**Association for Information Systems**
**AIS Electronic Library (AISeL)**

MCIS 2011 Proceedings | Mediterranean Conference on Information Systems (MCIS)

2011

# TOWARDS COMPUTER AIDED IT SERVICE ENGINEERING

Sebastian Dudek
*University of St. Gallen*, sebastian.dudek@unisg.ch

Falk Uebernickel
*University of St.Gallen*, falk.uebernickel@unisg.ch

Walter Brenner
*Institute of Information Management University of St. Gallen*, walter.brenner@unisg.ch

# TOWARDS COMPUTER AIDED IT SERVICE ENGINEERING

Dudek, Sebastian, University of St. Gallen, Mueller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland, sebastian.dudek@unisg.ch

Uebernickel, Falk, University of St. Gallen, Mueller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland, falk.uebernickel@unisg.ch

Brenner, Walter, University of St. Gallen, Mueller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland, walter.brenner@unisg.ch

## Abstract

*Increasing demand for cost-efficiency and customer-orientation leads to the need for professionalization in all stages of the information systems' lifecycle. Standardization and automatization of provision and delivery processes are commonly adequate means to reach a higher level of maturity in the operations and maintenance stage. A prerequisite is that corresponding processes are also designed in the development stage of information systems and reused – if possible. Commonly, IT services encapsulate these processes. Reusability of IT services is often achieved by decomposing them in modular IT service components. But customer-orientation and individualization decrease reusability of each IT service component. Hence, the complexity in the design stage rises due to the variety and the enormous amount of IT service components. Besides, permanent changes of the customer's needs also reinforce this situation. To assure the reusability of former created IT service components we suggest implementing a repository for these components. An efficient retrieval in the repository strongly relies on well-defined specifications of these components. In this paper we contribute to a reference data model to structure the repository of IT service components.*

*Keywords: IT service engineering, IT service design, IT service composition, IT service components, IT service repository.*

# 1 INTRODUCTION

IT-organizations are faced with a trade-off between the need for higher efficiency in all stages of the information systems' lifecycle and customer-individuality. Standardization and automatization of provision and delivery processes are commonly adequate means to reach a higher level of maturity in the operations and maintenance stage, whereas customer-individuality consequently leads to de-standardization of these processes (Corsten et al. 2007). Generally, processes are encapsulated in IT services that are specified in an IT service catalogue (Office of Governement Commerce 2007), whereas an IT service in this contribution must not be mixed up with services of service oriented architectures (SOA). Although, both service types should be designed according to the same principles, services of SOA are commonly used in the area of enterprise application integration and Business / IT-Alignment (Erl 2009). In contrast, IT services in our contribution represent the commercial view of IT service providers or IT organizations. Following ideas of the manufacturing industry IT services must integrate all information required for the following business areas: design engineering, process planning (e.g. including human resource planning), quality assurance, tooling / fixtures, costing, service, disposal / recycling, material management and control, sales and marketing (Scheer 1994).

Modularization of IT services is often suggested to ensure customer-orientation (Brocke et al. 2010; Böhmann et al. 2003). Hence, IT service components are deduced from IT services (Grawe and Fähnrich 2008). The set of IT service components are later used for composing new IT services. This idea is adopted by early concepts, e.g. from object-oriented programming (Balzert 2001) or the service-oriented architecture (Erl 2009). In more mature industries these principles are also known as modular product architecture and constitute basic beliefs in the context of mass-customization (Piller et al. 2001; Pine and Davis 1993). The objective is obvious: to create modular components that can be variously combined, locally modified and reused to increase standardization ability and hence to realize economies of scale.

In the areas of software-engineering (Overhage 2006) and construction of physical goods (Verein Deutscher Ingenieure 1982) (machine-readable) descriptions of the deduced components are one crucial point to ensure reusability. In the manufacturing industry the descriptions have finally converged in the design of IT systems – also known as computer aided design (CAD) or computer aided engineering (CAE) – that support the design of new products. The description or rather the properties that are used to distinguish between two variants of a product is a familiar mechanism in the area of variant configuration (Scheer 1994). Several interviews with experts indicate that, driven by the upcoming 'industrialization of IT management' (Zarnekow et al. 2006), similar mechanisms shall be established in the IT industry. Thus, we have identified the need for a structural description of IT services and its components that supports the design stage of information systems.

The current paper contributes to a reference model for IT service engineering (Fettke and Loos 2003) and addresses the issues of (1) how IT services and its components can be described and (2) how this information can be represented in a data model. Commonly, the IT service engineer is confronted with the situation to create new IT services by composing or configuring already existing IT services or IT service components. On the one hand he/she must be able to retrieve the appropriate IT service or IT service components and on the other hand he/she must be able to decide whether two IT service components can be combined and to which extent the selected IT service components may be configured. The presented data model enables the description and therefore easier retrieval of IT service and IT service components when assigning descriptive characteristics to each element. Secondly, the data model stores information about the compatibility of certain IT service components when introducing the "interface description" of IT services or IT service components.

