Generating Smart Glasses-based Information Systems with BPMN4SGA: A BPMN Extension for Smart Glasses Applications

Jannis Vogel[1] and Oliver Thomas[1]

[1] University of Osnabrueck, Chair of Information Management and Information Systems,
Katharinenstr. 3, 49074 Osnabrueck, Germany
{jannis.vogel,oliver.thomas}@uni-osnabrueck.de

Abstract. Although smart glasses allow hands-free interaction with information systems and can enhance business processes, they face problems with the adoption in businesses. Implementation challenges arise due to specific hardware conditions e.g. computational power, limited battery, small screen size and privacy issues caused by the camera. In addition, not many programmers are specialized for the development of smart glasses-based applications to conquer the mentioned challenges. We address this issue with a generation tool for smart glasses-based information systems. A BPMN extension for smart glasses applications allows the abstract specification. Specified processes are then integrated into a model-driven software development approach that transforms processes directly into smart glasses applications. This paper covers the design and development phase of the abstract and concrete syntax of the BPMN extension and the representation of the architecture to generate smart glasses-based information systems with the new developed BPMN extension.

Keywords: Smart Glasses, BPMN Extension, Model-driven Software Development, Domain-specific Modelling Language.

1 Introduction and Motivation

Due to the mobility, hands-free interaction with information systems and a range of different functionalities, smart glasses are suitable for various application areas [1]. Widespread applications are pick-by-vision [2], remote assistance [3] and step-by-step guidance [4]. In particular smart glasses are suitable as a service support system to guide users through a complex working process such as in the technical customer service domain [4]. In general, smart glasses are applied in businesses to enhance business processes due to information provided in the user's field of view.

However, different obstacles arise during the implementation of smart glasses in businesses. Hobert and Schumann identified 25 challenges with the integration of wearables (i.e. smart glasses and smart watches) into businesses through 21 semi-structured interviews. They recognized that organizations have an urgent demand for software developers that know the implementation of wearables into businesses.

14th International Conference on Wirtschaftsinformatik,
February 24-27, 2019, Siegen, Germany

Furthermore, interviewed experts in the study mentioned that only a few wearable third-party experts exist [5]. This circumstance hinders the adoption of smart glasses and hampers an adequate dealing with smart glasses specific implementation barriers such as the small screen size [6], limited battery capacity and privacy issues [5].

A promising approach to overcome the mentioned obstacles is the usage of domain-specific modelling languages (DSML) to design software [7], in this case smart glasses applications, and to steer the implementation process with a model-driven software development (MDSD) approach [8], i.e. models have a direct effect on the desired software. During the research project Glasshouse [9] we followed that approach by using the modelling language BPMN to describe smart glasses-based use cases in the logistics domain. Thereby, we identified that existing process modelling languages such as EPC or BPMN are not suitable for the technological representation of smart glasses functionalities [10]. Consequently, a proper model-based design of smart glasses applications and their utilization in a MDSD approach was not feasible.

Hence, we address implementation obstacles and the non-technical expressiveness of process modelling languages in terms of smart glasses with the development of an abstract and concrete syntax of a BPMN extension for smart glasses-based information systems. Thereby, adequately represented smart glasses-based process models lead to smart glasses-based information systems by a MDSD concept. The BPMN extension supports a precise technological representation of smart glasses functionalities into business processes and the MDSD approach allows non-technical experts the design and readjustment of smart glasses-based information systems. Thus, the superordinate research question is: *How can smart glasses-based functionalities be integrated within a BPMN extension, whose models allowing the generation of smart glasses applications with a model-driven software development approach?*

The paper is structured as follows: In the first section, we declare the relevance, motivate for a BPMN extension and explain the connection with a generating system for smart glasses-based information systems. In the second section, related work regarding smart glasses applications in the domain of process modelling is explained. Further, the objectives of the solution are outlined. Afterwards we describe the applied research method in section three. Excerpts from the results of the applied research method are presented in section four. In the next section, the implemented architecture for the MDSD approach is explained. Section six mentions discussions and limitations points that appeared during the research project and sums up the article.

2 Smart Glasses Applications related to Process Modelling and Objectives of BPMN4SGA

In the range of smart glasses applications, some have a close connection with process modelling in various forms. For instance, Metzger et al. [11] developed a smart glasses-based modelling system to model service processes during process execution. The worker uses specific speech commands and can take pictures with the smart glasses camera to create a process model. In addition, modelled processes can be used as step-by-step process guidance. Furthermore, Fellmann et al. [12] propose a process modelling recommender system to accelerate process modelling activities with smart glasses. A different approach presents Petersen et al. [13], their application allows the real-time elicitation of work processes with videos from the smart glasses user's view. The videos being used to generate an augmented reality manual that presents correct performed tasks with a green coloring of the worker's hands.

In comparison with the mentioned smart glasses applications, BPMN4SGA with an additional generating system provides step-by-step guidance based on process models like in [11], too. Though a distinction is the abstraction level, the smart glasses-based modelling system by Metzger et al. [11] uses simple workflow patterns like sequences or exclusive choices with the dedicated activities, whereas BPMN4SGA's objective is to support smart glasses functionalities within smart glasses information systems, e.g. to use the smart glasses camera or specific voice commands. Generally, BPMN4SGA allows a more detailed description of smart glasses applications. Besides a more precise description, other advantages come with the generation system and the applied MDSD approach.

