Explicating technological and organizational interfaces of modular IT service components to support the process of IT service composition

Sebastian Dudek
Institute of Information Management
University of St. Gallen
St. Gallen, Switzerland
sebastian.dudek@unisg.ch

Falk Uebernickel
Institute of Information Management
University of St. Gallen
St. Gallen, Switzerland
falk.uebernickel@unisg.ch

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

***Abstract*— IT organizations and IT service providers decompose monolithic IT services in IT service components for reaching a higher degree of reusability and therefore realizing higher economies of scales. Existing concepts of modularity are being adapted when creating these IT service components. In the area of software intensive systems modularity is a widely-used design principle to implement reusable and combinable elements. The elements (e.g. classes in object-oriented programming or services of a service-oriented architecture) commonly integrate the data and its behavior as well as supply interfaces for communication between them. Typically these elements are considered from a technological point of view regarding the possibilities to be composed. But IT service components are slightly different: they can be described by both technological and organizational aspects. The following paper examines both aspects of IT services and its components and demonstrates possibilities to integrate these facts in the description of IT service components. This approach is useful to decrease the effort for retrieving the necessary information in the process of IT service composition and therefore leads to less time-to-market, higher degree of reusability and higher quality of the delivered IT service.**

Keywords-IT service, IT service component, IT service composition, IT service engineering, IT service component description

I. INTRODUCTION

An increasing competitive environment in the IT industry leads to an enormous cost pressure for the delivery of IT services but concurrently IT organizations and IT service providers should also deliver high-quality, customer oriented IT services [31]. This arising trade-off is often faced by modularizing IT services or rather the components of which IT service are composed (e.g. [12,19]). The idea of modularization is adapted by former developed concepts e.g. from object-oriented programming or the service-oriented architecture. The approaches aim at loose coupled, high cohesive elements respectively components that only communicate via well-defined interfaces and therefore ensure the possibility of local modification without the need to change surrounding modules. In more mature industries these principle are also known as modular product or service architecture and are basic beliefs in the context of mass-customization [8,9,27,28]. Several publications deal with the idea to transfer either IT-centric or mass-customization approaches to the engineering or design of IT services [4,12]. The objective is obvious: to create modular IT service components that can be variously combined, locally modified and reused to increase the degree of standardization and hence realize economies of scale. These IT service components cover technological and organizational aspects, e.g. the delivery of a server system includes the server itself, all necessary production processes to get the server up and running and the corresponding business unit(s) must be assigned that operate the production processes [12]. As the example illustrates, a pure technological point of view is not sufficient to fully characterize an IT service component. Subsequently, mechanisms in creating, combining and describing IT service components should not only focus on technological aspects but on organizational issues, too. The possibility to combine two IT service components is accordingly dependent on both – the technological and organizational – characteristics of each IT service component and critical to reach a higher degree of reusability. Currently, the process of proofing the compatibility and effects when combining two IT service components is often operated by several business units and sparse structured.

We propose that a need exists to structure and to explicate corresponding information / knowledge to support the process of composing new IT services. This assumption is also based on research in the area of software engineering that cautions about the ‘modularity crises’. Overhage states that common development processes focus on creating modules but do not concentrate on reusing / composing them [26]. Hence the development processes must be adjusted and information about modules must be explicated to browse through the set of existing elements and finally find the appropriate one. We argue that such challenges also exist in the IT service industry when composing IT services out of IT service components.

The following paper will not focus on the process of IT service composition itself but on needed information during the process. We assume that process reference models that describe the process of IT service (de)composition (e.g. [19,23]) are already available but we do not deny that additional information could be necessary when further examination of the process of composing IT services is conducted.

We position this publication in the second stage of the information lifecycle suggested by [3]: (1) information retrieval, (2) information structuring and storing, (3) information maintenance, (4) information usage and enrichment, (5) information spreading and (6) information disposal. The shown artifact will provide a means to structure and store information about IT service components that are relevant when combining two or more IT service components to one IT service. On the one hand it focuses on properties of IT service components that influence the compatibility to other IT service components, technologically as well as organizational. On the other hand (data) structures are presented that are feasible for structuring and storing information about characteristics of IT service components.

