The Next Generation – Design and Implementation of a Smart Glasses-based Modelling System

Dirk Metzger[*,a], Christina Niemöller[a], Sven Jannaber[a], Lisa Berkemeier[a], Lukas Brenning[a], Oliver Thomas[a]

[a] Osnabrück University, Information Management and Information Systems, Osnabrück, Germany

Abstract. *Technical services in innovative business models are becoming increasingly complex. Thus, comprehensive IT-support is crucial for service delivery. Content for those IT-support systems is captured by modelling relevant service processes. Aside from how intangible and integrative they are, services are characterized by the complexity of their structure. Consequently, the traditional modelling approaches executed by modelling experts are challenging. To overcome those challenges, we developed a concept to model service processes at the point-of-service while executing the service itself. The process executer (e. g. the technician) is empowered by smart glasses that do not limit his scope of actions. Additionally, the glasses guide through the (runtime) modelling and allow easy capturing of service processes during the execution. We followed a design science-oriented approach. First, we identified relevant process blocks from literature for runtime modelling (analysis). Afterwards, we built related software components for the process blocks (design). We do so by proposing an implementation and an architecture for a smart glasses-based modelling system. Finally, we evaluated the concept by prototyping and demonstrating the system by means of a real-world service process (evaluation). Our approach tackles challenges on how new technology can enhance the modelling at the point-of-service, which process blocks are relevant and how domain experts can be integrated into the modelling process itself. The practical implications are geared towards new chances of capturing processes.*

Keywords. Smart Glasses • Modelling • Design Science Research • Domain Experts

Communicated by M. Nüttgens. Received 2017-03-10. Accepted after 2 revisions on 2018-06-10.

1 Introduction

Services become an increasing catalyst for innovative business models; service models become the crux of the matter for the design of service support systems (Thomas and Nüttgens 2012). Technical services, especially services like maintenance, become more complex, therefore making their deliveries inevitable without adequate support through IT (Becker and Neumann 2006; Matijacic et al. 2013). This IT-support should be process-oriented and meet the information needs of the service provider (Däuble et al. 2015; Matijacic et al. 2013). A sufficient device to support technicians during service tasks are smart glasses as they display information in the user's field of sight and allow a hands free navigation (Niemöller et al. 2017). To fill those process-oriented service support systems, it is necessary to model the service processes. The service characteristics, like the potential-, process- and result dimension (Scheer et al. 2006), as well as the industrial character of technical services (Becker and Neumann 2006),

* Corresponding author.
E-mail. dirk.metzger@uos.de
The research and development presented in this paper is part of the project Glassroom and is funded by the German Federal Ministry of Education and Research (BMBF, grant number 01PD14014A)
Note: This work is based on Metzger et al. (2016a).

imply specific requirements to the modelling process. Beside immateriality and interactivity, those services are difficult to model ex-post. This is due to the complexity of the dynamic process structure and the complex fault detection trees. Additionally, domain experts have the knowledge of the functional execution of the process, while the modelling experts have the knowledge for methodical process recording (Pendergast et al. 1999). Therefore, it is accompanied by a high resource expenditure and problems occur while joining the different knowledge bases (Riemer et al. 2011). To overcome the complexity, different approaches to simplify the modelling process are discussed in literature. For example, tool-assisted, collaborative modelling to involve different stakeholders (Riemer et al. 2011) or the simplification of modelling languages for novices (e. g. Becker et al. 2007; Recker et al. 2010; Wilmont et al. 2010) are suggested. However, those approaches do not solve the problem that the process sequence itself is difficult to capture and to reflect. Further, process sequences are frequently subject to change. Therefore, much thought was given on the modelling of processes in a flexible way (build time vs. runtime) (Weber et al. 2008). The following approach uses the idea of modelling at runtime and, in addition, faces the problem, that (a) the process does not have to be reflected and discussed with the modeller ex-post, but is recorded by the executor at the time of execution. (b) The process is recorded with the same technology that is used later on, when the process is supported with additional information. The artefact of this paper is the result of a design science-oriented research approach (Österle et al. 2011): a smart glasses-based system for runtime modelling of technical customer service (TCS) processes. To evaluate whether the developed design is feasible, we use prototyping and a demonstration case (Riege et al. 2009; Sonnenberg and vom Brocke 2012).

To guide our work we outline the following research questions. The first is based on the analysis of implications for modelling TCS processes on the scientific knowledge base of service science. The IHIP-characteristics of services in general, but also their complexity, due to the service processing primarily performed at machines or at plants, are investigated. We analyze the functional differentiation of TCS and the subsequent specific representation form of TCS processes, leading us to the first research question

- RQ1: What are the implications for modelling service processes based on the characteristics of TCS?

The analysis lead to a mobile, hands-free system used by the domain expert for modelling the excecuted process in realtime such as smart glasses. Based on that, we investigate further

- RQ2: How should a smart glasses-based modelling IS be designed to support TCS by modelling their processes in realtime?

This overarching question can be broken down into the following sub-questions, which will be addressed in this paper. The first prerequisite for business process modelling is the choice of an adequate business process modelling language (Weske et al. 2004). Second, in Business Process Management (BPM) literature, modelling conventions have to be applied in order to restrict or adjust business process modelling languages (BPML) towards a given modelling environment (Thomas and Scheer 2016). Naturally, process modelling within a smart glasses modelling environment requires the definition of a language convention that is tailored towards the specific characteristics of the intended target domain smart glasses e. g. small display size, limited interaction capabilities). Thus, the research question addresses a process modelling language convention for smart glasses:

- RQ2a: What kind of language constructs have to be implemented in a smart glasses-based modelling IS?

As no blueprint for a smart glasses-based modelling system exists, we examine an implementation and possible system architecture. Both are based on the previously described language constructs and characteristics, and further detail the proposed modelling concept by outlining the integration of a process modelling recommender system, which provides support with labeling suggestions and thus simplifies the run-time modelling task for end-users. To sum up, the final question is:

- RQ2b: What are the necessary components and how should the architecture be designed and implemented?

With this paper, we contribute to the knowledge base of Service Systems Engineering, BPM and the Design of IS. We do so, by (a) examining the requirements that arise in regards to modelling with the characteristics of services, (b) proposing which process modules have to be taken into account for runtime modelling and (c) how a mobile IT-system has to be designed and implemented in order to enable the documentation and modelling by the process executor in realtime. In practice, this results in new possibilities of process recording that meet the criticized aspects (resource expenditure through different stakeholders, communication effort, difficult descriptiveness).

The paper is structured as follows: In Sect. 2, the characteristics of TCS processes and the subsequent implications for modelling TCS processes are pointed out. In Sect. 3, related work is presented. In Sect. 4, we present our results. First, the language specification is defined. Second, the designed recommender component is shown. After that, the implementation and the architecture are given. In Sect. 5, we show the feasibility of our concept by demonstrating the prototype in a demonstration case. We conclude in Sect. 6 by discussing novelty, practical relevance, theoretical contributions, and limitations as well as giving an outlook for future work.

2 Characteristics of TCS processes

2.1 Classification of TCS Processes

Service dominant logic describes the paradigm shift from value creation with production to cooperative value creation processes or rather services (Vago and Lusch 2004). Simplified, a service is described as an activity that cannot be produced and stored in advance, due to its immateriality, and is characterized by an intense interaction between the service provider and receiver (Leimeister 2012; Meffert and Bruhn 2012; Thomas 2006). For services, four definitions have been established: enumerative, negative, institutional and constitutive delimitation (Meffert and Bruhn 2009; Scheer et al. 2006). The enumerative, the negative and the institutional delimitation do not meet the scientific claim because of missing criteria or the dismissing of hybrid forms (Burr and Stephan 2006; Leimeister 2012; Scheer et al. 2006).

Referring to the constitutional definition of services, four characteristics of a service, often termed as IHIP, are mentioned: *intangibility*, *heterogeneity*, *inseparability of production and consumption* as well as *perishability* (Zeithaml et al. 1985). The described IHIP-characteristics can be subordinated to the two essential constitutional features, *intangibility* and *interactivity*, because the differentiations determine each other. Although the IHIP-characteristics were considered the standard for a long time, in research it is discussed, whether the IHIP-characteristics reflect the process and the interactive character of a service appropriately, and for example, whether the intangibility as a core feature is sufficient (Leimeister 2012). Besides clear service processes, service bundles of products and services play an increasing role (e. g. Baines et al. 2007; Leimeister and Glauner 2008; Thomas et al. 2010) as well as the modularization (Leimeister 2012). In this context, for many manufacturers the TCS became a major value adding resource (Baines et al. 2013; Thomas et al. 2014). Systems that support the service provision co-create value as they configurate the interaction of resources and involved actors. The systematic engineering of such service systems is a current research topic in information systems; especially the mobilization of resources is one particular challenge of the service systems engineering domain (Böhmann et al. 2014). Due to bundling products and services to mobilise and provide sufficient information resources during service provision. To face this barrier further development of service systems are required.