This leads to several use cases that result in specific objectives of the solution. One use case is in the area where companies with no IT departments facing difficulties to adopt smart glasses from a technical point of view due to dependencies from third-party developers [5]. This circumstance ensures a rejection of new technologies such as smart glasses, although the advantages of the technology are well-known. A generating system for smart glasses-based systems can encounter technological rejection that is based on difficulties in software development. Therefore, especially small and medium-sized companies should benefit from the solution. Furthermore, smart glasses in a digitalized and mobile environment needing a fast readjustment of their information systems that can be accomplished with the proposed solution. Another tackled aspect is the reduction of investment costs and -risks into smart glasses-based systems. Assuming that one company provides the solution, economies of scale could be achieved that lead to lower and predictable costs. Particularly intangible benefits of smart glasses systems are hard to collect that makes a clear investment decision challenging [14]. BPMN4SGA and the associated generating system allow testable and fast running smart glasses systems that help to identify intangible benefits beforehand.

3 Research Method

The overall research project applies the Design Science Research Method (DSRM) by Peffers et al. [15]. This paper partly covers the design and development phase. For the actual phase, we use the method for the domain-oriented development of BPMN extensions by Braun and Schlieter [16] that extends the method for the technical development of BPMN extensions by Stroppi et al. [17] that is shown in figure 1. The domain-oriented analysis is covered in [10]. The outcome was the identification of 19 extension requirements for a BPMN extension that allows the representation of smart glasses-based processes, which can be transformed into smart glasses-based information systems with a MDSD approach.

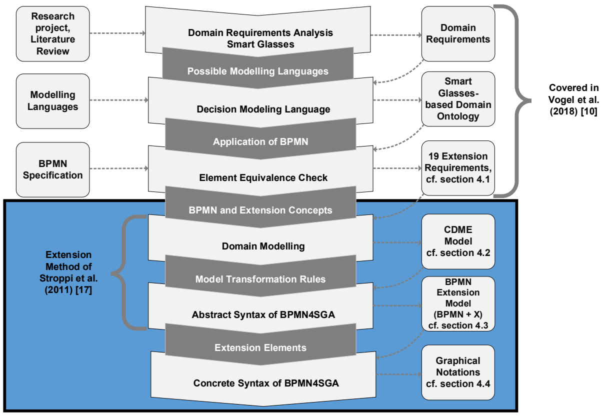

Figure 1. Applied method for the domain-oriented development of BPMN4SGA in accordance with Braun and Schlieter [16]

This paper focuses on the following steps highlighted in blue, which are overall three steps. The first step covers a domain model which is represented by a Conceptual Domain Model of the Extension (CDME) that is depicted by a UML class diagram. The CDME is the groundwork for the abstract syntax definition of the BPMN extension that is represented by a BPMN extension model (BPMN+X) and is also depicted by a UML class diagram. For the conversion of the CDME into the BPMN+X special model transformation rules are applied. Both steps are covered by the extension method of Stroppi et al. and are explained in detail in [17]. After that, the concrete graphical syntax is developed based on the BPMN+X diagram. Further design recommendations have to be covered. For instance, the shape of standard BPMN notation elements should not be altered [16].

4 Design and Development of BPMN4SGA

4.1 Extension Requirements for the BPMN Extension

Based upon insights from a one-year project phase for the identification of potential use cases in the logistic domain [9] and design guidelines from literature regarding smart glasses-based systems, we listed in an in-depth analysis technical- and software-based requirements for a dedicated BPMN extension that allows representing smart glasses-based information systems [10]. Thereby, we identified six technical-based (TB-DR) and six software-based (SB-DR) domain requirements (DR) (cf. table 1). In order to link the derived requirements with BPMN concepts, a further developed smart glasses-based domain ontology represents BPMN concepts and their linking with the developed DR. Apart from its use as a representation of knowledge, the ontology was used to steer the element equivalence check. In doing so, DR were compared with directly linked BPMN concepts. Thereby, the element equivalence check revealed 19 extension requirements where 14 requirements were not equal (NE) and 5 were conditional equail

(CE) regarding compared BPMN concepts. For all requirements occurs a specific extension, a graphical representation as an extension format is designated for smart glasses activities and their event types. Further, additional domain concepts that are not covered by standard BPMN are extended through new attributes. The design decisions were made based on the level of equality of domain to BPMN concepts and their novelty. This led only to new graphical notation elements for events and activities that will simplify learning BPMN4SGA.