The paper is divided in six parts: the next chapter summarizes the research methodology and related work. Following definitions of modularity, IT service product and IT service components we introduce an IT service component and process typology. Chapter 5 depicts the requirements and restrictions in the process of composition and examines compatibility of two IT service components. After applying the approach on a real life example we give a brief preview of future researches.

II. Related Work and Research Methodology

Related work in this area of research concentrates on service engineering and modularization in general [7,9,14], deriving IT services and particularly on methods and requirements needed for the creation of modular IT service components [12,19], packaging IT service components to IT services [22,23] or analyzing the necessity for a consistent service data management – partly focus on the IT service design stage [10,16,18,19,20,30]. An application-centric approach for describing and combining components is presented in [26]. Many authors highlight the advantages (e.g. standardization, reusability, economies of scale) arising from modular built components but only a few approaches provide a specific set of properties for describing and structuring IT service components. Mostly a technological hierarchy is proposed as exclusive sorting criterion [19]. Some work deal with the modeling of process interfaces between IT service components [11].

The research was conducted with leading IT service providers in Europe. We choose the design science research methodology [21] to construct the later presented IT artifacts. Concrete findings were gained during a research project with one IT service provider that offers IT services in each level of the IT technology stack (from telecommunication up to process related IT services). In the project we analyzed the existing description and structuring of IT service components. IT service components are typically specified in two ways: firstly a composite attribute (called short-description) encompass all defining properties of IT service components and secondly long descriptions exist in continuous text documents that are created and maintained with text processing software and enrich the information stored in the composite attributes. Hence the long descriptions are not able to be evaluated by applications that support the IT service engineer. We normalized (according to [13]) the composite attribute "short-description" and analyzed the continuous text descriptions to identify IT service components, its properties and its interdependencies.

III. IT Service Product and modular IT Service Components

Following ideas of the product architecture Burr defines service architecture as the decomposition of a service in components including the definition of technological and organizational interfaces between components [9]. It is also possible to adapt the concepts of service architecture to IT services [12].

As stated previously modularity is discussed in a lot of scientific contributions from different angles. Commonly, modules can be characterized by a few properties. The first characteristic of modules is that they are derived by decomposition of an IT service, i.e. each module contributes a part 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 to a module. The components only communicate via well-defined interfaces. These principles ensure reusability and possibilities of local modifications without changing surrounding components [12].

In the following we concentrate on the property of (nested) hierarchies of IT services. We therefore 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 their provided and delivered IT services. The emerging trend to bundle IT service components to IT services leads to "productized" IT services, labeled as IT service products [7,22,23]. It is noteworthy here that IT service products exist on every level of the IT technology stack (Figure 1), dependent on the considered customer. Hence IT service and IT service component are used equivalently, whereas an IT service product is a sellable (composition of) IT service component(s).

IV. IT Service Typology and Process Typology for Describing and Deriving IT Service Components

Before we propose our approach of supporting IT service composition we clarify our semantic understanding of IT service components.

Figure 1 illustrates an overview of so-called abstract IT service components. Abstract IT service components represent the templates for specific instances of IT service components. Each white rectangle represents an abstract IT service component and each grey rectangle groups several abstract IT service components. The four rows on the left side arrange all abstract IT service components along the IT technology stack. The two columns on the right side group abstract IT service components that cannot be assigned to a specific stage of the IT technology stack. These service are commonly referred as "professional services" [19] and consist of activities mainly operated by humans.

Various properties are assigned to every abstract IT service[1] (e.g. the database service is further described by database system capacity, database storage capacity, supported database languages, database conceptual model and so on). The groups of properties[2] are (partly based on [19]):

- Function – encompasses properties that further describe the functional aspects of IT service components (e.g. operating system, database model)
- Security – encompasses security relevant properties (e.g. end-to-end encryption or physical access restrictions to the data center)
- Quality – groups properties that further specify the quality aspects of IT service components. It is noteworthy here both technological quality (e.g. availability, reliability) and organizational quality (e.g. delivery time, support time) are considered.
- Capacity – stores the (technological and organizational) capacity requirements of IT service components (e.g. database capacity in TPC-C, capacity of a specified business unit)
- Qualification – describes the demand of necessary qualification of people (e.g. specified by complexity of the task, level of knowledge,)
- Customer integration – specifies if and how the customer is integrated in the delivery of the IT service component (e.g. integration of customer managed systems, jointly operated processes)
- Organizational structure – sets the delivering business unit and possible points of consumption (e.g. produced at the data center in Arizona, delivered at the office in Chicago)

