Towards an Early Warning Mechanism for Adaptation Needs of Digital Services

Kurt Sandkuhl[1] and Henderik A. Proper[2,3]
[1]*University of Rostock, Rostock, Germany*, kurt.sandkuhl@uni-rostock.de
[2]*Luxembourg Institute of Science and Technology, Luxembourg*, e.proper@acm.org
[3]*University of Luxembourg, Luxembourg*

***Abstract*— In many sectors, the ability to quickly respond to changes in the environment of an enterprise is considered as the key factor to long-term competitiveness. The focus of the paper is on enterprises providing digital services and the need for quick-responsiveness with respect to adapting these services. We investigate the feasibility of an "early warning" mechanism for the need of change in digital services, i.e., to detect events or actions which are likely to cause change requests. Such a warning mechanism is expected to allow for an earlier implementation of changes in comparison to a situation without early warning. The proposed approach to support early discovery of events and quick implementation of actions is based on context computing. Essentially, we propose to capture all factors relevant for quick-responsiveness of an enterprise in a computable context model and trigger organizational actions as soon as a relevant event is detected. One of the conceptual foundations for our work are the identified different aspects that constitute the context of digital service development.**

Keywords – Digital Service, Digital Service Development, Context, Early Warning Mechanism.

I. Introduction

New customer demands, changed regulations, competitors with innovative products entering the own market segment, shareholder demands for a more profitable business – there are many reasons why enterprises continuously need to adapt their products, services, operations or structures. In many industrial areas and economies, agility, i.e., the ability to quickly respond to changes in the environment, is considered as the key factor to long-term competitiveness.

Quick-responsiveness of enterprises essentially requires:

- *Early discovery of events or changes in the environment that require a response*. The earlier an event or a change can be detected reliably, and qualified as relevant or not, the earlier a valid response is possible. It should be noted that such events or changes can be of diverse nature: an increasing number of customer complaints about missing product features in social media, statistics of service downtime in server logs, or an utterance of the company's CEO that integration of services is required – to name but a few examples.
- *Quick implementation of actions that are an effective and sufficient response to the discovered events and changes.* Such actions usually require an organizational preparedness in terms of responsive processes and organizational structures, and a flexibility of the resources or platforms required for the actions.

The focus of our work is in particular on enterprises providing digital services. Marketing sciences (e.g. [1] and [2]), suggests that, now that the service economy has matured, the notion of economic exchange has also shifted from following a goods-dominant logic to a service-dominant logic. The former takes the view that economic exchange revolves around transactions, where value is exchanged, e.g. in terms of a transfer of goods versus monetary value. The latter shifts this towards the view that the creation of value involves the integration of resources, of providers and users, to the benefit of the latter. Most services delivered in the service economy are digital services in the sense that they are IT reliant / enabled. IT is generally seen as being the key enabler of the digital service economy [3]. Agility in a service-oriented enterprise means that there has to be a quick-responsiveness with respect to adapting the services provided.

Our work aims to contribute to the understanding of the dynamics of the socio-economical-technical environment in which digital services development and delivery takes place. The focus of this paper is on the possibility to conceptualize an early warning mechanism, i.e., to forecast change needs of digital services, allowing for an earlier reaction and, thus, an earlier reaction than without this early warning. The proposed approach to support early discovery of events and quick implementation of actions is based on context computing. It builds on earlier work on relevant aspects that constitute the context of digital service development.

Essentially, we propose to capture all factors relevant for quick-responsiveness of an enterprise in a computable context model and trigger organizational actions as soon as a relevant event is detected in the context represented by the model. Similar approaches are used in many IT-based systems, for example control systems in manufacturing environments reacting to situations discovered by sensors or decision support systems evaluating parameters in a situation. However, to identify context and required actions in IT-systems is much simpler than in a complex socio-technical system with many stakeholder groups such as an enterprise. Here, the research challenge is to understand what events are actually relevant to issue early warnings, how these events can be discovered, and how to react to them.

The main contributions of the paper are (a) to identify categories of events and their sources which can be used for early discovery of change needs, (b) to show the feasibility of capturing such events in a context model and associated context computing environment, and (c) two case studies for digital service design supporting our argumentation. The structure of the remainder of this paper is introduced in section 2 when presenting the research design.