Table 1. Element Equivalence Check and Extension Format for BPMN4SGA [10]

DR	Domain Concept	BPMN Concept	Equal?	Extension Format
TB-DR1	Activity and event support by the tracking concept	Activity Event	NE NE	Graphical Graphical
TB-DR2	Interactions steer the process flow, can trigger events and confirm activities.	Process flow Event Activity	NE NE NE	Attribute Graphical Graphical
TB-DR2	Manual decision through interaction inputs	XOR Gateway	CE	Attribute
TB-DR3	Visualisation of the process, brightness, and display size	Process flow Notation elements	CE	Attribute
TB-DR4	Camera to take pictures or videos	Activity	NE	Graphical
TB-DR5	Context via sensors	Event	NE	Graphical
TB-DR6	Communication	Activity Event	CE CE	Graphical Graphical
SB-DR1	Distractive visualisations	Process Flow	NE	Attribute
SB-DR2	Information provision Content type	Activity Activity	NE NE	Graphical Attribute
SB-DR3	Process improvements	Process	NE	Attribute
SB-DR4	Additional information	Data Object	CE	Attribute
SB-DR5	Object identification	Activity Event	NE NE	Graphical Graphical
SB-DR6	Communication system	Activity	NE	Attribute

4.2 Conceptual Domain Model of the Extension (CDME)

The Conceptual Domain Model of the Extension (CDME) represents the domain and BPMN concepts and their relationships (cf. figure 2). The CDME includes BPMN concepts and extension concepts that are derived from the identification of extension requirements (cf. section 4.1). Thereby, BPMN concepts are also covered in the BPMN metamodel and extension concepts, filled in grey, include concepts from the smart glasses domain [17]. The connection between standard BPMN concepts and extension concepts occurs through generalization relationships. The new graphical extension concepts have a generalization relationship to existing BPMN concepts e.g. TrackingTask has a generalization relationship to the BPMN concept task and are the new introduced notation elements (cf. section 4.4). The generalization relationships between BPMN concepts to extension concepts like GraphicalRepresentation, ProcessImprovement, SteeringProcessSetting, VisualisationSetting, ManualDecision and AdditionalInformation should add additional attributes to existing BPMN concepts. For instance, GraphicalRepresentation class attributes should be included in any task.

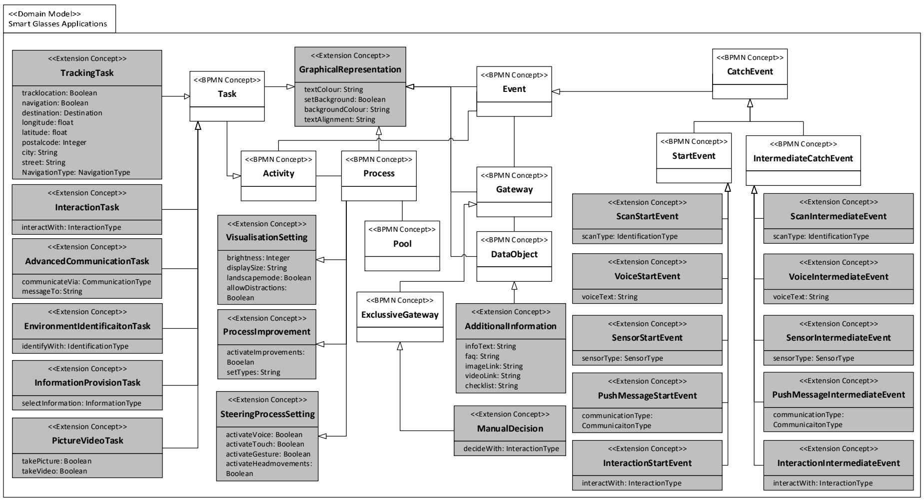

Figure 2. Conceptual Domain Model of BPMN4SGA

In summary, the CDME consists of six new tasks that have further unique attributes. They are getting complemented with distinctive graphical attributes such as text colour, background colour or alignment of the text content. Furthermore, additional information can be attached at tasks through data objects to provide the user with additional information if they request it. Events can be either catch or throw events. Thereby, catch events catching a trigger, whereas throw events throwing a result that can be caught [18]. We only covered catch events for several reasons. Firstly, the user throws events indirectly through the smart glasses functionalities that have to be caught by start or intermediate events. When the worker uses smart glasses in the work environment – he triggers different kinds of events that are also indirectly covered in the smart glasses specific tasks. Encountering this problem, we decided to avoid a doubling and we do not cover throw events in the BPMN extension. Further, with this division, a clearer understanding of the process flow is realized. Start events of the extension model can catch a trigger and starting a smart glasses-based process. This allows a technical selection of processes through e.g. identification of objects through scans, voice-based commands, sensors, interaction inputs or push messages.

Intermediate events can interrupt the process until a specific event is thrown. Additionally, intermediate events can steer the process flow when an event-based gateway is used. On the one hand, we are aware that the decision not to include throw events is debatable and have to be observed during the evaluation phase. On the other hand, it allows no confusion regarding the differences between smart glasses-based tasks and thrown events. Further comparable BPMN extensions, like uBPMN that represents ubiquitous computing processes, do only cover catch events too [19]. The manual decision class represents an exclusive decision that can be handled by the smart glasses user by interactions such as voice, touch or gestures. Further, the different kinds of process steering settings can be enabled in the process settings. Finally, process improvements can be activated to enable the functionality to retrieve feedback from the smart glasses user regarding the work process, and visual and graphical settings set the general design of the information system. Thereby, the CDME includes all listed requirements (cf. table 1) and brings the concepts in a sufficient order.

