

Towards a Method for IT Service Management

Heiko Kattenstroth and David Heise

Information Systems and Enterprise Modelling Research Group
University of Duisburg-Essen
Universitaetsstr. 9, 45141 Essen, Germany
{heiko.kattenstroth|david.heise}@uni-duisburg-essen.de

Abstract. To cope with the challenges in IT service management, methods are required that purposefully reduce the complexity inherent to enterprises – with regard to both business and IT –, facilitate communication among groups of stakeholders and support the management of IT services along their life cycle. In this paper we investigate the potentials of an enterprise modelling approach to IT service management and reflect upon design alternatives for corresponding modelling constructs. We present research in progress that is intended as foundation for discussion with and discursive evaluation by peers and domain experts.

Key words: IT service management, service life cycle, domain-specific modelling, enterprise modelling

1 Motivation

Today's IT organisations are confronted with remarkable challenges: First, they have to deal with the tremendous *complexity* of present day enterprises and their IT. A complexity that results from the multitude of IT platforms, networks, and information systems as well as their interrelations. At the same time they are expected to efficiently support the business and to drive innovations, which requires IT managers to account for the multifaceted *dependencies* between IT and the enterprise's goals, organisational structures, and business processes. Furthermore, aligning the IT to the business demands the participation of various stakeholders from business and IT, such as executives, end-users, and IT experts. Each of them has a different professional background, perspective on IT and business, and – even more important – technical language. As a result IT managers have to deal with language barriers, which hamper *communication* and collaboration between the stakeholders and consequently compromise the efficiency of the IT organisation.

In response to these challenges, the concept “IT service” emerged as an innovative approach to manage the relationship between business and IT. In current business practice, IT services are widely considered to be of outstanding relevance for IT management, as they, for example, represent the IT organisation's outcome that is used in the company's business processes and serve as a reference between IT and business stakeholders as well as between the different stakeholders inside the IT organisation. Unfortunately the introduction of this additional

concept seems to have an ambivalent effect: While it is suited to reduce complexity on the one hand, it also adds to the complexity of IT management on the other. IT organisations are, thus, urged to enforce development, implementation, and maintenance of their IT services – the IT service management (ITSM) – in a structured and systematic manner.

To cope with these challenges, methods are required that purposefully reduce the complexity inherent to enterprises – with regard to both business and IT –, facilitate communication among groups of stakeholders, and support the management of IT services along their life cycle. In this respect, several authors highlight the prospects of supporting ITSM through conceptual models in general and enterprise models in particular (e.g., [1, 14]). While current enterprise modelling approaches already provide preliminary conceptualisations for IT services (i.e., provide dedicated abstractions that, among others, reduce the complexity), their support for IT service management is still limited. Against this background we propose to extend a method for enterprise modelling to specifically addresses the requirements and challenges of IT service management. In this paper we present a part of this research in progress: First, we investigate the potentials of an enterprise model-based support for ITSM. In contrast to prior work, we deliberately analyse the potentials throughout the different phases of the service life cycle. And second, we propose corresponding modelling constructs in extension to existing approaches for enterprise modelling. Therefore, we synthesise and reflect upon promising enhancements and discuss design alternatives. The results of our analysis are intended as working drafts and foundation for discussion with and discursive evaluation by peers and domain experts.

In the next section we introduce the ITSM domain. In Section 3, we discuss approaches for enterprise modelling and how they can be used as a foundation for ITSM. The general prospects of such an approach are investigated in Section 4. In Section 5, we reflect on current work on domain-specific modelling constructs for ITSM and discuss design alternatives. Section 6 presents concluding remarks.

2 IT service management: Life Cycle and Requirements

IT service management seems to have evolved to an essential part of IT management that focuses on development, implementation, and maintenance of IT services. While in the last century many authors conceptualised (IT) services as “counterpart” to physical goods and products – applying the criteria of intangibility, heterogeneity, inseparability, and perishability – we follow Vargo and Lusch’s [32] arguments in that a segregation between goods and services cannot be maintained. Rather we see “IT service” as a broad term that covers *any IT-related outcome of the IT organisation* that is used by internal (e.g., business departments) or external customers (e.g., companies at the market). In this diction, the service term incorporates both tangible and intangible aspects and corresponds to ‘product’. Examples for IT services can be found at different levels of abstraction, ranging from fine-granular IT services like “Edit Customer Data” to broad ones like “Providing ERP Application”. A method

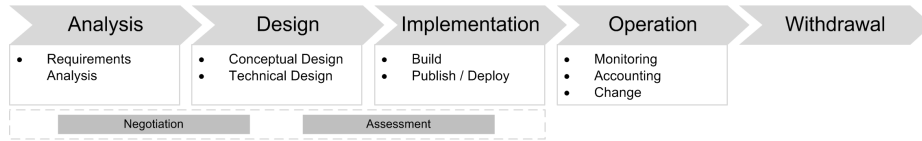


Fig. 1. Idealised life cycle with exemplary activities

for ITSM should therefore provide a conceptualisation of IT service that covers the various, generic interpretations of service, their dependencies and other interrelations.

