Context-Based Assignment and Execution of Human-Centric Mobile Services

Rüdiger Pryss, Manfred Reichert, Marc Schickler
Ulm University
Institute of Databases and Information Systems
{*ruediger.pryss,manfred.reichert,marc.schickler*}*@uni-ulm.de*

Thomas Bauer
University of Applied Sciences
Neu-Ulm
thomas.bauer@hs-neu-ulm.de

***Abstract*—Performing tasks with the help of smart mobile devices is demanded for various areas in everyday life. In business environments, for example, tasks requiring complex paper work (e.g., paper-based documentation in the context of machine maintenance) shall be digitally transformed with the use of smart mobile devices. However, the realization of respective mobile applications is challenging as coordination issues have to be addressed in this context. For example, mobile application A performing task A may have to be finished before mobile application B performing task B may be started, i.e., human-centric mobile services need to be coordinated. To accomplish the latter, a formal context capturing service dependencies is required, while at the same time considering the mobile context of each involved human-centric mobile service needs to be considered. The presented approach extends existing process management technology with mobile activities to enable this. More precisely, we developed a mobile context framework that allows for a robust and flexible execution of mobile activities. The feasibility of the approach is demonstrated through a prototypical implementation as well as case studies. Altogether, the support of human-centric mobile services is promising regarding work efficiency in numerous scenarios and application domains in everyday life.**

***Keywords*-mobile service, mobile human tasks, mobile context, mobile process, mobile worklist, mobile user assignment**

I. INTRODUCTION

The tremendous proliferation of smart mobile devices over the last years has affected many areas in everyday life. In line with this trend, the demand to use smart mobile devices in business scenarios is growing [1]–[3]. More and more tasks, which are hitherto paper-driven, shall be executed with applications running on smart mobile devices. For this purpose, numerous frameworks emerged enabling the rapid development of mobile applications [4].

Using process management technology in the context of process-centric business scenarios is another fundamental trend in enterprise computing [5]. However, approaches integrating smart mobile devices with process management technology are rather premature. To remedy this drawback, the approach presented in this paper enables the execution of mobile activities. The latter constitute process activities (i.e., single process steps) to be executed on smart mobile devices. Note that the technical integration of this activity type with existing process management technology in itself is challenging [6]. If a mobile context shall be additionally considered when executing the activities, the integration gets even more complex. However, the use of such a mobile context offers several advantages. For example, (mobile) activity execution time can be significantly decreased if mobile activities are only assigned to those users whose location is close to the one of the mobile activity.

We denote context-aware mobile activities running on smart mobile devices and controlled by a process management system as *human-centric mobile services* (cf. Fig. 2). To elaborate the requirements for the support of human-centric mobile services, we analyzed various real-world scenarios. Table I lists three examples for which we also implemented mobile applications. The latter supported the respective scenarios and constituted the base for identifying (1) parameters of the mobile context and (2) issues that arise when technically integrating mobile activities with process management technology.

Scenario	**Description**
Clinical Ward Round [6], [7] **Mobile Prototype**: iOS Tablet Application	Analysis of 3 clinical ward rounds to identify (1) required mobile data access demands and (2) activities to be executed in a mobile manner. **Characteristics**: High dynamics, many exceptions, potentially large number of users, many revealed context parameters, sensor data.
Airline Catering **Mobile Prototype**: Windows Phone Application	Analysis of an airline catering procedure with same goals like for the clincial ward round scenario. **Characteristics**: Low dynamics, frequent network connection losses, potentially large number of users, long-running activities, many revealed context parameters, sensor data.
Telecare **Mobile Prototype**: Android Tablet Application	Analysis of the telecare activities of practitioners, same goals as for the above scenarios. **Characteristics**: Low dynamics, frequent network connection losses, few users, sensor data, offline mode, many revealed context parameters, demanding data management.

Table I: Analyzed Real-World Scenarios

Based on the insights we gained in the context of these scenarios, the conceptual framework depicted in Fig. 1 is proposed to properly enable human-centric mobile services. *Seven* aspects are crucial in this context. The present paper focuses on the three aspects coloured in blue in Fig. 1. First, a mobile context (cf. Fig. 1, *Aspect 1*) needs to be elaborated. Second, it must be determined how to properly assign mobile users to mobile activities taking (1) the mobile context and (2) the process management system characteristics into account (cf. Fig. 1, *Aspect 4*). Third, it must be understood what aspects need to be considered when executing human-centric mobile services (cf. Fig. 1, *Aspect 5*). For example, a mobile context can be used to support the execution phase through a monitoring of context parameters. Therefore, the coloured aspects must be tightly integrated in a technical realization. For example, if an exception handling does not consider the mobile context, the overall approach is less useful. Altogether, to the best of our knowledge, no other approaches have proposed a conceptual framework that allows for a tight integration of a mobile context with process management technology to enable human-centric mobile services in the way we support them.