This mechanism is adapted by concepts of object-oriented programming where abstract classes ensure a higher degree of reusability and define a common interface for all derived subclasses [2]. Moreover abstract classes assign general properties and state which properties must be overwritten. In our approach abstract IT service components – and entire hierarchies of them – are used to specify common behavior of derived IT service components of one specific IT service component type and its specification. The set and hierarchies of abstract IT service components and its assigned properties creates a typology for IT service components. We assume that these properties can be used to define the interface of IT service components, too. This mechanism is shown below.

An abstract IT service component is instantiated by specifying the necessary, assigned properties. For example, an abstract IT service component "database service" is instantiated by setting the database system capacity to x tpmC[3], database storage capacity to y Gbyte and so on. The result is an instance of the abstract IT service component with certain behavior. These instances are called IT services or rather IT service components.

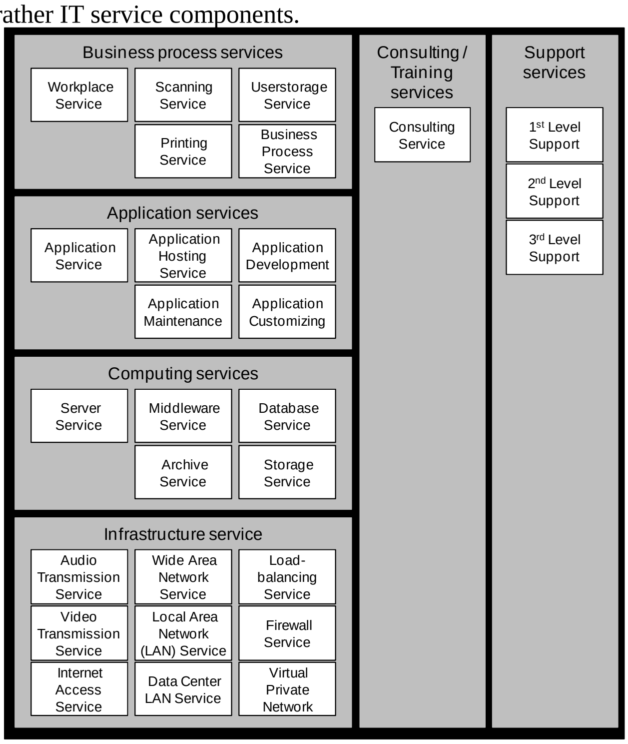

Figure 1: Typology of abstract IT service (components)

The typology (Figure 1) of abstract IT service components concentrates primarily on technological aspects of IT service products although we also integrated the "professional services" as non-technology oriented services. The underlying classification criterion is "required skills" of the delivering business unit. It is obvious that a database administrator generally needs different skills in contrast to a network administrator.

In addition to technological characteristics we previously mentioned that organizational aspects must be considered as well. We therefore introduce a process typology to provide a specification framework. These processes are fairly structured and typically operated in IT organizations while the IT service components are delivered: install/ setup/ register, move, add, change/ upgrade, remove/ dispose, test, backup, repair/ recover [1,6,17,25].

It is noteworthy here that one or more processes of one or more process types can be defined for each instantiated IT service component, i.e. there can be an install and a move process for an IT service component and there can be several

[1] Because of space restrictions we only present groups of properties. Each group consists of many properties.

[2] Common economic properties (e.g. tariffs, price models) are not listed

[3] tpmC is a database benchmark unit (www.tpc.org)

 ISBN: 978-1-61208-116-8

install processes (=process variants) for one IT service component. As IT service components include technological and organizational aspects both a different type of abstract IT service component (technological perspective) and a different process or process variant (organizational perspective) leads possibly to a new IT service component. The service engineer decides whether a new IT service component is created. This chapter has primarily focused on atomic IT service components and how they can be derived. The typologies and properties are used to structure the entire set of IT service components. The next chapter focuses on composite IT service components.