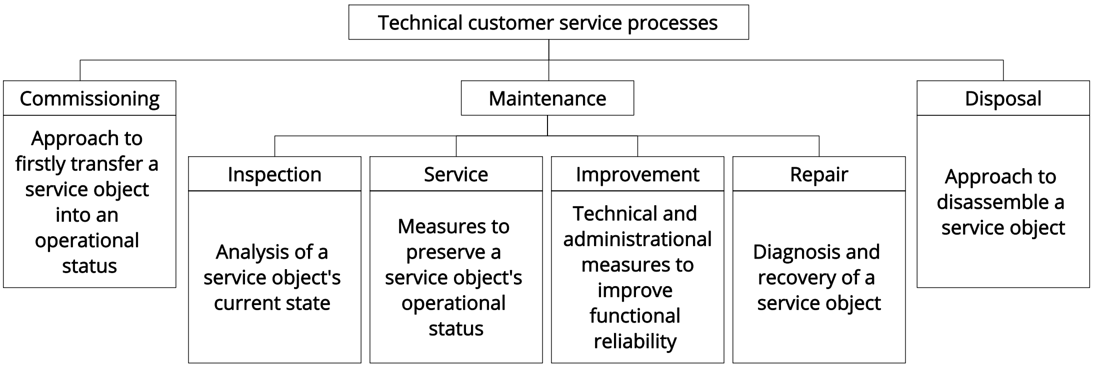

Fig. 1: Functional differentiation of TCS (referred to DIN 2003; Schlicker et al. 2010)

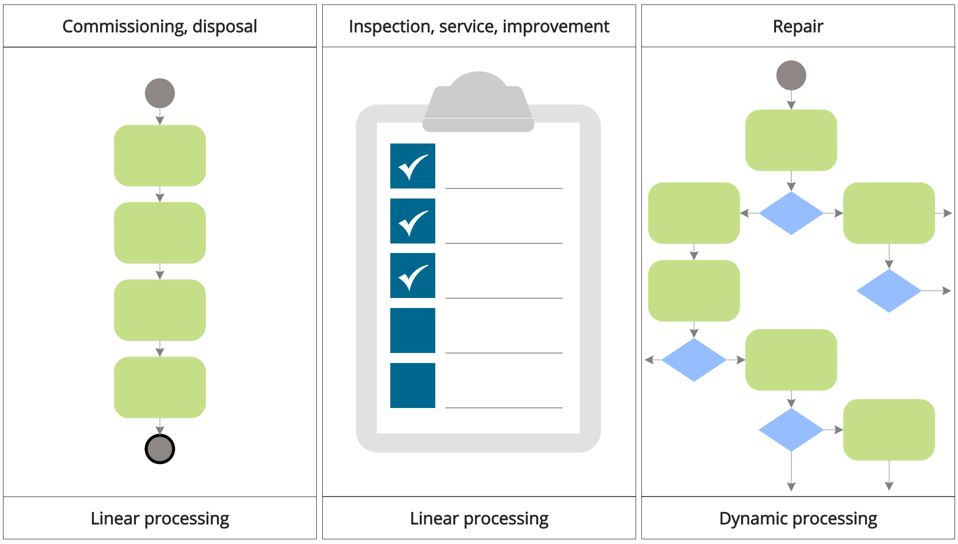

Fig. 2: Representation of TCS processes (referred to Schlicker et al. 2010)

User-generated content enables information mobilization and foster innovations (Leimeister et al. 2009) Furthermore, the interaction of users with the system has the potential to generate new information resources (Böhmann et al. 2014). Thus, the participation of users in information mobilization is crucial, but the practical application is still under ongoing investigation. In our research we investigate the involvement of users in information mobilization for smart glasses-based service systems in TCS. Thereby, a growing complexity arises that implicate new requirements to the modelling of technical services. Technical services are characterized by the fact that they are primarily performed at the customer's machines and plants (Becker and Neumann 2006). Examples for technical services are maintenance, spare parts supply as well as modifications, improvements and shutdowns and disassembles (Becker and Neumann 2006; Thomas et al. 2014; Walter 2010). Technical services like the ones performed by TCS are usually executed on-site at the machine, often in a short time window (Matijacic et al. 2013). By the example of technical service processes and their content-related classification, the complexity and the related influence to the model and its modelling can be shown. The DIN 31051 divides technical service processes in commissioning, maintenance and disposal. The maintenance itself is divided into inspection, service, improvement and repair (DIN 2003; Schlicker et al. 2010) (see Fig. 1).

Different representational forms are derived based on the content-related differentiation. Basically, the service processes can be structured in (a) a linear sequence, with an anticipated process, and (b) a non-linear structure, where the execution can hardly be predicted (Schlicker et al. 2010). Commissioning and disposal processes usually follow a linear structure that can be identified, brought into a reasonable order and documented as explicit knowledge during the construction and development of the final product. The same applies for the inspection, improvement and maintenance processes, which often take place in a checklist-styled process. In contrast, the repair process – hence the diagnosis and reparation processes – contains complex sequences, in which single diagnostic tasks alternate with repair tasks. Thus, a non-linear, dynamic and branched process structure is followed, depending on the context of the error. Due to the relationship of the parts, the result of the last performed task determines the next step. Therefore, the process sequence is determined ad hoc at runtime (Schlicker et al. 2010). Examples for the representational forms are given in Fig. 2.

2.2 Complexity of TCS processes

Based on the content-related differentiation, technical service processes can be grouped by their complexity (Meffert and Bruhn 2012; Schlicker et al. 2010). The dimensions for that are on the one hand, *the time needed for creating value*, and on the other hand, *the amount of distinct services and their heterogeneity*. A complexity matrix of technical service processes is given in Fig. 3.

Not only the heterogeneity of the services themselves, but also the heterogeneity of the objects (machines and plants) increase the complexity of creating value. The machines become more complex through hydraulic and electronic development. This leads to additional layers in the fault detection trees, which result in more complex modelling (Schlicker et al. 2010).

To meet the increased complexity, the service provider has to be supported with information regarding the process as well as the object. This can only be achieved by the use of IT (Matijacic et al. 2013), due to the huge amount of information that is necessary ad hoc at the point-of-service (Däuble et al. 2015). This requires support by a mobile, process-oriented service support system (Matijacic et al. 2013; Thomas et al. 2014). To fill those service support systems, the processes must be documented. In the following section, the implications will be identified and appropriate IT-support for runtime-modelling will be introduced, based on the characteristics of complex service processes.

2.3 Implications for Modelling TCS Processes

The characteristics of services as well as the industrial character of technical services have specific requirements for modelling the processes:

Intangibility
Regarding the intangible character of services, the difficult descriptiveness and measurability is often stated as a problem (Maleri 2013). Following Becker and Neumann (2006), several common systematization approaches exist for technical services; however, problems still occur while standardizing the order processing and the formalization of adequate process descriptions. This also increases the difficulty in transferring the processes to information systems (Becker and Neumann 2006). Hence, the difficulty to describe and to specify the TCS processes implicates that the process should be documented by the domain experts in the moment the service is provided. Because domain experts often lack skills in modelling, a simple way to create models, without knowing a modelling language, is required. For example, a textual description with several pictures may simplify the TCS process and increase comprehensibility (cf. Sect. 3). To capture the real process sequence at the point-of-service, the expert should not be distracted by the task of modelling. Therefore, (a) the hands should be free

for the service execution itself, (b) no additional IT-system should be needed and (c) the support should be mobile, ad hoc and easy to use.

Interactivity
The heterogeneity of services is influenced by external factors (e. g. the customer) (Leimeister 2012). Thus, the heterogeneity of the service implies a more difficult standardization. This is particularly relevant for technical services that include standardized tasks, like maintenance based on linear check lists, but also specialized tasks like fault detection, which require a lot of knowledge and creativity (Walter 2010). The inseparability of production and consumption (uno-actu-principle) has an impact on the flexibility of the service process. Due to the concurrent attendance during service (Leimeister 2012), the business processes contain service activities of the customer (interactivity), which leads, regarding to Becker and Neumann (2006), to spontaneous planning of tasks at runtime because of customer behavior or the condition of the technical objects. Planning and disposition activities must possibly be repeated (Becker and Neumann 2006). This implicates that, while modelling, several variations must be captured. The dynamic structure and the processes occurring at runtime (Schlicker et al. 2010) show that modelling in advance is too complex and that it should be done at runtime.

Modularization
Leimeister (2012) complements the constitutive characteristics by modularization, which means breaking down the processes into partial services. This procedure is called decomposition and has implications for the architecture of the backend system. It needs interfaces, so that highly dependent elements can be joined together. Individual modules can be reused, which has a positive effect on the economic benefit (Böhmann and Krcmar 2006). For administration purposes of the service portfolio, methods for removing redundant service processes have to be implemented (Leimeister 2012). Based on the identified implications, a mobile system for modelling those processes at runtime should be developed.

3 Related work

3.1 New Technologies to Model

New technologies do not only influence daily work, but they also change the possibilities for modelling business processes. Various authors entertain the idea of simplifying the modelling process with new technologies. In doing so, for example, Kolb et al. (2012) examined the usability of multi-touch devices to model and change processes through the introduction of standardized gestures. They considered concepts of interaction with multi-touch devices as requirements for relevant applications and formal modelling functions. The objective is the development of intuitive gestures. These gestures were derived based on a study with students without any experience in process modelling. The intuitive use of gestures offers new possibilities to model processes while being flexible for various devices (Kolb et al. 2012, p. 4). The enhanced usability throughout an intuitive control facilitates the conditions for inexperienced users; however, it is vital to have insight about formal modelling to create consistent processes. Similarly, Döweling et al. (2013) introduce a modelling system for collaborative process modelling on touchscreen devices such as tablet computers. Using device-specific forms of interaction, even modelling novices are enabled to create models of satisfying quality within a multi-user setting. In Alpers and Hellfeld (2016), a prototype for the modelling of petri net-based business processes on mobile devices is demonstrated. Again, the interaction capabilities of mobile devices are addressed via appropriate gestures, such as simplified drawings of petri net elements in order to place elements on the touchscreen. Another innovative approach for the creation of business process models through process elicitation has been proposed by Harman et al. (2015), who integrate 3D technology and virtual reality glasses with concepts of subject-oriented BPM (S-BPM). Hereby, stakeholders are enabled to recreate operational activities within a virtual requirement, which are logged and subsequently processed into process models.