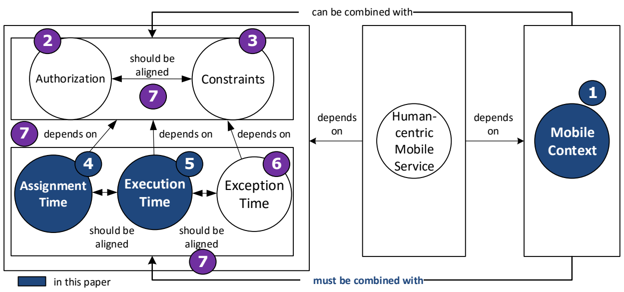

Figure 1: Conceptual Framework

The remainder of this paper is structured as follows: Section II introduces general aspects of human-centric mobile services. Section III discusses the support of human-centric mobile services based on process management technology. In Section IV, the mobile context is defined, while Section V presents the way mobile activities are assigned to mobile users utilizing the mobile context. Section VI discusses run time issues to be covered when executing human-centric mobile services. Section VII presents a proof-of-concept prototype, together with three case studies in which we applied the concepts. Finally, Section VIII discusses related work, while Section IX concludes the work with a summary and outlook.

II. Human-centric Mobile Services

Fig. 2 illustrates human-centric mobile services in a process context. As shown, the process management system assigns and manages process activities. Thereby, two basic types of activities are distinguished: the ones automatically executed and activities performed by users. The latter are denoted as human activities.[1] Their execution, in turn, is managed by the process management system. For this purpose, a process client is provided that may be deployed to desktop computers. The communication between the process client and the process management system is based on various protocols and concepts, which, in turn, enable the execution of robust and flexible human-centric services.

Based on the analyzed real-world scenarios (cf. Table I), the mobile execution of human-centric services requires novel concepts in two respects. First, the protocol of assigning human activities needs to be changed. Second, the notion of a mobile context must be elaborated. Finally, the changed assignment must be properly combined with the mobile context. In the following, human-centric mobile services are denoted as mobile activities.

Regarding the assignment of mobile activities, two aspects need to be distinguished. First, the communication between the process client and the process management system must be changed. This requires new protocols. Second, a novel worklist management for the process client is required. The worklist manages activity handling for users. In particular, it offers features to accept or decline activities assigned to users by the process management system. As shown in Fig. 3③, all activities for which a user qualifies will be added to the *ActivitiesAtHand* list. All activities in this list can then be claimed by the user. If one activity is actually claimed, it will be added to the *MyActivities* list and removed from the *ActivitiesAtHand* list. In addition, for all other users for whom this activity is contained in their *ActivitiesAtHand* list, it will be removed. Finally, if the user declines an activity, it will be solely removed from this particular *ActivitiesAtHand* user list.

Algorithms managing the worklists of mobile users were added. These algorithms, in turn, had to deal with the issues presented in Fig. 2①. For example, they utilize the mobile context to determine which users are qualified to execute a particular mobile activity. If more than one user is qualified, it must be further determined whether they are equally qualified.

For the mobile context, two major issues had to be addressed. First, we needed to identify those aspects required to capture the mobile context properly. Second, we elaborated in what way they can be evaluated by the algorithms. The analysis of the real-world scenarios as well as extensive studies of related work (cf. Table I) revealed important and useful context parameters (cf. Fig. 2②). For example, the location of a user must be managed. All parameters are summarized in a coarse-grained catalogue, which, in turn, is evaluated by the algorithms.

III. Process Context

First of all, a meta-model for mobile process activities was developed (cf Fig. 3②) that is based on an extensive literature review (e.g., [8], [9]). The meta-model therefore particularly considers (1) related work and includes (2) mobile activities as well as (3) the mobile context. Fig. 3 further illustrates mobile worklist management in the context of the extended meta-model (cf. Fig. 3③,④). Thereby, all meta-model entities concerned with worklist management are marked with a corresponding symbol. As can be seen, adaptations were required to enable a proper worklist management for mobile activities as well (cf. Fig. 3③). In the

[1] Alternatively, they are denoted as interactive activities.