II. Research Design

The structure of our research follows the design science research (DSR) methodology proposed by [4]. Our long-term vision is to be able to discover changes in the environment which may affecting the design of digital services, and to adapt digital service development accordingly. For example, by adjusting development processes, tools or practices or by offering assistive functions to the stakeholders involved. The artefact in focus of the DSR project is a framework for adaptive digital service design.

In the first part of our DSR project, we focused on problem identification and the initial design of the framework. For the problem identification, we performed a literature analysis on collaboration in information systems development, collaboration support for modelling and context modelling. Furthermore, we conducted a case study in digital service development in the domain of business process outsourcing. The design of the initial framework resulted in a model of the socio-technical context of digital service design (see section III.A) validated in the case study. The results of the first DSR phase are documented in [5].

This paper focuses on the second part of the DSR project, which basically represents the next design-evaluate cycle of the framework. The initial framework does not address what kinds of changes in the service design context are relevant and what kinds of (information) sources have to be considered to detect them. The second version of the framework extends the initial one by adding a classification of relevant events, potential ways of detection, and a classification of information sources. The extension is based on the analysis of two industrial cases of digital service design (see section IV.). In both cases, we analyze what causes of change in digital services are visible in industrial practice. These causes and the origin of the causes (e.g., stakeholders, documents or information systems) are used to derive a classification of causes and relevant sources. For evaluation purposes, the use of the framework in an early warning mechanism is demonstrated in section VI. Demonstration includes development of a context model for one of the cases and implementation of context monitoring using the context design and delivery platform of the CaaS[1] (**C**apability **a**s **a** **S**ervice for Digital Enterprises) project (see section III.C).

[1] http://caas-project.eu

In the last section of the paper, we reflect on the results of the case study analysis and the demonstration by deriving implications for future work.

III. Background and Related Work

The background for our work is the initial version of the framework for adaptive digital service design which is briefly summarized in section III.A.. Related work can be found in the context computing (section III.B.) and in digital service development (section III.C).

A. Context of Digital Service Design

Previous work on the context of digital service design resulted in the conclusion that digital service development happens in a complex socio-technical-economic environment [5]. The actual service design context has to be differentiated into three different aspects

- the context of the development life-cycle,
- the context of local practices of the involved stakeholder groups and
- the context of the external environment of the organization relevant for service deployment.

The local practices represent the various stakeholder groups involved in service design. Here, it is important to understand different work practices, concerns, roles, etc., including their input into and output from collaborative processes. The external environment represents the potentially different settings of deployment and operation of the digital service under consideration where it is relevant to identify changes and contextual variations affecting service design. The development life-cycle should not only respond to external environment changes and adapt to needs implied by local practices, but also has own procedural and artefact-related variations relevant for the collaboration process.

The existence and relevance of these aspects were confirmed in a case study and captured in a model. The digital service design aspects form the basis for continued work in the case studies presented in section IV.

B. Context Computing

Context computing plays an important role to detect changes in the environment relevant for early warning systems. The term “context” has been used and still is subject of research in various application areas and sectors of computer science. In the most general meaning, context describes what relates the entity under consideration to the environment surrounding this entity. What an “entity” is depends on the context’s interpretation. Hoffmann [6] provided a way to classify these interpretations as follows:

- Situational context includes any information characterizing state or situation of a person, object or location. This information has to support the purpose of understanding or being relevant for the interaction between user and application. Situational context models are often used in ubiquitous computing [7].

- Relational context includes any information pertinent to characterizing the relation of an entity to other entities, where this information is judged according to a given purpose. An example from problem solving is given in [8] using contextual graphs for this purpose.
- Formal representations of a perception or part of reality are like a model of an individual's viewpoint, which expresses a local view of the reality.

In this paper, we use the term context according to Dey, who defines context as "*any information that can be used to characterize the situation of an entity, where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves*" [7].

Context-computing was first introduced in 1994 by Schilit and Theimer [9]. They consider the context as the information about a located-object and the changes to this object over time. With the growth of mobility of users, the increase of performance and functionality of mobile devices and sensors, and the sheer explosion in the amount of available data, context computing also gains importance in order to integrate circumstances and situations of the users. The latter is often referred to as human related context.

Many different approaches were developed to represent *context* using a formal language (e.g., UML [10] or OWL [11]). In our work, we will apply the context modeling approach from CaaS (**C**apability **a**s **a** **S**ervice for Digital Enterprises; see section III.C) in order to capture the context information relevant for early warnings on changes in digital services.

