BPM to Go: Supporting Business Processes in a Mobile and Sensing World

Rüdiger Pryss, Manfred Reichert, Alexander Bachmeier, Johann Albach
Ulm University, Germany

Abstract

The growing maturity of smart mobile devices has fostered their prevalence in a multitude of business areas. As a consequence, business process management (BPM) technologies need to be enhanced with sophisticated and configurable mobile task support. Along characteristic use cases from different application domains (e.g., healthcare and logistics), this chapter will give insights into the challenges, concepts and technologies relevant for integrating mobile task support with business processes. Amongst others, we will show how mobile task support can be enhanced with location-based data, sensor integration, and mobile task configuration support. The latter is based on a 3D model for configuring mobile tasks on smart mobile devices.

Introduction

In the computer industry, the emergence of smart mobile devices has opened up new and exciting perspectives. Nowadays, we carry computers in our pockets that wouldn't have been out of place on a supercomputer ranking of the 1990s. The ever increasing ubiquity of these smart mobile devices and the dynamic nature of Business Process Management (BPM) technology demand new concepts and systems that may execute tasks on these smart mobile devices. For example, to assist a physician during her daily work through sophisticated mobile task and process support may ease her work significantly [19]. However, the smooth integration of smart mobile devices into the BPM landscape has revealed a multitude of specific challenges. One way to properly meet these challenges is through a sophisticated mobile task execution framework that can be smoothly integrated with existing BPM environments. A specific challenge in respect to such an integration concerns the proper configuration of mobile tasks. In this context, the chapter proposes an approach that enables the domain expert (i.e., the end user) to configure mobile tasks by a 3D process model, which is displayed in an augmented reality view on his smart mobile device.

Motivation

While smart mobile devices have evolved rapidly [15], they still show limitations that need to be considered when integrating them with BPM systems [17]. Current limitations include, amongst others, limited battery power, instantaneous shutdowns, data inconsistency, and unreliable network connectivity. As a consequence, one cannot simply migrate the execution of complete processes and their tasks onto mobile devices without coping with these issues.

The last years have shown a divergence towards smart mobile devices, with only a small number of tasks not ported to a smartphone or tablet in one way or another. Accordingly, users more and more expect from their smart mobile devices to assist them in fulfilling almost every task they have processed on their stationary PC. BPM technology should pick up this trend, not only due to emerging customer demands, but also because it opens up new and promising opportunities.

In general, BPM serves as an approach to analyze, model, automate, monitor, and optimize business processes in a variety of application domains [11]. Particularly, BPM improves business IT alignment and serves as a glue between information technology on one hand and the various business stakeholders (e.g., staff, customers, and business partners) on the other [8]. In this context, *BPM to Go* represents our vision of coupling smart mobile devices with BPM technology in order to enable flexible process and task assistance of mobile (knowledge) workers. Amongst others, the following areas need to be touched to make this vision a reality:

1. Mobile task execution & configuration
    a. Smooth integration of mobile tasks in existing BPM environments [19]
    b. Tackling challenges of a mobile execution context (e.g., instantaneous shutdowns) [17].
2. Distributed mobile processes (e.g. cross-departmental as well as cross- organizational scenarios) [7].
3. Collaborative mobile processes (e.g., mobile checklists for collaborating knowledge workers) [16].
4. Mobile office in combination with BPM (e.g., using personalized smart mobile processes) [4].
5. Cyberphysical systems and Internet of Things in combination with BPM [5].

Contribution

This chapter focuses on mobile task assistance in general and mobile task configuration in particular.

The primary goal of mobile task assistance is to enable end users to work on their business tasks using smart mobile devices instead of stationary PCs. Smart mobile devices not only allow performing these tasks almost everywhere, but also enable measurements on the spot. Furthermore, the interaction of mobile workers with smart mobile devices fosters process and task flexibility as well as a faster completion of business processes. Figure 1: Mobile task approaches depicts three approaches for integrating smart mobile devices with BPM. The support of these mobile scenarios reveals challenging issues that need to be properly addressed, e.g., in respect to process exception handling [13] and task failure management [17].

Beyond the collaborative and mobile aspect, the connectedness of smart mobile devices and their sensors allows for the integration of additional context information and parameters with business process execution. For example, a location parameter may be used to store information about the location a particular task has been performed [13].

The usage of smart mobile devices needs not be limited to the execution of single tasks. As demonstrated later in this chapter, the ability to perform task configurations on the *Go* opens up new opportunities. Our idea for mobile task configuration revolves around the concept of a 3-dimensional, augmented reality view of processes, which is generated using a common Android smart mobile device. Based on this view, domain experts and users shall be enabled to configure and optimize mobile tasks on the *Go*.

To the best of our knowledge, there are no comparable approaches dealing with mobile task execution and configuration. In particular, the configuration of mobile tasks directly on a smart mobile device has not been properly addressed so far. However, there exist approaches that characterize a *mobile context* through a set of parameters [9]. Usually, only few contextual parameters are considered and these mainly focus on process characteristics [12]; e.g., to be able to decide what shall happen with a task if a device is unavailable. In turn, no parameters are maintained to characterize the different kinds of failures (e.g., low battery power).