Moreover, IT service management can be understood as a paradigm that acknowledges the importance of aligning IT with the business (“IT/Business Alignment”). Thus, the key focus of IT management shifted from technology, i.e., hardware and software, to IT as an instrument that aims to support people in creating a business value and is strongly motivated by the demand for cost control and customer orientation. Thereby it seems accepted that IT does not provide a business value per se, but “must be part of a business value creating process with ‘other’ IS and organisational factors operating in a systematic manner” and that it includes “IT people and management, routines, and policies or the organisational system including non IT-people and management, business processes, knowledge assets, relationship assets, culture structure, and policies” [18, p. 26]. Accordingly, IT is not an end in itself. Instead, it is supposed to support an organisation’s business processes, enterprise goals, and – in general – its competitiveness. Hence, adequate management of IT services requires a profound understanding of the organisational context and interdependencies between business and IT. At the same time, ITSM has to account for the technical context, too, since many IT services will be realised by hardware and software. Consequently, a method for IT service management has also to address the technical perspective and provide concepts representing the IT domain.

ITSM frameworks such as ITIL [21] aim at systematically managing the offerings of the IT organisation as services for the business by means of managing the service portfolio. Service portfolio management focuses, amongst others, on reuse of existing services and the strategic development of new offerings and capabilities to serve the customers, which recommends managing IT services during all phases of their life cycle. Moreover, if instruments do *not* support IT service management at all stages of the life cycle – i.e., different instruments have to be applied to cover the different phases – negative effects such as lack of integrity, loss of information, and frictions in communication are likely that finally will hamper the efficiency of IT service management in general. Based on the work from Jin and Ray [15], Kohlborn et al. [17], and Riedl et al. [24], Figure 1 illustrates a synopsis of IT service life cycle phases along with exemplary activities. A method for ITSM thus has to provide support for all phases.

Table 1 summarises the requirements for a method that aims to support IT service management. Based on our experience with ITSM frameworks, methods, and tools, we conclude that there is currently no instrument or methodical sup-

<i>Requirement</i>
<i>Req. 1</i> – Reduce complexity inherent to ITSM domain
<i>Req. 2</i> – Foster communication among business and IT stakeholders
<i>Req. 3</i> – Provide concepts for IT services and their interrelations
<i>Req. 4</i> – Account for the organisational context
<i>Req. 5</i> – Offer concepts of the IT domain (technical perspective)
<i>Req. 6</i> – Provide support over entire service life cycle

Table 1. Requirements for a method in support of IT service management

port available that addresses all requirements. For instance, the ITIL framework provides high-level guidelines for IT service management processes and explicitly spans the service life cycle, but does neither address the business context nor provide concepts at detailed technical level; approaches like “Model Driven SOA” [35] support the technical design and implementation of (web) services, but do hardly address the complexity of the ITSM domain, foster communication with business or account for organisational context. Our research is therefore based on the assumption that both demand and potentials exist for methods and tools that fulfill the described requirements and support IT service management.

3 Conceptual Background

Enterprise modelling (EM) approaches such as ARIS [25], MEMO [11] or ArchiMate [19, 29] seem to provide a promising foundation for several reasons: (1) They serve to structure an enterprise by providing purposeful abstractions of IT and the surrounding action systems (*Req. 1*). (2) They, therefore, make use of domain-specific modelling languages (DSMLs), thereby offering a consistent, intuitive and semantically rich modelling [16]. (3) They usually build upon an elaborate meta model that already includes organisational and technical context, such as for goal, business process or organisational structure modelling on the one hand and for modelling the IT organisation on the other (*Req. 4 & 5*). (4) They provide stakeholders with specific and illustrative views on a company at various organisational levels, such as on value chains, business processes, or IT landscapes. Since the DSMLs are seamlessly integrated, they allow for comprehensive analyses fostering IS/Business Alignment and communication among business and IT stakeholders (*Req. 2*). Figure 2 illustrates an example of an enterprise model based on the MEMO family of modelling languages. The excerpt shows models at different levels of an organisational hierarchy, which is intended to address different stakeholders: A value chain and goal model at strategic level, a business landscape and business process model at organisational level, and models of the IT organisation at IT level.

In this regard, some approaches to enterprise modelling already provide preliminary meta types representing an “IT service”, though their conceptualisations of the (IT) service term, the resulting meta types and the corresponding