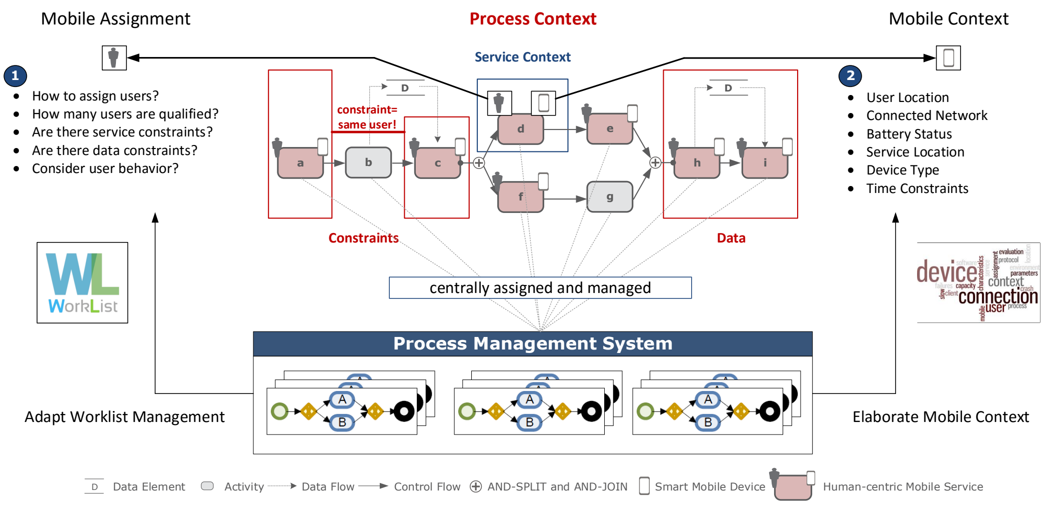

Figure 2: Human-centric Mobile Services based on Mobile Process Activities

following, the extended meta-model is denoted as *mobile process meta-model*.

In addition to the changes regarding worklist management, other adaptations became necessary as well: First, entity *mobile activity* was added. Second, for enacting the mobile activity, two algorithms were developed dealing with mobile worklist management (cf. Fig. 3⑤,⑥).

The **first algorithm** selects all mobile users qualified to execute a mobile activity (cf. Fig. 3⑤), this algorithm is denoted as *Selection Algorithm*. Selection means that all mobile users determined by the algorithm are qualified to execute the mobile activity. In this context, the goal is to find as many qualified mobile users as possible in order to allow for a flexible and robust execution of mobile activities. In mobile environments, with frequently changing circumstances and limited resources, the goal to achieve a high degree of flexibility and robustness is of utmost importance. Consequently, the algorithms mainly pursue robustness and flexibility by taking the mobile context appropriately into account. Note that the *Selection Algorithm* is applied when mobile activities become instantiated (cf. Fig. 3⑤).

The **second algorithm**, the *Ranking Algorithm*, handles exceptions (e.g., smart mobile device crashes) during the execution of mobile activities (cf. Fig. 3⑥). As shown in Fig. 3, both algorithms must additionally consider authorization issues and constraints. For example, a concept like *role* [10] can be used to govern which users are allowed to execute which activities. Constraints, in turn, allow for more complex authorization access rules. For example, they can be used to ensure that two activities need to be executed by the same user [2]. Note that the consideration of both authorization and constraint issues in the context of mobile activities are important contributions of the proposed conceptual framework as well (cf. Fig. 1②,③).

Furthermore, changes to the state model of mobile activities (cf. Fig. 3⑧) became necessary. More precisely, two adaptations were required compared to non-mobile activities. First, the behavior of specific state transitions was changed (cf. Fig. 3 red-coloured transitions). For example, there exist scenarios, in which only one mobile user is qualified to execute a mobile activity. If she not claims it, but urgent execution is required, the activity will be automatically assigned to her. To enable this feature, a change was applied to the transition between state *Selected* to state *Started*. Second, new states were required in the context of exception handling. Finally, the mobile context was added to the meta-model (cf. Fig. 3⑦).