The paper is structured in five sections: In the next section we discuss the related work in this area of research. The third section describes the research methodology and its application. The model is derived and presented in the fourth section. Finally, we conclude this paper by highlighting the benefits of our approach and discussing restrictions as well as further research.

# 2 RELATED WORK

IT service engineering in general is an often discussed topic in this area of research and many authors highlight the potentials of modular IT service components as a reaction to the trade-off between standardization and customer-orientation. Second literature concentrates on composing and configuring IT services. Overall, the IT service data management literature aims at establishing a consistent view on IT services during the entire lifecycle of IT services.

## 2.1 (IT) Service Engineering / modularization / mass customization of IT services

Services in general are commonly characterized by three dimensions: structure, process and outcome. The structural dimension addresses the ability and willingness to deliver the service; the process dimension describes the processes that are operated to deliver a certain service; the outcome dimension depicts the results of the processes performed for the external factor (Bullinger et al. 2003). On the other hand, a main characteristic of IT services is that technological and organizational aspects are incorporated in all three dimensions, i.e. IT systems themselves and their processes to provision and deliver must be taken into account (Böhmann et al. 2004).

Modularization of IT services emerges from literature about modularization and modular product architectures (Sanchez and Mahoney 1996; Ulrich 1995; Baldwin and Clark 2000) and the application of these ideas on services (Burr 2005; Bullinger et al. 2003; Corsten et al. 2007). Besides, literature in the area of software engineering (e.g. (Balzert 2001), especially in the context of service oriented architectures e.g. (Erl 2009)) also emphasizes the advantages of modularly built components. The transfer of modularity to IT services is discussed in various research works about IT service engineering (Böhmann 2004; Grawe and Fähnrich 2008). The conflict of objective between standardization and customization is also part of mass customization. Mass customization seeks to bridge the gap by modularizing both products and processes to combine high external variety at low internal variety (Pine and Davis 1993; Piller 2004; Duray 2002). Brocke et al. (2010) adopts mass customization concepts to IT service agreements with regard to customer-orientation and standardized producibility of IT services.

## 2.2 Composition / configuration of IT services

The composition or packaging of IT services has been examined by several researchers (e.g. (Salmi et al. 2008; Kaitovaara 2001; Nieminen and Auer 1998)). Grawe and Fähnrich (2003) distinguish between IT service composition (combining one or more IT service components to IT services) and configuration of IT services (selection of IT service variants by corresponding properties). An application-centric approach of combining components is presented in Overhage (2006). The author accentuates the necessity to comprehensively specifying application modules in order to enhance existing development methodologies. In Overhage’s point of view existing development methodologies aim at creating modular components, but do not consider how to reuse them. This would inevitably lead to the so-called “modularity crisis”. Interfaces are used to standardize the communication between modules. Other research works concentrate on modelling process interfaces between IT service components to ensure proper composition (Böhmann et al. 2005).

## 2.3 IT service data management / computer aided IT service engineering software

In order to support the IT service engineer the composition process requires an efficient management of the derived modular IT service components. Many hundreds or sometimes thousands of IT service components exist and are requisite to fulfil separate customer requirements (Grawe and Fähnrich 2008). A software-based solution is expedient to cope with the evolving complexity. Research about product data management systems addresses similar requirements in industries of physical products. These systems store product defining data and support related processes. Workflow management as well as configuration management often constitute core functionalities of these systems (Eigner and Stelzer 2009). Boehmann and his colleagues transferred this idea to services in general (Böhmann and Krcmar 2004) and IT services in particular (Böhmann et al. 2004). Both contributions aim at developing a reference model of (IT) service data management. In this context they have introduced an exemplary representation of modular IT services in a software tool. This tool supports the parties involved in the service engineering process by providing the necessary information about each IT service component. While these publications predominantly focus on modular (IT) services and its components, the "Computer Aided Service Engineering Tool (CASET)" also integrates a project management component (Herrmann et al. 2006). The authors state that current available tools only support parts of the service engineering process and claim that a need for a holistic approach exists. Based on the ideas of the Y-CIM-Model (Scheer 1994) the authors develop the functional, data und control view of the service engineering tool. The described solution emphasizes the communication, coordination and cooperation aspects of the service engineering process and also introduces a module database, similar to the presented approach by Böhmann et al. (2004). Thomas and Scheer (2006) also deal with the transfer of knowledge of the manufacturing industry in order to improve the service data management. In comparison to more explored industries the authors identified a lack of systematic service data management. On the basis of service decomposition and modularization the authors present detailed data models that eventually converge in a software tool for a model-based service customization. The researches of Grawe and Fähnrich (2003, 2008) also emphasize the need for a machine-readable service description to support the IT service data management. In this connection, they present the mechanism of a service hierarchy that relies on abstract classes similar to the ones used in the object-oriented programming paradigm.