C. Context Computing in CaaS

The development of digital services builds on a long tradition of information systems engineering [12], business process engineering [13], and enterprise architecture [15]. Each of these specializations of, what might be broadly called, enterprise and information systems engineering have developed their own strategies to deal with the need to tackle change requests and requirements from different stakeholders, in particular when confronted with fast changing contexts. Systems and service innovation is an inherently collaborative process involving different partners and stakeholders.

An explicit on focus on conceptualizing *context* as part of development of digital services and on adapting to changes in context has been taken in work on capability management in the CaaS project, see e.g. [14]. Capability management defines a systematic way to plan, design, develop, deploy, operate, and adjust capabilities. Management processes are often organized in cycles, typically including a PDCA-like structure. For capability management, three integrated cycles are proposed which partly reflect the different time horizons and partly the different levels of abstraction relevant in management [16]: (1) business model development, (2) capability design, and (3) capability delivery and adjustment. All three cycles are tightly interrelated and require mutual feedback.

In the business model development cycle, the strategic business goals of an enterprise and the strategy regarding business model development are maintained, continuously updated and translated into structures and processes required for their implementation. In established enterprises, this usually includes a transformation process from the current situation into a future state. During this cycle, an essential element of the operationalization of strategic goals is the planning of capabilities, i.e. the decision about new capabilities to be created, capabilities to be modified, or capabilities to be terminated. The planning process includes definition or identification of KPIs to be used in capability design and delivery for deciding on goal achievement. An important input for the strategic planning comes from the capability design cycle and includes the performance of the capabilities. The planning also provides input to the capability design cycle.

The capability design process is initiated by a business request for a new capability (the request might be initiated by strategic business planning, changes in context or discovery of new business opportunities). The design cycle starts with a formalized definition of requested capabilities and definition of the relevant context. The requested capability is matched to enterprise architecture and required business services and variants are identified. If IT solutions or process variants are missing, IT development for the missing elements is started. The capability designed is assessed for its business and technical feasibility. If capability delivery is deemed feasible, business structures and software enabling capability delivery are put in place. The capability delivery application is developed following the development process used by a company, and the capability driven approach provides the necessary inputs characterizing capability delivery goals, context data and algorithms for dealing with changes in the context.

The delivery cycle starts with the IT-solutions ready and deployed in the target platform. During delivery, business goals and KPIs of the capability are monitored. Context changes are captured and alternative capability delivery evolution scenarios are evaluated. If capability is not delivered as requested, delivery adjustment is invoked to modify the capability delivery. The iterative development is used mainly due to the need to account for additional context factors because not all relevant factors can be identified during the first development iteration. Updates in the capability are initiated according to the results of capability delivery monitoring and adjustment. These results indicate validity of context model and business service variants.

IV. Industrial Case Studies

Two case studies on adaptation needs in digital service design were conducted to study events and actions motivating an early warning mechanism and to classify such events and the sources they originate from. The overarching purpose of the case studies can be captured in terms of two research questions:

1. What kinds of events and actions can be observed in industrial practice which cause changes in digital services and motivate implementation of an early warning mechanism?
2. What are the origins or sources of the events and actions causing the changes?

Both questions aim at extending the model of service design context.

In order to investigate the above questions, we performed an ex-post analysis of two industrial cases. Case A focused on the development of a new category of digital services for a manufacturer of elevators in the area of digital signage. Case B addresses the extension of digital services for managing fleet of lawn mowers, trimmers and other garden products for housing companies. Within both cases, we had access to several information sources: Both cases were part of research projects investigating business model development and service design and development. As part of the projects, the companies' digital service offerings, their development processes and aspects relevant for deployment were investigates. This information is included in the projects' deliverables and accessible to the researchers. Furthermore, in case B one researcher worked during 3 months one or two days per week at the company's facilities. The researcher was part of the service development and maintenance team. The researcher collected information about the work processes, technologies used and practices by participating in the work and taking notes. The company management agreed to this and the co-workers were informed about the purpose of the data collection.