V. Component-Based Construction / Design of IT Service Products

The component-based construction or design of IT service products relies on two design principles: the composition of IT service components and the substitution of IT service components [26]. Adjustments of the internal structure of IT service components are only allowed on copies of the considered IT service components. This restriction bases upon the black-box principle of modularity where IT service components hide their internal structure [2,9,12,14]. We do not deny that several reuse mechanisms can also be applied (e.g. as described in [5]) at the design stage of new IT service components but this contribution focuses on the sole composition of yet developed IT service components, its description and its structured storing.

Precondition for applying the design principles (composition and substitution) is conformity or compatibility of the interfaces of two or more IT service components to be combined. 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 [26]. The composed IT service product finally 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 (e.g. 100 Gbyte of storage are required, the used IT service component only provides 50 Gbyte of storage) or the IT service product inherits the remaining demands. This consequently leads to dependencies between IT service products. The second solution 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 [29]. The demands on IT service product level are fulfilled by preliminary ordered IT service products, e.g. before the installation of a database system the installation of all necessary network and storage services is required. It is therefore inevitable to manage the customer’s preceding bought IT service products in a so-called installed base [6].

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) [29].

IT service component types as well as interface types are specified in help with the previously illustrated typologies and properties. Specifying the interface in depth will restrict and reduce the number of suitable IT service components. For instance, if an application service component requires data encryption an applicable network service component must provide corresponding techniques. All “non-encrypting” network service components cannot be chosen.

As stated before the interface of IT service components must also cover organizational issues. According to Corsten et al. we assume that modularization of IT service components results in the necessity of coordination [14]. Coordination implies the alignment of activities to achieve a common target whereas objective and social 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 [15]. Resource interdependencies emerge if two or more production 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 [24]. This topic is addressed by the properties of capacity assigned to IT service components. Target interdependencies are observed if the contribution to profit of production processes of IT service components is interdependent of the parameterization of other production 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 production processes of two or more IT service components are interdependent. This type of interdependencies leads to several restrictions concerning (1) the execution order of production processes (chronological with / without given sequence or parallel), (2) the compatibility of two production processes (mutual required, mutual exclusive) and (3) possible process variants. We therefore propose to enhance the previously presented definition of interfaces to cope with output interdependencies of IT service components.

First of all we enhance the interface definition to address the first restriction. Each interface is therefore typed with the corresponding process execution restriction, e.g. the demand interface “network service” of a “server service” is typed as “in advance” in order to ensure that an IP address is preliminarily reserved when the “server service” is deployed. For instance a demand interface is typed as “parallel” if an “application service” requires x Gbyte of database storage but the execution sequence of the install processes is negligible as long as all install processes are finalized when the user wants to consume the IT service product.

 ISBN: 978-1-61208-116-8

Secondly we introduce a "*constraining interface*" that describes other IT service components that cannot be composed with the considered IT service component, e.g. it is notpossible to combine a "network attached storage" (a further specified IT service component of type "storage service" in Figure 1) with a "shared server service" (of type "server service"). This addresses the restriction "mutual exclusive" of two production processes whereas the restriction "mutual required" can already be represented by the previously adapted mechanisms of supply and demand interfaces.

The third restriction is addressed by two mechanisms: firstly process properties that determine which variant is chosen are handled in the same way as technological properties, i.e. a demanding IT service component sets the property to its corresponding values. This is often done by requesting a specific quantity or quality. Secondly the decision is delegated to the IT service product and either resolved by establishing interdependencies between IT service products or it must be specified by the service engineer (in the design stage) or the service consumer (when ordering the IT service product). The interdependencies are handled with concepts of variant configuration.

VI. Application of the Proposed Approach on a Company Example

Due to space restrictions, we focus in the following example on the interface specification of the presented approach. It is self-evident that the IT service component typology (Figure 1) and the presented process typology can be used to structure the set of IT service components. The following example is based on data of a leading IT service provider in Europe. For reasons of confidentiality we alienated some quantity and quality data and simplified the interface descriptions to avoid waste of space.

Figure 2 gives an insight how we applied the proposed approach to the interface description of the IT service component of an IT service provider. The presented examples are different variants of a "loadbalancer service" that possess a demand interface and a supply interface. The second row describes possible properties how the interfaces are defined (e.g. LAN Ports, Data center LAN and so on). The substitutability of the defined IT service components is evaluated in regard to the first specified IT service component ("Loadbalancer (substitutable component)"). Each property can be specified with technological and/ or organizational values, e.g. the required "IPs in data center" are characterized by the "technological required quantity" and whether the process of "IP registration" must be operated "in advance" or "parallel" to the installation of the "Loadbalancer service".