# 3 RESEARCH METHODOLOGY

The research findings of this paper have been gathered in close cooperation with four overall and internal IT-service providers of international companies. In accordance to the direct interaction of the researchers and the representatives of the companies we applied the "action research" methodology promoted by Checkland and Holwell (1998). The evaluation of the concept has been derived from the results of semi-structured and in-depth interviews with experts, several workshops and four research projects conducted with two different IT service providers. Currently, the artefacts are applied partially at the involved companies, i.e. that the description mechanisms are used but an integrated information system is not implemented yet. Our current research project uses the concepts to implement a variant configuration process for IT services at a major IT service provider based on the presented approach.

The companies applied similar mechanisms and concepts in the context of IT service engineering: an IT service is decomposed into IT service components that are to be reused when composing a new or adjusted IT service. In-depth interviews with experts and relevant literature show that previously developed knowledge about (the composition of) IT service components is rarely being made explicit and therefore sparsely reused. On the basis of literature and several existing service catalogues, service level agreements and the corresponding IT service component catalogues we have derived a set of properties and interdependencies between IT service components that must be considered when describing and combining IT service components. Moreover, we have identified a set of common tasks (=processes) that were normally operated in the operation and maintenance stage of IT services or IT service components. Currently, we have identified two data sources at the companies concerning IT service components: firstly, lists of IT service components with some – mostly cost accounting relevant – attributes (e.g. material number, material short description, cost centre, tariff, charging units and pricing model). The lists are often exported from financial accounting systems and consist of approximately 3000 items depending on the offered service portfolio (e.g. network, storage, server, middleware, database, application management, client / workplace, service desk services). Furthermore, IT service components are described in continuous text documents that are maintained by text-processing software, approximately 400 pages depending on the offered service portfolio. These descriptions are more detailed and encompass for instance quality attributes, processes, hardware realisation, technological limitations and interdependencies. Further investigation of processes of IT service providers (service offering: network, storage, server, middleware, databases, workplace, service desk services) revealed that approximately 180,000 possible processes can be distinguished. One of these processes is for instance operated to provision (install) a server service with a specific operating system, quality, and capacity and so on. We have developed – predominantly derived from reference models of the manufacturing industry – corresponding data models that are able to store the aforementioned information about IT services and IT service components and applied the approach in four research projects with two different IT service providers.

# 4 DERIVING THE DATA MODEL

## 4.1 Modularity of IT services

One of the major success factors to reach higher reusability, less variety and therefore realizing economies of scale is a strict modularization of IT services and IT service components (Duray 2002; Pine and Davis 1993). Commonly, modules can be characterized by a few properties. The first characteristic of modules is that they are derived from the decomposition of an IT service, i.e. each module partly contributes to the entire IT service. Secondly, modular IT services are built of nested hierarchies, i.e. one module can be used to compose another module and vice versa. Thirdly, the underlying design principles of modules are loose coupling and high cohesion, i.e. strongly-related elements should be merged into a module. The components only communicate via well-defined interfaces, i.e. modules hide their internal structure (=black-box principle) (Böhmann et al. 2003). These characteristics consequently lead to the ability of modules to be locally changed without affecting surrounding modules.

## 4.2 Framework for specification of IT service components

The next section describes the developed engineering data model for IT services and IT service components. We structured the section by the defining characteristics of modules. Since we focus on the existing set of IT services and its components the decomposition process is not further investigated. The first chapter’s topic is the (nested) hierarchy of IT service components. Secondly, we introduce a set of properties for further description before we discuss standard procedures that are used to derive necessary processes for operation and maintenance stage. Finally, we investigate the technological and organizational interfaces of IT service components and present the data model. The data model is based on the reference data model of the manufacturing industry presented by Scheer (1994). Therefore the data model consists of similar terminology although it is untypical for the IT service management literature.

### *4.2.1 (Nested) hierarchies*

In the following we concentrate on the property of (nested) hierarchies of IT services. For that we distinguish between atomic IT service components and compositions of IT service components. Atomic IT service components can be defined from two different points of view: firstly, the focused IT service component cannot truly be further decomposed (e.g. infrastructure IT services). Secondly, the considered IT organization or unit does not have knowledge about the further decomposition of the IT service component. This can be observed if the IT organization or unit sources some parts of its provided and delivered IT services. The emerging trend to bundle IT service components to IT services leads to "productized" IT services, labelled as IT service products (Bullinger et al. 2003; Kaitovaara 2001; Nieminen and Auer 1998). It is noteworthy that IT service products exist on every level of the IT technology stack (infrastructure services, server service, application services or business services), dependent on the considered customer. Hence, the IT service, IT service component and IT service product can be used equivalently, whereas an IT service product is always a sellable element. This circumstance is included in the data model (see Figure 1) by the relationship type "Structure" as a self-reference on the entity type "IT service component" and the possibility to assign "Sales Characteristics" to the IT service component.

