A Framework to Model and Implement Mobile Context-Aware Business Applications

Julian, Dörndorfer[1] and Christian Seel[1]

Abstract: The success and ubiquity of mobile devices like smartphones and tablets changed the daily work activities of many employees and employers. With the data provided by mobile devices in combination with external data from databases or web services, it is possible to recognize the context of a business process. Context recognition allows to adapt the business process by selecting the next process step or providing the necessary data, like the next customer near the current location. However, to use the advantages of context recognition, mobile business processes have to be designed, implemented and executed in a context-aware manner. Therefore, this paper presents a comprehensive framework for modeling context-aware business processes, which comprise the business process as well as the information collection to evaluate the needed context. Furthermore, it presents an architecture for the realization of context-aware applications.

Keywords: Context-aware Business Processes, BPMN, Domain Specific Modeling Language (DSML), Mobile Architecture

1 Introduction

With the introduction of the iPhone ten years ago a boost of mobile devices like smartphones or tablets took place, which increasingly leads to a disruption of "traditional" work conditions and executions [Pr16], [In13b], [Mo14], [KK14], [Rh13], such as waiters using smartphones to accept orders from the customer and billing the meal with it. The emerging generation of digital natives which are entering more and more the job market will further drive this change from stationary to mobile working conditions [Pr16]. The widespread use of mobile devices is leading to business processes being executed independent of the location.

The design, implementation, execution and controlling of business processes is a standard approach in theory and practice [HC93], [BKR11], [VR10], [Sc00], [Bi16]. The nearly ubiquitous presence of mobile devices can be used to support the execution of business processes. The use of mobile devices improves the quality and flexibility of business processes, and also saves time and costs during the execution [FL14], [HL15]. Another aspect is that mobile devices are providing many sensors and can be additionally equipped via *Bluetooth* or other proprietary protocols with more sensors. Moreover, mobile devices are capable to request additional data from other sources, like databases or web services, via their internet connection.

[1]University of Applied Sciences Landshut, Computer Sciences, Am Lurzenhof 1, 84036 Landshut,
julian.doerndorfer|christian.seel@haw-landshut.de

The aggregation and interpretation of sensor data enable detecting the context of the users and support them during the execution of a business process. DEY [De01] describes context as "any information that can be used to characterize the situation of an entity." Therefore, context recognition can be used to automatically present useful information for the user or adapt the application behavior and moreover pre-select the next possible process steps. To support the design and implementation of supportive mobile context-aware applications, the business process languages also have to consider contextual influences [DS16]. In addition, the aggregation of context through sensor data can be complex. For instance, the context parameter *weather* basically consists of the sensor data *humidity*, *wind speed* and *temperature*. Moreover, *weather* can also be a sensor for another context parameter, like the *street conditions* in a navigation system. To use the opportunities of context recognition via mobile devices and ease the design of mobile context-aware business processes, this paper answers the following research questions.

> **RQ.1**: How can a modeling framework for mobile context-aware applications be designed?
>
> **RQ.2:** How can the architecture of a mobile context-aware application be built?

The remainder of this paper is structured as follows: In section 2 a brief overview of the existing literature will be given. Afterwards, the modeling approach will be presented, which comprises a Business Process Model and Notation (BPMN) extension and a sensor modeling language (**RQ.1**). Section 3.3 discusses an approach for an architecture for mobile context-aware applications (**RQ.2**). Hereafter, section 4 shows an example application which supports the business process from the previous section. The paper ends with a conclusion and outlook to further research in mobile context-aware applications.

2 Related Work

Besides DEY context was also defined by others [SAW94], [SBG99], [We91]. Not only is DEY's definition well known and accepted in the scientific community, but he also declares when an application is context-aware. It is context-aware "if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task" [De00]. Some approaches to make business processes more flexible have been made. ROSEMANN et al. [RRF08] claim that modeling languages have to be more flexible to model context. Further, they state that an increased attention on flexibility took place in the research area, which leads to a decreasing time-to-market for products [RRF11]. Therefore, the result is a demand for higher process flexibility [So05]. In particular, ROSEMANN et al. show the limitation of the actual event-driven process chain (EPC) language and the lack of supporting context modeling. However, they do not present an appropriate way to integrate the identified context in business processes. The extension C-EPC is an approach to make EPC more configurable for decisions at runtime, but it doesn't address the context in particular [Rv07]. The approach from LA VARA et al. aims to reveal all possibilities of a business process and integrate them into one process model. However, this leads to large and complex models. In [SN07] an approach to identify and apply context in a business process is introduced, but it is more of a theoretical framework to identify context and does not show how a context-aware business process could be designed. In [HS15] HEINRICH and SCHÖN mention that business processes have to consider “not-static” context events that change the process execution. Moreover, they present an algorithm which supports automated process planning for context-aware processes, but no modeling representation in BPMN. CONFORTI et al. present an approach to manage process risks by sensor evaluation [Co13]. Furthermore, they show a way to model these risks and when they occur. They only consider sensor data and context evaluation for process risks, but context and its evaluation can be used for more than risk analysis for business processes. DÖRNDORFER and SEEL present in their paper a BPMN extension for business processes, which is able to model context in business processes via a complete modeling technique [DS17]. They also developed a context-free grammar to state brief context expression for decisions depending on context. Furthermore, two extensions for UML are published by AL-ALSHUHAI and SIEWE. In the first paper, they are extending the class diagram with additional annotations [AS15b]. The second paper [AS15a] expands the activity diagram to mark context-aware areas or sequences. All the presented papers do not consider how complex context evaluation is and how context-aware business processes can be designed. Furthermore, the implementation of supportive mobile applications is not supported by the modeling languages. The area of domain specific (modeling) languages (DS(M)L) is partly covered by the following articles. A DSL for multiple mobile platforms [KCO10] and for context modeling in the context-aware system [HGB13] were found. Both articles introduce a DSL which enables a description of context. However, both approaches do not present a possibility to model the context aggregation graphically. SHENG and BENATALLAH [SB05] are introducing an adapted UML to enable context modeling. In addition, BERARDINELLI et al. [BCD10] also present an extension for the UML. However, both approaches are not integrated into BPMN, which seems to be difficult. Therefore, they cannot be used for further development.