4.3 BPMN Extension Model (BPMN+X)

The evolved CDME is transformed into the BPMN extension model (BPMN+X) (cf. figure 3). BPMN+X is a specific language developed by Stroppi et al. [17] that is based on the BPMN extension mechanism [18]. Classes covered by the BPMN metamodel are in white and classes representing the extension domain are in grey. To evolve the BPMN+X model, we applied the transformation rules defined by Stroppi et al. [17]. Especially rules five and seven are used to represent a CDME generalization relationship in the BPMN+X model. Rule 5 implied that the BPMN generalization relationship remains by two BPMN concepts. In contrast, if a generalization relationship exists in the CDME with an extension concept, then the relationship is presented in the BPMN+X model through an ExtensionRelationship. According to Stroppi et al. [17], an ExtensionRelationship represents the conceptualization of extensions and their connected extensions are understood as customization of existing elements from the BPMN metamodel [17]. Further extension enumerations (Extension Enum) are represented for new attribute types e.g. SensorType.

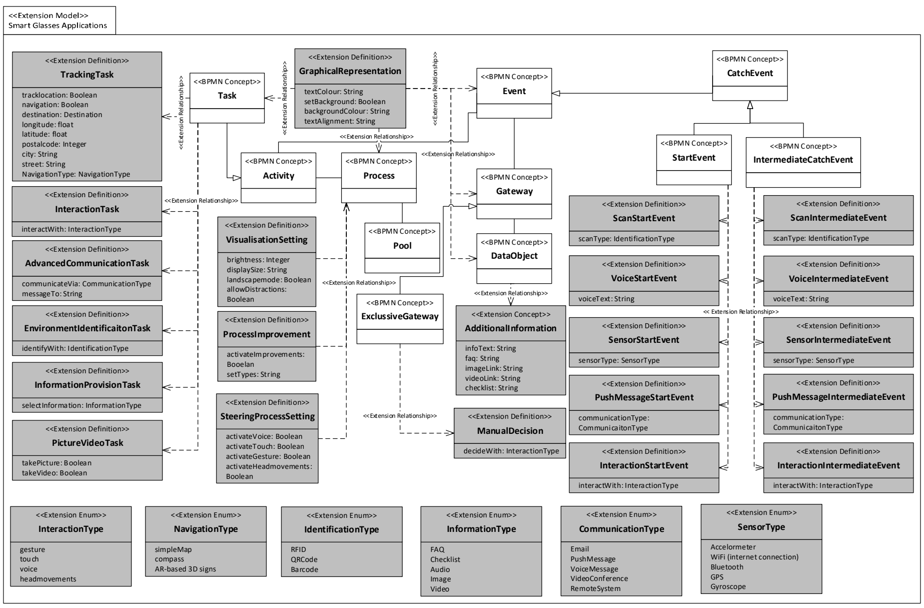

Figure 3. BPMN Extension Model (BPMN+X) of BPMN4SGA

In a next step, Stroppi et al. [17] suggest two further transformation steps. The created BPMN+X can be used to create an XML schema extension definition model that can be transferred into an XML schema document. Similar to the development in [20], we do not execute these steps, because the research project focuses primarily on the development of a DSML. To cover the extension model into a new visual BPMN extension, we propose a graphical representation of the new tasks and events (cf. section 4.4).

4.4 Graphical Representation of BPMN4SGA

The graphical representation of BPMN4SGA is represented through additional events (cf. figure 4) and activities (cf. figure 5). Further extension elements are represented through additional attributes. We include the design guidelines for DSML by Frank [21] to design the specific graphical notation elements. Because of the usage of standard BPMN shapes and just the alteration of symbols, we indirectly fulfilled all design guidelines, except design recommendations in regard to symbols. Therefore, we checked popular icon databases such as NounProject [22] to identify common icon designs for specific word phrases. Assuming that many persons use these icons to represent specific domains an approximation will increase the understandability of the graphical notation of BPMN4SGA. Based upon a selection, we adapted the icons for smart glasses applications and created scalable vector graphics (SVG).

In summary, ten start and intermediate events are added as graphical elements. Figure 4 shows the new events of the BPMN extension for smart glasses applications. The sensor (1), voice (2) and scan (3) start and intermediate events are in accordance with the uBPMN [19]. Sensor-based events (1) react to build in smart glasses sensors. For instance, the Vuzix M300 is equipped with the sensors: acceleration, Bluetooth, internet connection, GPS or gyroscope [23]. Voice-based events (2) capture voice commands and can start new smart glasses processes and can have an impact on the process flow. The same applies to scan-based events (3) regarding the identification of objects e.g. through QR- or barcodes. Start and intermediate events catch specific user-generated interaction-based events (4). These could be gestures, voices, touches on buttons or head movements. Voices are also covered by this extension element to depict combinations with other user-generated interactions. Push message-related events (5) trigger new smart glasses processes for a user or can have an impact on the process flow by using communication systems. They allow in combination with the advanced communication task the collaboration between other smart glasses users.