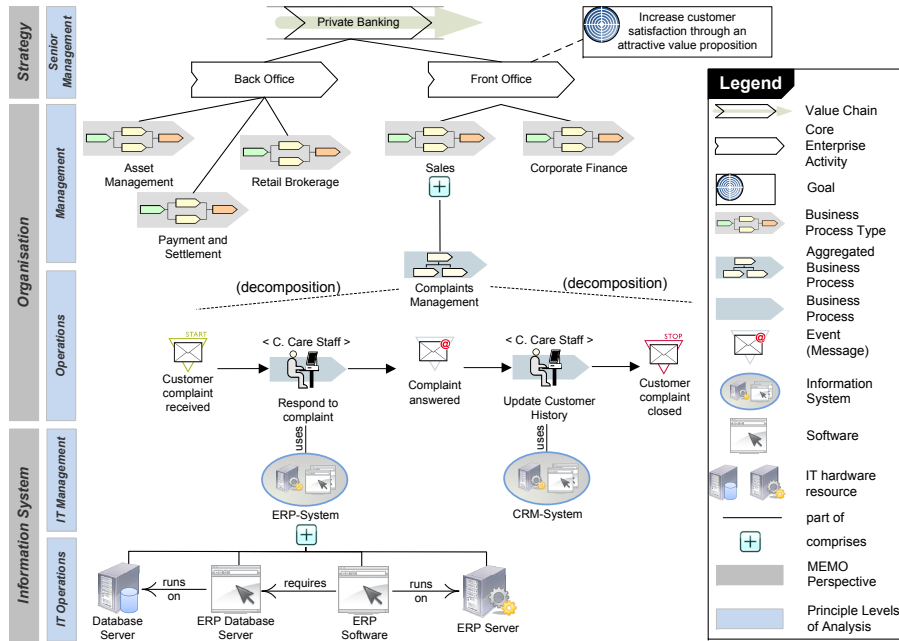


Fig. 2. Example of an Enterprise Model that integrates models of an IT landscape with models of associated business processes from various perspectives

associations vary notably. Furthermore, these concepts have usually not been developed with dedicated ITSM support in mind. Moreover, to our best knowledge, approaches that propose an enterprise modelling-based approach as foundation for IT service management, such as Braun and Winter [1] and Frank et al. [14], do not illustrate how to support IT service management in detail, for instance, for service portfolio management.

At the same time, various authors outside the EM community propose specific DSMLs – at least in the broadest sense – for conceptual modelling of services. For instance, Quartel et al. [22] introduce modelling concepts and a preliminary notation that allow to model services as sequences of user-provider interactions. Hence, they conceptualise services as counterpart to products and goods; technical or organisational context is not considered. An approach that specifically addresses *IT* service management is presented by Correia and Abreu [3, 4] as work in progress. They suggest a DSML that aims to support the management of service level agreements over their life cycle; a meta model is not (yet) provided. A meta model for ITSM is, instead, provided by Uebernickel et al. and Ebert et al. respectively [7, 31]. Their metatype “IT Service” is embedded in organisational and technical context in that it is associated to meta types representing business processes and users as well as IT resources (e.g., “Information”, or “Hardware”). It seems their meta model is not intended to be used for graphical modelling, since they do neither discuss a notation nor provide examples of

how to apply the concepts. As attributes are missing from the meta model, the semantics of its concepts are mainly up to interpretation.

Against this background, the paper at hand extends prior work in two aspects: (1) We exemplarily illustrate how enterprise modelling can support ITSM during different phases of the life cycle; (2) we enhance current work by synthesising, reflecting, and enhancing current language specifications for IT service management as well as discussing corresponding design alternatives. Our research follows a design research process that is grounded on a research method configured for the epistemological particularity of research on modelling methods (for a detailed description, see [10] and [27]) and applies a method for designing domain-specific modelling languages [12].

4 Outline of EM-based IT service management

This section illustrates the prospects of supporting life cycle spanning IT service management with a dedicated method, which is based on a domain specific modelling language and is integrated with an approach for enterprise modelling as described in Sec. 3. With respect to space restrictions, the example is limited to selected activities of the life cycle. The application scenario, as well as the specification of the design alternatives in Sec. 5, are based on the MEMO approach. Thus the present models are illustrated presuming modelling languages and notation of the MEMO language family, for instance, for strategic, organisational, and IT landscape modelling [11, 14] as well as specific extensions like for risk or indicator modelling [27, 28]. It is, however, important to note that shown diagrams and meta models are not intended to predetermine a specific enterprise modelling approach; instead, they serve as an illustration of principle application and design decisions in context of enterprise model-based ITSM.

Figure 3 illustrates the application scenario: In the *analysis* phase the desired functionalities, constraints, and expectations of the business stakeholders have to be captured as requirements. With regard to *Req. 4* it seems promising to use models of the organisational context (e.g., business processes, organisational structure, and business goals) since they provide an overview of the business processes the IT service will be used for and the business goals (and restrictions) it aims to support. These models are enriched with a description of the required functionalities (see ① in Fig. 3). In contrast to common requirements engineering approaches such as [5, 34] these requirements are *integrated* with the enterprise models in that they are associated to the concepts they affect. The required functionalities are then structured as potential IT services (“service candidates”) based on their scope, purpose and similarities. To foster reuse, the resulting candidates are mapped against existing IT services, and the functionalities they offer, using the IT service model (②). The IT service model contains the services and their functionalities, dependencies (such as “uses” or “requires”) as well as similarities (“similar to”) and specialisation/generalisation relationships between IT services. In this regard, the objective of this service mapping