For the architecture of mobile context-aware applications two main paradigms exist. Firstly, the client approach, which is to recognize context information on the mobile device. The architecture reminds of a traditional *model view controller* (MVC) approach [BA11], [Ch08], [KKC11], [Sh12]. Recognizing and preparing context information on a server is the second approach. All devices which can collect data via sensors send their information to a backend application that evaluates the context information [VL12], [JKR01], [DAS01], [He05]. Both approaches have specific disadvantages. With the server approach all collected data have to be sent to the server, the data have to be evaluated and at last the result (context) has to be sent back to the mobile device. This requires an internet connection, which is not always available for a mobile device. In addition, the described data transmission sometimes leads to unnecessary delays, when the context evaluation can also be conducted on the mobile device. The client approach, on the other hand, is limited to the resources of these devices. If big data packages have to be evaluated, the processor or the memory can be pushed to their limits, thus delaying the process. In addition, some information relevant to evaluate the correct context, might not be accessed from a mobile device.

Therefore, this paper presents a context-aware framework which firstly enables to design mobile context-aware business processes. Secondly, it shows a modeling language for evaluation of context. And thirdly it shows how context-aware application can be designed to support the conduction of context-aware business processes. Thus, this paper presents a new artifact in the design science paradigm by HEVNER et al. [He04].

3 Modeling a Mobile Context-aware Application

To increase the efficiency of business processes and ease the planning and implementation of mobile context-aware applications, this paper introduces a framework (**RQ.1**). On the left side of **Fig. 1** are the context parameter which influences the process. The first step is to create a context-aware business process (first layer). The modeling language of choice is Context4BPMN [DS17] because it extends the standard BPMN to enable the creation of context-aware business processes. It uses so-called *Context Expressions* as conditions for paths or elements depending on context. To evaluate the *Context Expression* the domain specific modeling language *SenSoMod* can be used (second layer). It enables to model how information from sensors can be aggregated to context information (cf. section 3.2). The model hereafter can be used to create a mobile context-aware application which supports the execution of the context-aware business process (third layer).

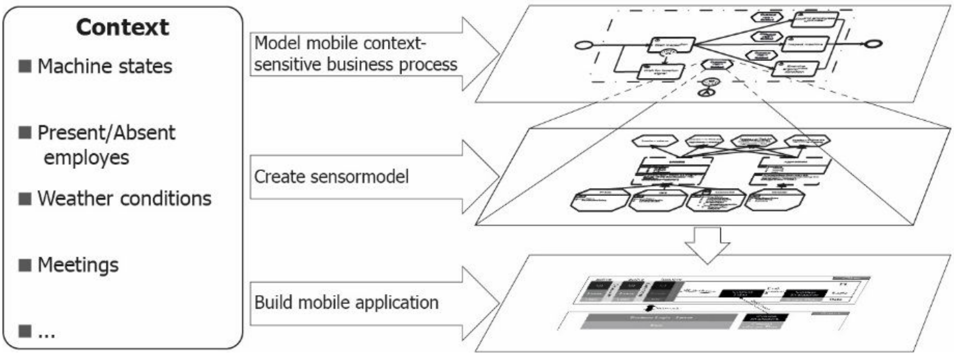


Fig. 1. The framework to model and create a mobile context-aware application

An example of a mobile context-aware business process are traveling salespersons in a sales department of a company. They partly work at the customer's to sell the products from the company as well as in the company to participate in meetings or further training. Moreover, some paperwork can also be done at home. Therefore, in this example the first context parameter is the *location* with three states *at the customer, at the office* and *at home*. Other parameters are the *season* and the *weather* because some goods can be better sold in specific seasons or in certain weather condition [Mu10], [Bu12]. In addition, with the context parameter *customer history,* the mobile application is capable to adapt the list of goods to the preferences of the customer. For instance, the non-food products can be deactivated if the customer only bought food-related products so far. Therefore, we use this example to demonstrate the framework.

3.1 Context-Aware Business Processes

To reach higher flexibility in business processes, context has to be considered in modeling languages. BPMN is a standard modeling language for business processes [In13a] and has a built-in extension mechanism [Ob11]. The mechanism was used to create the Context4BPMN extension [DS17]. **Fig. 2** depicts the business process of the salesperson mentioned in section 3 at design time.