A. Programmatic Advertising on Digital Signage

The first industrial case involves a company developing and producing elevators with global market presence and more than 100 years of tradition. In 2016, the company started a new "digital" business line outside the established manufacturing value chain. The group of digital services is to offer targeted advertisements and information on elevator doors or screens, i.e. digital signage on elevators (EDS; elevator door signage) [20]. The case study company has detailed knowledge who operates the elevators and what kinds of users (i.e., target groups for the ads) are frequently using the elevator. This makes the elevator an interesting space to sell in marketing campaigns which is the core of the business model.

Subject of the case study was the development of front end, back office and infrastructure for EDS. One example is to add gesture recognition to the elevator door which made it an interactive EDS and improved user experience. A second example is the so called black-board which is offered to customers as alternative to the elevator door solution. Housing industry is considered as one of the main target market for the black-board.

The principal components of the EDS solution are as follows: Above the elevator door a data projector is mounted or, alternatively, on one side of the door, a shock-proof large touch-screen fixed. The data projector (or the touch screen) uses the data communication facility of the elevator shaft to connect to a communication device for all elevators in the building. This communication device exchanges information with the back-office, including content and interactions with EDS and for the purpose of elevator maintenance. A content management system in the back office offers functionality to clients for booking certain EDS solutions, defining the content to be displayed based on various parameters (time of the day, type of building, amount of visitors), reporting and billing. This kind of dynamic targeted advertising is also known as programmatic advertising.

B. Fleet Service for Power Garden Products

The enterprise considered in the second case is a world-leading producer of outdoor power products including chainsaws, trimmers, robotic lawn mowers, garden tractors, watering systems, cutting equipment, and diamond tools for the construction industries. The case study company offers products and services for both the private and industrial market. Many of the products for professional customers do not only have built-in electronics or embedded systems but also networking and communication capabilities. The built-in IT is in many cases used for controlling the different mechatronic components of the product and for collecting data when the product is in use, either performance parameters or the environment of the product. The networking features are used for communicating usage statistics, license information or location information to either the product owner or the back-office of the manufacturer [21]. Other functions are software upgrades and functionality add-ons, e.g. for optimizing energy consumption. The features and functions are used in digital services allowing for operating, monitoring and maintain large fleets of devices. Target groups for such fleet services are housing companies or service providers.

A typical scenario from a professional customer perspective includes different products installed for supporting maintenance of parks or green areas, all of them equipped with wireless communication. This could include a fleet of robotic mowers as well as man-operated trimmers or cutting devices. The robotic mowers and man-operated devices provide operations data to a base station in the buildings and receive software updates or schedules from it. The base station is connected to the cloud by using the customer's Internet access. In the cloud, backend and customer services are available. Thus, the owner of the fleet has access to services for operating, supervising and planning garden maintenance using mobile devices.

Since many of the products offer similar functionality regarding networking and communication, the case company designed and implemented reusable services and components for either the product or the back-office infrastructure which comprise an IT and service architecture. Customer frequently request new features which requires a high degree of business involvement, fast turnaround, and frequent update of business ideas into applications.

V. Case Study Results

Based on the two research questions defined in section IV. we analyzed the available documentation of both cases. The results are presented in the next sections with a focus on events in V.A. and on Information sources in V.B. Section V.C. contains a discussion of the observations.

A. Events and Actions

The identified events and actions that caused changes in the digital services of case A and B are shown in the next three tables. Table 1 starts with a selection of events discovered in the case material which are related to stakeholders and their local practices. In total, 9 events were detected in the material of case A and 17 in Case B. In case A, a very visible stakeholder outside the actual service development team was the product manager who produced many inputs to the service design process which were documented in diverse artefacts, such as the business plan or minutes from the bi-weekly product board meeting. In case B, the number of stakeholder groups working independently from each other in their local practices was larger. Here, sales and marketing were responsible for market-related aspects of the service development (target setting); controlling was providing inputs used for feature optimization; customer relations provided a channel for customer inputs leading to features or change requests.

Table 1: Excerpt of changes caused in the local practice contexts

Case	*Event or Action*	*Source*
A	New product feature in EDS by integrating additional hardware	Business plan
B	Change in package “fleet service”: new combination of services	Target setting document
B	Change in pricing for end customer to separate connection fee	Change request
A	Differentiation of market segments in hospitality area	Product Design Board minutes
A	Definition of performance requirements and possibility of remote maintenance	SLA with client
B	Data protection requirements limiting remote access	Policy document
B	Architecture platform decision	Strategy document
A	New target market announced as plan for next year	Chairman of the board

In the context of the development life-cycle, quality manager, enterprise architect and the members of the DevOps were the primary originators of changes in Case B which were documented in various artefacts and systems. In Case A, the development team was smaller and platform and tools were newly established. This explains why also team composition and platform changes were visible. Table 2 shows an excerpt of the events discovered which can be traced to the development life-cycle context. In total, 11 events were detected in the material of case A and 23 in Case B.