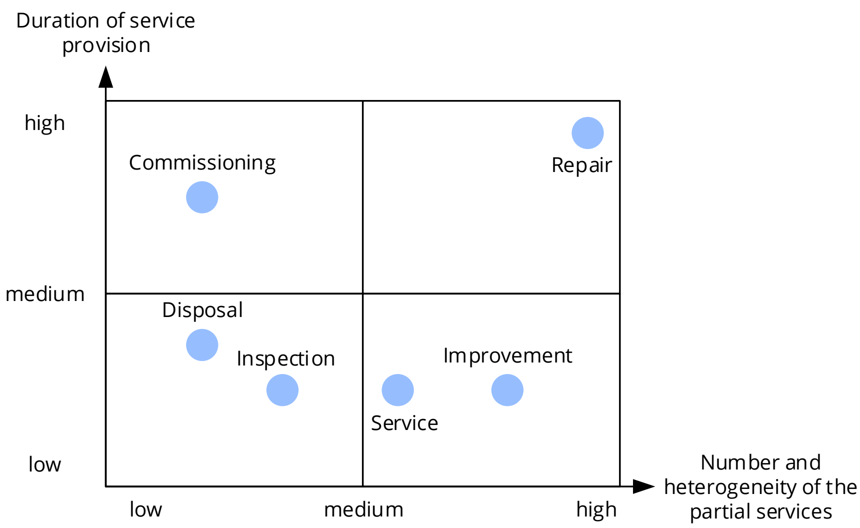

Fig. 3: Complexity matrix (referred to Meffert and Bruhn 2012; Schlicker et al. 2010)

3.2 Flexibility of Models

Process models can be differed into build time and runtime models (Thomas 2009). The designated build-time and runtime, known from software engineering, were transferred to process modelling due to the increasing importance of model-based software development since the mid-1990s (Remme and Scheer 1996). The distinction aims less at time, but rather the use of the model (design vs. reuse) (vom Brocke 2002). While models on build-time level do not need to be executable, models on the runtime level only contain executable constructs (Schütte 1998). Ordinarily, a runtime model is created out of a build time model by reducing the model to an executable version (vom Brocke 2002). Certainly, the procedure in the following concept is reverse, which means that the models are recorded directly and could be transformed, if necessary, to a build time model to reuse. Another aspect that literature presents is the flexibility of models. In doing so, Weber et al. (2008) define change patterns for information systems with process awareness to ensure process flexibility and, therefore, make process changes manageable. The identified change patters are reviewed in practice, and the systems which were used are examined in terms of their flexibility. The existing systems did not cope with the claim of changeability (Weber et al. 2008). The approach of adapting runtime models during the process could be improved by modelling directly during process execution. Therefore, processes could not only be adapted by experts, but also be created gradually, without access to build time models.

3.3 Integration of Domain Experts

Recker et al. (2010) and Wilmont et al. (2010) emphasize the value of integrating domain experts in the modelling process. In an experimental study, Recker et al. (2010) examine the basic understanding of modelling business processes by observing modellers who do not have any previous or formal knowledge. The participants of the experiment modelled their processes using pen and paper. These processes were visualized in written form (text, prose), graphically as well as hybrid. The experiment revealed an increased acceptance of the models as soon as an intuitive approach along with graphics and text were used by the modeller (Recker et al. 2010). For our suggested approch, this implicates the integration and the improvement of visualization aspects. In addition to abstract graphics, processes could be enriched during the modelling by voice recording or taking pictures, for example, by using the recording function of smart glasses. In line with the researchers, Wilmont et al. (2010) emphasize the advantage of integrating domain experts rather than the use of conventional modelling methods in collaboration with inexperienced users. The authors develop modelling approaches in which the perspective of domain experts without any modelling experience is considered. Therefore, the modelling behavior and the applied concepts of modelling experts and domain experts are compared. The researchers confirm the thesis that the various perspectives on the reality of both groups lead to different abstractions of processes. Modelling experts do not provide the daily know-how within this domain, to define an appropriate degree of abstraction. However, domain experts face the difficulty of communicating all of the relevant aspects of their actions due to the dynamic of those actions. In the case of independent modelling by an expert, detailed support from the modelling tool is required (Wilmont et al. 2010). Smart glasses offer experts the opportunity to interactively design and support the process of modelling. Changes in the process could be adapted dynamically and parallel to the execution of the appropriate action, which adresses challenges that arise by trying to adequately capture the service process in the back-office subsequent to the actual service execution.

In summation, the subject of runtime modelling is discussed in literature from different points of views. The adaption of new technologies, the flexibility of creating models during runtime and the integration of experts in the modelling process are already topics under examination. However, so far no technologies exist that offer the possibility to create the model without any interruption and influence from the service execution. Hence, the use of such technology to create models by domain experts during runtime was not considered yet. This paper contributes to the previous topics by designing a smart glasses-based system for runtime modelling during the service execution by the domain experts.

4 Realtime modelling with smart glasses

4.1 Language definition

A crucial prerequisite for business process modelling is the choice of an adequate BPML (Weske et al. 2004). Common BPMLs are, for example, the Business Process Model and Notation (BPMN) and the Event-driven Process Chain (EPC). For the purpose of this paper, the EPC language has been chosen for modelling service processes using smart glasses. The choice of the EPC as BPML in this work is primarily based on two reasons: First, the EPC language represents a widely used and well researched language for business process modelling (Fellmann et al. 2013; Fettke 2009; Houy et al. 2009). Studies show that the EPC is one of the most popular notations for business process modelling (Harmon and Wolf 2016). Additionally, the EPC is implemented in major process modelling tools and BPM software, which further underlines its relevance in practice (Karhof et al. 2016). Second, the EPC is of moderate complexity due to a limited set of modelling elements. Hence, an EPC model is rather simple to construct, which suites the given application scenario of this paper. On the one hand, wearable technology comes along with specific requirements regarding usability, both from a BPML point of view (for example, a small screen size and limited user interactions require a clear defined set of language constructs; see Döweling et al. (2013)) as well as a general modelling perspective (e. g. known functionality from BPM tools cannot simply be applied to mobile or wearable technology (Kolb et al. 2012)). On the other hand, the presented concept of this work specifically addresses employees working as service technicians, who often only possess limited knowledge of modelling sound business processes (Pendergast et al. 1999). Furthermore, the proposed concept provides that the modelling of processes must be done in parallel to the actual service activity. Accordingly, the modelling task needs to be as efficient and simple as possible to not distort service execution. Following the stated reasons, we will briefly introduce the EPC modelling language. Based on the smart glasses application scenario, we will then evaluate EPC language constructs regarding the applicability and argue for an adjusted EPC language specification to meet the outlined needs.

Developed in 1992, the EPC has been one of the most dominant business process modelling languages in research and practice over the last decades, being applied in various scenarios and resulting in numerous scientific publications (e. g. Fellmann et al. 2013; Fettke et al. 2010; Riehle et al. 2016). Due to its maturity, the EPC language has been refined and modified by various authors, resulting in a plethora of different EPC extensions and variants. A brief overview of relevant EPC extensions can be found in Riehle et al. (2016). Nowadays, the term "EPC" refers, in most cases, to the extended EPC (eEPC), initially proposed by Galler and Scheer (1994), Hoffmann et al. (1993), and Keller and Teufel (1997). Subsequently, in accordance to Becker et al. (2009b), this paper uses the term "EPC" synonymously for the eEPC extension. The EPC is defined as a business process modelling language "for the representation of temporal and logical dependencies of activities in a business process" (Mendling 2007, p. 36).

In BPM literature, modelling conventions are being applied in order to restrict or adjust BPMLs towards a given modelling environment. Thomas and Scheer (2016) define language conventions as a "determination of language constructs allowed when using a specific modelling language". Due to previously stated reasons in terms of the application scenario and technology used in this paper, we present a descriptive and meta model based language convention for EPC modelling with smart glasses. Fig. 4 depicts alterations that have been made to the EPC meta model, which is based on an integrated EPC meta model proposed by Jannaber et al. (2016). Changes in the meta model are highlighted according to the presented key.

In particular, the proposed convention used in this contribution omits the OR operator (Adjunction) and does not allow for a resource and resource relationship type hierarchy. The OR operator has been removed because its semantics are complex and mostly still not well defined (for a detailed elaboration on this issue see Gruhn and Laue (2007), Kindler (2006), and Mendling and van der Aalst (2007)). Henceforth, the OR operator increases the cognitive effort necessary by the users, who are often novice modellers. Naturally, this lowers the quality of the resulting models, since the complex modelling task needs to be conducted in parallel to service execution. In fact Furthermore, the resulting models are intended to be used as instructions for a single user. Accordingly, a parallel execution of different process paths, which is enabled by the OR operator, is not considered in the smart glasses modelling concept. Lastly, leaving out the OR connector due to its ambiguity is also suggested in literature (Fellmann et al. 2013). Further, we postponed the support for AND operators, as parallel work on objects requires additional elaboration on collaboration and synchronization. Since the proposed modelling concept focuses on single users that document the service process from their point of view, a parallelization of the process flow is not considered at this stage of development. The hierarchies of resources and process-resource relationships have been left out due to pragmatic issues that primarily concern the limited display size of smart glasses. The remaining elements in the meta model basically keep their inherited semantics:

- *Function*. A function captures a single activity executed by a service technician during service delivery. Each function has a meaningful label which expresses the function's main concerns. Functions may be assigned with resources to highlight resource usage or consumption.
- *Event*. Events are passive elements that represent specific states which occur during the process flow. In the given scenario, we relinquished trivial events and instead only allow

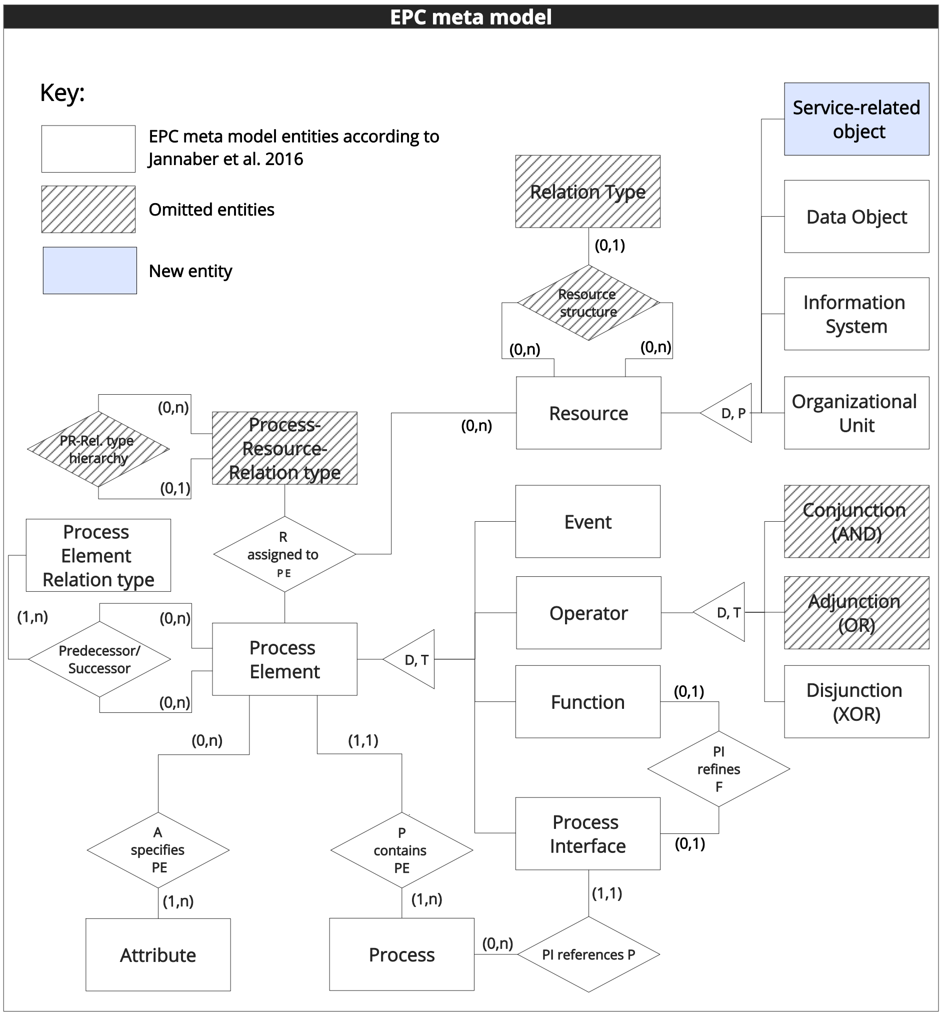

Fig. 4: Restricted EPC meta model

usage for meaningful notifications (e. g. mile stone events, warnings or service triggers) that facilitate the understanding or the service process.