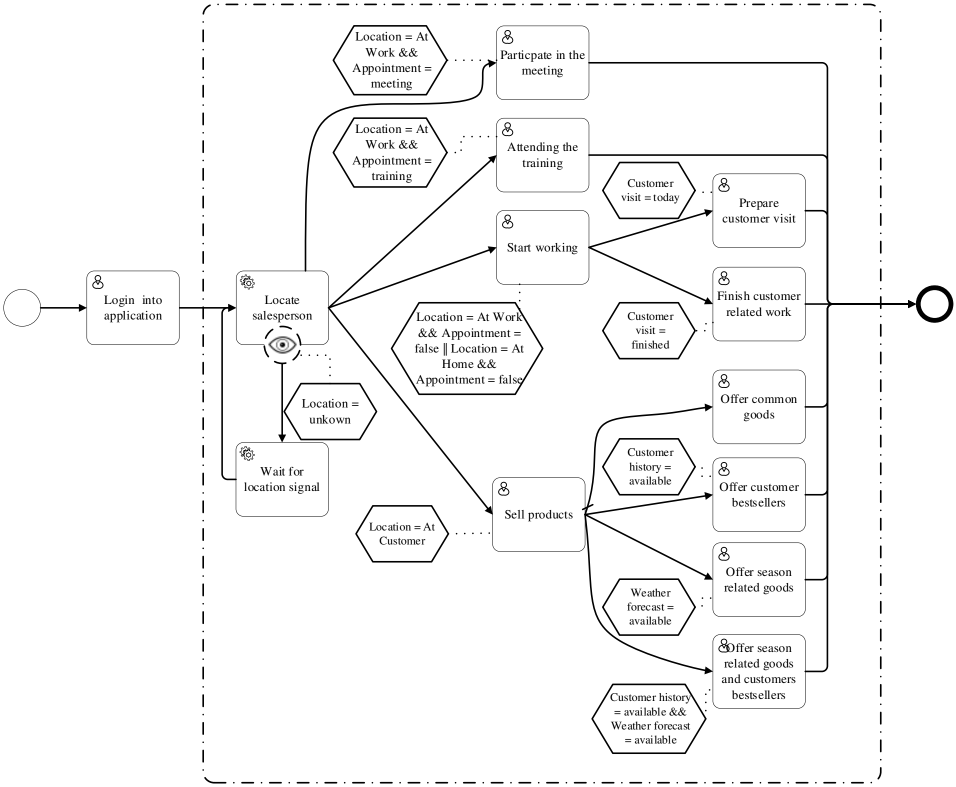
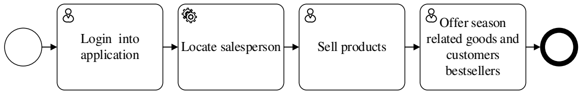
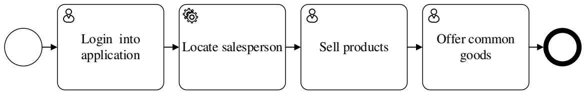
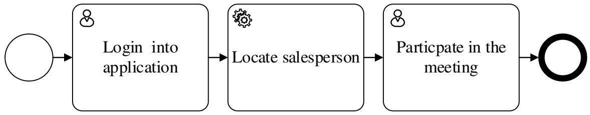


Fig. 2. The salesperson business process designed with the BPMN extension *Context4BPMN*

The first action after starting the application is log in. Hereafter, a dotted line marks the ongoing business process sequence as context depending. After the login, the application locates the salesperson. Attached to the task *locate salesperson* is the *context event* symbolized through an eye. It is a *non-interrupting context event* which is triggered when the location is undefined. The activation condition is stated in the hexagon with a context-free grammar to formalize the conditions. A parallel activity starts in which the application tries to determine the location of the salesperson. If a location is identified, four different tasks are possible. They all mainly depend on location. For example, the salesperson takes part in the sales training if s/he is at work and a training is noted on the calendar. A different path is when the salesperson is at the customer. Obviously, s/he wants to sell the products of the company to the customer. Depending on the accessible data, the mobile application can support this process by showing the customer's bestsellers and/or seasonal depending goods. The first action needs the customer history which can be queried via a connection to the customer's relationship management system (CRM). The seasonal goods can be shown when the weather forecast for the local area is available.

No.	Context	Run-time business process
1	Location = At Customer && Customer history = available && Weather forecast = available	
2	Location = At Customer && Customer history = unknown && Weather forecast = unknown	
3	Location = At Work && Appointment = meeting	

Tab. 1: Variations of the salesperson business process at run-time

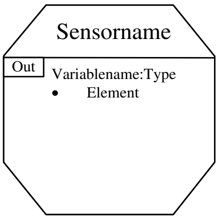
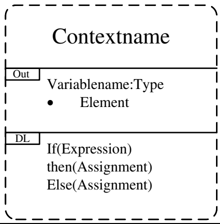
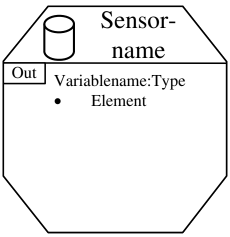
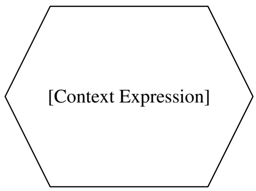
The business process will be transformed at run-time into a standard BPMN workflow. **Tab. 1** depicts three variations of the business process for three different context situations. The first row depicts the process when the salesperson is at the customer, the customer history is available and also the weather forecast is available. Therefore, the salesperson offers the customer his bestsellers and seasonal related goods. In the second row, the only determined context is that the salesperson is at the customer. Hence, s/he sells the common goods of the company. In the third variation, the customer is at work and has an appointment with colleagues on the calendar. So s/he participates in the meeting.