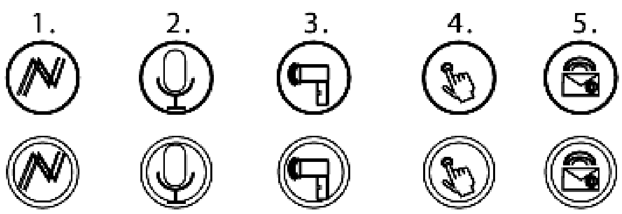

Figure 4. Events of the BPMN Extension for Smart Glasses Applications (BPMN4SGA)

The smart glasses specific tasks are depicted in figure 5. The tracking task represents the identification of the user's position. The user should be navigated to a specific location. The location can be declared as longitude and latitude attributes or as an address. The concrete navigation type can be a simple map, a compass or augmented reality-based 3D sign in the user's field of view. The interaction task is especially suitable to gain user-aware interactions. They are suitable to retrieve interactions to create checks e.g. retrieve a special interaction command to confirm the receiving of goods in the logistics domain. The environment identification task provides the declaration of the task in combination with an object identification e.g. for a package with a specific barcode some value-added-services have to be carried out. The information provision task allows the representation of information for the smart glasses user that is for instance necessary to execute complex tasks. The different kinds of information types could be an image, video, audio, checklist or FAQ. A picture or video-based task enables the recording of the environment e.g. pictures can be taken to collect defections in the process receipt of goods. As mentioned before the advanced communication task represents the communication possibilities of smart glasses for instance via e-mail, audio messages, video conferences or remote systems.

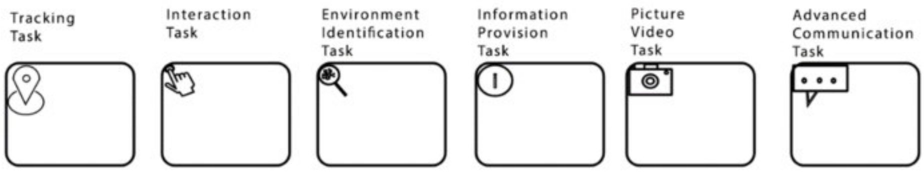

Figure 5. Activities of the BPMN Extension for Smart Glasses Applications (BPMN4SGA)

4.5 Implementation into a suitable Modelling Tool

To realize an application of the BPMN extension for smart glasses applications, we implemented the extension model into a modelling tool and extended the BPMN metamodel [24]. Business users should document smart glasses-based processes effortlessly without the need to install a new program. Therefore we favour web-based modelling tools over desktop-based modelling tools that need additional software modules e.g. language packets such as Java. To identify a suitable BPMN modelling tool that fulfills our requirements, we conducted a market analysis. Hence we do not use established desktop-based meta modelling platforms such as ADOxx [25] or MetaEdit+ [26]. Instead, we build upon BPMN-js a rendering toolkit for BPMN 2.0 that is based on JavaScript [24]. It allows a comfortable integration into web applications and is further extendable. The project is supported by the company Camunda and an active community. Further example projects simplify the customization and extension of the web-based modelling tool [27].

In figure 6 a screenshot from the adjusted modelling tool is presented. The left side shows a palette with the standard BPMN notation elements. The right side provides options to edit the element specific attributes. In the middle is the drawing surface. The domain-specific notation elements can be reached through a click on a standard BPMN element e.g. task or event. After that, a context menu opens and through a further click on the screw-wrench icon the BPMN element can be changed into a more specific form e.g. a start event into a voice start event. We represented BPMN4SGA elements in blue for precise identification. We do not include the concrete syntax in the palette (left side) to keep the clarity of the task/event notation elements and to retain the same user interface as for standard BPMN elements.

The icons are implemented into JavaScript code as SVG graphics. For a correct representation of the BPMN extension, the BPMN metamodel was enhanced with the domain-specific elements. This step ensured a correct transformation of the formal model into an XML document. Therefore, we edit the JSON sources that are representing the BPMN 2.0 metamodel. This is realized in bpmn.io through the bpmn-moddle library [28].

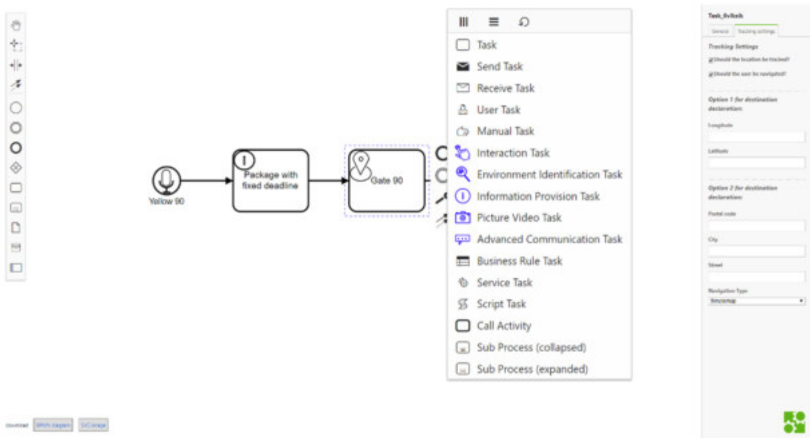

Figure 6. Demonstration of the BPMN4SGA in bpmn.io (screenshot)

5 Architectural Integration of the Model-driven Software Development Concept

MDSD aims to "automatically transforming formal models into executable code" [29]. This can be realized through the generation of code or interpretation. The generation of code is widely-used regarding structural parts, in contrast interpreters are used for behavior characteristics of a system [29]. Behavioral-oriented information system models are representing process and functional characteristics [30]. Because of, BPMN4SGA models focus on the representation of behavioral-oriented information systems, i.e. smart glasses applications, we favour the interpretation approach. Further reasons are the two differences between generators and interpreters that makes interpreters more suitable for our research project. The first reason is the time of analysis. The analysis of generators occurs during build time, whereas interpreter can parse models at runtime. Therefore, changes in BPMN4SGA models have a direct effect on smart glasses applications through direct interpretation. Second, the mode of execution differs. Interpreters execute directly different kind of code based on the read model. In contrast, generators link code fragments that are compiled in a next step [29]. Because the solution tries to contribute non-technical experts with a tool to readjust smart glasses applications directly (cf. section 2), we decided to implement an interpreter that binds code during runtime and does not need a further compilation step.