The remainder of this chapter is structured as follows: First, we discuss lessons learned in the context of real-world scenarios and relate them to the "*BPM to Go*" vision. Second, we describe relevant aspects of mobile task execution along an example from the healthcare domain. Finally, we present an augmented reality engine being able to display a 3D representation of processes on a smart mobile device to assist users in configuring mobile tasks.

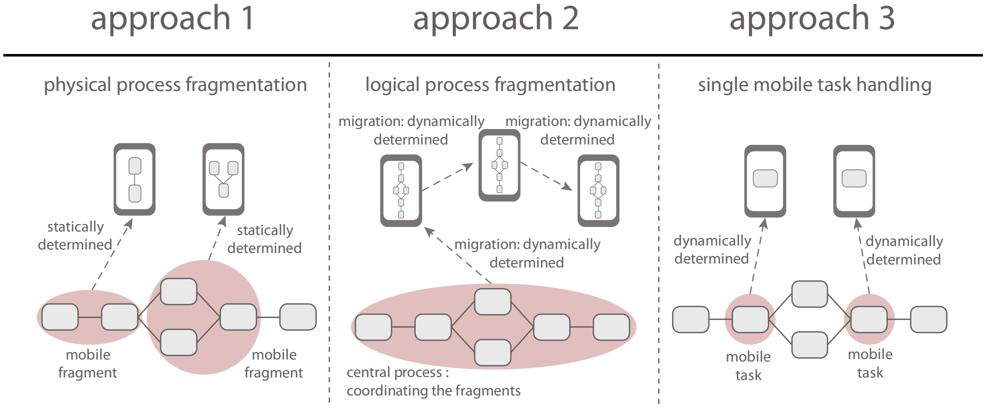

FIGURE 1: MOBILE TASK APPROACHES

Background Information and Considered Scenarios

In the context of our research on mobile processes [6, 17], a number of approaches have been designed that provide robust ways to deal with the challenges of mobile task support. To further evolve these approaches (cf. Figure 1), both their feasibility and their limitations were investigated in a number of real-world scenarios. The three considered approaches differ in respect to the part of the process known to a device, the information exchanged, and the way the tasks are synchronized between the devices.

To confirm the importance of mobile task support, different application scenarios from the healthcare, automotive and psychology domains were investigated. As will be shown in the course of this chapter, the benefits of enhancing scenarios with smart mobile devices are manifold. For example, through the use of smart mobile devices, the efficiency of how tasks are performed can be improved. As another benefit, data can be collected and recorded at the right time and place [18]. Furthermore, the occurrence of erroneous data can be decreased since data can be processed and checked on the device actually running the data collection procedure.

While some of the discussed challenges are known to mobile app developers in general, the execution of mobile tasks in the context of business processes particularly raises demands in respect to robustness and usability. We present four categories of challenges to be addressed in this context.

Challenge 1: Process-related

Regarding process-related challenges, the focus should be on data consistency. In general, network connectivity will be unstable, compared to a physically immobile system like a workstation or server. During any point in the execution or configuration of a process, network connection problems or power issues might arise. Consequently, the challenge is to keep the data of executed process consistent even when unexpected exceptions occur.

Challenge 2: User Behavior

An issue not specific to mobile environments, but of higher relevance compared to stationary devices, concerns the user as a source of irregularities. For example, a user might inadvertently shut down his device or put an application currently executing a task into a sleeping state. Since the multi-tasking capabilities of smart mobile devices are not yet up to the standards known from other systems, the application will have to respond in a safe way to, for example, an unexpected shutdown of the device. Note that this is particularly important in the context of Challenge 1 (i.e., data consistency) as well.

Challenge 3: Mobile Context

The mobile space opens up new opportunities regarding process execution. Specifically, the location and time context may be utilized to foster process and task support. In general, the location of smart mobile devices can be determined within a reasonable margin of error using sensors like GPS. In turn, this data provides a contextual factor of the process, which may be utilized to restrict task execution to those devices located near the work place this task shall be performed. Note that this might decrease the distance to be covered by an actor in the process when performing individual tasks.

Challenge 4: Sensors

Nowadays, most smart mobile devices are equipped with a plethora of sensors. This includes sensors to locate a device using a satellite positioning system, cameras and microphone. Other sensors available are heartbeat sensors, thermometers, or blood sugar sensors. Smart mobile devices equipped with them may be used to provide physical data during process and task execution, mitigating the need to capture this data by using specialized devices or – even worse – requiring manual user input.

The particularities of smart mobile devices should be considered as specific challenges when targeting at a mobile task execution. However, the sensors of these mobile devices are also able to provide valuable parameters in the context of process execution. When dealing with real-world scenarios, several of these parameters could be used as an integral part of the processes, proofing their validity and importance in a business process context. When investigating the real-world scenarios, we identified a large number of parameters that can be related to the four categories mentioned above. To give an impression, for each considered real-world scenario,

Table 1: parameter validty in real-world scenarios shows parameters of the four categories that turned out to be relevant.