IV. Mobile Context

The mobile context is represented by a parameter catalogue evaluated by the *Selection* and *Ranking* algorithms. The parameters were identified when the developed mobile prototypes (e.g., [6], [7]) were applied to real-world scenarios (cf. Table I). In addition, the catalogue is based on a comprehensive literature study (e.g., [8]).

For several reasons, we assign parameters to four categories: First, parameters related to the smart mobile device of a mobile user (*SMD* parameters) are managed. For example, a battery status is managed for smart mobile devices. Second, all parameters associated with mobile activities (*MA* parameters) are managed. For example, the execution location of the mobile activity is captured. Third, parameters that can be related to the overall process execution are managed. This category became necessary to cope with the complexity of managing a multitude of parameters. For several parameters it would be costly to manage them separately for each mobile activity. Therefore, a parameter applying to all mobile process activities is used, e.g., to manage a generally demanded battery status for the activation of all

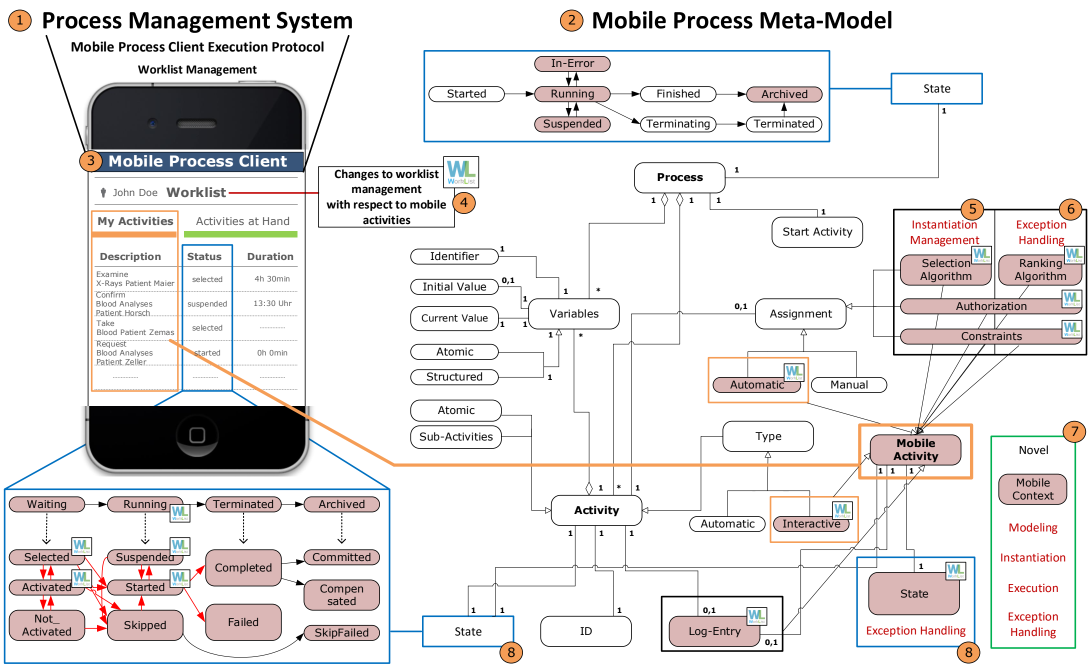

Figure 3: Mobile Process Meta-Model and Worklist Management

mobile activities of a process. Finally, parameters associated with mobile users (*MU* parameters) are managed, e.g., the usual location of a mobile user.

The entire parameter catalogue is presented in Table II. Note that Column *T* indicates whether a parameter is of type *symbolic* or *measured*. From the considered application scenarios we revealed that such differentation is useful. Symbolic parameters are used in related work to define parameters on an abstract level [11]. For example, regarding the location of a mobile activity, the symbolic parameter *emergency room* might be used. First, symbolic parameters are considered as they can already be evaluated before starting a process. For example, consider the following scenario: Symbolic parameters are managed for mobile users and mobile activities. Hence, it can be indicated before the execution of a mobile activity takes places how many mobile users are closely located to the location of the mobile activity that shall be performed based on a simple comparison of the symbolic parameters. Apparently, it is just an indication, but has shown its advantages in practice. Second, for the assignment of mobile users to mobile activities symbolic parameters can be advantageous as well. For example, if precise location information (e.g., GPS) cannot be obtained for a user location due to a connection loss, the symbolic parameter may be used instead. Conversely, *measured* parameters are automatically determined after starting the process instance. For example, if a mobile acticity shall be executed, the battery status of all mobile users are gathered.