The following enumeration explains the evaluation:

- Variant A – substitution prohibited; Reason(s): supplying less service ("virtual server")
- Variant B – substitution prohibited; Reason(s): supply interface is generalization ("not set virtual server") although offering more service and specialized supply interface ("real server")
- Variant C – substitution prohibited: Reason(s): demanding more service ("IPs in data center")
- Variant D – substitution prohibited; Reason(s): demanding more service and demand interface is specialized ("LAN Ports")
- Variant E – substitution possible; Reason(s): demanding less service and less organizational restriction ("IPs in data center", amount and parallel execution of processes possible); supplying more service ("virtual server", "real server"); supply interface is specialized ("real server")

	Demand interface			Supply interface			
Property	LAN Ports	Data center LAN	IPs in data center	Load-balanced LAN	Concurrent Sessions	Virtual Server	Real Server
Loadbalancer (substitutable component)	not set	(1000 Mbit/s | parallel)	(10 | in advance)	1000 Mbit/s	200.000	300	not set
Loadbalancer (Variant A) - prohibited	not set	(1000 Mbit/s | parallel)	(10 | in advance)	1000 Mbit/s	200.000	**200**	not set
Loadbalancer (Variant B) - prohibited	not set	(1000 Mbit/s | parallel)	(10 | in advance)	1000 Mbit/s	200.000	**not set**	**100**
Loadbalancer (Variant C) – prohibited	not set	(1000 Mbit/s | parallel)	**(20 | in advance)**	1000 Mbit/s	200.000	300	not set
Loadbalancer (Variant D) - prohibited	**5**	(1000 Mbit/s | parallel)	(10 | in advance)	1000 Mbit/s	200.000	300	not set
Loadbalancer (Variant E) - possible	**not set**	(1000 Mbit/s | parallel)	**(8 | parallel)**	1000 Mbit/s	200.000	**500**	**100**

Figure 2: Interface defintion and substitutability of IT service components

As illustrated, the approach can be used for the specification of the technological and organizational interface as well as the evaluation of substitutability of two IT service components. "Constraining interfaces" are specified in the same way, e.g. a demand interface of a "network attached storage" possess the property "customer assignment" with the value "shared | mutual exclusive". This parameterization implies that all IT service components that define "shared customer assignment" in their supply interface cannot be combined.

In Figure 2 we combined technological and organizational aspects in one interface. Another possible solution is to split the interface and assign two interfaces to each IT service component.

VII. Conclusion, Benefits, Restrictions and Further Research

As response to the trade-off between customer-orientation and standardization IT organizations and IT service providers decompose their monolithic IT services in modular IT service components. The IT service components are afterwards used to compose and configure IT service products. Customer-specific variants, especially in the area of process-oriented IT service products, leads to an increasing number of IT service components. To realize potentials of standardization and reusability the set of IT service components must be structured to support the IT service engineer in the process of composition.

In this context the presented approach addressed mainly two challenges: the description/ derivation of IT service components and the interdependencies between two or more

IT service components. Both aspects were examined regarding technological and organizational characteristics. The approach can be used to structure the entire set of IT service components. This was also demonstrated on a real world example.

Implementing such an approach will enormously decrease the time-to-market of new IT service products and the time-to-delivery of configurable IT service products because the effort for coordination and communication declines. Furthermore, the quality of IT service products will increase, especially of new IT service products because each demand of an IT service component is explicated and must be satisfied. This leads to a predictable behavior of the composed IT service product.

The approach must be evaluated and enhanced in a real company environment to gain further information on structuring IT service components and its interfaces. The huge number of IT service components can only be managed by corresponding IT systems. A next step will be to design and implement the presented mechanisms.

References

[1] Bailey, J., Kandogan, E., Haber, E., and Maglio, P.P. Activity-based management of IT service delivery. Proceedings of the 2007 symposium on Computer human interaction for the management of information technology - CHIMIT ’07, (2007), 5.

[2] Balzert, H. Lehrbuch der Software-Technik - Software Entwicklung. Heidelberg, 2001.

[3] Bodendorf, F. Daten- und Wissensmanagement. 2005.