Scenario	Data Consistency	Shutd owns	Location	Camera
HEALTHCARE (WARD ROUNDS)	✓	✓	✓	✓
AVIATION (AIRLINE CATERING)	✓	✓	✓	✓
LOGISTICS (WAREHOUSING)	✓	✓	✓	✓
AUTOMOTIVE (PRODUCTION)	✓	✗	✓	✓
PSYCHOLOGY (QUESTIONNAIRES)	✓	✗	✓	✗

TABLE 1: PARAMETER VALIDTY IN REAL-WORLD SCENARIOS

Mobile Task Execution and Configuration Support

BPM not only deals with the modeling, configuration and execution of business processes, but also with their monitoring and evolution. The presented work deals with task configuration support, specifically the implementation of this support on smart mobile devices. We have designed an example of a process demonstrating how smart mobile devices and features enabling mobile configuration support may serve to enhance scenarios.

We consider a scenario involving a nursing home and a patient with dementia. A nurse performs her scheduled rounds to check the status of all patients she is responsible for. To assist her in accomplishing this procedure, she is equipped with a smart mobile device that is linked with a BPM system. Thereby, the procedure around a single ward round shall be implemented as a process in the BPM system and its corresponding tasks be executed as mobile tasks on her smart mobile device. The simplified scenario referring to a particular patient is depicted in Figure 2 (in terms of the BPMN notation). The latter also shows how data is exchanged between the smart mobile devices of the actors involved in the process.

After finishing the care of a patient, the nurse shall visit the patient scheduled next, as displayed on her smart mobile device. As part of the process, the device can display the tasks that need to be performed for a particular patient. In particular, we consider patients suffering from dementia and – most of them – from diabetes as well. To reflect the latter one of the tasks involves a routine check of the patient's blood sugar level. When the nurse enters the room of a specific patient, an automated check is performed as to the whereabouts of the patient. To facilitate this check, every patient has been equipped with a small tracking device that permits indoor and outdoor localization, thus decreasing the possibility of a disoriented patient that might wonder off and get lost outside the area of the nursing home.

Assume that the smart mobile device of the nurse queries the patient's tracking device and determines that the patient is currently not available in her room. The task execution engine on the mobile phone is further able to indicate to the nurse that the respective patient is currently in the cafeteria. The nurse may now choose to send a notification to the patient's device, informing her that she is needed in her room. The nurse declines the request and decides to walk to the cafeteria and take the patient with her back to the room in order to be able to administer the required tests.

The nurse administers a blood sugar test on the patient using a computerized measurement strip. The blood sugar level is then automatically stored in the electronic patient record. At the same time, the blood sugar value is compared with thresholds set by a doctor. Assume that the system discovers that the levels are elevated, but not high enough to warrant immediate action. As defined by the parameters of the process, the doctor will be notified about the blood sugar levels and an appointment be automatically added to calendar of the doctor.

The nurse finishes the test and proceeds with the next patient (i.e., the next process will be started and executed). Without any interaction of the nurse, an event was created by the device after the test strip had been used as the number of test strips has reached a level below the threshold set. As part of this event, an automatic request is sent to the person managing the devices that the supply of test strips for this device needs to be replenished.

A logistics expert is notified about the situation and proceeds with the nursing home's storage facility to gather the required test strips (i.e., the device may be restocked at the next opportunity). He is also equipped with a smart mobile device that includes augmented reality features. Once notified about the task, the device determines the location of its owner and can, if requested, show the location of these test strips in the warehouse and navigate him to the aisle in question.

Once the logistics expert reaches the location of the test strips, the camera of his smart mobile device is able to scan the barcode in the field of view of the camera. Once the camera recognizes the barcode of the needed package of test strips, the package is marked on the augmented reality view. This feature serves to minimize the possibility that a wrong set of test strips is taken from the warehouse. The expert takes the visually marked package of test strips and delivers them to the desk, where the blood sugar testing device can be restocked in the near future.

In summary, the scenario makes use of the following parameters:

- *Mobile Context*: Location of patient, nurse, logistics expert
- *Mobile Context*: Date and Time
- *Mobile Context & Process Related*: Device crashes
- *Sensors*: Blood sugar level
- *Sensors*: Camera (product barcodes)
- *User Behavior*: Instantaneous shutdowns