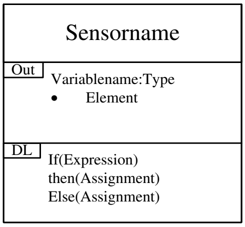
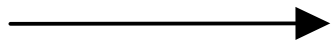
3.2 Sensor Modeling

The extension of the BPMN enables to consider static and non-static context events. It is therefore possible to design context-aware business processes. However, the extension of the BPMN does not specify how context can be evaluated by sensors. For instance, how can the context *location* be measured? This sounds like an easy task because the obvious answer would be via GPS, but where is the location *at home*? When can it switch to *at work*? To address these questions the sensor modeling language has been developed. To further ease the design and implementation of mobile context-aware applications which support the execution of context-aware business processes, an additional language is needed.

Notation of SenSoMod

Context is measured through sensors. We understand a sensor as any source of information for context. This can be a “usual” physical sensor, – like a hygrometer or a temperature sensor – a database, or an application from which information could be requested. Even a machine in an assembly hall that is accessible through a network connection can be a source of information. Therefore, different types of sensors have to be distinguished. There are atomic sensors which cannot be aggregated from other sensors. Two kinds of atomic sensors exist: *Physical sensors,* which measure physical quantities like temperature or humidity, and *virtual sensors* which are dedicated to non-physical quantities like databases, machines or stock states. Besides the atomic sensor, there is the computed sensor. It is a sensor that relies on other sensors, which could be atomic sensors or other computed sensors. For example, the sensor *weather* could be the combination of the atomic sensors *humidity* and *temperature*. To address the different kind of sensor types we gave each of them a graphical representation (cf. **Tab. 2**).

Element	**Notation**	**Element**	**Notation**
Physical atomic sensor		Context	
Virtual atomic sensor		Context description	

Element	**Notation**	**Element**	**Notation**
Computed sensor		Flow	
Multiple Instances	**III**		

Tab. 2: The notation of SenSoMod

The *physical sensor* is designated to depict physical quantities, whereas the *virtual sensors* are dedicated to non-physical quantities. To differentiate these two atomic sensors, the virtual sensors are marked by a database symbol to the left of the sensor name field. Beneath the name is the output area marked with *Out*. The area is intended to describe the structure of the outgoing objects of the sensor like the type and elements of it. First the name and – separated by a colon – the type of the return object have to be stated. If the object type has specific values – like enums, arrays or lists – they can be stated in a bullet list. Alternatively, the return values can be specified in the JavaScript Object Notation (JSON) RFC-7159 [In14] notation. Next to the two atomic sensors is the *computed sensor*. It can be used to combine different atomic sensors and/or *computed sensors*. Weather, for example, can be aggregated from different types of sensors. *Computed sensors* also have an area for describing outgoing elements. Furthermore, they have a *DL* area dedicated to describing the decision logic for their outgoing objects. This is necessary to express when a certain state from the *Out* area of a sensor will be returned. For example, the location *at work* will be returned when the logged-in network name is *companyNetwork*. The accurate description of the decision logic language is presented in **Fig. 3**. The *context* notation is the next element in the table. The name of the element has to match with the name of the involved context in the *context description*. Only *context* elements can be connected with a *context description*. Like the sensors, the *context* has the *DL* and *Out* areas. A *context* is based on at least one sensor, of some type. The *context description* is the graphical representation of an expression to describe a contextual influence in a business process. The context expression language is part of the Context4BPMN extension [DS17]. To connect the different elements and show the sequence stream, the *flow* element has to be used. *Multiple instances* is the last element in the table and is an attribute for any type of sensors. It indicates that a sensor occurs in more than one instance, and is placed to the right of a sensor name. An example, using the introduced elements, is given in **Fig. 4**. **Fig. 3** depicts the logical decision language which has been developed. The language is a context-free grammar in the Extended Backus Naur Form (EBNF) [Ba60] and serves to briefly and precisely express the decision in the *DL* area of the *sensor* and *context* elements. A decision logic consists of a *LogicTerm* and can have a default assignment. The *LogicTerm* itself is built out of an *Expression* and an *Assignment*. If the expression is evaluated as *true*, the variable will be assigned to the designated value. The assigned variable has to exist in the output area of the computed sensor or context. To shorten some basic definitions, like integer numbers or date, we link with “-->” to standards, like strings. In order to express the structure of the object which a sensor returns to a request, the “Out” area is provided. First the name and – separated by a colon – the type of the return object have to be stated. Optionally, the values of the type (if existing) could be stated in a bullet list. Alternatively, the return values could be specified in the JSON notation.

```
<Decisionlogic>::= <LogicTerm> ["else(" <Assignment> ")"] |
   <LogicTerm>", "<LogicTerm> ["else(" <Assignment> ")"]
<LogicTerm>::= "If(" <Expression> ")then(" <Assignment> ")"
<Assignment> ::= <Variable> "=" <Value>
<Expression>::= <Variable> [<MathematicalOperator> <Constant>]
   <Comparison> <Value> | <Expression> <LogicOperator> <Expression>
<Comparison>::= "=" | "!=" | "<=" | ">=" | "<" | ">"
<MathematicalOperator>::= "*" | "/" | "+" | "-"
<LogicOperator>::= "&&" | "||"
<Constant>::= --> "DoubleNumber"
<Interval>::= -->"IntegerNumber" "msec" | "sec" |"min" | "hours" | "days"
<Variable>::= -->"StringIdentifier in UTF-8"
<Value>::= -->"StringIdentifier in UTF-8"
```