- *Process interface*. Process interfaces are optional elements that are applied when a particular function is detailed via a sub process. Hence, process interfaces may replace functions during the process flow and allow the construction of a process hierarchy.
- *XOR operator*. Exclusive split/join of the process/service flow. The XOR operator indicates alternative path choices during service execution.
- *Organizational unit*. Optional resource. A service technician may attach an organizational unit element to a function to explicate i. e. specific responsibilities for its execution.
- *Data object*. Optional resource. General data/-media/information object to be attached to functions. In the context of smart glasses, data objects may represent audio or video files as well as pictures that provide additional details for a particular activity.

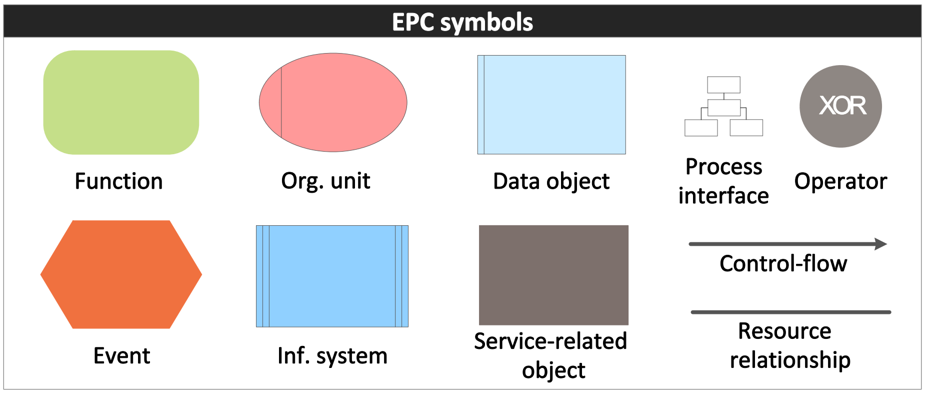

Fig. 5: EPC Symbols

- *Information system*. Optional resource. Modellers may highlight the usage of information systems that support the execution of an activity.
- *Control-flow and resource relationship*. The basic control-flow connects the main elements function, event and operator, and indicates the sequence of service activities necessary to accomplish the service objective. Resource relationships connect resources to functions.

Service-related modelling elements for the EPC have been frequently discussed in literature (e. g. Huth and Wieland 2008). For example, in their EPC meta model, Scheer et al. (2005) introduce various service objects such as information services or financial services. To meet the requirements of service modelling in the given case, we propose a new resource element that is added to the meta model. The *service-related object* is an object that consolidates previous efforts in defining EPC service resources and is specific to the given context, which is TCS. Subsequently, a service-related object may represent either tools used in a service activity (e. g. screwdriver for activity "Remove engine") or parts that are being consumed during e. g. maintenance ("Screw type LT405") (cf. Metzger et al. 2016b).

- *Service-related object*. The service-related object is an additional resource element that can be attached to functions. Using this element, the service technician is able to explicate tools that are required for service execution or (spare) parts that are being consumed during an activity.

With the addition of the service-related object, the specified language convention allows service technicians to capture information about domain-specific entities that goes beyond the expressiveness of regular EPC resource elements. As technical service processes that are performed on-site, such as maintenance and repair, not only require the documentation of data objects to be processed, details on the organizational unit or used information systems, the service-related object is considered as a necessary addition to cope with TCS-specific requirements regarding information provision. The documentation of e. g. tools and spare parts is particularly crucial for the intended application scenario, which focuses on the reuse of modelled processes for service support and training purposes. In this context, knowledge about service-related objects that are needed to execute a specific service process is essential.

All EPC elements specified in the proposed language convention and henceforward being used for process modelling are summarized in Fig. 5. For a graphical representation of the service-related object, we adhere to notations from similar service objects introduced in literature. In particular, the service-related object shares its graphical notation with service objects presented in Scheer et al.

(2005), thus being a double-layered rectangle. As we enforce a restriction on resource relationships, there are only two connector types: A simple control-flow, which connects the main process elements, and a single resource relationships that specifies the annotation of resources to functions. In contrast to popular EPC definitions, the type of resource relationship is the same for all resources to reduce model complexity.

Changes in the meta model, as well as slight abbreviations regarding the EPC element specifications influence the construction process of the presented EPC restriction. Since the primary purpose of this paper is the documentation of service processes during runtime, focus is being put on syntactical issues. Therefore, we refer to the consolidated set of EPC syntax rules as given by Fellmann et al. (2013), and highlight changes to these rules that must be made in order to conform to the previous language specification. Tab. 1 summarizes the proposed syntax rules and indicates whether they apply in the presented case (full circle) or are omitted (blank circle).

Tab. 1: Syntax rules (referred to Fellmann et al. 2013)

EPC syntax rules	*Relevance*
1. There is at least one start event and one end event	•
2. Functions and events need to be alternating	◦
3. Resources are attached to functions	•
4. Join and split connectors have the same type	•
5. Join and split connectors have at least one incoming and one outgoing control-flow	•
6. Events and functions possess only one incoming and one outgoing control-flow	•
7. OR and XOR connectors must not be followed by a function	•
8. Process paths are only connected to events	•

Tab. 1 demonstrates that the changes that have been made to the EPC specification solely address the alteration of functions and events. In literature, events that directly follow functions and thus represent a simple change of state are called trivial events (cf. Davis 2001; Kopp et al. 2006). For the presented language convention, we decided to remove these trivial events in order to allow sequences of EPC functions in the smart glasses-based modelling scenario. The added complexity, in terms of modelling effort and model comprehensibility, increases when incorporating trivial events in created EPC models.

Henceforth, the proposed language convention for EPC modelling is being applied to the smart glasses scenario. Subsequently, all demonstrated features and resulting process models adhere to the stated specification.

4.2 Recommender component

Nowadays, the use of recommender systems for process modelling is heavily featured in literature (Fellmann et al. 2015; Koschmider et al. 2011). Initially, recommender systems have been primarily applied in disciplines such as e-commerce to support the decision making of users with respect to certain products or items (Melville and Sindhwani 2010). In general, recommender systems try to “generate meaningful recommendations to a collection of users” (Melville and Sindhwani 2010, p. 829). For a fitting recommendation, implemented algorithms make use of the increased availability of data in order to apply advanced statistics. The BPM domain has recognized the capabilities of recommender systems for modelling assistance. One example is the application of recommender systems in the form of auto-complete functionality during process model design. Here, possible recommendations may be succeeding process elements (Clever et al. 2013) or process patterns (Deng et al. 2016; Wieloch et al. 2011), which are determined based on parameters such as process context and semantics, language syntax or previously modelled processes. In addition to mere process elements, element labels have been a frequent subject of discussion in the field of

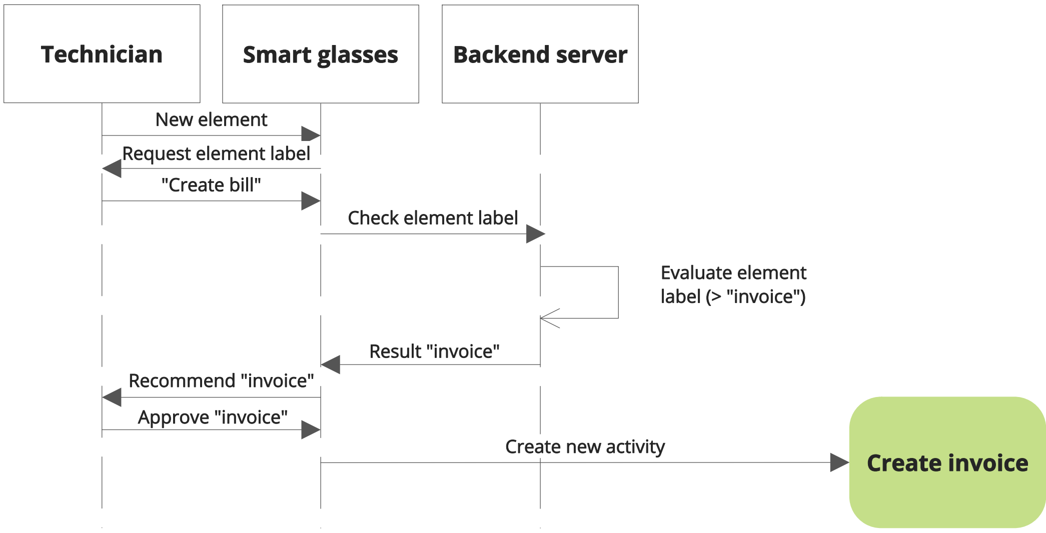

Fig. 6: Recommender concept

process modelling recommender systems. Proposed approaches in this regard range from formal specifications of linguistic conventions for business process models (Becker et al. 2009a) to text corpora and linguistic knowledge (Leopold et al. 2013) and terminological ontologies (Delfmann et al. 2009).