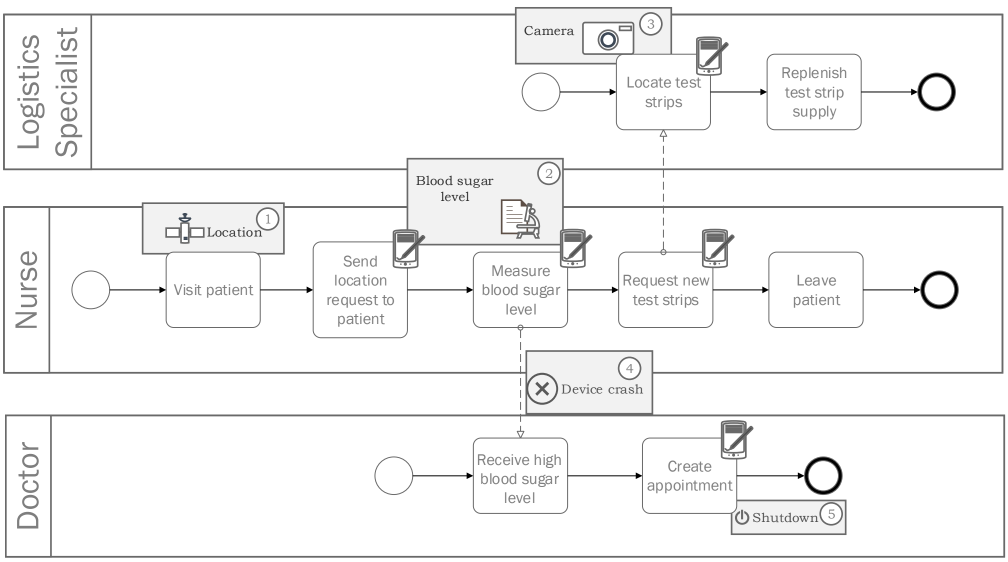

FIGURE 2: NURSING HOME SCENARIO

REGARDING THE PARAMETERS AND CHALLENGES ADVOCATED IN THE PREVIOUS SECTION,

Figure 2: Nursing home Scenario shows specific parameters (numbered rectangles) that may be applied to this scenario. We briefly summarize their significance as follows:

(1) *(Mobile Context)* The location of the nurse and the patient needs to be determined as required by the possible situations described for the scenario (e.g., patient is not in her room).

(2) *(Sensors)* Blood sugar levels are determined using a smart mobile device.

(3) *(Mobile Context & Process Related)* During the transfer of the blood sugar levels, the smart mobile device of the nurse might crash. Regarding this scenario, data consistency is crucial as incomplete or erroneous data might have severe consequences on the patient's health. Once the device is safely restarted, therefore, the data must be completely transmitted to the BPM backend.

(4) *(Sensors)* The logistics actor uses the camera of a smart mobile device to correctly identify the correct test strips.
(5) *(User Behavior)* Assume that, when using the smart mobile device, the doctor receives the required data to make an appointment, the doctor shuts down his device. In such a case, exception handling techniques must be applied to determine whether the appointment has been created correctly once the device becomes available again, or another doctor must be notified about the situation.

As the scenario illustrates, the use of smart mobile devices provides new possibilities in terms of available process parameters. This scenario only serves to illuminate the task of executing processes on smart mobile devices.

In general, in many business processes scenarios various difficulties and exceptions must be tackled at the time a process leaves the planning stage and is implemented for real-world execution. In this context, particularly for mobile scenarios, the sheer number of parameters and execution anomalies are often not foreseeable during design time. By equipping users with the possibility to view and modify their specific tasks on the smart mobile devices, the quality of processes can be significantly improved. In this context, the ability to modify, remove or add parameters becomes an issue. In particular, we propose that the parameterization may improve overall process execution in case the parameter operations (i.e., add, remove, and modify) may be applied at the place a task will be executed. Such a configuration scenario constitutes mobile task configuration on the *Go*.

As we envision mobile task configuration on the *Go*, a scenario comprising the following three steps can be realized:

(1) A user is working on a task and wants to modify one of the parameters since she discovered settings which are not properly configured. Therefore, she uses her smart mobile device to identify the task corresponding to the process she is working on, using the current location and known parameters. Based on this information, a model is created which provides information about the process and the mobile task the user is working on as well as configured parameters.
(2) Using the model, the user may modify, remove or add parameters.
(3) Once the parameters are changed, the user may view the modified task in the context of the process. Following this, the parameters can be discarded or saved. In the letter case, all new process instances based on the modifications will be executed.

3D Mobile Task Configuration Support

Our implementation of this vision is based on a 3D augmented reality engine that can display processes and parameters and that is able to offer the ability to modify these processes directly on the smart mobile device.

Currently, we provide two ways to determine the tasks to be modified. First, for tasks which are currently executed, we use the parameters and the current location of a user. Second, for tasks which are not executed, the smart mobile device uses a marker approach (1) to recognize a task. At the time, a task is known to the smart mobile device for mobile configuration, it can use its internal or server side storage to fetch the data for this task and display this task on the mobile screen (2). Using gestures, the user is now able to view the task and its process based on the 3D augmented reality engine (3).

This section focuses on our implementation of a 3D augmented reality engine that can be used to display a process using a marker or running process information to determine the process that needs to be displayed. We provide an in depth explanation of the architecture and the underlying computational model. We show how our engine can display processes and tasks using a system of markers as task identifiers. Additionally, we present the way users can modify the representation of the model, including the possibilities of zooming into specific sections of the model. Furthermore, the task configuration view is shown. Finally, we discuss what aspects we revealed when using our approach in practice.

Architecture

The architecture of the prototype is shown in Figure 3. It can be divided into three parts: (1) an augmented reality framework, (2) a graphics engine and a (3) process parser. The AR-component allows us to recognize different markers (e.g., QR-Codes) using the smart mobile device's camera system. We restrict our explanations to this variant of task detection due to the lack of space. As the next step, after the marker is recognized, a process is loaded, parsed and computed. After this procedure, it can be visualized by the graphical component. Each instance of those components operates in its own thread (one for the AR-component, one for every parsed process, and one for the renderer). In practice, this approach became necessary to deal with user demands regarding the interaction response quality. Usually, after the process is loaded and parsed, only up to two threads are running. Consequently, there are still enough capabilities for fluid interactions, considering that many contemporary smart mobile devices make use of a quad core CPU architecture.