Fig. 3. EBNF Grammar for the Decision Logic (DL)

Fig. 4 shows the sensor model for the example from section 3. At the bottom line, the sensors are represented. For the context *location*, which is represented in the dotted rectangle above, the physical sensors *WiFi*, *GPS* and the virtual sensor *Location-DB* are needed. The GPS sensor returns an object consisting of a long- and latitude double, to compute the position of the salesperson. In the center of the figure, the context elements are represented. The *DL* areas show when an *appointment* is a *meeting* or *training,* respectively when the *location* is *at work, at home, at the customer* or *unknown*. For example if the calendar entry is today and the entry description contains the name *training,* the salesperson should be part of the training. At the top of the figure are the context expressions related to the context elements. They are also the connection to the related context-aware business process. This example shows that it is possible to model the context evaluation for mobile context-aware business processes with SenSoMod. It enables to model the aggregation of information from the basic sensor information to the context expressions.

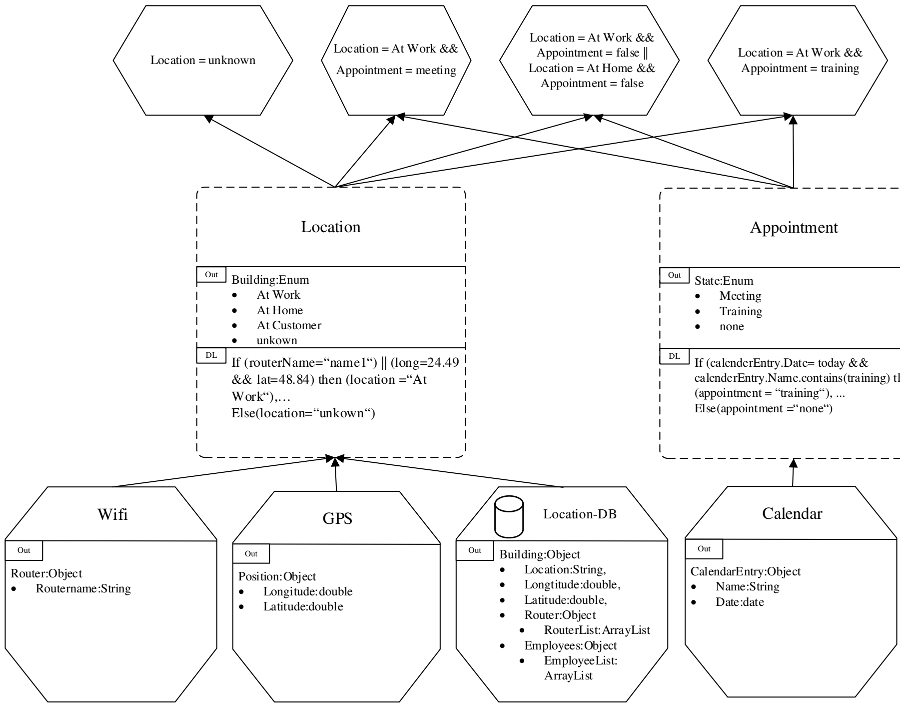


Fig. 4. Sensor model for the contexts *location* and *appointment*

3.3 Architecture of the Mobile Context-Aware Application

To implement a mobile application an architecture for context-aware applications has to be designed (**RQ.2**). Unfortunately, the standard architectures of the operating systems for mobile applications do not support context-awareness [Go17], [Ap17]. Therefore, to implement the application, an architecture considering context has to be created. The context recognition allows adapting the application to the context, which means that certain parts of the application have to be activated, whereas other parts can be deactivated to reduce the information overload for the user. The result is that logical components have to be encapsulated. **Fig. 5** shows the architecture of the supportive mobile context-aware application. The application is divided into a server and a client part. The server contains the normal logic and data layer to exchange data with the client applications and handle the data storage. In some cases, the server also has a graphical user interface (GUI) layer, but this does not matter for the architecture of mobile context-aware devices. In a separated part the evaluation of the context and the storage of the context relevant data is encapsulated. This is necessary because not every context can be evaluated on the mobile device. For instance, the context *weather* will be evaluated by sensors in a weather station. The evaluated context on the server side will be sent to the mobile context-aware application. The client is usually separated into a 3-layer architecture consisting of a user-interface (UI), logic, and data-layer. To realize the adaptive components, the layers are divided vertically. The vertical parts are *context components* which can be activated or deactivated depending on the evaluated context. The *context components* also have a UI-, logic- and data-layer dedicated to their specific task, but they have to be independent of the other context components to achieve activation or deactivation.