For process modelling with smart glasses, a label-based recommendation is beneficial for two main reasons: First, it reduces the complexity the modeller has to face when constructing a business process model. Since the intended purpose of the presented concept is to model during service execution, there is a specific need to simplify the modelling process as much as possible to reduce distraction from the service task and to prevent time consuming design choices. Via labelling recommendations, the user is supported in the construction of valid process models as, for example, label ambiguity is automatically adjusted. Second, recommending element labels results in a higher process model quality, due to naming conventions that need to be followed. Enterprises often maintain a specific set of vocabulary or an organizational glossary containing and standardizing the notation of the most dominant business objects. Hence, not adhering to these standardized terms renders business processes incomprehensible for further usage. Additionally, distorting element labels hampers the application of process analytics, since algorithms, such as the automatic check for compliance violations, are strictly dependent on the standardized usage of element labels across all process models (Delfmann and Hübers 2015). Due to the stated reasons, a recommender system concept for business process element labels is introduced as part of the smart glasses-based modelling environment in Fig. 6.

The recommendation of labels is triggered every time the user signals the modelling of a new process element via voice command. The smart glasses-based modelling environment then requests an element label, as provided by the language specification. Subsequently, the user chooses a label, which will be forwarded to a recommender system running on a background server. Applying the concept of process label ontologies (as characterized in e. g. Delfmann et al. (2009)), the input label is evaluated against standardized organizational terms predefined in sources such as an enterprise glossary or an openly accessible language thesaurus. In the presented case, this evaluation is limited to nouns only. Fig. 6 depicts the evaluation of the input term *bill*. In consulting the connected sources with respect to the underlying ontology, the system recognizes the desired usage of the term *invoice* instead of *bill*. Accordingly, the recommender system returns the term *invoice*, which is then forwarded to the user in the form of a process label recommendation. As soon as the user approves the suggested term, a new process element with the corresponding label is created. The characterized recommendation process is identical for all primary modelling elements (function, event and resource) specified in the language convention.

The implementation of a label-based recommender system requires the system architecture to provide a server component that is capable of running the recommender system and corresponding algorithms. In parallel, the server component requires access to both internal and external data sources. Regarding internal data sources, the recommender system works on existing processes stored in a process database. In this case, the recommendation of process element labels can be deducted from previously modelled processes and facilitate organization wide terminological standardization. In terms of external data sources, the server running the recommender system provides interfaces to openly accessible terminological databases.

4.3 Implementation

Next, we describe the implementation in our smart glasses-based modelling system based on the previously clarified components (cf. Fig. 7). In the corresponding figures the screens and voice commands of the system are shown together with the EPC model. The dotted arrows show the relevant change to the model.

New Activity

In order to create a simple sequence of functions, we built a software component that allows new activities to be added to the model. The component provides two possibilities. First, the user of the smart glasses-based modelling system can take a

Fig. 7: Smart glasses-based modelling system

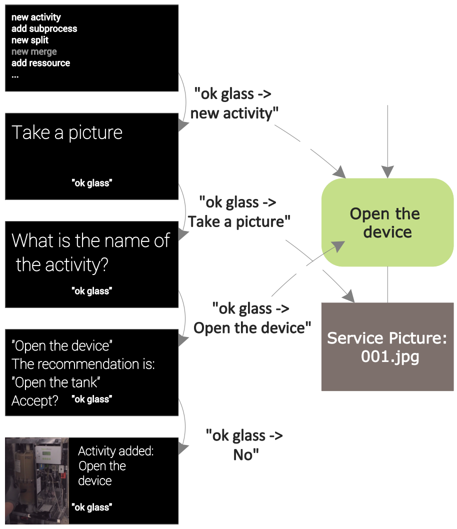

Fig. 8: Software component: New activity

picture to illustrate the function. Second, through the use of voice recognition and speech-to-text (activated by “ok glass”), the labeling of the function is implemented. The labeling of the function is assisted by a recommending component (cf. Sect. 4.2). The user gets a recommendation on how the system would label it and is asked whether or not the recommendation is accepted.

Additionally, if the user is altering an existing process, and there is already a function following the current one, the new function is put in between the existing functions. If no function exists at all (as when the user is starting a new process), the added function is taken as the first one.

Add Subprocess

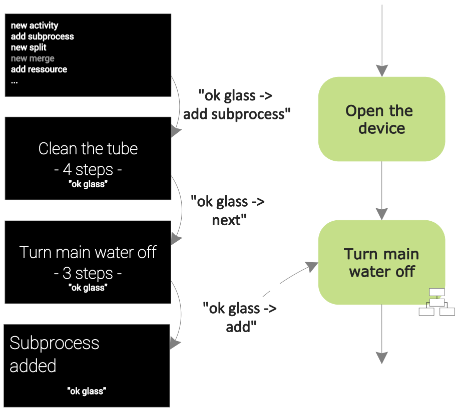

Fig. 9: Software component: Add subprocess

With an increased amount of available processes, the need to reuse certain process parts is present (cf. Sect. 2, "modularization"). Thus, we added a software component to include existing processes into the current one. The component provides an interface to browse through existing processes (with "next") and add one of them to the sequence (in this example "Turn main water off"). Further information is given about the length of every existing process to give the modeller an idea about how many steps are added. Within the EPC model, a process interface is used to represent the added subprocess.

New Disjunction (XOR) – Split

Besides the simple sequence of functions, sometimes the necessity for choices and different process paths arises. Thereby, we implemented the software component for a new split. For the sake of simplicity the call of the component is only usable when at least one additional process element after the current one exists. Consequently, the split takes the existing path as given and continues with the new one. The split is put inbetween the current process element and the next one.

The operation first asks what the condition for the split is. The system also recognizes speech-to-text to define the choice. This also benefits

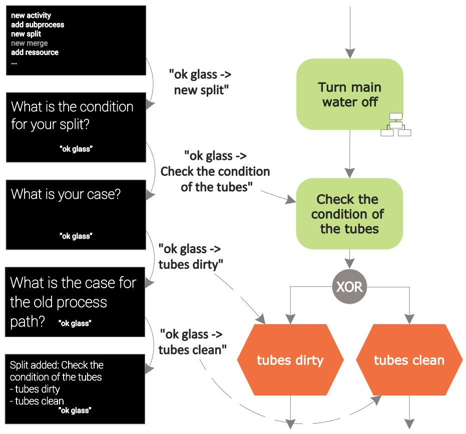

Fig. 10: Software component: New split

the recommendation component that helps to find the appropriate label for the events (for space restrictions not shown in Fig. 10). Afterwards, the user is asked about the possible answer for the existing path and, finally, the decided answer for the new path. The process modelling then continues with the new path.

New Disjunction (XOR) – Merge

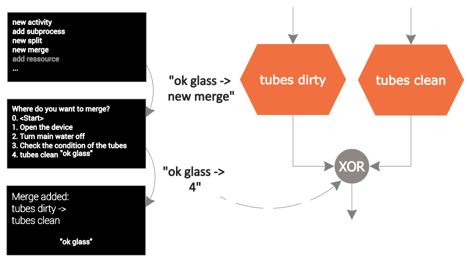

Fig. 11: Software component: New merge

Consequently, the need for merging paths arises when the system is able to split paths, and, thereby, enhance the complexity of the model. The software component integrates new merges. As soon as the model splits into two paths, the option to merge them is given to the user. To do so, the user is asked where in the existing process the merge should take place. As the system does not restrict process elements following the initial merge, both multiple paths and circles (when the user merges on an earlier process element after the initial split) are possible.

Add Resource

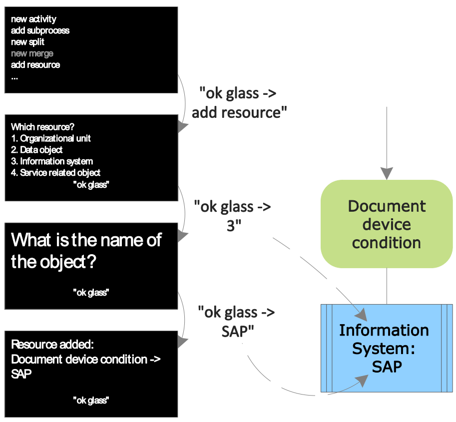

Fig. 12: Software component: Add resource

To attach additional information to activities, such as organizational units, data objects, information systems, or service-related objects, an additional component was designed. When the last element was an activity, it is possible to attach multiple resources. Therefore, the type and name of the resource (via speech-to-text) is asked. If the "data object" is chosen, the service technician can also take pictures or record videos. The same applies to "service-related objects" where pictures of tools, as well as speech-to-text can be used analogously with the software component *new activity*.

Change Element

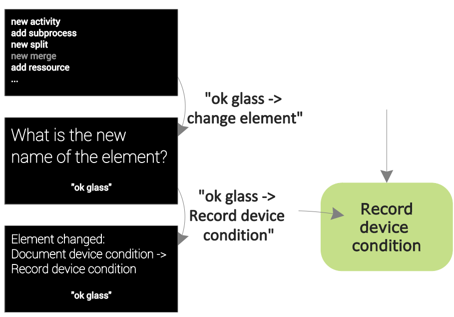

Fig. 13: Software component: Change element

When errors or misleading terms are used, the modeller is able to change the current activity and add a new caption. The system asks for a new caption and displays a confirmation. This is encouraged by the recommendation component (analogously to *new activity* and *new split*).

Delete Element

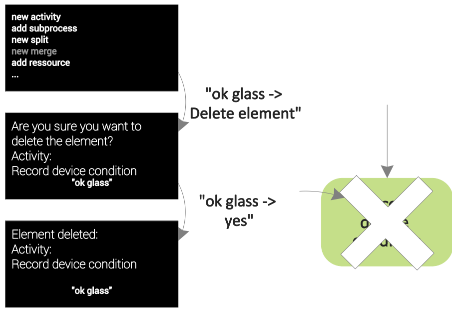

Fig. 14: Software component: Delete element

Finally, the system allows the deletion of model elements. To ensure integrity, the removal is only possible if no other elements (such as splits or merges) depend upon them. When deleting splits, the user needs to decide which process path to omit. Subsequently, elements in the omittted path have to be deleted first to enable the deletion of the split itself. If following elements exist, the next element is connected to the previous one to keep the model coherent.

4.4 Architecture

The architecture illustrated in Fig. 15 includes the communication between the different components of the system. Namely, the components are the *technician* wearing smart glasses, a *server* where all the communication converges, a *database* for storing the processes, a *backend* where a modeller can check and revise the process models, *(public) thesauri* and an *interface* to other systems.