#### *4.2.2 Hiding internal structure (black-box principle) – parameterization of IT service components*

Hiding the internal structure of IT service components implies that the characteristic of an element is only influenceable by predefined parameters (Böhmann et al. 2005), i.e. each IT service component has its own set of assigned parameters that define its behaviour. On the basis of several IT service catalogues, corresponding IT service components and relevant literature we identified eight types of engineering relevant parameters or properties. Engineering or design relevant parameters encompass all parameters that are used to distinguish functional or non-functional behaviour of IT service component. For instance, price models or different vendors do not directly affect the functional or non-functional characteristics of IT service components, i.e. these types of properties are secondarily important in the stage of IT service engineering. This kind of separation of different views on products is also well-known in more mature industries. The manufacturing industry, for instance, distinguish between ten perspectives: properties regarding design engineering, process planning, quality assurance, tooling / fixtures, costing, service, disposal / recycling, material management and control, sales and marketing (Scheer 1994). Although "Sales", "Cost" and "Sourcing" Characteristics are modelled we focus on engineering relevant characteristics:

- Function – properties that describes the functional behaviour of the IT service component, e.g. a database service or a certain application. The functional aspect is represented in the data model (Figure 1) by an own entity type called "Function". This design alternative was chosen because functional aspects can further be described by properties (e.g. by the sourcing property "Software Vendor", possibly Oracle or IBM). As stated in Grawe and Fähnrich (2008) a functional taxonomy of abstract classes can further characterize functional aspects of IT service components. The relationship between the entity types "Function" and "Solution Principle" is interpreted as follows: for instance, a customer relationship management process (function) can be implemented in different IT systems (solution principle, e.g. a SalesForce and an Oracle software product). This might also represent parts of the IS/IT architecture and existing software engineering methodologies.
- Capacity – properties that describe the requirement of technological (e.g. 1000 SAPS[1]) or organizational (e.g. in full time equivalents and level of knowledge) capacity. This property is represented in the data model (Figure 1) by a subtype of the entity type "IT Service Component Characteristic". Another design alternative is to model an own entity type "Profile of Capacity Utilization" where the required capacity is assigned along a time axis (Corsten et al. 2007; Ebert 2009). In this case a 1:N relationship between "IT Service Component" and "Profile of Capacity Utilization" is modeled.
- Quality – properties that describe both technological (e.g. availability, system response time) and organizational (e.g. provision time, service time) quality aspects of IT service components. Our

[1] SAPS is a standard application performance benchmark by SAP AG (http://www.sap.com/solutions/benchmark/glossary.epx#SAPS)

analysis has revealed that each combination of IT service component and quality characteristics is strictly linked to one or more methods of measurement to enable the reporting of quality towards the customer (Office of Governement Commerce 2007).

- Security – properties that describe both technological (e.g. encryption level) and organizational (e.g. physical access rules) characteristics of IT service components.
- Business Criticality Level – describes the business impact in case of malfunction of the IT service component. These properties have major influence on the technologically configuration of the underlying hardware and the design of recovery processes.
- Customer-Platform Coherence – sets if the underlying hardware is used dedicated for one customer or shared across several customers. For instance, an IT service component that indicates a dedicated use cannot be operated in virtual hosting environments.
- Customer Integration – properties that describe the customer integration in the stage of operation and maintenance. Customer integration can also be related to technological (e.g. customers have to provide several IP addresses of their IP pool or certain application modules) or organizational (e.g. corporate operation of release management processes) aspects (Böhmann 2004).
- Production / Delivery Structure – properties that describe the geographic structure of IT service component operation and / or delivery. The usage of these properties is twofold: on the one hand it is ascertainable if the chosen IT service component can be consumed at a certain location. For example, an application that is hosted in India but used in Chicago. On the other hand these properties can be used to control where an IT service component should be produced. This is useful regarding the upcoming trend of "global production" (Mastakar and Bowonder 2005).