The architecture of our system is in accordance with Metzger et al. [11] and is shown in figure 7. The domain expert can model smart glasses-based processes through the web-modelling tool that includes the BPMN4SGA notation elements and attributes. The generated XML file can be uploaded through a web interface that saves metadata e.g. process id, process name and creation date from the process in the database. The web and REST interfaces for the platform are implemented with the PHP framework Symfony [31]. Further, an authentication system was implemented with Symfony that enables user-specific storing and client-side loading of BPMN4SGA models. Additionally, an Android PacKage (APK) was programmed and installed on the smart glasses. We implemented the generation system for the Vuzix M100. Thereby, the installed mobile app represents the generation system. The app includes the interpreter, which consists of an individual programmed parser that is implemented into the application. In doing so, the parser is written in Java and it is adjusted to our needs to read BPMN4SGA XML files. Based on the interpretations specific Android activities are called. We decided to download the XML files completely and execute the parsing steps locally on the smart glasses device to enable offline functionality and to minimize the number of requests. During the process execution, the generation system loads activity specific content through REST request e.g. pictures. Also, the system sends data to the server e.g. messages or log files and has further connections to additional systems e.g. to handle speech recognition.

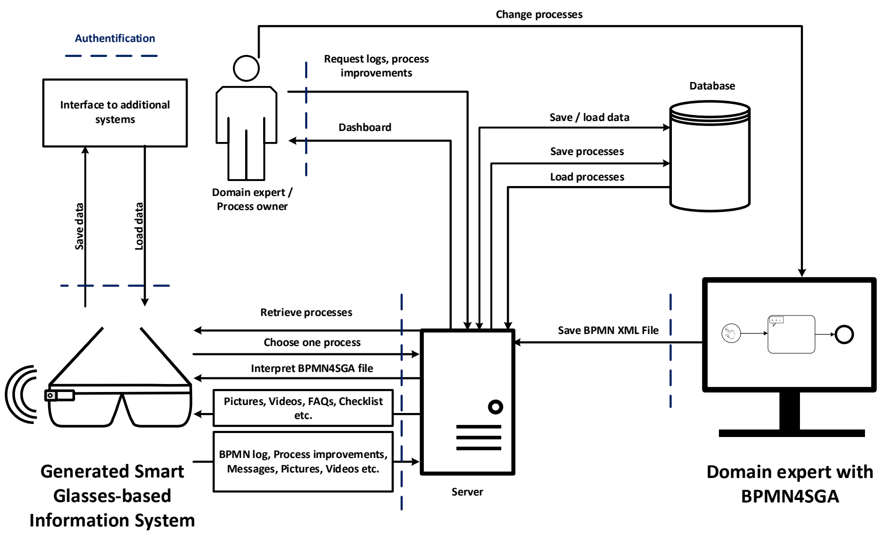

Figure 7. Architecture to generate Smart Glasses-based Information Systems with BPMN4SGA in accordance with Metzger et al. [11]

In close relation to figure 7 shows figure 8 the concrete MDSD approach: Thereby, a modelled process based on BPMN4SGA (1) will be interpreted by a parser (2) and calls specific views and methods. The triggered functions will be reflected in the user's field of view (3). The smart glasses use case "receipt of goods" is a common use case in the logistics domain [9] and is used to demonstrate the solution. The logistician starts the process with a specific speech command that can be specified in further attributes. Next, the worker has to scan the qrcode of the product to identify it. After that, he should give feedback regarding the condition of the goods. For instance, if the product is damaged, he creates a picture and does a damage classification. Finally, after the completion of the process tasks, the data is accessible and new processes can be triggered or selected by the smart glasses user [32].

The presented approach ensures high flexible information systems that can be changed by business experts without programming skills due to a direct transformation of the BPMN4SGA model into the desired information system. Thereby smart glasses-based information systems can in particular facilitate service processes [4].

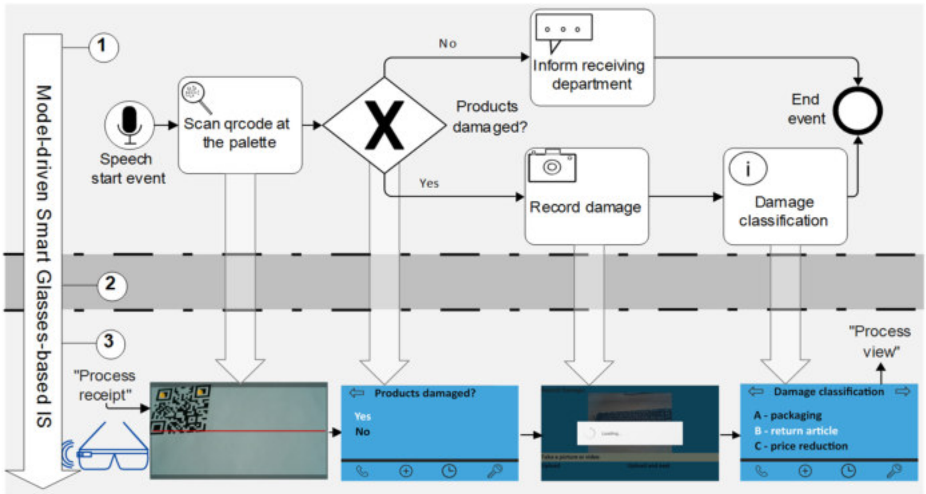

Figure 8. Demonstrating Smart Glasses-based Information System based on BPMN4SGA [32]