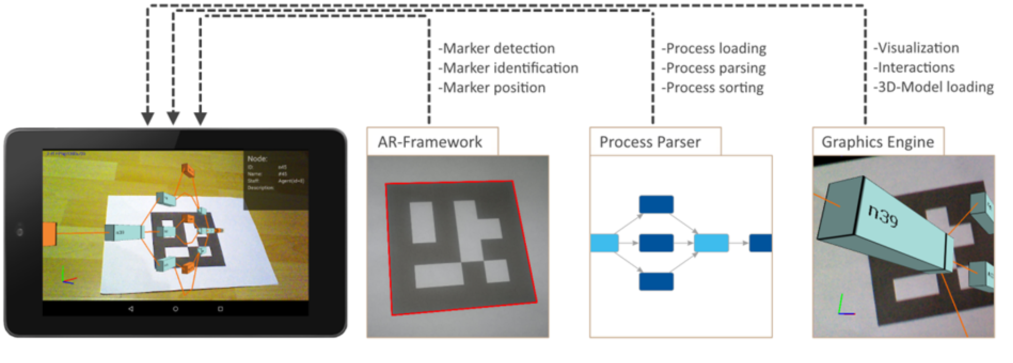

FIGURE 3: ARCHITECTURAL MODEL

Computational Model

Before a user may interact with a graphical representation of a process, it is necessary to create separate objects, which - once linked together - will form a 3D representation of a process graph. To simplify this task, objects can be produced using a variety of 3D modeling tools. The current prototype is able to load the exported *.OBJ model format (cf. Figure 4), which stores a model's geometries and its materials in two plaintext files along with the needed textures. Those files are loaded in the prototype's own parser and represent single nodes of a process graph. The second input information are XML files that contain the BPM data (we currently read ADEPT [2] process models), comprised of nodes with several attributes and edges controlling the flow. The parser for such graphs creates single objects out of the BPM's components, fills them with their attributes, computes them properly to minimize crossovers in edges and notifies the renderer that new graph data exists. Once the renderer gets informed, the loading of 3D models that correspond to the process graph's nodes is initialized and edges between them are established using lines. Once a previously linked marker is visible to the camera, the process graph can be drawn onto the screen. Thereby, a mapping between graph data and its visual representation exists. This mapping is used to get additional information about visible objects, e.g., on touch events, or vice versa.

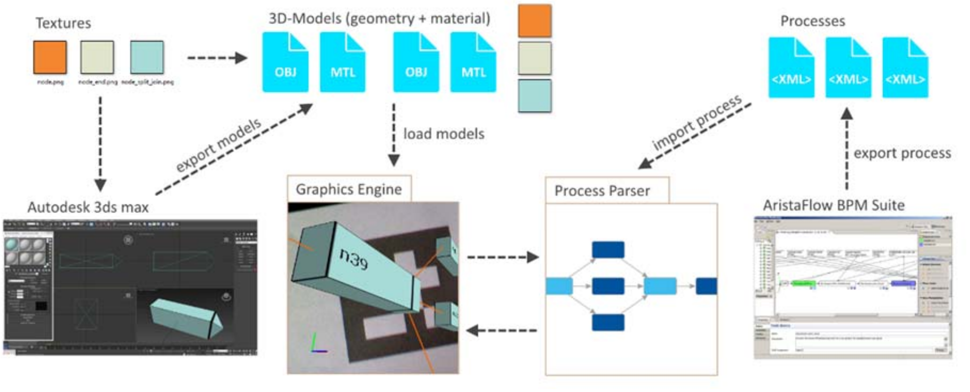

FIGURE 4: COMPUTATIONAL MODEL

Process Representation

The visual representation of the graph consists of the previously loaded 3D models, which represent the nodes of the graph (cf. Figure 5). Each node has its own visual counterpart, thus every node type has a different visual representation. At the moment, four types are used: *start/end*, *conditional join/split*, *parallel join/split* and *plain tasks*. Additionally, a second model in front of nodes contains the name of the node using a bitmap texture. All nodes are connected by edges according to the edge type.

All nodes and edges are placed on the plane, spanned by the x- and y-axis. Only backwards directed edge types are placed behind a node's plane. This allows us to provide the user with the complete graph without overlapping edges on the first plane, and limiting the possible overlaps on the second one.