#### *4.2.3 Deriving processes out of standard procedures*

According to Davenport (1993) a process is a "structured, measured set of activities designed to produce a specified output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what". When applying this definition on IT services and its components the previously discussed properties describe what (outcome dimension) is delivered but do not explain how (process dimension) work is done. The analysis of IT service components and their descriptions indicates that IT organizations also delineate processes that are operated in the operation and maintenance stage. We classify these processes as standard procedures (lower left corner of the data model (Figure 1), inheritance structure of the entity type "Standard Procedure") that are used to derive standard operations (entity type "Standard Operation"). Standard operations are valid in a specific resource environment (e.g. installing a MySQL database Version 5.1) and can be used to describe specific process steps of IT service components (Scheer 1994). In this context, we have identified ten different standard procedures (partly based on (Bailey et al. 2007; Brocke 2010; Garschhammer et al. 2001; Oliveira et al. 2006)):

- INSTALL / SETUP / REGISTER – derived processes are operated in the deployment or provisioning stage of IT service components, e.g. deploy a certain application or register a new user
- MOVE – standard procedures that geographically or logically move IT assets, e.g. move a workplace device (geographically) or move data from one storage area to another (logically)
- ADD – derived processes enhance the current capacity, e.g. enhance database storage or network throughput. This class of standard procedures is strongly related to "INSTALL / SETUP / REGISTER" standard procedures because "adding capacity" can also be represented by "installing more capacity". We distinguish the two classes by the characteristic that "ADD" procedures are always be operated on previously provisioned IT service components.

- CHANGE / UPGRADE – derived processes change or upgrade the functional characteristics of a previously provisioned IT service component, e.g. changing the accounting standards in a business application or upgrading an application by patching known bugs
- REMOVE / DISPOSE – derived processes are performed to stop the delivery of an IT service component, e.g. remove database storage (capacity adjustment) or remove user (revoke user rights). This class of standard procedures is also strongly related to "INSTALL / SETUP / REGISTER" and "ADD" standard procedures as they encompass opposed activities.
- DOCUMENT – derived processes for creating necessary documentation, e.g. changes on applications
- ARCHIVE – derived processes are performed to permanently store data, e.g. archiving records of a business application. These processes are often operated periodically.
- BACKUP – derived processes backup certain data. These processes are often operated periodically but in contrast to "ARCHIVE" processes backup data is not stored permanently.
- REPAIR / RECOVER – derived processes for repairing or recovering an IT service components state. "REPAIR" processes are commonly operated if a system failure previously occurred whereas "RECOVER" processes are often initiated due to faulty operation by the user, e.g. accidently deletion of important data.
- TEST – derived processes to test new or changed IT service components.

### *4.2.4 Defining interfaces between IT service components*

Modular IT service component design results in the necessity of coordination that is addressed by two concepts (Corsten et al. 2007): decreasing coordination by creating high cohesive components and secondly structuring residual coordination by defining interfaces (Baldwin and Clark 2000; Burr 2005; Böhmann et al. 2003; Sanchez and Mahoney 1996; Ulrich 1995). As the current contribution focuses on structuring a given set of IT service components we do not examine the creation process or methodologies of modular IT service components but concentrate on interfaces of IT service components. Therefore two design principles are applied: composition of one or more IT service components and substitution of IT service components (Overhage 2006). Other reuse mechanisms (e.g. as stated in (Brocke et al. 2010)) that modify the internal structure of IT service components can only be applied on copies of the corresponding IT service components. This arises due to the black-box principle of IT service components. In the following we will derive different types of interfaces before we focus on the composition and substitution process of IT service components. As seen in the data model (Figure 1) the interfaces are also specified by the same properties used for IT service component description.

We identified six different types of interfaces of IT service components in our research (upper right corner in Figure 1). The specialization of the entity type "Interface" in its subtypes is overlapping and total, i.e. each interface must be of one or more subtypes. According to Overhage (2006) and Seco and Caires (2000) interfaces are distinguished in demanding and supplying interfaces. Whereas demanding interfaces express which services are needed from other IT service components the supply interfaces describe which services are offered by an IT service component. The previously discussed description of IT service components in help with properties can therefore be interpreted as the supply interfaces of IT service components.

Other types of interfaces are derived by the previously mentioned necessity of coordination. Coordination implies the alignment of activities to achieve a common target whereas social and objective causes can be considered. Social causes (asymmetry of information and complementary personal targets) are not in scope of the current contribution. Objective causes are further distinguished in resource, target and output interdependencies (Ewert and Wagenhofer 2008).