6 Discussion and Conclusion

Discussion. During the research project, some points for discussion arose. Additionally, we noticed limitations that are mentioned here. We are aware that the development and design of the BPMN4SGA are mostly based on the insights from the research project and our understanding of smart glasses applications. While we included requirements based on additional smart glasses literature, an overall requirement elicitation was not fully covered. In our point of view, the developed BPMN extension offers the possibilities to discuss with external experts and evaluate the implemented notation elements and attributes. Additional adjustments can be implemented in a further step. Moreover, we do not claim a full applicability of the developed generation system for every smart glasses-based use case – to cover most of them will be a further challenge in the future. Furthermore, the developed BPMN4SGA is mainly based on smart glasses functionalities and it is questionable if a broader software-based focus has to be included. For instance, the communication system was yet not implemented due to the complexity and further relationships with additional interfaces. We developed a closed ecosystem that is mainly focused on visualization purposes and does not reflect business boundaries e.g. to enterprise resource planning systems. So far the representation of businesses interfaces is underrepresented. Additionally, an evaluation with practice experts and field test can deliver valuable insights and are the next research steps.

Conclusion. This research project presents BPMN4SGA, the BPMN extension for smart glasses applications. The paper describes the technical development and implementation of the abstract and concrete syntax, and the implementation steps into a web-based modelling tool. The modelling tool is added into an architecture to realize smart glasses applications based on formal models through a MDSD approach. The implemented artifact enables the direct transformation of BPMN4SGA models into smart glasses applications. As technical DSML are still underrepresented in the range of various DSML, the research project contributes to the development and usage of these. Further, the research addresses a relevant topic to enhance the adoption of new technologies in businesses and delivers a solution to implement business specific smart glasses applications. The implemented solution enables non-technical domain experts to create and modify smart glasses applications for business use cases without the need to write one single code line.

7 Acknowledgements

This research is partly funded by the German Federal Ministry of Education and Research (BMBF) in the scope of the research project Glasshouse (ID: 02K14A090) and the Graduiertenkolleg va-eva: Vertrauen und Akzeptanz in erweiterten und virtuellen Arbeitswelten at the University of Osnabrueck. We would like to thank them for their support.

References

1. Klinker, K., Berkemeier, L., Zobel, B., Wüller, H., Huck-, V., Wiesche, M., Remmers, H., Thomas, O., Krcmar, H.: Structure for innovations : A use case taxonomy for smart glasses in service processes. In: Multikonferenz Wirtschaftsinformatik 2018, pp. 1599–1610. Lüneburg (2018)
2. Reif, R., Günthner, W.A., Schwerdtfeger, B., Klinker, G.: Pick-by-Vision comes on Age : Evaluation of an Augmented Reality supported Picking System in a real Storage Environment. In: Proccedings of the 6th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, pp. 23–31. Pretoria (2009)
3. Streibl, M., Brandl, P.: Einsatz eines Live Video Remote Systems in der Industrie. In: Wyers, B. and Dittmar, A. (eds.) Mensch und Computer 2016, pp. 1–6. Gesellschaft für Informatik e.V., Aachen (2016)