Due to lack of space, only selected parameters are discussed in more detail. The *form factor* parameter is used

CPM	Description	T	UA	EH	CH
Category I: Smart Mobile Device (SMD)					
SMD_{BS}	Battery Status	M	✓	✓	✓
SMD_{FF}	Form Factor	S	✓	✓	✓
SMD_{NT}	Network Type	M	✓	✓	✓
SMD_{GC}	Geometric Coordinate	M	✓	✓	✓
Category II: Mobile Activity (MA)					
MA_{SC}	Symbolic Coordinate(s)	S	✓	✓	✓
MA_{GC}	Geometric Coordinate	M	✓	✓	✓
MA_{LR}	Location Range	S	✓	✓	✓
MA_{BS}	Battery Status	M	✓	✓	✓
MA_{U}	Urgency Value	S	✓	✓	✓
MA_{OFF}	Offline Mode	S	✓	✓	✓
MA_{FF}	Form Factor	S	✓	✓	✓
MA_{RF}	Response Frequency	S	✓	✓	✓
MA_{UT}	User Threshold	S	✓	✓	✓
Category III: Process (P)					
P_{IST}	Instant Shutdown Threshold	S	✓	✓	✓
Category III IV: Mobile User (MU)					
MU_{SC}	Symbolic Coordinate(s)	S	✓	✓	✓
MU_{IS}	Instant Shutdowns	S	✓	✓	✓

T=Type ⇒ M=Measured, S=Symbolic,
UA=User Assignment, EH=Exception Handling, CH=Constraint Handling,
√=holds, X =not holds

Table II: Mobile Context Parameters

to indicate the smart mobile device type (i.e., tablet or smartphone). *Geometric* coordinates, in turn, are measured and correspond to GPS coordinates in an outdoor scenario and WLAN coordinates in an indoor scenario. Parameter *location range* refers to the location of a mobile activity defining a radius around its geometric coordinates. Mobile users located inside this radius are considered for executing this mobile activity. *Urgency value* defines a period (or point in time) during which the mobile activity must be executed. Furthermore, *response frequency* is a value that determines the frequency with which the mobile device of a particular user must report its online status to the process management system. If the smart mobile device does not obey this reporting frequency, an exception handling is triggered. Related to *response frequency* is parameter *offline mode*. If the latter is set to *true* for any mobile activity, the smart mobile device may ignore the *response frequency* to enable offline execution of this mobile activity. In practice, this was frequently demanded. Finally, *network type* is used to capture the network connection of a smart mobile device (e.g., WLAN or UMTS).

In addition, the parameter catalogue contains two threshold parameters: *User threshold* indicates the number of users that need to be available to activate a mobile activity. In turn, the *instant shutdowns* parameter captures the behavior of mobile users. Note that in practice users may instantly shutdown their mobile device without reflecting on the consequences of this shutdown. Usually, this constitutes a short-term problem and the device can be restarted soon in most cases. If a user exhibits many instant shutdowns, however, this misbehavior should be considered in the context of mobile activity assignments. To cope with such *careless* shutdowns, a smart mobile device sends a message to a service that an instant shutdown will take place soon. In this context, several mobile development frameworks were evaluated (i.e., Google Android, Apple iOS, Microsoft Windows Mobile) and it could be demonstrated that this solution for detecting instant shutdowns is feasible for all of them. Finally, to assess user behavior over time (e.g., whether or not a user performs many instant shutdowns), parameter *instant shutdowns* is managed. Related to it, a general process parameter *instant shutdown threshold* is managed for all mobile activities that is compared with the parameter managed for a particular mobile user. If the parameter value of a mobile user is above the *instant shutdown threshold* parameter, he or she will be particularly considered for the execution of the mobile activity.

Regarding the presented parameter catalogue, we do not claim that it captures all possible or required parameters. It rather reflects insights we gathered from our analysis of real-world scenarios. Furthermore, the application of the parameters identified in other practical scenarios was promising. Future analyses potentially will reveal additional parameters or invalidate existing ones. Second, since less related approaches aim to capture a mobile context to be able to properly assign and execute mobile activities in a process context, we do not claim that the catalogue is the most appropriate instrument for representing a mobile context. In practice, its feasibility, taking maintenance, robustness, and flexibility into account could be shown. Furthermore, domain experts were able to determine useful parameter values. Currently, we conduct case studies and work on metrics to formally show the effectiveness of the catalogue.

