the usage and acceptance of our approach in the presented case study. Over time, we will add further and more complex services of this institution in order to gain more experience. As a second step, a further case study will be conducted in order to demonstrate practicability in different application areas.

Conducting this case study, we have learned that during the creation of a service model we need to determine how detailed the service descriptions and models need to be. This question is a tradeoff between accuracy, precision and correctness on the one hand, and invested effort, usability and simplicity on the other. This observation gave rise to the idea, that the level of information depth could be determined by iteration loops following our procedure. The taught is, that starting with simple models and procedures will expose missing information, ineffective work processes etc. via reverse and mediation procedures. This deficit in systematic service development will be then addressed in a second iteration where more sophisticated determined resources and procedures are installed depending on the communicated demand. Following several iteration loops, we will achieve the information depth and procedures that are most appropriate to the given institution and task.

ACKNOWLEDGEMENT

The research described within this paper has been conducted as part of a project funded by the Germany Ministry of Education and Research (FKZ 01XZ12001 – 01XZ12004). More information can be found at www.routis.de

REFERENCES

Aßmann, U.(2003): *"Automatic Roundtrip Engineering"*. In: Electronic Notes in Theoretical Computer Science, vol. 82, pp. 33–41

Barros, A.; Decker, G.; Dumas, M. (2007): "Multi-staged and Multi-viewpoint Service Choreography Modelling", In: SEMSOA 2007

Böttcher, M. (2009): *„Architektur integrierter Dienstleistungssysteme. Konzeption, Metamodell und technikraumspezifische Konkretisierung"*. Dissertation, Universität Leipzig

Böttcher, M.; Fähnrich, K.-P. (2009): *"Service Systems Modelling"*. Proceedings First International Symposium on Services Science (ISSS´09). In: Alt, R.; Fähnrich, K.-P.; Franczyk, B. (ed.): Berlin: Logos.

Bullinger, H.-J.; Fähnrich, K.-P.; Meiren, T. (2003). *„Service engineering—methodical development of new service products."* International Journal of Production Eco-nomics, 85(3), pp. 275–287

Chesbrough, H.; Spohrer, J. (2006): *"A research manifesto for services science"*. Commun. ACM 49(7),pp. 35-40.

Demeyer, S.; Ducasse, S.; Tichelaar, S. (1999): *"Why Unified Is not Universal. UML Shortcomings for Coping with Round-Trip Engineering"*. In (France, R.; Rumpe, B. Eds.): «UML» 99. The Unified Modeling Language. Beyond the Standard. Springer-Verlag, Berlin; Heidelberg, 1999; pp. 630–644.

Edvardsson, B. (2006): *"Development of Service Research in Europe against the background of Global economic Change: Experiences, Challenges and Trends"*. Moderne Dienstleistungen. In: Streich, D.; Wahl, D. (ed.): Frankfurt/Main: Campus, pp. 23-26.

Fähnrich, K.-P. (1998): *„Service Engineering - Perspektiven einer noch jungen Fachdisziplin"*. Information Management and Consulting 13 (Sonderausgabe),pp. 37-39.

Henriksson, A., Larsson, H.(2003): *"A Definition of Round-trip Engineering"*, Technical report, Linköpings University, Sweden

Hettel, T.; Lawley, M.; Raymond, K.: *"Model Synchronisation. Definitions for Round-Trip Engineering"*. In (Vallecillo, A.; Gray, J.; Pierantonio, A. Eds.): ICMT. Springer-Verlag, Berlin; Hei-delberg, 2008; pp. 31–45.

Maglio, P. (2006): *"Service systems, servcie scientists, SSME, and innovation"*. Communications of the ACM 49(7), pp. 81-85.

Meyer, K.(2009): *"Software-Service Co-Design. Eine Methodik für die Entwicklung komponentenorientierter IT-basierter Dienstleistungen"*. LIV, Leipzig, 2009.

Meyer, K. (2011): *"A classification of NSD process models"*. In: Service Research & Innovation Institute (ed.) Proceedings of the Annual SRII global conference (SRII). Annual SRII global conference, San Jose, California, USA, March 29, 2011 - April 2, 2011, NJ

Meyer, K.; Böttcher, M. (2011): *„Entwicklungspfad Service Engineering 2.0 : neue Perspektiven für die Dienstleistungsentwicklung"*. Leipzig: Univ.

Meyer, K., Thieme, M. (2012): *"High-Tech-Services, Clustermanagement und Dienstleistungsengineering. Potentiale, Trends und Perspektiven"*. Leipziger Beitäge zur Informatik, LIV, vol. 35

Meyer, L. P.; Thieme, M.; Meyer, K. (2013): *"Round-trip Engineering for System Services";* Proceedings Fifth International Symposium on Services Science (ISSS 2013). In: Theory and Practice for System Services Providers in Complex Value Chains and Service Systems;Meyer, K.; Thieme, M. (ed.): Leipzig.

Knublauch, H.; Rose, T. (2000): *"Round-Trip Engineering of Ontologies for Knowledge-Based Systems"*. In (Ruhe, G.; Bomarius, F. Eds.): Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE'2000). Springer, Berlin; Heidelberg, 2000.

Rust, R.T., Miu, C. (2006): *"What academic research tells us about service"*. Commun. ACM 49(7), 49.

Schneider, K.; Bullinger, H.-J.; Scheer, A.-W. (2006): *„Service Engineering : Entwicklung und Gestaltung innovativer Dienstleistungen"*. Berlin, Heidelberg, Springer Berlin Heidelberg.

Sturm, R.; Morris, W.; Jander, M. (2000): "Foundations of service level management". SAMS, Indianapolis, Ind, 2000.

Salvendy, G.; Karwowski, W. (2010): *"Introduction to service engineering"*. John Wiley & Sons, Hoboken, N.J, 2010.

Thieme, M., Schletz, A., Meyer, L.-P., Meyer, K. (2013): *"System Services Providers – An explorative study on networked providers of complex services"*. In: 23rd International RESER Conference (ed.) Finding growth through service activities in barren times, Aix en Provence, 19–21. September 2013