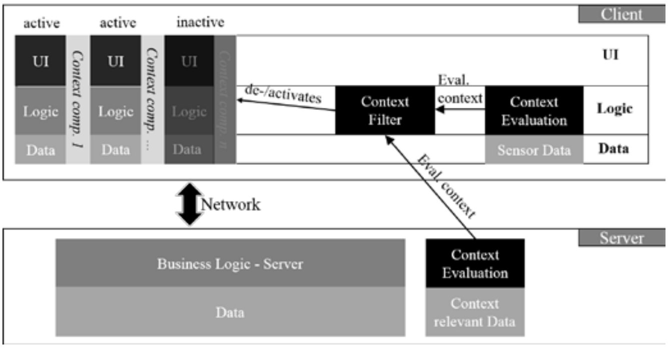


Fig. 5. Schematics of the architecture of the mobile context-aware application

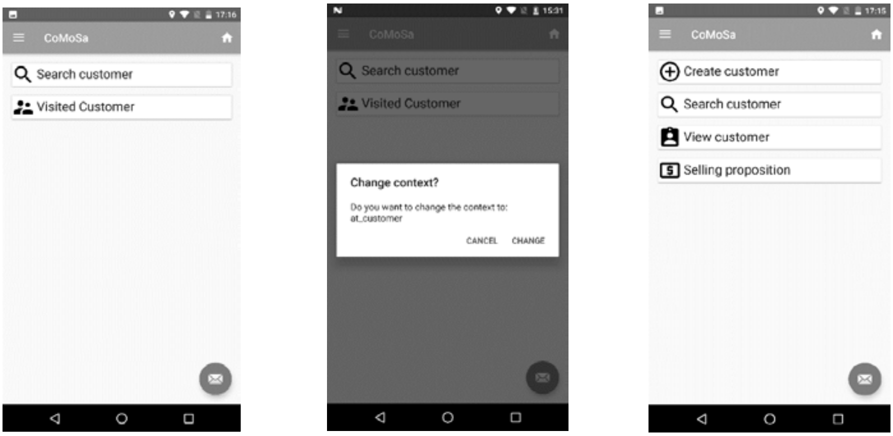
Context, which can be evaluated on the mobile devices, is encapsulated in the *context evaluation* component which also has its own sensor data. A typical example for this is *location*. It can be evaluated via the GPS or WiFi sensor. The decision taken when a certain context is recognized can be seen in the sensor model in the *DL* areas. Furthermore, the model supports the programmer by showing how many different context objects exist and what the return values of the context objects are. The evaluated context from the mobile device and the server will be sent to the *context filter* component. It decides to activate or deactivate a certain *context component* depending on the evaluated context.

4 Implementation of a Prototype

The designed mobile context-aware business processes and its sensor model eases the implementation of a supportive mobile application. The sensor model can be used as a blueprint for the sensor and context classes which will be needed to evaluate the context of the user. Furthermore, the return types of the sensor and context objects, as well as the decision logic, can be used in the source code of the context evaluation. The context-aware business process can be used to derive the scope of the application and which parts of the application are needed under which condition.

Proof of Concept

Based on the modeled mobile context-aware business process, the sensor model and the architecture of the application were developed. In **Fig. 6** three screenshots are depicted demonstrating how the mobile application reacts to the context recognition. Screenshot a) shows the home screen when the context *at work* was recognized. It supports the user with his tasks to *prepare a customer visit* or *finish customer related work*. The first task will be supported by researching the customer to get relevant data, like address, contact or sell history. By showing the visited customers the user can finish related work like editing customer details, confirm discounts or set shipping dates. The next screenshot shows a pop-up when a new context was recognized. The mobile application does not force the adaption of the UI, instead the pop-up lets the user decide to adapt the UI. If the user declines, the mobile application remains unchanged, otherwise the UI will be changed. The home screen when the context *at customer* was recognized is depicted in screenshot c). In contrast to the screenshot a) the component *visited customer* is deactivated and the components *create customer*, *view customer* and *selling proposition* are activated. The user will be supported to sell the company's products by the component *selling proposition*. By clicking on it, a submenu will be shown and depending on the availability of the *weather data* and *customer history* it then recommends products.



The mobile context-aware application shows how the user can be supported in executing their tasks. Furthermore, the modeling languages help to implement these kinds of applications by clarifying how the context will be evaluated (SenSoMod) and what the user needs during the execution of the business process (Context4BPMN).

5 Outlook and Further Research

The main contribution of this article is to show how the framework helps to model mobile context-aware application to support the execution of business processes. This comprises an extension of the BPMN, a domain specific modeling language and an architecture for mobile context-aware applications. It provides the possibility for model engineers to plan such processes in a precise, detailed and comprehensive way. It also enables programmers to reuse the decision logic in the sensor model in the source code of the application.