Table 2: Excerpt of changes caused in the development life-cycle context

Case	*Event or Action*	*Source*
A	Migration to new operating system platform	Software configuration
A	Separation of responsibilities for user experience and interaction devices	Development team composition
A	Additional external services and interfaces to be integrated	Product specification
B	Modified common service specifications	Architecture specification
B	Changes quality requirements and test specifications on enterprise level	Quality handbook
B	Changed technical requirements to the platform on individual service and service package level	Requirements management system
B	Changes distribution of responsibilities in the DevOps tea,	Task specification

The enterprise-external deployment environment showed many events originating from the customers or their user groups. Furthermore, service providers and regulators or supervising authorities caused changes in the design context. Table 3 shows an excerpt of changes caused from the deployment environment. In total, 4 events were detected in the material of case A and 7 in Case B.

Table 3: Excerpt of changes caused in the deployment environment

Case	*Event or Action*	*Source*
B	Customer request for error correction	Trouble Ticket System
B	Modified priorities of features in product roadmap	Notes from User group meeting
A	Customer request for new features	Product manager’s notes
A	Changes in specification triggered by new data protection regulations	Product manager’s notes
B	Platform and configuration change in client’s computing center	Fleet service product manager
B	License expiry of external service	Error log
B	New privacy policy of service provider	E-Mail from service provider

B. Information Sources

In a second analysis step, we focused on research question 2, i.e. the origins or sources of the events and actions causing the changes. We went through the material collected (see section V.A) one more time and grouped the context information into sources. Table 4 shows a categorization of kinds of information sources and examples from the cases.

Table 4: Sources of context information

Kind of Source	*Examples*
Enterprise information system	Trouble Ticket System Requirements Management System
Documents	Notes from User group meeting Product manager's notes
Models	Architecture model
Social Media	Product manager's notes
Public web services	New regulations published by standardization bodies or regulator
Stakeholders oral statements	Announcements of the chairman of the board
Personal information management	E-Mail to the project manager about policy change

C. Discussion

All in all, the analysis resulted in a substantial number of events and actions, which were visible in the case documentation and caused adaptations of the digital service development or the digital services as such. All three aspects of the digital service context identified in our previous work were also visible in the two cases. The events have different levels of granularity: some indicate bigger structural changes actually made up by more than a dozen change requests; others cause only minor adaptations. Many occurrences of events in documents seem actually to be based on earlier events which caused the change but were not documented. Example: the decision to introduce new services in case B was a result of a longer discussion process between two stakeholder groups (sales and controlling) which was documented only in the notes of these stakeholder groups. From an early warning perspective, relying on service design documentation therefore probably would not be sufficient. Even the local practices of the stakeholders would have to be integrated in the early warning mechanism.

One of the aims of this paper is to categorize the events and actions as a contribution to extending the framework for service design context. Such a categorization can be made from different perspectives and according to different criteria. The most relevant ones from our perspective are the topic of the event and the relevance for an early warning mechanism. The topic of the event is required for deciding on the required adaption of the digital service development, i.e., has the functionality of digital services to be adapted, or the development process, the group of stakeholders involved in the development, the way of deployment, the customer interaction – to mention a few examples. The relevance for early warning basically treats if the events are critical from a timing perspective and can be detected.

When categorizing the events from a topic perspective, we observed that all events discovered in the local practice context of stakeholders are related to the dimensions typically discussed in business model innovation and design (see e.g [18] and [19]). These business model dimensions are

- *Value proposition*: what kind of value proposition does a digital service offer to potential customers? Changes in functionality or non-functional features of services typically are originating from a new or modified value proposition in the business model. Detecting shifts in the value proposition of a digital service before the actual change requests are formulated would be highly relevant for an early warning mechanism.
- *Packaging of the value proposition*: what value proposition or combinations of value propositions are offered to what customer segments? Changes in the packaging often result in adapting the configuration of digital services with effects on user interfaces.
- *Suppliers or partners*: what parts of the service are contributed by suppliers or in cooperation with partners? If new alliances with partner are formed or changes in suppliers are upcoming, interfaces, data flows and technical platforms often are affected.
- *Distribution channels*: how does the service offering reach the customer? When changing customer touch points or including new delivery channels, the user interface oriented part of the digital services often need modification or extension.