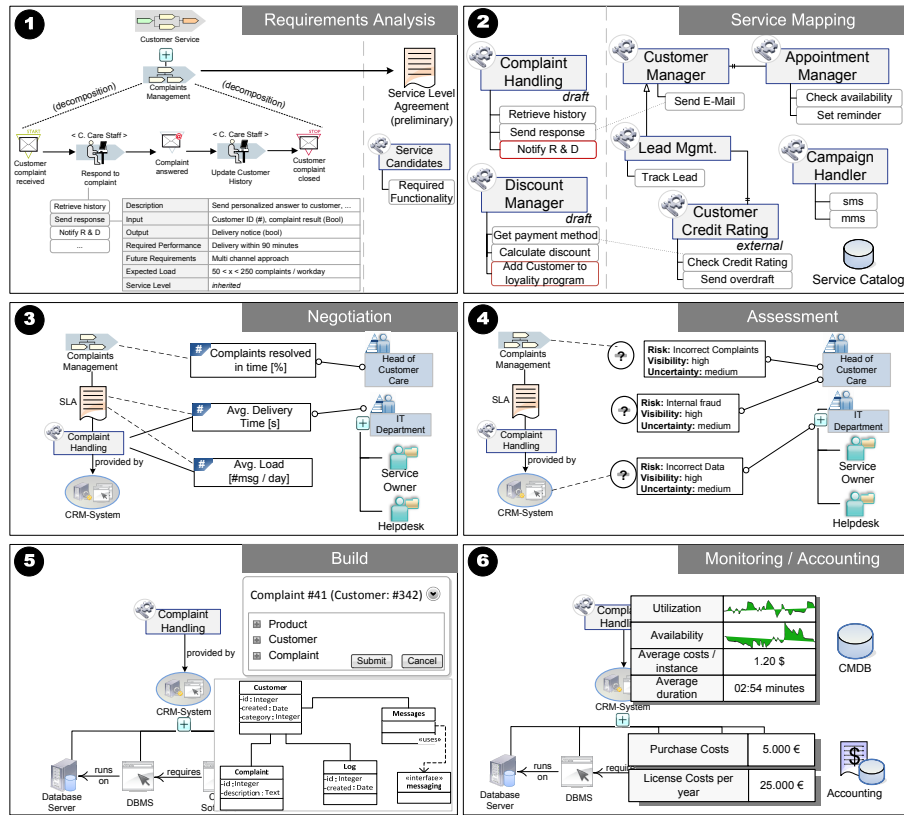


Fig. 3. Example for a model based ITSM

is to identify services that can be reused, extended or adapted to fulfill the requirements. This allows for a comprehensive and sustainable management of the IT service portfolio. To support the negotiation between service provider and service customer, service level agreements act as contracts that specify responsibilities, additional IT services (e.g., a help desk) and, in general, their expected quality. For this purpose the IT service model is enriched with, for instance, (key performance) indicators and corresponding thresholds the service provider has to comply with (3). Enterprise models provide an overview of potential sources from which risks originate and thereby support risk identification. The identified risks and corresponding cause-and-effect relations can be added to the models (4). This allows for evaluation and prioritisation of risks related to IT services, e.g., to make decisions on their treatment and to implement appropriate measures. For the implementation of new IT services and functionalities, the IT service model – along with corresponding SLAs and architectural and implementation standards – is used for the conceptual and the technical design of the prospective service. Furthermore, the model can be enriched with implementation details, e.g., about hardware, software, or network resources as

well as data models, interfaces, and GUI drafts (⑤). Again, models promise to foster the communication, in this case between different groups in the IT department and the “service owner”, for example, to get feedback and to ensure the alignment between business requirements and service implementation. For higher productivity, models can also be used to generate (parts of) the service realisation such as interface descriptors, schema for databases, and program code [16]. Once deployed and running, the IT services need to be monitored, maintained and evaluated. In this context, Frank et al. [13] employ the idea of using *models at run-time*. Accordingly, the service model is used for operations in the run phase (⑥), for instance, to enforce quality and availability as guaranteed in the SLA. Whereas models in the plan and build phases remain on type level, the models in the run phases are enriched with data about the service execution (i.e., instance data). Thus the models are used as frontend to instance data (“dashboard”), which promotes a more proactive IT service management. Presuming an integration with corresponding operational information systems, these models could also be enhanced by their running costs, current availability and utilisation as well as other relevant information.

5 Considerations on Method Design: Reflections on Service Modelling

Development of a method for IT service management requires reconstructing the key concepts of the domain as modelling concepts and providing a process model that guides the application of these concepts. The key domain-specific concepts seem obvious: The method has to provide concepts that represent IT services (*Req. 3*), the business context the IT services are used in (*Req. 4*) as well as concepts that represent the underlying IT infrastructure (*Req. 5*). The conceptualisation in detail, however, depends on design decisions between alternative specifications. In this section we therefore discuss considerations on the language specification and design alternatives (Sec. 5.1), which are based on existing conceptualisations, and outline open issues for a method for IT service management (Sec. 5.2). The specifications are presented as metamodel excerpts in notation of the MEMO Meta Modelling Language [9]. The meta models are intended as working drafts and foundation for discussion with and discursive evaluation by peers and domain experts. The reuse of modelling concepts from existing modelling languages in the MEMO language family is visualised by a colored rectangle attached to the meta type header (as suggested in [9]).