Therefore, it can improve the interaction between modelers and programmers and accelerate the adaption of business processes. There are some tasks for further research in this area. The introduced framework need an evaluation with practitioners to get feedback from the target group and improve its usefulness. Since mobile context-aware business processes obviously need to measure context and are supported by an application on smart devices, an automated or semi-automated way to generate code from the business process would be helpful to increase the flow between modeling and implementation phase. The logical context expressions can be used to generate decisions in the application program. Furthermore, the sensor model can be utilized to pre-generate classes and interfaces. In an additional step, the use of the gathered context data from the execution of a business process will be investigated. The data could be used to identify problems in the execution and therefore be interesting for the controlling phase.

Literaturverzeichnis

[Ap17] Apple: About the iOS Technologies. https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html, 03.03.2017.

[AS15a] Al-alshuhai, A.; Siewe, F.: An Extension of UML Activity Diagram to Model the Behaviour of Context-Aware Systems: Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015; S. 431–437.

[AS15b] Al-alshuhai, A.; Siewe, F.: An Extension of Class Diagram to Model the Structure of Context-Aware Systems: The Sixth International Joint Conference on Advances in Engineering and Technology (AET), 2015.

[BA11] Barrenechea, E. S.; Alencar, P. S.C.: An Adaptive Context-Aware and Event-Based Framework Design Model. In Procedia Computer Science, 2011, 5; S. 593–600.

[Ba60] Backus, J. W. et al.: Report on the algorithmic language ALGOL 60. In Communications of the ACM, 1960, 3; S. 299–314.

[BCD10] Berardinelli, L.; Cortellessa, V.; Di Marco, A.: Performance Modeling and Analysis of Context-Aware Mobile Software Systems. In (Rosenblum, D. S.; Taentzer, G. Hrsg.): Fundamental approaches to software engineering. 13th international conference, FASE 2010, held as part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20 - 28, 2010 ; proceedings. Springer, Berlin, 2010; S. 353–367.

[Bi16] Bichler, M. et al.: Erratum to. Theories in Business and Information Systems Engineering. In Business & Information Systems Engineering, 2016.

[BKR11] Becker, J.; Kugeler, M.; Rosemann, M. Hrsg.: Process management. A guide for the design of business processes. Springer, Berlin, 2011.

[Bu12] Busse, M. et al.: Projection Bias in the Car and Housing Markets. National Bureau of Economic Research, Cambridge, MA, 2012.

[Ch08] Choi, J.: Context. From Birth to Design.: International Conference on Advanced Language Processing and Web Information Technology, 2008; S. 347–352.

[Co13] Conforti, R. et al.: Real-time risk monitoring in business processes. A sensor-based approach. In Journal of Systems and Software, 2013, 86; S. 2939–2965.

[DAS01] Dey, A.; Abowd, G.; Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In Human-Computer Interaction, 2001, 16; S. 97–166.

[De00] Dey, A. K.: Providing Architectural Support for Building Context-aware Applications. Georgia Institute of Technology, Atlanta, GA, USA, 2000.

[De01] Dey, A. K.: Understanding and Using Context. In Personal and Ubiquitous Computing, 2001, 5; S. 4–7.

[DS16] Dörndorfer, J.; Seel, C.: The impact of mobile devices and applications on business process management. In (Barton, T. et al. Hrsg.): Prozesse, Technologie, Anwendungen, Systeme und Management 2016. Angewandte Forschung in der Wirtschaftsinformatik, 2016; S. 10–19.

[DS17] Dörndorfer, J.; Seel, C.: A Meta Model Based Extension of BPMN 2.0 for Mobile Context Sensitive Business Processes and Applications. In (Leimeister, J. M.; Brenner, W. Hrsg.): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI), St. Gallen, 2017; S. 301–315.

[FL14] Falk, T.; Leist, S.: Effects of mobile solutions for improving business processes. In (Avital, M.; Leimeister, J. M.; Schultze, U. Hrsg.): ECIS 2014 proceedings. 22th European Conference on Information Systems ; Tel Aviv, Israel, June 9-11, 2014, AIS Electronic Library, 2014.

[Go17] Google: Android Platform Architecture. https://developer.android.com/guide/platform/index.html, 03.03.2017.

[HC93] Hammer, M.; Champy, J.: Reengineering the corporation. A manifesto for business revolution. Harper Business, New York, NY, 1993.

[He04] Hevner, A. R. et al.: Design Science in Information Systems Research. In MIS Q, 2004, 28; S. 75–105.

[He05] Henricksen, K. et al.: Middleware for Distributed Context-Aware Systems. In (Hutchison, D. et al. Hrsg.): On the Move to Meaningful Internet Systems 2005. CoopIS, DOA, and ODBASE. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005; S. 846–863.

[HGB13] Hoyos, J. R.; García-Molina, J.; Botía, J. A.: A domain-specific language for context modeling in context-aware systems. In Journal of Systems and Software, 2013, 86; S. 2890–2905.

[HL15] Heinrich, B.; Lewerenz, L.: A Novel Concept for the Usage of Mobile Information Services. In (Linnhoff-Popien, C.; Zaddach, M.; Grahl, A. Hrsg.): Marktplätze im Umbruch. Digitale Strategien für Services im mobilen Internet. Springer Vieweg, Berlin, 2015; S. 319–329.

[HS15] Heinrich, B.; Schön, D.: Automated Planning of Context-aware Process Models. University of Münster, Münster, Germany, 2015.