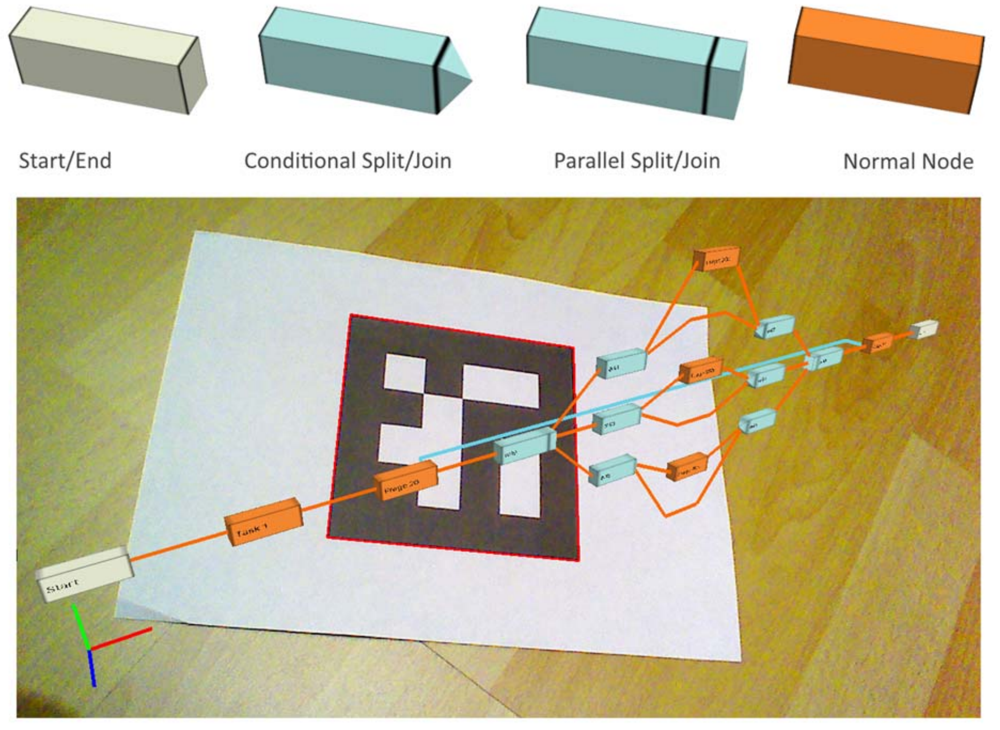

FIGURE 5: EXAMPLE PROCESS WITH CORRESPONDING MARKER

Interaction Dimensions

As we lifted the two dimensional graphical process representation to the third dimension, we gained additional interaction possibilities (cf. Figure 6). The third dimension allows us to move a process, like it's two dimensional counterpart on two axes and zoom into it, which corresponds with a translation on the third axis. Additionally, it is possible to rotate the complete process visualization along those three axes, which provides a useful feature (reported by users) to interact with process graphs. In particular, this feature provides a better overview on the backwards directed edge types on the second plane (and behind the actual graph). Furthermore, with a single touch, it is possible to extract information about nodes and present the details for mobile task configuration on the screen.

To provide a better experience during the interaction with those processes, an option to pause the marker detection was introduced. The user is not required to keep pointing the device's camera towards the marker, and may operate with the processes more easily.