5.1 Outline of a Metamodel: Design Alternatives

IT service concept. Pertinent literature, ITSM approaches and ITSM standards such as [1, 2, 21, 23] apply a generic definition wherein an IT service is used as an abstraction over any (IT-related) outcome of the IT organisation that is used by a customer. This is mainly for one reason: A vague definition of “IT service”

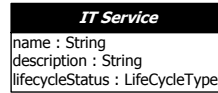


Fig. 4. Basic Conceptualisation of IT service

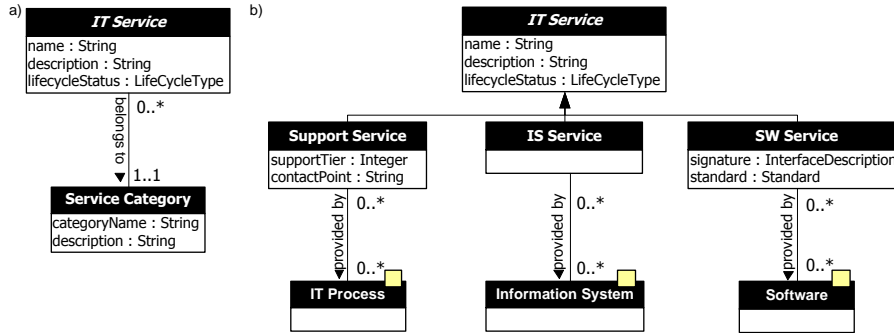


Fig. 5. Design alternatives for IT service

enables stakeholders from different hierarchical levels (e.g., management vs. operations), organisations, and domains (e.g., business vs. IT) to communicate *without* clarifying their (maybe different) conceptualisations in detail. Current modelling languages like [1, 14] seem to acknowledge this practice and offer a generic service concept as illustrated in Fig. 4: A meta type *IT service* serves to describe a service at least by means of a name, a description and its status in the life cycle (such as “planned”, “in implementation”, “operated” or “inactive”).

However, despite the advantages of such a broad service concept it also comes with various deficiencies: Though stakeholders might be able to communicate for a start, it bears the risk of (maybe unrecognised) misunderstandings, especially if stakeholders apply a different conceptualisation of IT service in detail; and since it does not provide much semantic, it does not allow for a specific support throughout the different life cycle phases as illustrated in Sec. 4. Thus it seems reasonable to provide a more specific conceptualisation that accounts for different connotations of “IT service”. We reason that the IT-related service term is used on at least three different levels of abstraction (for a similar conclusion, see [22]). On the one hand an IT service is used as an abstraction over *processes* performed by the IT organisation. Examples for such IT services are “Help Desk” or “PC installation” (e.g., [30]). On the other hand, it refers to a more technical conception with a focus on features that are provided from *information systems*, like “Invoice Printing” or “Manage Customers” offered by an ERP system. Finally, the IT service term is used to describe functions (i.e., operations) of *software*; prominent examples are services in context of SOA as in [8] (e.g., “Retrieve Customer” or “Add Customer”).

Against this background, we recommend providing a differentiated conception of IT service in the modelling language as well. One solution would be to offer

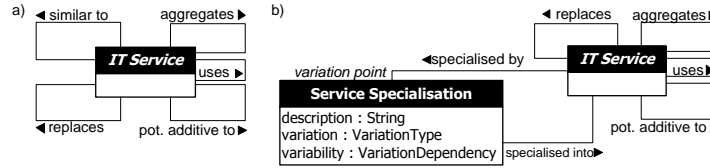


Fig. 6. Design alternatives for service relations (cardinalities & constraints omitted)

a meta type for *IT service categories* that allows organisations to define categories of IT services, such as “IS-based services” or “software-based services” (cf. fig. 5a). Though this solution supports stakeholders in interpreting the semantic of IT service conceptualisations, it does still not provide service-specific semantic and thus no support for tool-based analyses or code generation. Figure 5b illustrates a design alternative that reconstructs the different types of IT services *as meta types* (a similar conceptualisation has been chosen in ArchiMate [29]). In contrast to the service categorisation this alternative offers semantically rich concepts, since it allows to (1) specify IT service type-specific attributes and to (2) associate the meta types to other meta types that further describe them, e.g., associate *IS-based services* to a meta type that represents information systems etc. This promotes a more precise modelling, enables tool-supported analyses and supports code generation, e.g., for software services.

IT service relations. Analysing and managing IT services recommends to account for the various relations between IT services. For instance, Braun and Winter [1] suggest an *isPartOf* relation between services; Brocke et al. [2] differentiate the service meta type into core services and associate services, that customise the core services; while Weigand et al. [33] classify services into, e.g., core, enhancing, complementary and support services. Since this distinction might be a matter of perspective – while one customer regards a service as complement, others might consider it core – we refrain from such separation in the language specification; instead we regard them as context specific *roles*. However, it might be promising to introduce an association *potentiallyAdditiveTo* between IT services. Following the proposal from Ebert et al. [7], we also suggest to differentiate the *isPartOf* relation: On the one hand an IT service might be an abstraction over other IT services, without providing own functionality (*aggregates*). On the other hand an IT service might, in turn of its execution, optionally or mandatory revert to other IT services (*uses*). In extension to current work, two further relations seem promising: In support of the withdrawal of IT services an IT service can be dedicated to *replace* other IT services; and an IT service can be *similarTo* other IT services, which might be feasible especially for larger corporations with several local IT organisations. Figure 6a illustrates the design alternative.