4. Niemöller, C., Metzger, D., Thomas, O.: Design and Evaluation of a Smart-Glasses-based Service Support System. In: 13th International Conference on Wirtschaftsinformatik (WI), pp. 106–120. St. Gallen (2017)
5. Hobert, S., Schumann, M.: Enabling the Adoption of Wearable Computers in Enterprises – Results of Analyzing Influencing Factors and Challenges in the Industrial Sector. In: Proceedings of the 50th Hawaii International Conference on System Sciences, pp. 4276–4285. (2017)
6. Uchiyama, T., Tanuma, K., Fukuda, Y., Nakanishi, M.: GUI Design Solution for a Monocular, See-through Head-Mounted Display Based on Users' Eye Movement Characteristics. In: Shumaker, R. (ed.) Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments. VAMR 2013, LNCS, vol. 8021, pp. 231–240. Springer, Berlin Heidelberg (2013)
7. Clark, T., Barn, B.S.: Domain Engineering for Software Tools. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., and Bettin, J. (eds.) Domain Engineering: Product Lines, Languages, and Conceptual Models, pp. 187–209. Springer, Berlin Heidelberg (2013)
8. Trompeter, J., Pietrek, G., Beltran, J.C.F., Holzer, B., Kamann, T., Kloss, M., Mork, S.A., Niehues, B., Thoms, K.: Modellgetriebene Softwareentwicklung - MDA und MDSD in der Praxis. entwickler.press, Frankfurt am Main (2007)
9. Niemöller, C., Zobel, B., Berkemeier, L., Metzger, D., Werning, S., Adelmeyer, T., Ickerott, I., Thomas, O.: Sind Smart Glasses die Zukunft der Digitalisierung von Arbeitsprozessen? Explorative Fallstudien zukünftiger Einsatzszenarien in der Logistik. In: Leimeister, J.M. and Brenner, W. (eds.) 13. International Conference on Wirtschaftsinformatik (WI), pp. 410–424. St.Gallen (2017)
10. Vogel, J., Zobel, B., Jannaber, S., Thomas, O.: BPMN4SGA: A BPMN Extension for Smart Glasses Applications to enable Process Visualisations. In: Czarnecki, C., Brockmann, C., Sultanow, E., Koschmider, A., and Selzer, A. (eds.) Workshops der Informatik 2018 - Architekturen, Prozesse, Sicherheit und Nachhaltigkeit, pp. 259–273. Köllen Druck+Verlag GmbH, Bonn (2018)
11. Metzger, D., Niemöller, C., Berkemeier, L., Brenning, L., Thomas, O.: Vom Techniker zum Modellierer – Konzeption und Entwicklung eines Smart Glasses Systems zur Laufzeitmodellierung von Dienstleistungsprozessen. In: Smart Service Engineering, pp. 193–213. Springer Fachmedien Wiesbaden, Wiesbaden (2017)
12. Fellmann, M., Metzger, D., Jannaber, S., Zarvic, N., Thomas, O.: Process Modeling Recommender Systems. Bus. Inf. Syst. Eng. 60, 21–38 (2018)
13. Petersen, N., Pagani, A., Stricker, D.: Real-time modeling and tracking manual workflows from first-person vision. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 117–124. IEEE (2013).
14. Oesterreich, T.D., Teuteberg, F.: Why one big picture is worth a thousand numbers: measuring intangible benefits of investments in augmented reality based assistive technology using utility effect chains and system dynamics. Inf. Syst. E-bus. Manag. 16, 407–441 (2018)
15. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24, 45–77 (2008)
16. Braun, R., Schlieter, H.: Requirements-based development of BPMN extensions: The case of clinical pathways. In: 2014 IEEE 1st International Workshop on the Interrelations between Requirements Engineering and Business Process Management (REBPM), pp. 39–44. IEEE, Karlskrona, Sweden (2014)

17. Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: Extending BPMN 2.0: Method and Tool Support. In: Dijkman, R., Hofstetter, J., and Koehler, J. (eds.) Business Process Model and Notation. BPMN 2011, LNBIP, vol. 95, pp. 59–73. Springer, Berlin Heidelberg (2011)
18. Object Management Group (OMG): Business Process Model and Notation (BPMN), http://www.omg.org/spec/BPMN/2.0.2/PDF/ (Accessed: 20.12.2018).
19. Yousfi, A., Bauer, C., Saidi, R., Dey, A.K.: UBPMN: A BPMN extension for modeling ubiquitous business processes. Inf. Softw. Technol. 74, 55–68 (2016)
20. Braun, R., Schlieter, H., Burwitz, M., Esswein, W.: BPMN4CP: Design and Implementation of a BPMN Extension for Clinical Pathways. In: 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 9–16. IEEE (2014)
21. Frank, U.: Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., and Bettin, J. (eds.) Domain Engineering: Product Lines, Languages, and Conceptual Models, pp. 133–157. Springer, Berlin Heidelberg (2013)
22. NounProject, https://thenounproject.com/ (Accessed: 20.12.2018).
23. Vuzix M300 Technical Specifications, https://www.vuzix.com/products/m300-smart-glasses (Accessed: 20.12.2018).
24. Bpmn.io: BPMN 2.0 rendering toolkit and web modeler., https://bpmn.io/toolkit/bpmn-js/ (Accessed: 20.12.2018)
25. Fill, H., Karagiannis, D.: On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. Enterp. Model. Inf. Syst. Archit. 8, 4–25 (2013)
26. Kelly, S., Lyytinen, K., Rossi, M., Tolvanen, J.P.: MetaEdit+ at the Age of 20. In: Bubenko, J., Krogstie, J., Pastor, O., Pernici, B., Rolland, C., and Sølvberg, A. (eds.) Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE, pp. 131–137. Springer, Berlin Heidelberg (2013)
27. BPMN-JS-Examples, https://github.com/bpmn-io/bpmn-js-examples (Accessed: 20.12.2018)
28. BPMN-MODDLE, https://github.com/bpmn-io/bpmn-moddle (Accessed: 20.12.2018)
29. Stahl, T., Völter, M.: Model-Driven Software Development. John Wiley & Sons, Southern Gate, Chichester (2006)
30. Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Gabler Verlag (1998).
31. Symfony The Book, https://symfony.com/pdf/Symfony_book_2.8.pdf (Accessed: 20.12.2018)
32. Vogel, J., Jannaber, S., Zobel, B., Thomas, O.: Design and Development of a Process Modelling Environment for Business Process Utilization within Smart Glasses. In: Fellmann, M. and Sandkuhl, K. (eds.) 9th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA), pp. 85–89. Rostock (2018)