The workflow for technicians starts with the capture/change of a process. (1) When they use speech-to-text for a new label of a process element, the system is generating a request for similar labels. (2) The server then queries the database with known processes. (3) Additionally, (public)

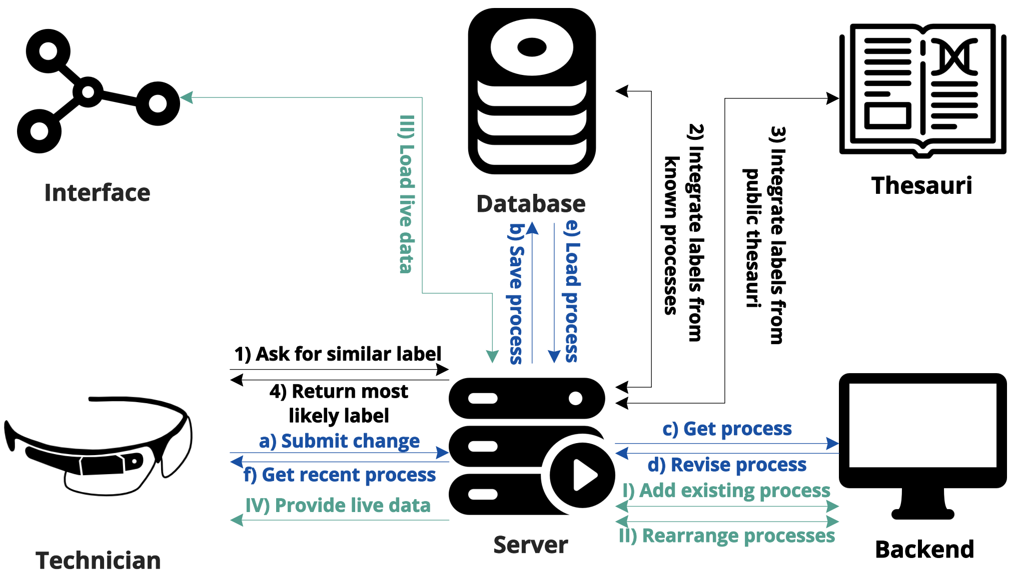

Fig. 15: System architecture

thesauri might be queried for similar labels as well. (4) Afterwards, the recommender component of the server calculates a similarity score (e. g. as discussed by Mihalcea et al. 2006) for every entry that was send back in order to pick the most likely label. The highest rated one is sent back to the technician.

(a) After the technician finishes the capture/change, the changes are submitted to the server. (b) The server processes the data, extracts the labels for the recommender component and saves all to the database. (c) To ensure process quality, the processes are reviewed by a modeller and (d), in case alteration is needed, a revised version is saved. The modeller that is working in a web-based back-end modelling system has more possibilities to change processes or add more resources (e. g. pictures). Furthermore, he/she can consult additional meta data to the process. The modeller can therefore determine whether the last changes might be due to a particular maintenance object or contractor. For the technician, when the maintenance process at the point-of-service is started, the newest process is (e) loaded from the database and (f) delivered to the smart glasses.

(I) When pre-existing digital processes are present before the smart glasses system is rolled out, the backend modeller has the possibility to add them to the database, making them available to all technicians with smart glasses. (II) Further, the modeller has the possibility to rearrange processes to build reusable processes.

(III) The communication server also provides an additional interface to legacy systems such as ERP, CRM or those that are alike. (IV) They enable the technician to load live data for the particular machine, sensor or customer.

Overall, the presented architecture is meant to serve as a reference for building similar systems and should be adapted for the particular scenario. However, our implementation based on Google Glass uses the same architecture. Both, the architecture and the implementation are meant to be a reference design and answer the given research question *RQ2b* (What are the necessary components and how should the architecture be designed and implemented?).

5 Demonstration case

The previously presented system is illustrated in this section by a demonstration case (as proposed by Riege et al. 2009). Therefore, we demonstrate the feasibility (as discussed by Sonnenberg and vom Brocke 2012) of the defined constructs based on a real process from air conditioning and heating technology. The process has been collected as a part of the research project Glassroom and describes the procedure of changing a tank. The service provider is specialized in B2B with about 140 employees. The service provision includes mainly maintenance processes that are very diverse in terms of different manufacturers and variety of the plants.

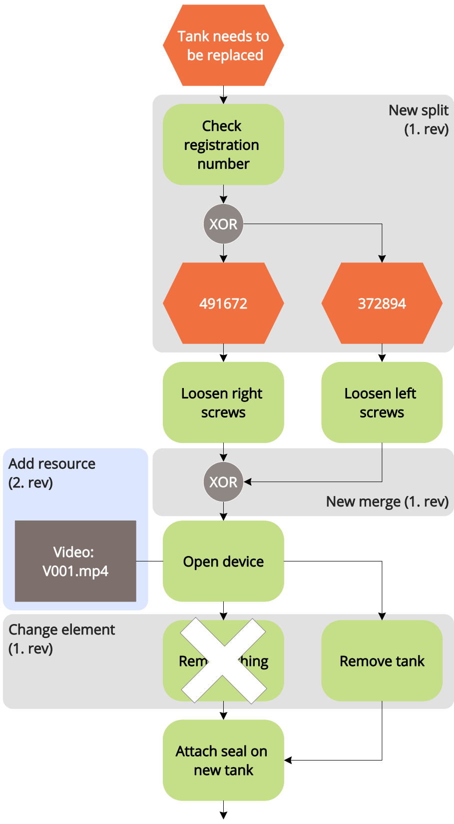

Fig. 16: Example of service process - part 1

In Fig. 16, the first part of a process of a tank change is illustrated. The individual process steps are observed by the technician with smart glasses,

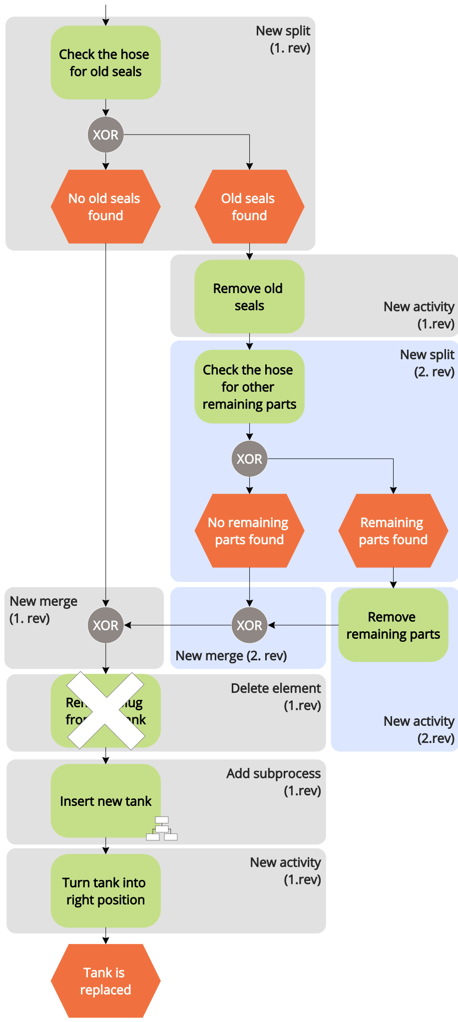

Fig. 17: Example of service process - part 2

who photographs the individual steps and adds voice notes. The process capturing is adapted twice, illustrated in 1st revision (1. rev.) and 2nd revision (2. rev.). The process was initially captured by a technician as a simple sequence. The individual steps were captured with the new activity software component during the execution of a tank change. Afterwards, the process was available for all technicians as instruction. During a tank change carried out by another technician (1. rev), he/she noticed a deviation of the process and adapted the process immediately. First, the technician expanded the process and added the new split *Check registration number* to determine the mounting of the screw. Depending on the model series, the technician has to *Loosen right screws* or *Loosen left screws*. With a new merge the process returns. In the step *Remove thing*, the technician replaces the imprecise description of the activity and details it with the new label *Remove tank*. In the second part (Fig. 17), the technician (still 1. rev) adds the check for old seals by using the new split *Check the hose for old seals*. If there is an old seal in the hose, the new alternative is specified with the new activity *Remove old seals*. With a new merge, the process returns back to the original path. Later on, the unneeded activity "Remove plug from tank" follows. The technician reduces the process by using *delete element*. Afterwards, he/she adds a subprocess to the process with the name *Insert new tank*. Concluding the process, the technician adds a new activity *Turn tank into right position* to the process before it ends. Another technician (2. rev), also equipped with smart glasses, is able to profit from the initial capturing and the changes made during revision 1. This technicians' adaptions are described afterwards. In the first part (Fig. 16), he/she adds a resource (video) to the step *Open device* that is named "V001.mp4". In the second part (Fig. 17), during the step *Remove old seals*, he/she adds a new split if additional parts remain in the hose (*Check the hose for other remaining parts*). When parts are found they need to be removed (*Remove remaining parts* through new activity). The path returns to the old process through a merge and follows the process order from the initial capturing and the 1st revision. Overall, the process was revised twice to provide a more detailed explanation of the replacement of the tank.

6 Discussion, conclusion & outlook

Problem Statement and Novelty of the Solution
The underlying idea of allowing the technician to capture and change processes during runtime has been limited until now. The need to interrupt the activity to take a mobile device and capture the process does not provide a satisfactory solution. Hence, the processes were adapted and modelled independently from their execution. The required cooperation of the process and modelling experts is often mentioned as cost- and time-intensive. Moreover, based on the complexity of technical services, the adaption of these processes is difficult. The maintenance processes are difficult to capture ex-ante and too complex to describe ex-post. To face this challenge, we proposed an intuitive and practicable approach to model processes at runtime.

Summary
We started by working out the theoretical foundation of the system; the characteristics of technical services and requirements that arise from them (cf. Sect. 2). Additionally, we investigated related work on modelling with new technologies, on novices and runtime modelling to get further implications for the design of our concept (e. g. picture and text, relevance of recommender functionalities for novices) (cf. Sect. 3) *(RQ1)*. Based on these factors, we developed an overview of relevant modelling constructs (cf. Sect. 4.1) that need to be taken into consideration for a process capturing system *(RQ2a)*. Afterwards, we added the recommender component to reduce the complexity of the modelling process for the technician (cf. Sect. 4.2). Based on the conceptual grounding, the implementation of the smart glasses-based modelling system is presented (cf. Sect. 4.3) and reference architecture is provided (cf. Sect. 4.4) *(RQ2b)*. Finally, we demonstrated the feasibility of the concept by the implementation and demonstration that includes all relevant software components (cf. Sect. 5). Thereby, we put all of the pieces together to answer *RQ2* on a conceptual level.