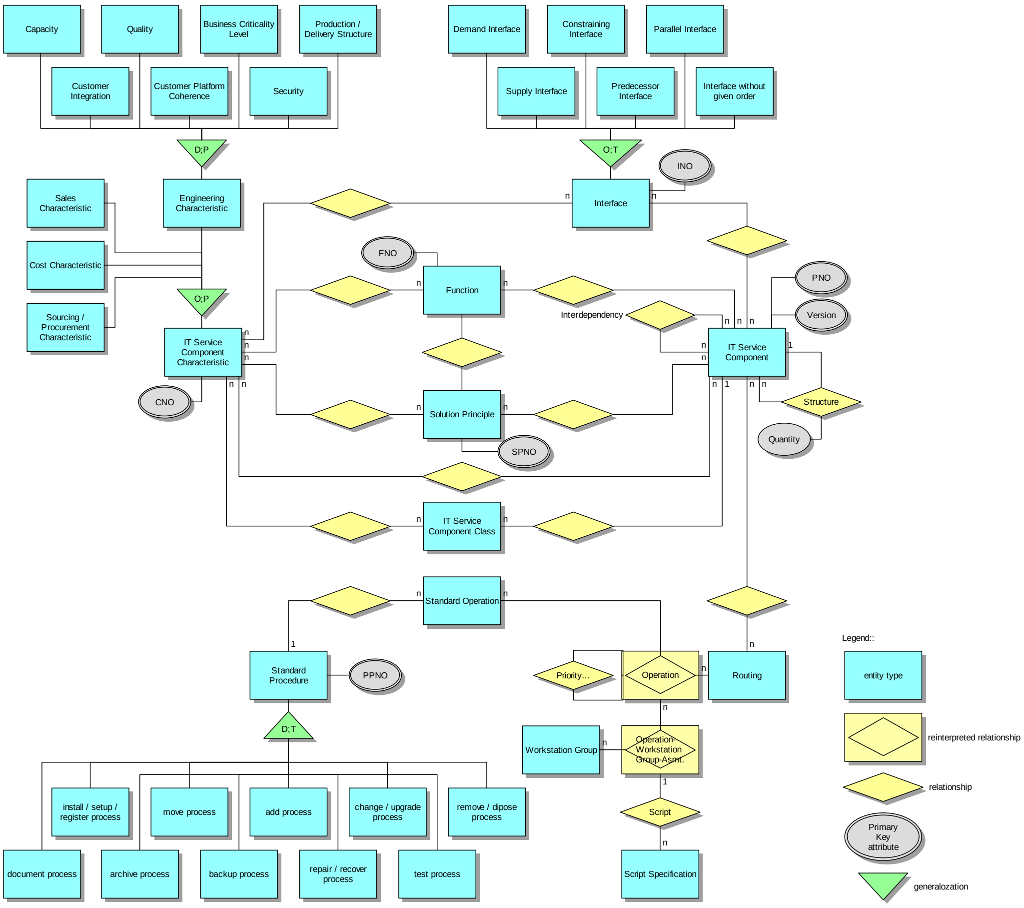


*Figure 1: IT Service Engineering Data Model (notation in eERM based on (Chen, 1976) and (Scheer, 1994))*

Resource interdependencies emerge if two or more processes of IT service components demand the same, limited resource, e.g. a database server instance or high qualified employees. Although this is indeed determined in the design stage of IT service products it is typically object of capacity management processes (Office of Government Commerce 2007). This topic is addressed by the properties of capacity assigned to IT service components. Target interdependencies are observed if the contribution to profit of processes of IT service components is interdependent of the parameterization of other processes. This commonly leads to cost and profit effects and does not further restrict the technological or organizational compatibility of two IT service components. We therefore do not focus on this type of interdependencies. Output interdependencies exist if the output of processes of two or more IT service components is interdependent. This type of interdependencies leads to several restrictions concerning (1) the execution order of processes (chronological with / without given sequence or parallel), (2) the compatibility of two processes (mutual required, mutual exclusive) and (3) possible process variants. Consequently it is necessary to address these interdependencies by formulating interfaces.

Prerequisite for composition is conformity or compatibility of the interfaces when combining two or more IT service components that communicate among each other. In the case of composition this is fulfilled if a component A entirely or partly provides the demanded services of component B. Furthermore, component A must own a supplier interface that matches or is conform to the demand interface of component B (Overhage 2006). The composed IT service consists of IT service components where demands are partly or fully satisfied by provided services of other IT service components that were combined together.  If not all demands are satisfied, two possibilities exist to handle the situation: the demand can be fulfilled by increasing the quantity of several IT service components or the composed IT service inherits the remaining demands. The first solution claims that the underlying data structure is able to store the quantity of IT service components. The presented data model (Figure 1) satisfies this requirement by the attribute “Quantity” at the “Structure” relationship. If the IT service inherits the remaining demands this consequently leads to dependencies between IT services. This is called “factoring out” of IT service components and allows extra potential of reuse. The inherited demands are treated in the same way as demands between IT service components (Seco and Caires 2000). The demands on IT service level must be fulfilled by preliminary ordered IT service products, e.g. 100 Gbyte of storage has been procured before and can now be used for a database service. It is therefore inevitable to manage the customer’s preceding bought IT service products in a so-called installed base that also reflects the current contractual relationship between the IT organization and its customer (Brocke 2010). The dependencies are represented in our data model (Figure 1) by the relationship “Interdependency” between IT service components. As the data model focuses on the engineering stage of IT service components we did not model the installed base.

The second design principle is substitution of IT service components and is handled similar: an IT service component A can be substituted by another IT service component B if both components are of the same type. Type conformity persists if component B provides equal or more services and requests fewer services than component A. Moreover, component B must supply a specialized interface for its provided services and a generalized interface for its demanded service(s) (Seco and Caires 2000).