With respect to ongoing research in product modelling (see, e.g., [26]), one design alternative seems worthwhile that builds upon the idea of generalisation/specialisation. IT services that are similar to a large extent are generalised to an IT service that describes their common features; additionally, features

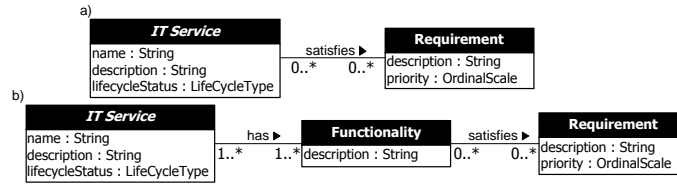


Fig. 7. Design alternatives for service functionalities

that vary among the specialised IT services are explicitly modelled as “variation points” (fig. 6b). This alternative contributes to transparency and fosters communication with business stakeholders, as it supports abstracting over all the different variations. Moreover, it specifically promotes a systematic and elaborate portfolio management of IT services. However, although fig. 6 indicates corresponding language specifications (and, for instance, ArchiMate already offers service specialisation), the semantic of service specialisation, service variation points and similar-to relations has not been clarified yet and is subject to further research. As for similar-to relations, we assume that this issue might – at least partly – be addressed by particularising IT services into *functionalities*.

Functionality. An IT service is used as abstraction over IT processes, information systems and software. At the same time an IT service is used to describe and bundle functionalities that address customers’ requirements. Two design alternatives are feasible to represent a service’s functionality. First, the concept of functionality is considered being part of the IT service, esp. the description attribute (fig. 7a); this alternative conforms to most current state of the art in EM approaches, e.g., [14, 29]. Second, a dedicated meta type represents the functionalities of an IT service (fig. 7b), as suggested by Brocke et al. [2]. The first alternative requires organisations to formulate their services in a specific, functionality-oriented manner and, thus, restricts to describing one functionality. As a result, an IT service that offers several functionalities has to be modelled by using the above mentioned aggregation mechanism, i.e., the IT service has to comprise several other IT services that each provide one of the required functionalities. The second alternative, in contrast, supports to (1) assign several functionalities to one IT service and (2) allows to separate a general, maybe more business-intuitive description of the IT service and the description of the functionality. Furthermore, a separate modelling of functionalities might be a prerequisite to using service-specialisation and variation points – though this depends on the semantic of these concepts.

Business context. To account for the organisational context IT services are used in, we propose the integration with existing modelling languages, e.g. for business process and goal modelling. Besides “traditional” concepts such as business processes, goals, and organisational structures, we further propose to integrate concepts for risks and indicators as these aspects play a crucial role for today’s IT management (fig. 8): Against this background, we suggest integrating the IT service concept with concepts representing risks [28] and indicators [27].

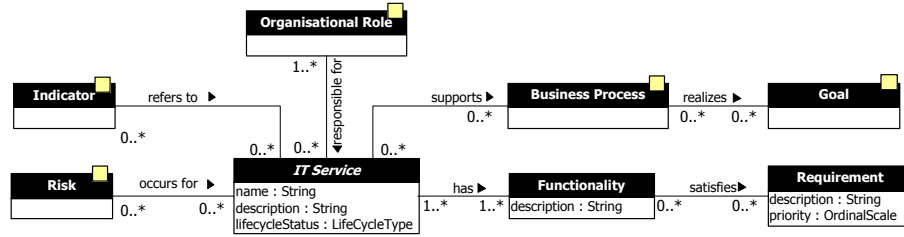


Fig. 8. Business context for IT services

Service level agreement. The relations between a service provider, the IT services it provides and the customers of these IT services are often contractually agreed in service level agreements (SLAs). At least two alternatives seem feasible: First, SLAs are considered to govern IT services in that IT services have to be provided with the agreed quality (fig. 9). Similar conceptualisations have been proposed in [1, 6]. Since a SLA might be specific for the relation between service provider and service customer, a second design alternative is to refine the association between IT service and business process, which acts as a surrogate for the service customer (fig. 9). Whereas Moser and Bayer [20] propose to associate SLA with an IT service and a business unit, we use the concept for business processes as this more precisely describes the point of use of the service.

In both alternatives a SLA is constituted, amongst others, by a set of indicators, responsible organisational roles, and goals. Usually these aspects will be a subset of indicators, roles, and goals of the IT service the SLA refers to, which recommends corresponding constraints to assure consistency in the model. Additionally a SLA can contain billing details for the IT service (e.g., *costs*). Although the second solution promotes a more precise definition of the relation between IT service and business process and thereby of a SLA, this alternative might be counterintuitive for users that are accustomed to assign IT services directly to a business process.