Practical Relevance
Stopping the maintenance process to pick up a smart phone, tablet or use a laptop in order to capture processes at runtime is not feasible. With smart glasses and hands-free interaction, the potential to allow a new way of modelling processes is given. To ensure the modelling quality, we added a revision option through an experienced modeller.

This allows service technicians to capture complex processes and use them for documentation or training purposes. As a result, new technicians can be assisted by smart glasses while learning, which simplifies training and improves efficiency. In sum, our practical contribution is to give a blueprint of a runtime modelling system based on smart glasses that could help technicians increase the quality of service.

Theoretical Contribution
With this paper, we contribute to the knowledge base of service science. For the aforementioned reason, a smart glasses-based modelling system is beneficial for the discipline, but is not yet present. Hence, the presented concept could be used as a template for this kind of system. Further, it is shown how process modelling language modifications, such as the addition of the service-related object to the set of EPC elements, can be used to capture domain-specific knowledge of TCS processes within a runtime modelling environment. In addition, the paper also contributes to the discipline of IS design as we present how smart glasses systems might look like in general. Finally, our major contribution is to the knowledge base of BPM, as the system utilizes a novel technology that allows a new approach of capturing business processes while executing them. Furthermore, our research has outlined how a process modelling language convention can be specified that is suitable to capture business processes on smart glasses. Additionally, process recommender systems have proven to be a valuable addition to the modelling environment, since the cognitive effort needed for modelling can be reduced, which in return ensures the quality of the actual service process. This addresses the discussion about modelling procedures and add a potential approach for process modelling and elicitation.

Limitations, Outlook and Future Work
First, the modified EPC language convention in combination with hardware-specific restrictions (e. g. small screen size) provide that resulting models are of rather limited expressiveness and complexity, which is also facilitated by the omission of parallel activities with more persons executing processes simultaneously. Hence, further work need to address this issue by exploring ways that allow for the creation of adequately complex processes, for example by expanding the recommender system component towards the recommendation of process elements or by putting further emphasis on the backend modelling system. Additionally, enhancements may be implemented for later versions of the smart glasses hardware that allows more information on the screen, thus enabling users to model more complex processes that consist a larger amount of elements. Furthermore, since synchronization and collaboration play an important role in parallel activities, subsequent research needs to investigate possibilities to synchronize and harmonize processes that have been documented using two or more smart glasses modelling environments. As a second limitation, the proposed system is primarily build upon a scientific point of view. Consequently, further research need to focus on the exposure of the smart glasses modelling system towards real-world application scenarios to reveal further insights of practitioners, which also encompasses usability (interaction capabilities). Here, challenges arise in the context of the modelling system (e. g. cognitive load), the modelling language itself (e. g. process element visualization) and the resulting models (e. g. sufficient level of detail).

Overall, the given concept builds a foundation for addressing the challenge of process modelling being too time- and cost-intensive. Henceforth the modelling of processes while executing them has potential to minimize the effort of creating and maintaining service support systems.

References

Alpers S., Hellfeld S. (2016) Werkzeug zur mobilen Modellierungvon Geschäftsprozessen mittels Petri-Netzen. In: Betz S., Reimer U. (eds.) Modellierung 2016 Workshopband. Gesellschaft für Informatik e.V., Bonn, pp. 147–152

Baines T. S., Lightfoot H. W., Evans S., Neely A., Greenough R., Peppard J., Roy R., Shehab E., Braganza A., Tiwari A., Alcock J. R., Angus J. P., Bastl M., Cousens A., Irving P., Johnson M., Kingston J., Lockett H., Martinez V., Michele P., Tranfield D., Walton I. M., Wilson H. (2007) State-of-the-art in product-service systems. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 221(10), pp. 1543–1552

Baines T. S., Lightfoot H., Smart P., Fletcher S. (Apr. 2013) Servitization of manufacture - Exploring the deployment and skills of people critical to the delivery of advanced services. In: Journal of Manufacturing Technology Management 24(4), pp. 637–646

Becker J., Delfmann P., Herwig S., Lis L., Stein A. (2009a) Formalizing linguistic conventions for conceptual models. In: Laender A., Castano S., Dayal U., Casati F., de Oliveira J. (eds.) Conceptual Modeling - ER 2009. ER 2009. Vol. Lecture Notes in Computer Science, vol. 5829. Springer, Berlin, Heidelberg, pp. 70–83

Becker J., Mathas C., Winkelmann A. (2009b) Geschäftsprozessmanagement, 7th edition. Springer, Berlin, Heidelberg

Becker J., Neumann S. (2006) Referenzmodelle für Workflow-Applikationen in technischen Dienstleistungen. In: Bullinger H.-J., Scheer A.-W. (eds.) Service Engineering SE - 26. Springer, Berlin, Heidelberg, pp. 623–647

Becker J., Pfeiffer D., Räckers M. (2007) Domain Specific Process Modelling in Public Administrations – The PICTURE-Approach. In: Wimmer M. A., Scholl J., Grönlund A. (eds.) Electronic Government. EGOV 2007 Vol. Lecture Notes in Computer Science, vol. 4656. Springer, Berlin, Heidelberg, pp. 68–79

Böhmann T., Krcmar H. (2006) Modulare Servicearchitekturen. In: Bullinger H.-J., Scheer A.-W. (eds.) Service Engineering SE - 26. Springer, Berlin, Heidelberg, pp. 377–401

Böhmann T., Leimeister J. M., Möslein K. (Apr. 2014) Service-Systems-Engineering. In: WIRTSCHAFTSINFORMATIK 56(2), pp. 83–90

Burr W., Stephan M. (2006) Dienstleistungsmanagement: innovative Wertschöpfungskonzepte im Dienstleistungssektor. W. Kohlhammer Verlag, Stuttgart

Clever N., Holler J., Shitkova M., Becker J. (2013) Towards Auto-Suggested Process Modeling – Prototypical Development of an Auto-Suggest Component for Process Modeling Tools. In: Proceedings of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2013) Lecture Notes in Informatics (LNI) 222, pp. 133–145

Däuble G., Özcan D., Niemöller C., Fellmann M., Nüttgens M., Thomas O. (2015) Information Needs of the Mobile Technical Customer Service - A Case Study in the Field of Machinery and Plant Engineering. In: Proceedings of the 48th Annual Hawaii International Conference on System Sciences (HICSS 2015). Koloa, Hawaii, USA, pp. 1018–1027

Davis R. (2001) Business Process Modelling with ARIS. Springer, London

Delfmann P., Herwig S., Lis L. (2009) Konfliktäre Bezeichnungen in Ereignisgesteuerten Prozessketten–Linguistische Analyse und Vorschlag eines Lösungsansatzes. In: EPK 2009 Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten Nüttgens M., Rump F., Mendling J., N. G. (eds.), pp. 178–194

Delfmann P., Hübers M. (2015) Towards Supporting Business Process Compliance Checking with Compliance Pattern Catalogues. In: Enterprise Modelling and Information Systems Architectures 10(1), pp. 67–88

Deng S., Wang D., Li Y., Cao B., Yin J., Wu Z., Zhou M. (2016) A Recommendation System to Facilitate Business Process Modeling. In: IEEE Transactions on Cybernetics, pp. 1–15

DIN (June 2003) 31051:2003-06: Grundlagen der Instandhaltung (31051:1985-01)

Döweling S., Tahiri T., Schmidt B., Nolte A., Khalilbeigi M. (2013) Collaborative Business Process Modeling on Interactive Tabletops. In: 21st European Conference on Information Systems (ECIS 2013). Paper 29. Utrecht, The Netherlands

Fellmann M., Bittmann S., Karhof A., Stolze C., Thomas O. (2013) Do we need a standard for EPC modelling? The state of syntactic, semantic and pragmatic quality. In: Proceedings of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2013). Lecture Notes in Informatics (LNI) Vol. 222, pp. 103–117

Fellmann M., Delfmann P., Koschmider A., Laue R., Leopold H., Schoknecht A. (2015) Semantic Technology in Business Process Modeling and Analysis. Part 1: Matching, Modeling Support , Correctness and Compliance. In: EMISA Forum 35

Fettke P. (2009) Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. In: ZfbF 61, pp. 550–580

Fettke P., Houy C., Loos P. (2010) Zur Bedeutung von Gestaltungswissen für die gestaltungsorientierte Wirtschaftsinformatik: Konzeptionelle Grundlagen, Anwendungsbeispiel und Implikationen. In: WIRTSCHAFTSINFORMATIK 52(6), pp. 339–352

Galler J., Scheer A.-W. (1994) Workflow-Management - Die ARIS-Architektur als Basis eines multimedialen Workflow-Systems.. Universität des Saarlandes. Universität des Saarlandes

Gruhn V., Laue R. (2007) Einfache EPK-Semantik durch praxistaugliche Stilregeln. In: Nüttgens M., Rump F. (eds.) EPK 2005 Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Gesellschaft für Informatik, Bonn, pp. 176–189

Harman J., Brown R., Kannengiesser U., Meyer N., Rothschädl T. (2015) Model as You Do: Engaging an S-BPM Vendor on Process Modelling in 3D Virtual Worlds In: S-BPM in the Wild: Practical Value Creation Fleischmann A., Schmidt W., Stary C. (eds.) Springer, Cham, pp. 113–133

Harmon P., Wolf C. (2016) The State of Business Process Management 2016. In: A BPTtrends Report

Hoffmann W., Kirsch J., Scheer A.-W. (1993) Modellierung mit Ereignisgesteuerten Prozeßketten In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes 101

Houy C., Fettke P., Loos P. (2009) Stilisierte Fakten der Ereignisgesteuerten Prozesskette – Anwendung einer Methode zur Theoriebildung in der Wirtschaftsinformatik. In: Nüttgens M., Rump F. J., Mendling J. (eds.) EPK 2009 Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Gesellschaft für Informatik, Bonn, pp. 22–41

Huth S., Wieland T. (2008) Geschäftsprozessmodellierung mittels Software-Services auf Basis der EPK In: Service-orientierte Architekturen: Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen Nissen V., Petsch M., Schorcht H. (eds.) Gabler, Wiesbaden, pp. 61–76