# 5 CONCLUSION, BENEFITS, RESTRICTIONS AND FURTHER RESEARCH

The paper demonstrates an approach and its corresponding reference data model to support the design stage with the focus on the operation and maintenance of information systems. The aim was to create a framework that allows the composition of already existing modular IT service components. As IT services consist of technological as well as organizational aspects we combined both views to enable a complete description of IT services and its components. We denoted the integration of software engineering methodologies as a part of the entire approach whereas we did not focus in detail the interaction between our framework and a specific software engineering methodology. By analysing the required information and common processes of the operation and maintenance stage, our approach is able to enhance the alignment of software development and its operation. This is a possible subject to further research.

The main benefit of our approach is to support engineering / design units by explicating knowledge and hence help to cope with emerging complexity. When implementing service data management systems like ours the rate of reuse and standardization may be increased and thus, less time-to-market and time-to-delivery can be achieved. Our framework contributes to the goal to design and deliver standardized IT solutions while concurrently preserve customer-orientation and individualization. The trade off of the effort and granularity of the explicated knowledge against the emerging benefit of reusability and standardisation / automation must be subject to individual managerial decisions. This strongly relies on the targeted level of standardization / automation and the planning granularity and interval. As new service delivery models like cloud computing base upon automated / automatable processes (standard procedures) our approach also contributes to this context.

It is subject to further research how the ratio of reusability may be increased with our concept. Our findings have been derived from research projects with leading European IT service provider and IT organizations. We have succeeded in describing all known IT service components, their interdependencies and their “behaviour”. However, a current research project with one of the previously mentioned IT service provider focuses on the applicability and impact of variant configuration in the IT service industry. For that a subset of the derived properties and interdependencies of IT service components are applied to variant configuration systems to achieve better insights of the applicability of our approach. As mentioned before the concepts are partially implemented at a major IT service provider (mainly in Excel sheets and the SAP variant configuration module) but an integrated IT service data management system is still missing. This may also be subject to further research and may refine our research findings. Future researches shall also examine the usage of our approach in the context of individual, business oriented, on demand IT services and in the area of application management.

## References


Bailey, J. et al., (2007). Activity-based management of IT service delivery. Proceedings of the 2007 symposium on Computer human interaction for the management of information technology - CHIMIT ’07, p.5.

Baldwin, C.Y. and Clark, K.B., (2000). Design Rules - the power of modularity, MIT Press.

Balzert, H., (2001). Lehrbuch der Software-Technik - Software Entwicklung, Heidelberg.

Brocke, H., (2010). Managing the Current Customization of Process Related IT-Services. In Proceedings of the Hawaii International Conference on System Sciences (HICSS-43).

Brocke, H., Uebernickel, F. and Brenner, W., (2010). Mass customizing IT-service agreements - towards individualized on-demand services. In Proceedings of the 18th European Conference on Information Systems (ECIS 2010).

Bullinger, H.-J., Fähnrich, K.-P. and Meiren, T., (2003). Service engineering - methodical development of new service products. International Journal of Production Economics, 85(3), pp.275-287.

Burr, W., (2005). Chancen und Risiken der Modularisierung von Dienstleistungen aus betriebswirtschaftlicher Sicht. In T. Herrmann, U. Kleinbeck, and H. Krcmar, eds. Konzepte für das Service Engineering. Heidelberg: Physica-Verlag, p. 17–44.

Böhmann, T., (2004). Modularisierung von IT-Dienstleistungen: eine Methode für das Service-Engineering, Gaber Edition Wissenschaft.

Böhmann, T., Junginger, M. and Krcmar, H., (2003). Modular service architectures: a concept and method for engineering IT services. In Proceedings of the Hawaii International Conference on System Sciences (HICSS-36).

Böhmann, T. and Krcmar, H., (2004). Servicedatenmanagement für modulare Dienstleistungen. In H. Luczak, ed. Betriebliche Tertiarisierung. Gaber Edition Wissenschaft, pp. 149-178.

Böhmann, T., Loser, K.-U. and Krcmar, H., (2005). Modellierung von Prozessschnittstellen modularer Servicearchitekturen. In T. Herrmann, U. Kleinbeck, and H. Krcmar, eds. Konzepte für das Service Engineering. Heidelberg: Physica-Verlag, pp. 167 - 186.

Böhmann, T. et al., (2004). Servicedatenmanagement für IT- Dienstleistungen: Ansatzpunkte für ein fachkonzeptionelles Referenzmodell. In J. Becker and P. Delfmann, eds. Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung. Heidelberg: Physica-Verlag, pp. 99-124.

Checkland, P. and Holwell, S., (1998). Action Research: Its Nature and Validity. Systemic Practice and Action Research, 11(1), pp.9-21.