V. Mobile Assignment

For assigning mobile activities to mobile users, the *Selection Algorithm* was developed. It considers the presented mobile context to determine those mobile users being appropriate for executing the mobile activity. The respective calculation starts when a mobile activity gets activated (cf. Fig. 3⑧). For the users determined by the *Selection Algorithm*, the respective worklists are updated by adding the mobile activity to the *ActivitiesAtHand* list (cf. Fig. 3). The algorithm uses the parameters presented in Table II. Its a prerequisite for applying this algorithm that all parameters of type *symbolic* are manually specified. In turn, the ones of type *measured* are automatically determined by the process management system. Algorithm 1 sketches the *Selection Algorithm*. It consists of four calculations evaluating the appropriateness of mobile users to execute mobile activity n. These users are stored in the list $coMU(n)$ for mobile activity n. **First**, Algorithm 1 determines all mobile users that are (1) inside the location range of the considered mobile activity $(0 < nLR(mu) \leq 1)$[2] and (2) for whom the battery status of their mobile device is appropriate $(SMD_{BS}(mu) \geq MA_{BS}(n))$. The latter is determined by comparing the battery status of a mobile user with the battery status demanded for the mobile activity. In summary, the first calculation determines all mobile users with appropriate physical location and battery status, and adds them to $coMU(n)$ (cf. Alg. 1, Lines 5-13). The **second** part is only performed if the threshold of required users for mobile activity n is not reached by the first calculation $(c1 < MA_{UT}(n))$. In the latter case, all mobile users whose symbolic coordinates as well as battery status are appropriate will be added to $coMU(n)$ (cf. Alg. 1, Lines 14-24). **Third**, the urgency parameter is considered. If it is set to *true* for mobile acitivity n $(MA_U \neq 0)$, all mobile users whose device has an inappropriate form factor (cf. Alg. 1, Lines 25-35) are removed from $coMU(n)$. The scenario analysis revealed that an appropriate form factor usually speeds up activity execution duration. **Finally**, if the number of determined mobile users in $coMU(n)$ is still above the required threshold, all mobile users in $coMU(n)$ are ranked based on instant shutdown behavior and battery status. This ranking, in turn, uses formula $SMD_{BS}(mu) + \frac{1}{MU_{IS}(mu)}$. Following this, all mobile users having lower rank compared to all other users in $coMU(n)$ (cf. Alg. 1, Lines 36-37) are removed from $coMU(n)$. Due to lack of space, the removement of users with lower rank is not presented in more detail.

The way authorization and security constraints can be combined with the *Selection Algorithm* has been not considered in this paper. Algorithm 1 uses the list $aMU(n)$ to represent all available users authorized to execute the mobile activity, while meeting the defined constraints. The

[2]See Table III for nLR formula.

Algorithm 1: Selection Algorithm

```
Data: Relevant context parameters
      aMU: Set of all available mobile users
Result: coMU(n): Set of context-based appropriate mobile users to instantiate mobile activity n
begin
    coMU(n) ←− ∅;                                   /* initialize variable */
    c1 ←− 0;                          /* counter for elements in coMU(n) */
    c2 ←− 0;                            /* counter for MA_FF = SMD_FF */
    /* Match geometric coordinates and battery status */
    foreach mobile user mu ∈ aMU do
        if (0 < nLR(mu) ≤ 1) ∧ (SMD_BS(mu) ≥ MA_BS(n)) then
            c1 + +;
            coMU(n) ←− coMU(n) ∪ {mu};
            if MA_FF(n) = SMD_FF(mu) then
                c2 + +;
            end
        end
    end
    /* Match symbolic coordinates and battery status if needed */
    if c1 < MA_UT(n) then
        foreach mobile user mu ∈ (aMU \ coMU(n)) do
            if (MA_SC(n) = MU_SC(mu)) ∧ (SMD_BS(mu) ≥
              MA_BS(n)) then
                c1 + +;
                coMU(n) ←− coMU(n) ∪ {mu};
                if MA_FF(n) = SMD_FF(mu) then
                    c2 + +;
                end
            end
        end
    end
    /* Consider urgency */
    if MA_U ≠ 0 ∧ c2 > MA_UT(n) then
        foreach mobile user mu ∈ aMU do
            if MA_FF(n) ≠ SMD_FF(mu) then
                coMU(n) ←− coMU(n) ∩ {mu};
                c1 − −;
            end
            if c2 = MA_UT(n) then
                exit();
            end
        end
    end
    /* Consider battery status again with instant shutdowns */
    if c1 > MA_UT(n) then
        /* Call method to remove all mobile users from coMU(n) until
           c1 = MA_UT(n) due to SMD_BS(mu) + 1/MU_IS(mu) */
    end
end
```