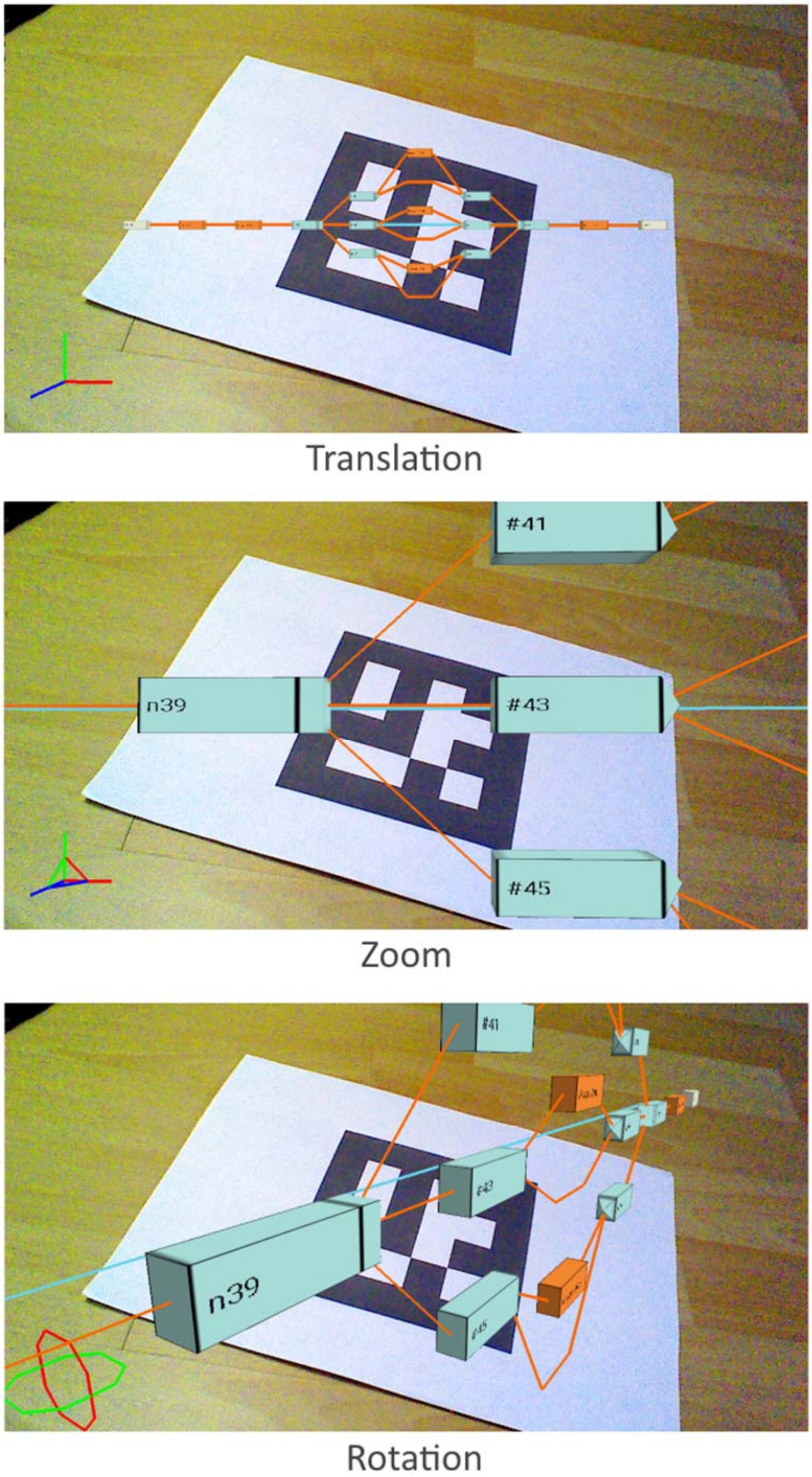

FIGURE 6: USER INTERFACE TO MODIFY PROCESS REPRESENTATION

Task and Process Operations

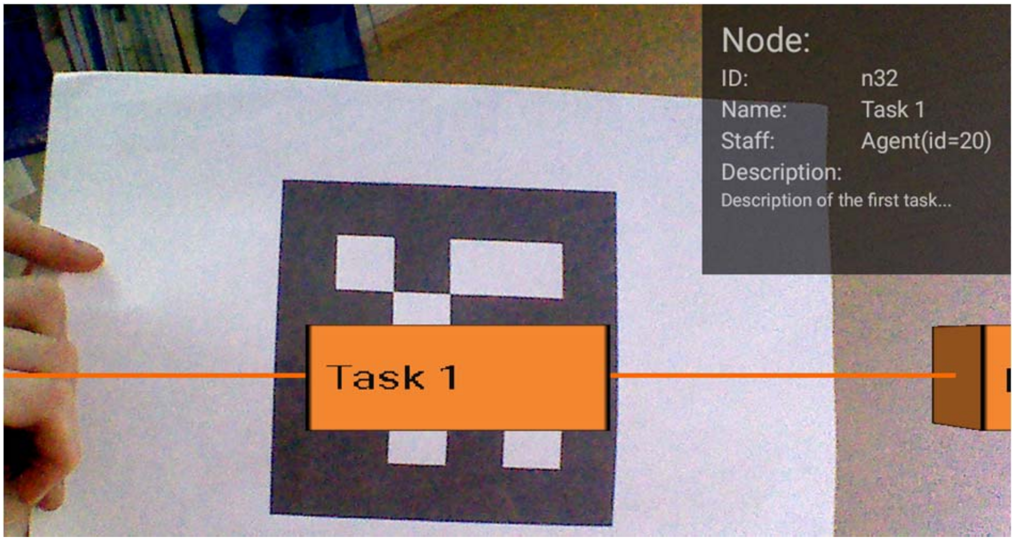

FIGURE 7: TASK VIEW WITH CORRESPONDING MARKER

The framework includes basic procedures to extract information from tasks that are mapped with its visually represented 3D nodes. At the moment, this information consists of attributes and parameters stored for the mobile tasks and being extracted by the parsing procedure.

Lessons Learned

The currently implemented 3D model has a few limits. Models, seen as single objects, should have a suitable size. If models are exported with wrong scales, the renderer will use those without rescaling, resulting in a misleading or miscalled representation. Additionally, the parser for the *.OBJ file format does not support the full feature set offered by the exported models. This means that single geometries of those models should have one material with at least a diffuse map as texture. Furthermore, single texture maps should be stored in a format supported by android. Only the PNG format was used during testing, but JPEG and many other formats should work as well, depending on the Android version and its support for the format in question.

Looking at a complete graphical screen consisting of multiple models, the GPU usage has to stay within limits. The texture sizes and triangle counts of geometries should stay within a reasonable limit to maintain a fluid experience during usage.

Mobile Task Configuration Using 3D Models

The architecture (cf. Figure 3) and computational model (cf. Figure 4) we presented for realizing 3D models for process graphs on smart mobile devices have shown their feasibility in practice. In particular, we were able to meet the requirements of a fluid experience for users working with our approach to configure mobile tasks. Furthermore, the overall approach has revealed the following other aspects:

- 3D models are a proper technique to show process graphs on smart mobile devices. In particular, domain experts have reported that using such models combined with the interaction possibilities meet their requirements to configure mobile process tasks as well as improve the overall process execution quality.
- Our overall approach has shown its flexibility. We started with process models based on ADEPT [3]. Currently, we integrate process models based on BPMN [11] and learned that this could be easily done. To be flexible in the context of BPM is a major challenge [1, 10] which must be properly addressed.
- The integration of a marker component has been reported as very useful. Assume that within a warehouse, a warehouseman wants to check whether a shelf in his warehouse is related to a task (e.g., to manage goods for this shelf). Furthermore, he wants to check the parameters of the corresponding mobile task of the process that manages the goods of the warehouse. For this purpose, the shelf has a QR-code which will be recognized by a smart mobile device that shows based on the information of the QR-code the corresponding mobile task and its parameters.

However, in practice we learned that many more challenges have to be tackled. For example, at the moment, we only support the Android mobile operating system. Other mobile operating systems must be addresses as well. In addition, the different ways of implementing mobile applications [14] must be also considered carefully in this context to meet the requirement of integrating sophisticated mobile task execution as well as configuration with existing BPM environments.