The analysis of events from the development life-cycle showed that existing categorizations from software engineering could be applied, for example the categories published in [17]. Thus, we will use this categorization and evaluate it for completeness in future work.

When analyzing the information sources, we made several observations. Much information available in documents would actually be suitable for capturing in models and were also represented in model-like representation in the documents. An example is a process model for documenting future work flows in a description of changes to be made in the system provided by the sales team in case B. Another example is an architecture sketch originating from the product owner in case A when indicating the changes required due to the new operating system version. Furthermore, some very important events originating from stakeholders were oral statements, informal information or non-official e-mails which only were visible in the case documents because members of the service design team decided to write a note and enter the note into the project documentation. It can be expected that much more oral and informal information actually is relevant for an early warning mechanism but not captured.

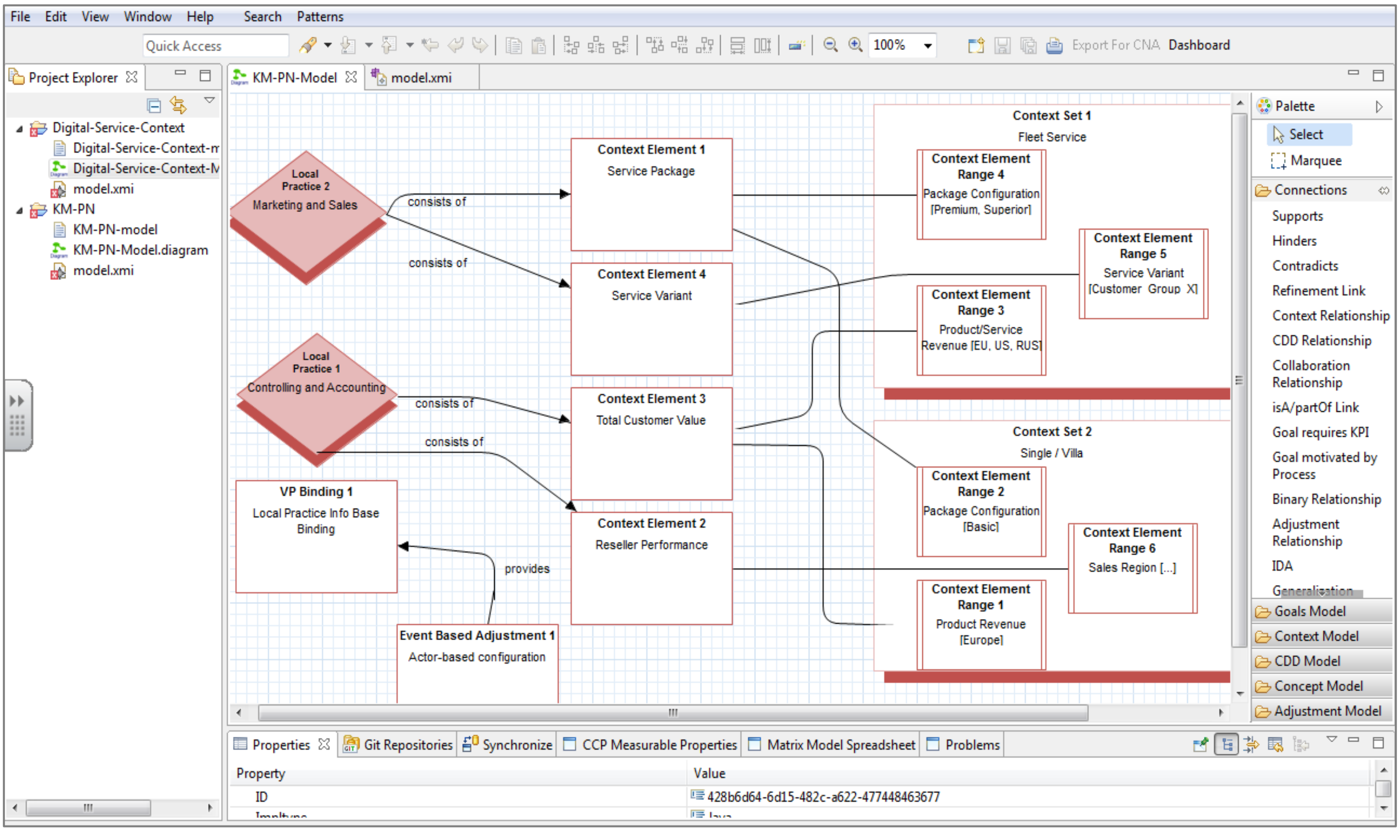

Fig. 1. Model of Digital Service Context in Case B

VI. Feasibility Study of an Early Warning Mechanism

The case study results show what kind of events and sources have to be considered in the development of an early warning mechanism and the results also identify the actual events and sources for the two industrial cases. In order to show the feasibility of using these results for implementing an early warning mechanism, we decided to CaaS approach which requires two general steps:

1. In CaaS, basically the context to be monitored has to be captured in a context model which clearly identifies, which context elements are relevant, what context indicators operationalize every context element, what sources can be used to compute the value of an indicator and what changes in indicators or indicator constellations have to trigger what action in response.
2. The context model is used to configure a monitoring application. Configuration means that the binding of context indicator sources in the model to actual sources in an IT infrastructure (e.g., a web service computing and delivering the value) has to be done and that the binding of the response action to an actual service has to happen. This could, in a simple case, be to change the values in a dashboard.

To demonstrate feasibility of the first step, we developed a context model for case B which is depicted in figure 1. The excerpt of the context model visible in the screenshot in figure 1 shows context elements for two stakeholder groups, "Marketing and Sales" and "Controlling". These stakeholder groups are modeled as local practices and represented with red rhombuses in the upper left part of the figure. "Marketing and Sales" is responsible for service packages and service variants for customer groups. Changes in packages and variants are relevant for digital service design and thus modeled as context elements (no. 1 and 4, upper middle of fig.1). As service packages are relevant for the two digital