Chen, P.P.S., (1976). The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), p.9–36.

Corsten, H., Dresch, K.-M. and Gössinger, R., (2007). Gestaltung modularer Dienstleistungsproduktion. In M. Bruhn and B. Stauss, eds. Wertschöpfungsprozesse bei Dienstleistungen. Gabler Verlag, p. 95–117.

Davenport, T.H., (1993). Process Innovation - Reengineering Work through Information Technology, Harvard Business School Press.

Duray, R., (2002). Mass customization origins: mass or custom manufacturing? International Journal of Operations and Production Management, 22(3), pp.314-328.

Ebert, N., (2009). Produktionsplanung und -steuerung bei IT-Dienstleistern.

Eigner, M. and Stelzer, R., (2009). Product Lifecycle Management, Berlin, Heidelberg: Springer.

Erl, T., (2009). SOA Design Patterns, Prentice Hall.

Ewert, R. and Wagenhofer, A., (2008). Interne Unternehmensrechnung, Berlin, Heidelberg: Springer Berlin Heidelberg.
Fettke, P. and Loos, P., (2003). Classification of reference models: a methodology and its application. Information Systems and e-Business Management, 1(1), pp.35-53.
Garschhammer, M. et al., (2001). Towards generic service management concepts a service model based approach. 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No.01EX470), pp.719-732.
Grawe, T. and Fähnrich, K.-P., (2008). Service Engineering bei IT-Dienstleistern. In K.-P. Fähnrich and C. Husen, eds. Entwicklung IT-basierter Dienstleistungen. Heidelberg: Physica-Verlag, pp. 281-301.
Grawe, T. and Fähnrich, K.-P., (2003). Wissensgestützte Konfiguration komponentenbasierter IT-Dienstleistungen in Wertschöpfungsnetzen. In K.-P. Fähnrich and H. Herre, eds. Content- und Wissensmanagement. Leipzig, pp. 139-147.
Herrmann, K., Klein, R. and The, T.-S., (2006). Computer Aided Service Engineering – Konzeption eines Service Engineering Tools. In H.-J. Bullinger and A.-W. Scheer, eds. Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Berlin, Heidelberg: Springer, pp. 649-678.
Kaitovaara, P., (2001). Increasing Business-Relevancy to the IT Service Product with the Support of Packaging of IT Services, Turku.
Mastakar, N. and Bowonder, B., (2005). Transformation of an entrepreneurial firm to a global service provider: the case study of Infosys. International Journal of Technology Management, 32(1/2), p.34.
Nieminen, P. and Auer, T., (1998). Packaging of IT services, Turku.
Office of Government Commerce, (2007). IT Infrastructure Library - ITIL V3 (Service Strategy, Service Design, Service Transition, Service Operations, Continuous Service Improvement), London: The Stationery Office.
Oliveira, F. et al., (2006). Understanding and validating database system administration. Proceedings of the 2006 USENIX Annual Technical Conference, pp.213-228.
Overhage, S., (2006). Vereinheitlichte Spezifikation von Komponenten: Grundlagen, UNSCOM Spezifikationsrahmen und Anwendung.
Piller, F., (2004). Mass Customization: Reflections on the State of the Concept. International Journal of Flexible Manufacturing Systems, 16(4), pp.313-334.
Piller, F., Zanner, S. and Jäger, S., (2001). Mass Customization und Personalisierung im Electronic Business. WISU (Das Wirtschaftsstudium), 30(1), pp.88-96.
Pine, B.J. and Davis, S., (1993). Mass Customization: The new frontier in business competition, Harvard Business School Press.
Salmi, P. et al., (2008). New product creation process of KIBS firms: a case study. International Journal of Services and Standards, 4(1), p.16–32.
Sanchez, R. and Mahoney, J.T., 1996. Modularity, flexibility and knowledge management in product and organization design. Strategic Management Journal, 17, pp.63-76.
Scheer, A.-W., (1994). Business Process Engineering 2nd ed., Springer.
Seco, J.C. and Caires, L., (2000). A basic model of typed components. ECOOP 2000—Object-Oriented Programming, pp.108-128.
Thomas, O. and Scheer, A.-W., (2006). Customizing von Dienstleistungsinformationssystemen. In H.-J. Bullinger and A.-W. Scheer, eds. Service Engineering. Berlin, Heidelberg: Springer, p. 679–720.
Ulrich, K., (1995). The role of product architecture in the manufacturing firm. Research Policy, 24(3), pp.419-440.
Verein Deutscher Ingenieure, (1982). VDI Richtlinie 2222 (Blatt 2) - Konstruktionsmethodik - Erstellung und Anwendung von Konstruktionskatalogen,
Zarnekow, R., Brenner, W. and Pilgram, U., (2006). Integrated Information Management, Berlin, Heidelberg: Springer.