Jannaber S., Karhof A., Riehle D. M., Thomas O., Delfmann P., Becker J. (2016) Invigorating Event-driven Process Chains - Towards an intgrated meta model for EPC standardization. In: Proceedings of the 1st International Workshop on Adequacy of Modelling Methods (AQEMO'2016). Karlsruhe, Deutschland

Karhof A., Jannaber S., Riehle D. M., Thomas O., Delfmann P. (2016) On the de-facto Standard of Event-driven Process Chains : Reviewing EPC Implementations in Process Modelling Tools. In: Oberweis A., Reussner R. (eds.) Modellierung 2016. Gesellschaft für Informatik e.V., Bonn, pp. 77–91

Keller G., Teufel T. (1997) SAP R/3 prozeßorientiert anwenden. Addison-Wesley, Bonn

Kindler E. (2006) On the semantics of EPCs: Resolving the vicious circle. In: Data and Knowledge Engineering 56(1), pp. 23–40

Kolb J., Rudner B., Reichert M. (2012) Towards Gesture-based Process Modeling on Multi-Touch Devices. In: Advanced Information Systems Engineering Workshops, pp. 280–293

Kopp O., Unger T., Leymann F. (2006) Nautilus Event-driven Process Chains : Syntax, Semantics and their mapping to BPEL. In: Nüttgens M., Rump F., Mendling J. (eds.) EPK 2006 Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten Vol. 224. Gesellschaft für Informatik, Bonn, pp. 85–104

Koschmider A., Hornung T., Oberweis A. (2011) Recommendation-based editor for business process modeling. In: Data and Knowledge Engineering 70(6), pp. 483–503

Leimeister J. M. (2012) Dienstleistungsengineering und -management. Springer, Berlin, Heidelberg

Leimeister J. M., Glauner C. (2008) Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. In: WIRTSCHAFTSINFORMATIK 50(3), pp. 248–251

Leimeister J. M., Huber M., Bretschneider U., Krcmar H. (2009) Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. In: Journal of Management Information Systems 26(1), pp. 197–224

Leopold H., Eid-Sabbagh R. H., Mendling J., Azevedo L. G., Baiao F. A. (2013) Detection of naming convention violations in process models for different languages. In: Decision Support Systems 56(1), pp. 310–325

Maleri R. (2013) Grundlagen der Dienstleistungsproduktion, 4th edition. Springer, Berlin, Heidelberg

Matijacic M., Nüttgens M., Fellmann M., Thomas O. (15.-18. Dezember 2013 2013) Elicitation and Consolidation of Requirements for Mobile Technical Customer Services Support Systems. In: International Conference on Information Systems (ICIS 2013). Milan, Italy, pp. 1–16

Meffert H., Bruhn M. (2009) Dienstleistungsmarketing: Grundlagen-Konzepte-Methoden. In: Meffert-Marketing-Edition. Wiesbaden: Gabler

Meffert H., Bruhn M. (2012) Konzepte und theoretische Grundlagen des Dienstleistungsmarketing. In: Dienstleistungsmarketing. Gabler, Wiesbaden, pp. 43–80

Melville P., Sindhwani V. (2010) Recommender systems. In: Encyclopedia of Machine Learning, pp. 829–837

Mendling J. (2007) Detection and prediction of errors in EPC business process models.. PhD thesis, WU Wien, Vienna, p. 525

Mendling J., van der Aalst W. M. P. (2007) Formalization and Verification of EPCs with OR-Joins Based on State and Context. In: Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE'07) 4495, pp. 439–453

Metzger D., Niemöller C., Berkemeier L., Brenning L. (Feb. 2016a) Vom Techniker zum Modellierer - Konzeption und Entwicklung eines Smart Glasses Systems zur Laufzeitmodellierung von Dienstleistungsprozessen. In: Thomas O., Nüttgens M., Fellmann M. (eds.) Dienstleistungsmodellierung (DLM 2016). Springer, Berlin, Heidelberg, pp. 193–213

Metzger D., Niemöller C., Thomas O. (2016b) Design and demonstration of an engineering method for service support systems. In: Information Systems and e-Business Management 14(4), pp. 1–35

Mihalcea R., Corley C., Strapparava C. (2006) Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1. AAAI'06. AAAI Press, Boston, Massachusetts, pp. 775–780

Niemöller C., Metzger D., Thomas O. (2017) Design and Evaluation of a Smart-Glasses-based Service Support System. In: Leimeister J. M., Brenner W. (eds.) 13th International Conference on Wirtschaftsinformatik. 2017, pp. 106–120

Österle H., Becker J., Frank U., Hess T., Karagiannis D., Krcmar H., Loos P., Mertens P., Oberweis A., Sinz E. J. (2011) Memorandum on design-oriented information systems research. In: European Journal of Information Systems 20(1), pp. 7–10

Pendergast M., Aytes K., Lee J. D. (1999) Supporting the group creation of formal and informal graphics during business process modeling. In: Interacting with Computers 11(4), pp. 355–373

Recker J., Safrudin N., Rosemann M. (2010) How Novices Model Business Processes. In: Hutchison D., Kanade T., Kittler J., Kleinberg J. M., Mattern F., Mitchell J. C., Naor M., Nierstrasz O., Pandu Rangan C., Steffen B., Sudan M., Terzopoulos D., Tygar D., Vardi M. Y., Weikum G., Hull R., Mendling J., Tai S. (eds.) Business Process Management Vol. 6336. Springer, Berlin, Heidelberg, Berlin and Heidelberg, pp. 29–44

Remme M., Scheer A.-W. (1996) Konstruktion von Prozeßmodellen. In: Scheer A.-W. (ed.) Veröffentlichungen des Instituts für Wirtschaftsinformatik - Nr. 125. Saarbrücken

Riege C., Saat J., Bucher T. (2009) Systematisierung von Evaluationsmethoden in der gestaltungsorientierten Wirtschaftsinformatik, German. In: Becker J., Krcmar H., Niehaves B. (eds.) Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Physica-Verlag HD, pp. 69–86

Riehle D. M., Jannaber S., Karhof A., Delfmann P., Thomas O., Becker J. (2016) Towards an EPC Standardization – A Literature Review on Exchange Formats for EPC Models. In: Nissen V., Stelzer D., Straßburger S., Fischer F. (eds.) Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2016). Ilmenau, Germany

Riemer K., Holler J., Indulska M. (2011) Collaborative process modelling-tool analysis and design implications. In: 19th European Conference on Information Systems (ECIS 2011). Paper 39. Helsinki, Finland

Scheer A.-W., Grieble O., Klein R. (2006) Modellbasiertes Dienstleistungsmanagement, German. In: Bullinger H.-J., Scheer A.-W. (eds.) Service Engineering SE - 2. Springer, Berlin, Heidelberg, pp. 19–51

Scheer A.-W., Thomas O., Adam O. (2005) Process Modeling using Event-Driven Process Chains, pp. 119–145

Schlicker M., Blinn N., Nüttgens M. (2010) Modellierung technischer Serviceprozesse im Kontext hybrider Wertschöpfung. In: Thomas O., Loos P., Nüttgens M. (eds.) Hybride Wertschöpfung. Springer, Berlin, Heidelberg, Berlin, Heidelberg, pp. 144–175

Schütte R. (1998) Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Springer Fachmedien, Wiesbaden

Sonnenberg C., vom Brocke J. (2012) Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In: DESRIST 2012, pp. 381–397

Thomas O. (2006) Management von Referenzmodellen: Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen In: Wirtschaftsinformatik - Theorie und Anwendung Loos P. (ed.) Schriftenreihe: Wirtschaftsinformatik - Theorie und Anwendung, Berlin

Thomas O. (2009) Fuzzy Process Engineering. Gabler, Wiesbaden

Thomas O., Loos P., Nüttgens M. (2010) Hybride Wertschöpfung - Mobile Anwendungssysteme für effiziente Dienstleistungsproduktivität im technischen Kundendienst. Springer, Berlin, Heidelberg

Thomas O., Nüttgens M. (2012) Dienstleistungsmodellierung 2012 - Product-Service Systems und Produktivität. Springer Gabler, Wiesbaden

Thomas O., Nüttgens M., Fellmann M., Krumeich J., Hucke S., Breitschwerdt R., Rosenkranz N., Schlicker M., Özcan D., Peris M. (2014) Empower Mobile Technical Customer Services (EMOTEC) – Produktivitätssteigerung durch intelligente mobile Assistenzsysteme im Technischen Kundendienst. In: Nüttgens M., Thomas O., Fellmann M. (eds.) Dienstleistungsproduktivität. Springer Fachmedien, Wiesbaden, pp. 2–17

Thomas O., Scheer A.-W. (2016) Verfahren und Werkzeuge zur Informationsmodellierung In: Handbuch Unternehmensorganisation: Strategien, Planung, Umsetzung Spath D., Westkämper E. (eds.) Springer, Berlin, Heidelberg, pp. 1–30

Vago S. L., Lusch R. F. (2004) Evolving to a New Dominant Logic for Marketing. In: Journal of Marketing 68(1), pp. 1–17

vom Brocke J. (2002) Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen. Logos, Berlin, p. 404

Walter P. (2010) Technische Kundendienstleistungen: Einordnung, Charakterisierung und Klassifikation. In: Thomas O., Loos P., Nüttgens M. (eds.) Hybride Wertschöpfung. Springer, Berlin, Heidelberg, pp. 24–41

Weber B., Reichert M., Rinderle-Ma S. (Sept. 2008) Change patterns and change support features – Enhancing flexibility in process-aware information systems. In: Data & Knowledge Engineering 66(3), pp. 438–466

Weske M., van der Aalst W. M. P., Verbeek H. M. W. (2004) Advances in business process management. In: Data & Knowledge Engineering 50(1), pp. 1–8

Wieloch K., Filipowska A., Kaczmarek M. (2011) Autocompletion for Business Process Modelling. In: Abramowicz W. Maciaszek L. W. K. (ed.) Business Information Systems Workshops. BIS 2011. Vol. Lecture Notes in Business Information Processing, vol. 97. Springer, Berlin, Heidelberg, pp. 30–40

Wilmont I., Brinkkemper S., van de Weerd I. (2010) Exploring Intuitive Modelling Behaviour. In: pp. 301–313

Zeithaml V. A., Parasuraman A., Berry L. L. (1985) Problems and strategies in services marketing. In: The Journal of Marketing, pp. 33–46