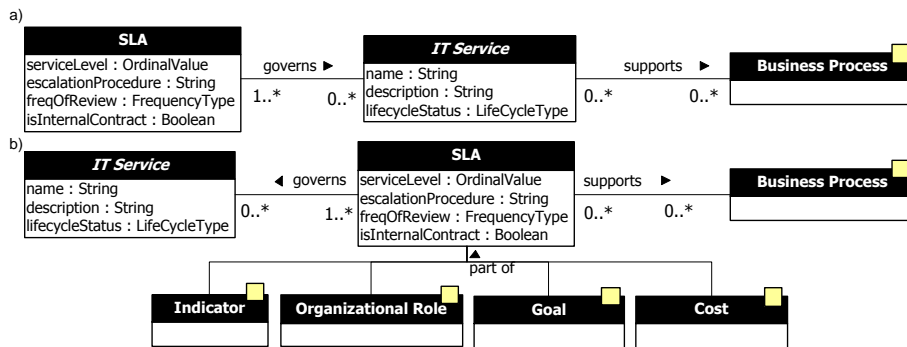


Fig. 9. Design alternatives for service level agreements

5.2 Open Issues & Research Agenda

The presented reflections on IT service modelling already indicate challenging questions that need to be investigated. Although we presented alternative design decisions and, as far as reasonable, decision criteria and rationales, we highlighted just a selection of open questions. Development of an enterprise modelling method for IT service management requires considering further design issues. Among others, open research questions are:

- Some authors introduced a differentiation of IT services and IT products (e.g., [7]). While this separation seems reasonable – a product bundles services to be sold, thus applying a marketing perspective – it entails a problem if the specialisation/variation point concepts (cf. fig. 7b) are applied: Assuming that these concepts make sense for product modelling, too, the relation between product, specialised products, IT services and specialised IT services has to be investigated.
- In fig. 5 we assumed a meta type representing IT processes. However, yet there seems no consensus – nor analysis – whether there is need for a specific meta type or if existing modelling concepts and languages for business processes can be reused to also model IT processes.
- With regard to the requirements analysis it needs to be investigated (1) to what extent requirements can be derived from process models and (2) how additional details such as capacity requirements and anticipated changes can be added to process models and used in later life cycle phases.
- Similar and reusable services (and functionalities) can only be identified if they are described *consistently* and at the same level of abstraction – at best using (semi-)formal concepts that support tool-based comparison. In context of enterprise modelling such concepts and corresponding heuristics are, to our knowledge, subject to future research.
- Finally, modelling SLAs, as indicated in fig. 9, recommends accounting for juridical aspects, too. However, we are not aware of work that addresses the formalisation juridical aspects, yet.

Beside issues pertaining the language specification, the application of the presented modelling concepts needs to be embedded in a process model that envisions the use of the concepts; in addition, heuristics and guidelines are recommended. These aspects were not in focus of this paper.

6 Conclusions

This paper investigates the potentials of an enterprise modelling based method in support of IT service management. In this regard, our contribution in this paper is twofold: First, we illustrate consistent support for ITSM over the entire life cycle of IT services. Second, we prepare for further research on such a method by reflecting key considerations and decisions pertaining to IT service modelling. Therefore, we focus on language specification and discuss design alternatives

as enhancements to existing approaches as research in progress. However, developing a method requires further considerations, for instance, guidelines, an accompanying process model, and support of a modelling tool. Since corporate culture and the distribution of power in an organisation have an impact on IT management [23, 30] it should also be investigated, whether these aspects, which apparently resist against formalisation, could be somehow accounted for in the method, too.

References

1. C. Braun and R. Winter. Integration of IT service management into enterprise architecture. In *Proc. of the 2007 ACM Symposium on Applied Computing (SAC '07)*, pages 1215–1219, New York, NY, USA, 2007. ACM.
2. H. Brocke, F. Uebernickel, and W. Brenner. Managing the current customization of process related it-services. In *Proc. of the 43rd Hawaii Int. Conf. on System Sciences (HICSS-43)*, Koloa, Kauai, Hawaii, 2010.
3. A. Correia and F. B. e. Abreu. Integrating IT Service Management within the Enterprise Architecture. In *4th Int. Conf. Software Engineering Advances (ICSEA '09)*, pages 553–558, Los Alamitos, Calif, 2009. IEEE.
4. A. Correia and F. B. e. Abreu. Defining and Observing the Compliance of Service Level Agreements: A Model Driven Approach. In *Proc. of the 7th Int. Conf. on the Quality of Information and Communications Technology*, pages 165–170, Washington, DC, USA, 2010. IEEE.
5. A. M. Davis. *Software requirements: Objects, functions, and states*. PTR Prentice Hall, Englewood Cliffs, N.J, rev. edition, 1993.
6. M. Debusmann, R. Kroger, and K. Geihs. Unifying service level management using an MDA-based approach. In *IEEE/IFIP Network Operations and Management Symposium (NOMS '04)*, pages 801–814, Piscataway, N.J, 2004. IEEE.
7. N. Ebert, F. Uebernickel, A. Hochstein, and W. Brenner. A service model for the development of management systems for it-enabled services. In *Proc. of the 13th Americas Conference on Information Systems (ACIS '07)*, 2007.
8. T. Erl. *Service-oriented architecture: Concepts, technology, and design*. Prentice Hall Professional Technical Reference, Upper Saddle River, NJ, 2005.
9. U. Frank. The MEMO Meta Modelling Language (MML) and Language Architecture. Technical report, ICB, University Duisburg-Essen.
10. U. Frank. Towards a Pluralistic Conception of Research Methods in Information Systems Research. Technical report, ICB, University Duisburg-Essen.
11. U. Frank. Multi-Perspective Enterprise Modeling (MEMO): Conceptual Framework and Modeling Languages. In *Proc. of the 35th Hawaii Int. Conf. on System Sciences (HICSS-35): Honolulu*, 2002.
12. U. Frank. Outline of a Method for Designing Modelling Languages. Technical Report 42, ICB, University Duisburg-Essen, 2010.
13. U. Frank, D. Heise, and H. Kattenstroth. Use of a domain specific modeling language for realizing versatile dashboards. In *Proc. of the 9th OOPSLA workshop on domain-specific modeling (DSM '09)*, Orlando, Florida, USA, 2009.
14. U. Frank, D. Heise, H. Kattenstroth, D. Ferguson, E. Hadar, and M. Waschke. ITML: A Domain-Specific Modeling Language for Supporting Business Driven IT Management. In *Proc. of the 9th OOPSLA workshop on domain-specific modeling (DSM '09)*, Orlando, Florida, USA, 2009.