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

[5] Brocke, H., Uebernickel, F., and Brenner, W. Reuse-Mechanisms for Mass Customizing IT-Service Agreements. Proceedings of the Sixteenth Americas Conference on Information Systems, (2010).

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

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

[8] Bullinger, H.-J. Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. In Service Engineering. 2006, 3-18.

[9] Burr, W. Chancen und Risiken der Modularisierung von Dienstleistungen aus betriebswirtschaftlicher Sicht. In Konzepte für das Service Engineering. Springer, 2005, 17–44.

[10] Böhmann, T. and Krcmar, H. Servicedatenmanagement für modulare Dienstleistungen. In Betriebliche Tertiarisierung. Duv, 2004, 149-178.

[11] Böhmann, T., Loser, K.-U., and Krcmar, H. Modellierung von Prozessschnittstellen modularer Servicearchitekturen. In Konzepte für das Service Engineering. 2005, 167 - 186.

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

[13] Codd, E.F. A relational model of data for large shared data banks. Communications of the ACM 13, 6 (1970), 377-387.

[14] Corsten, H., Dresch, K.-M., and Gössinger, R. Gestaltung modularer Dienstleistungsproduktion. In Wertschöpfungsprozesse bei Dienstleistungen. Springer, 2007, 95–117.

[15] Ewert, R. and Wagenhofer, A. Interne Unternehmensrechnung. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[16] Fogl, F., Winkler, T., Böhmann, T., and Krcmar, H. MoSES – Baukastensystem für modulare Dienstleistungen Ebenen und Beteiligte. In Konzepte für das Service Engineering. 2005, 85-100.

[17] Garschhammer, M., Hauck, R., Hegering, H.-G., et al. 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), , 719-732.

[18] Grawe, T. and Fähnrich, K.-P. Wissensgestützte Konfiguration komponentenbasierter IT-Dienstleistungen in Wertschöpfungsnetzen. In Content- und Wissensmanagement. 2003, 139-147.

[19] Grawe, T. and Fähnrich, K.-P. Service Engineering bei IT-Dienstleistern. In Entwicklung IT-basierter Dienstleistungen. 2008, 281-301.

[20] Herrmann, K., Klein, R., and The, T.-S. Computer Aided Service Engineering – Konzeption eines Service Engineering Tools. In Service Engineering. 2006, 649-678.

[21] Hevner, A.R., March, S.T., Park, J., and Ram, S. Design science in information systems research. Mis Quarterly 28, 1 (2004), 75-105.

[22] Kaitovaara, P. Increasing Business-Relevancy to the IT Service Product with the Support of Packaging of IT Services. 2001.

[23] Nieminen, P. and Auer, T. Packaging of IT services. 190 (1998).

[24] Office of Governement Commerce. IT Infrastructure Library - ITIL V3 (Service Strategy, Service Design, Service Transistion, Service Operations, Continous Service Improvement). 2007.

[25] Oliveira, F., Nagaraja, K., Bachwani, R., Bianchini, R., Martin, R.P., and Nguyen, T.D. Understanding and validating database system administration. Proceedings of the 2006 USENIX Annual Technical Conferenc, (2006), 213-228.

[26] Overhage, S. Vereinheitlichte Spezifikation von Komponenten: Grundlagen, UNSCOM Spezifikationsrahmen und Anwendung. 2006. http://deposit.ddb.de/cgi-bin/dokserv?idn=983172935&dok_var=d1&dok_ext=pdf&filename=983172935.pdf.

[27] Piller, F. and Waringer, D. Modularisierung in der Automobilindustrie: neue Formen und Prinzipien: modular sourcing, Plattformkonzept und Fertigungssegmentierung als Mittel des Komplexitatsmanagements. 1999.

[28] Pine, B.J. and Davis, S. Mass Customization: The new frontier in business competition. 1993.

[29] Seco, J.C. and Caires, L. A basic model of typed components. ECOOP 2000—Object-Oriented Programming, (2000), 108-128.

[30] Shimomura, Y., Sakao, T., Sundin, E., and Lindahl, M. A design process model and a computer tool for service design. ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Las Vegas, NV, (2007), 1-13.

[31] Zarnekow, R., Brenner, W., and Pilgram, U. Integrated Information Management. 2006.