service areas "fleet service" and "single/villa", the context element is part of two context sets (fleet service and single/villa, right side in fig.1). This is specified by defined context element ranges in both sets. Context set 1 also includes the service variants for customer groups and the "Product/Service Revenue" context elements which originates from "Controlling" and is included in the context model because underperforming product/service packages will lead to changes in digital service offerings. "Controlling" monitors "Total customer value" and "Reseller Performance" (context elements 2 and 3), which both are linked to context element ranges in context sets 1 and 2. The indicators used to capture the context elements are also part of the context model, but not visible in the excerpt depicted in figure 1.

As we performed an ex-post analysis of case B, step 2 (configuration of a monitoring application) cannot be implemented in a real-life scenario. In principle, the practical applicability of using context models for monitoring real-world applications has been proven in CaaS in various cases.

VII. Conclusion

Based on our observation, we have some preliminary conclusions which require further work for elaboration and evaluation:

- Detecting events probably is not sufficient for implementing an early warning mechanism. A more complex mechanism seems to be required which could be characterized as understanding situations. Situations would be constellations of stakeholders and discussion topics leading to events or actions.
- Adaptation mechanisms in digital service developments have to go far beyond functional or non-functional change requests, but also to include team compositions, development process changes or tool composition.

Although the model of digital service design context and the identification of kinds of events and actions and their sources indicate the feasibility of an early warning mechanism for change needs in digital services, much more work is required to implement our long-term vision. Our work showed isolated events causing changes in digital services but did not investigate or capture the underlying collaborative structures and socio-technical interaction. We envision a much broader approach which will have to be part of the future work. More concrete, we envision an extension of our framework for collaborative development of digital services, comprising:

Collaborative dynamics: An explanatory theory identifying performance indicators of collaborative service development processes, factors that influence these indicators, and their (causal) relationships. This part of the framework, will allow us to express / understand the dynamics of the socio-economical-technical environment in which the development of digital services takes place.

Methodological support: Methodological support for doing collaborative service development, involving (situational) guidelines and heuristics to structure the collaborative aspects of (digital) services development. The focus should be on the collaborative aspects, thus complementing existing development methods. The situational guidelines and heuristics will be motivated in terms of the relationships and causalities of the explanatory theory part of the framework.