Conclusion

This chapter introduced an approach for configuring mobile task of business processes. The presented approach using 3D models allows for a completely new support of users to configure mobile tasks. Providing a sophisticated mobile task configuration support is challenging. We have shown that such a support is demanded by many BPM scenarios. The particular challenges in this context stem from the four presented categories, which we identified in the context of the investigated real-world scenarios. We have shown that particularly for mobile scenarios, a sheer number of parameters and execution anomalies are required to capture the mobile context properly. Furthermore, we have shown that in a mobile context these parameters are often not foreseeable at build time. Consequently, new techniques must be provided to configure these parameters on the *Go*. Moreover, we have learned that user acceptance is crucial in the context of mobile task configuration support. Although we have demonstrated the feasibility of our approach, many other challenges need to be tackled as well. We consider mobile task execution and configuration as shown in this chapter as one of the most challenging as well as most promising aspects of *BPM to Go*. Finally, our future work will consider distributed and collaborative processes in the context of *BPM to Go* as well.

References

1. Reichert, M., Dadam, P.: ADEPTflex - supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems 10(2), 93-129 (1998)
2. Dadam, P., Reichert, M.: The ADEPT project: a decade of research and development for robust and flexible process support - challenges and achievements. Comp Science Research & Developmt 23(2), 81-97 (2009)
3. Dadam, P., Reichert, M., Rinderle-Ma, S., Lanz, A., Pryss, R., Predeschly, M., Kolb, J., Ly, T., Jurisch, M., Kreher, U., Goeser, K.: From ADEPT to AristaFlow BPM Suite – a research vision has become reality. In: Proceedings BPM'09 Workshops, LNBIP 43, Springer, pp. 529-531 (2009)
4. Ko, R., Lee, S., Lee, W.: Business process management (BPM) standards – a survey. Business Process Management Journal 15(5), 744-791 (2009)
5. Atzori, L., Iera, A., Morabito, G.: The Internet of Things – a survey. Computer Networks 54(15), 2787-2805 (2010)
6. Pryss, R., Tiedeken, J., Kreher, U., Reichert, M.: Towards flexible process support on mobile devices. In: Proceedings CAiSE'10 Forum, Hammamet, Tunisia, LNBIP 72, Springer, pp. 150-165 (2010)
7. Zaplata, S., Hamann, K., Kottke, K., Lamersdorf, W.: Flexible execution of distributed business processes based on process instance migration. Journal of Systems Integration 1(3), 3-16 (2010)
8. Knuplesch, D., Pryss, R., Reichert, M.: Data-aware interaction in distributed and collaborative workflows: modeling, semantics, correctness. In: Proceedings 8th IEEE Int'l Conference on Collaborative Computing, Pittsburgh, Pennsylvania, pp. 223-232 (2012)
9. Kocurova, A., Oussena, S., Komisarczuk, P., Clark, T.: Context-aware content-centric collaborative workflow management for mobile devices. In: Proceedings 2nd Int'l Conference on Advanced Collaborative Networks, Systems and Applications, pp. 54-57 (2012)
10. Reichert, M., Weber, B.: Enabling Flexibility in Process-aware Information Systems: Challenges, Methods, Technologies. Springer (2012)
11. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2012)
12. Yang, Y., Lu, W., Domack, J., Li, T., Chen, S., Luis, S., Navlakha, J.: Madis: A multimedia-aided disaster information integration system for emergency management. In: Proceedings 8th Int'l Conf on Collaborative Computing (2012)
13. Pryss, R., Musiol, S., Reichert, M.: Collaboration support through mobile processes and entailment constraints. In: Proceedings 9th Int'l Conf on Collaborative Computing (2013)
14. Schobel, J., Schickler, M., Pryss, R., Nienhaus, H., Reichert, M.: Using vital sensors in mobile healthcare business applications: challenges, examples, lessons learned. In: Proceedings 9th Int'l Conf on Web Information Systems and Technologies, Aachen, pp. 509-518 (2013)

15. Geiger, P., Schickler, M., Pryss, R., Schobel, J., Reichert, M.: Location-based mobile augmented reality applications: challenges, examples, lessons learned. In: Proceedings 10th Int'l Conf on Web Information Systems and Technologies, Barcelona, pp. 383-394 (2014)
16. Mundbrod, N., Reichert, M.: Process-aware task management support for knowledge-intensive business processes: findings, challenges, requirements. In: Proceedings IEEE 18th Int'l Distributed Object Computing Conference - Workshops and Demonstrations (EDOCW'14), Ulm, Germany, IEEE Computer Society Press, pp. 116-125 (2014)
17. Pryss, R., Musiol, S., Reichert, M.: Integrating mobile tasks with business processes: a self-healing approach. In: Handbook of Research on Architectural Trends in Service-Driven Comp, pp. 103-135 (2014)
18. Schobel, J., Schickler, M., Pryss, R., Maier, F., Reichert, M.: Towards process-driven mobile data collection applications: requirements, challenges, lessons learned. In: Proceedings 10th Int'l Conf on Web Information Systems and Technologies, Barcelona, pp. 371-382 (2014)
19. Pryss, R., Mundbrod, N., Langer, D., Reichert, M.: Supporting medical ward rounds through mobile task and process management. Information Systems and e-Business Management 13(1), 107-146, Springer (2015)