[In13a] International organization for standardization (iso): Information technology. Object Management Group Business Process Model and Notation, 2013.

[In13b] Intel IT Center: Mobile Computing Trends: Insight into Today's Workforce, 2013.

[In14] Internet Engineering Task Force: The JavaScript Object Notation (JSON) Data Interchange Format, 2014.

[JKR01] Jang, S.-I.; Kim, J.-H.; Ramakrishna, R. S.: Framework for Building Mobile Context-Aware Applications. In (Kim, W. et al. Hrsg.): The Human Society and the Internet Internet-Related Socio-Economic Issues. First International Conference, Human.SocietyInternet 2001 Seoul, Korea, July 4-6, 2001 Proceedings. Springer, Berlin, Heidelberg, 2001; S. 139–150.

[KCO10] Kramer, D.; Clark, T.; Oussena, S.: MobDSL. A Domain Specific Language for multiple mobile platform deployment: 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications. IEEE, 2010; S. 1–7.

[KK14] Kerr, D.; Koch, C.: A Creative and Useful Tension? Large Companies Using "Bring Your Own Device". In (Bergvall-Kåreborn, B.; Nielsen, P. A. Hrsg.): Creating Value for All Through IT. IFIP WG 8.6 International Conference on Transfer and Diffusion of IT, TDIT 2014, Aalborg, Denmark, June 2-4, 2014. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, s.l., 2014; S. 166–178.

[KKC11] Kim, S.; Kim, E.; Choi, Y.: Composite context information design and model approach for adaptive service decision: 13th International Conference on Advanced Communication Technology (ICACT), 2011. 13 - 16 Feb. 2011, Phoenix Park, Gangwon-Do, Republic of Korea ; proceeding. IEEE, Piscataway, NJ, 2011; S. 1593–1598.

[Mo14] Morabito, V. Hrsg.: Trends and challenges in digital business innovation. Springer, Cham, 2014.

[Mu10] Murray, K. B. et al.: The effect of weather on consumer spending. In Journal of Retailing and Consumer Services, 2010, 17; S. 512–520.

[Ob11] Object Management Group (OMG): Business Process Model and Notation (BPMN), Version 2.0, 2011.

[Pr16] Prümper, J. et al.: Abschlussbericht der Studie: „Mobiles Arbeiten". spring Messe Management GmbH, Frankfurt am Main, 2016.

[Rh13] Rhee, K. et al.: High-Level Design for a Secure Mobile Device Management System. In (Marinos, L.; Askoxylakis, I. Hrsg.): Human aspects of information security, privacy, and trust. First international conference, HAS 2013, held as part of HCI International 2013, Las Vegas, NV, USA, July 21 - 26, 2013 ; proceedings. Springer, Berlin, 2013; S. 348–356.

[RRF08] Rosemann, M.; Recker, J. C.; Flender, C.: Contextualisation of business processes. In International Journal of Business Process Integration and Management, 2008, 3 (1); S. 47–60.

[RRF11] Rosemann, M.; Recker, J.; Flender, C.: Designing context-aware Business Processes. In (Siau, K.; Chiang, R.; Hardgrave, B. C. Hrsg.): Systems analysis and design. People, processes and projects. M.E. Sharpe, Armonk, NY u.a, 2011; S. 51–73.

[Rv07] Rosemann, M.; van der Aalst, W.M.P.: A configurable reference modelling language. In Information Systems, 2007, 32; S. 1–23.

[SAW94] Schilit, B.; Adams, N.; Want, R.: Context-Aware Computing Applications: First Workshop on Mobile Computing Systems and Applications (WMCSA), 1994; S. 85–90.

[SB05] Sheng, Q. Z.; Benatallah, B.: ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development: International Conference on Mobile Business (ICMB), 2005; S. 206–212.

[SBG99] Schmidt, A.; Beigl, M.; Gellersen, H.-W.: There is more to context than location. In Computers & Graphics, 1999, 23; S. 893–901.

[Sc00] Scheer, A.-W.: ARIS - Business Process Modeling. Springer, Berlin u.a., 2000.

[Sh12] Shrestha, A. et al.: A Framework for Building and Operating Context-Aware Mobile Applications. In (Venkatasubramanian, N. Hrsg.): Mobile wireless middleware, operating systems, and applications. 4th international ICST conference, Mobilware 2011, London, UK, June 22 - 24, 2011 ; revised selected papers. Springer, Heidelberg, 2012; S. 135–142.

[SN07] Saidani, O.; Nurcan, S.: Towards Context Aware Business Process Modelling: Workshop on Business Process Modelling, Development, and Support, Norway, 2007; S. 1.

[So05] Soffer, P.: On the Notion of Flexibility in Business Processes: Proceedings of the CAiSE’05 Workshops, 2005; S. 35–42.

[VL12] Verclas, S.; Linnhoff-Popien, C. Hrsg.: Smart Mobile Apps. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[VR10] Vom Brocke, J.; Rosemann, M. Hrsg.: Handbook on Business Process Management 2. Strategic Alignment, Governance, People and Culture. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010.

[We91] Weiser, M.: The Computer for the 21st Century. In Scientific American, 1991, 265; S. 94–104.