Situational instrumentation: Situational instrumentation for doing collaborative service development, involving instruments, such as modelling frameworks, canvassing techniques, collaboration support tools, and other IT-based tools. The situational aspect will result in additional guidelines providing guidance in selecting / tuning the tools for specific situations at hand. These situational guidelines will also be motivated in terms of the relationships and causalities of the explanatory theory part of the framework.

VIII. References

[1] Vargo, S.L., Lusch, R.F.: Institutions and axioms: an extension and update of service-dominant logic. Journal of the Academy of Marketing Science 44 (2016)

[2] Grönroos, C., Ravald, A.: Service as Business Logic: Implications for Value Cre-ation and Marketing. Journal of Service Man 22, 5–22 (2011)

[3] Maglio, P.P., Vargo, S.L., Caswel, N., Spohrer, J.: The Service System is the Basic Abstraction of Service Science. Information Systems and e-business Man-agement 7, 395–406 (2009)

[4] Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24, 45–77 (2007)

[5] Proper, H.A., Sandkuhl, K.: The Context of Collaborative Digital Service Development. Proceedings of the 2nd International Workshop on Practicing Open Enterprise Modelling at PoEM 2018, Vienna, Austria, October 31, 2018, Vol. 2238: CEUR-WS.org (CEUR Workshop Proceedings), pp. 73–87.

[6] Hoffmann, P.: *Context-based Semantic Similarity across Ontologies*, PhD-thesis, Claude Bernard University of Lyon, 2008.

[7] Dey, A.K.: “Understanding and Using Context,” *Personal Ubiquitous Computing,* Vol. 5, No. 1, 2001, pp. 4-7.

[8] Brézillon, P.: “Context Modeling: Task Model and Practice Model,” Springer, LNCS 4635, 2007, pp. 122-135.

[9] Schilit, B.N. and Theimer, M.M.: Disseminating Active Map Information to Mobile Hosts, *Network*, IEEE, Vol. 8, No. 5, 1994, pp. 22-32.

[10] Henricksen, K., Indulska, J., Rakotonirainy, A.: “Modeling context information in Pervasive Computing Systems,” *Pervasive ’02: Proceedings of the First International Conference on Pervasive Computing*, Springer-Verlag, 2002, pp. 167-180.

[11] Wang, X., Zhang, D., Gu, T., Pung, H.: “Ontology Based Context Modeling and Reasoning Using OWL,” *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, 2004, pp. 18-22.

[12] Bernus, P., Mertins, K., Schmidt, G.: Handbook on architectures of information systems. Springer Science & Business Media (2013)

[13] van der Aalst, W. M. P., ter Hofstede, A. H. M., Weske, M.: Business Process Management: A Survey. In:, pp. 1–12

[14] Bērziša, S., Bravos, G., Gonzalez, T. C., Czubayko, U., España, S., Grabis, J., Henkel, M., Jokste, L., Kampars, J., Koç, H., Kuhr, J.-C., Llorca, C., Loucopoulos, P., Juanes Pascual, R., Pastor, O., Sandkuhl, K., Simic, H., Stirna, J., Giromé Valverde, F., Zdravkovic, J. (2015). Capability driven development: an approach to designing digital enterprises. Business & Information Systems Engineering, 57(1), 15-25.

[15] Lankhorst, M.M. et al: Enterprise architecture at work: Modelling, communication and analysis. Springer (2009)

[16] Sandkuhl, K., & Stirna, J. (Eds.). (2018). Capability Management in Digital Enterprises (No. 1). Springer.

[17] Williams, B. J., Carver, J. C. (2010). Characterizing software architecture changes: A systematic review. Information and Software Technology, 52(1), 31-51.

[18] Schallmo, D. R., Williams, C. A. (2018). Digital Transformation Now!: Guiding the Successful Digitalization of Your Business Model. Springer.

[19] Wirtz, B. W., Pistoia, A., Ullrich, S., Göttel, V. (2016). Business models: Origin, development and future research perspectives. Long range planning, 49(1), 36-54.

[20] Wißotzki, M., Sandkuhl, K., Smirnov, A., Kashevnik, A., & Shilov, N. (2017). Digital signage and targeted advertisement based on personal preferences and digital business models. In 2017 21st Conference of Open Innovations Association (FRUCT) (pp. 374-381). IEEE.

[21] Sandkuhl, K., Seigerroth, U., & Kaidalova, J. (2017, October). Towards Integration Methods of Product-IT into Enterprise Architectures. In 2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW) (pp. 23-28). IEEE.