15. K. Jin and P. Ray. Business-oriented development methodology for it service management. In *Proc. of the 41st Hawaii Int. Conf. on System Sciences (HICSS-41)*, page 99, Waikoloa, Hawaii, USA, 2008.
16. S. Kelly and J.-P. Tolvanen. *Domain-specific modeling: Enabling full code generation*. Wiley-Interscience and IEEE Computer Society, Hoboken, N.J, 2008.
17. T. Kohlborn, A. Korthaus, and M. Rosemann. Business and software service life-cycle management. In *13th Int. Conf. on Enterprise Computing (EDOC '09)*, Auckland, New Zealand, 2009.
18. R. Kohli and V. Grover. Business Value of IT: An essay on expanding research directions to keep up with the times. *J of the AIS*, 9(1):23–39, 2008.
19. M. Lankhorst. *Enterprise architecture at work: Modelling, communication and analysis*. Springer, 2005.
20. C. Moser and F. Bayer. IT Architecture Management : A Framework for IT Services. In *Proc. of the Workshop on Enterprise Modelling and Information Systems Architectures (EMISA)*, pages 137–151, 2005.
21. Office of Government Commerce. *ITIL – Service Design*. London, 2007.
22. D. A. C. Quartel, M. W. A. Steen, S. Pokraev, and M. J. Sinderen. COSMO: A conceptual framework for service modelling and refinement. *Information Systems Frontiers*, 9(2-3):225–244, 2007.
23. T. Rands. Information technology as a service operation. *J of Information Technology*, 7(4):189–201, 1992.
24. C. Riedl, T. Böhmman, M. Rosemann, and H. Krcmar. Quality management in service ecosystems. *ISeB*, 2(7):199–221, 2009.
25. A.-W. Scheer. *Architecture of integrated information systems: Foundations of enterprise modelling*. Springer, Berlin, 1992.
26. M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch. Covamof: A framework for modeling variability in software product families. In *Proc. of the 3rd Software Product Line Conference (SPLC04)*, San Diego, CA, 2004.
27. S. Strecker, U. Frank, D. Heise, and H. Kattenstroth. MetricM: A Modeling Method in support of the Reflective Design and Use of Performance Measurement Systems. *ISeB*, 2011.
28. S. Strecker, D. Heise, and U. Frank. RiskM: A Multi-Perspective Modeling for IT Risk Assessment. *Information Systems Frontiers*, 2010.
29. The Open Group. *ArchiMate 1.0 specification*. Van Haren Pub., Zaltbommel, 2009.
30. J. J. Trienekens, J. J. Bouman, and M. van der Zwan. Specification of service level agreements: Problems, principles and practices. *Software Quality J*, 12(1):43–57, 2004.
31. F. Uebernickel, C. Bravo-Sanchez, R. Zarnekow, and W. Brenner. Is service-engineering: A process model for the development of is services. In *European and Mediterranean Conference on Information Systems (EMCIS)*, 2006.
32. S. L. Vargo and R. F. Lusch. The 4 service marketing myths: Remnants of a goods-based, manufacturing model. *J of Service Research*, 6(4):324–335, 2004.
33. H. Weigand, P. Johannesson, B. Andersson, and M. Bergholtz. Value-Based Service Modeling and Design: Toward a Unified View of Services. In *Proc. of the 21st Int. Conf. on Advanced Information Systems Engineering (CAiSE '09)*, 2009.
34. R. Wieringa. *Requirements engineering: Frameworks for understanding*. Wiley, Chichester, New York, 1996.
35. L.-J. Zhang, N. Zhou, Y.-M. Chee, A. Jalaldeen, K. Ponnalagu, R. R. Sindhgatta, A. Arsanjani, and F. Bernardini. SOMA-ME: A platform for the model-driven design of SOA solutions. *IBM Systems J*, 47(3):397–413, 2008.