procedure how $aMU(n)$ can be calculated is not shown as the presented work focuses on the mobile context and its use for executing mobile activities. It should be kept in mind that this procedure ensures the overall and dynamic assignment of mobile activities to mobile users in the process context.

Altogether, when applying Algorithm 1 to real-world scenarios, its applicability could be shown particularly with respect to the better and context-aware assignment of mobile users to mobile activities. Consequently, the robustness and flexibility of mobile activities and their execution could be improved. Currently, a comprehensive case study elaborates metrics for robustness and flexibility. Another ongoing aspect considers user expertise. For the exception handling we already factor user expertise, while the assignment of mobile users solely factors instant shutdowns with respect to user behavior. Therefore, user expertise should be also covered for the assignment of mobile users.

VI. Mobile Execution

When executing mobile activities, the mobile context is utilized to increase robustness and flexibility. In this context, three goals are pursued. First, as many parameters as possible shall be (1) continously measured and (2) evaluated during the execution of mobile activities. Table III(*Parameters*) lists the operations that will be accomplished during run time. The second goal was to increase robustness through changing worklist management. In particular, the management of the *MyActivities* list (cf. Fig. 3) was adapted. Recall that with this list all activities a particular user has claimed are managed. The user then decides for each activity individually about the point in time he actually will perform it. Consequently, resource situation may change between claiming an activity actually performing it. To assist users in deciding which activity they shall be perform next, the worklist is prioritized based on the rules shown in Table III(*Worklist*). The first rules considers the current location of a user, whereas the second one takes the battery status into account. As third goal, we address dependencies among the parameters. For example, we automatically derive a parameter value based on the value of another parameter (cf. Table III, *Functional Dependencies*). Finally, if a device is no longer available or crashes during run time, the process activity will be dynamically switched to another device. This is ensured by our exception handling, which is not discussed in this paper. Recent related work also addresses exception handling [12] deeply.

	Description
Parameters	
MU_{IS}	Will be increased in case of an instant shutdown (M).
MA_U	Activities must be not suspended if urgency is set (E).
MA_{OFF}	No response by the smart mobile device is required (E).
MA_{RF}	Determines response frequency of smart mobile device (E).
SMD_{NT}	Will be evaluated due to dependency $SMD_{NT} \rightarrow SMD_{BS}$ (E).
Worklist Management	
$0 < normEZB(MU) \leq 1$: To prioritize mobile activities in worklist	
$SMD_{BS} \geq MA_{BS}$: To prioritize mobile activities in worklist	
Applied Rules	
$\text{nLR(MU)}=\begin{cases} \frac{MA_{GC}-SMD_{GC}(MU)}{MA_{LR}}, & if\ (1) \\ 0, & if\ (2) \end{cases}$ (1)	
Used Functional Dependencies	
$SMD_{NT} \rightarrow SMD_{BS}$	Network type determines required battery status.
$MA_{GC} \rightarrow MA_{LR}$	Geometric coordinates determine location range.
$MA_{OFF} \rightarrow MA_U$	Offline mode determines urgency value.
$MA_{OFF} \rightarrow MA_{RF}$	Offline mode determines response frequency.

$\surd$=holds, X=not holds, $-$=not relevant,
$(1) = if\ (MA_{GC} - SMD_{GC}(MU)) \neq 0,$
$(2) = if\ (MA_{GC} - SMD_{GC}(MU)) = 0$
$M = measured, E = evaluated$

Table III: Context Evaluation during Execution

VII. Evaluation

The concepts for defining the mobile context as well as for the assignment and execution of mobile activities were evaluated through a prototypical implementation. In addition, the prototype was applied in three case studies. Fig. 4 summarizes major parts of the prototype. First, on the left side, the architecture of the prototype is presented. Second, in the middle, the protocol which was developed to govern the communication between the process management system and the mobile process client is shown.

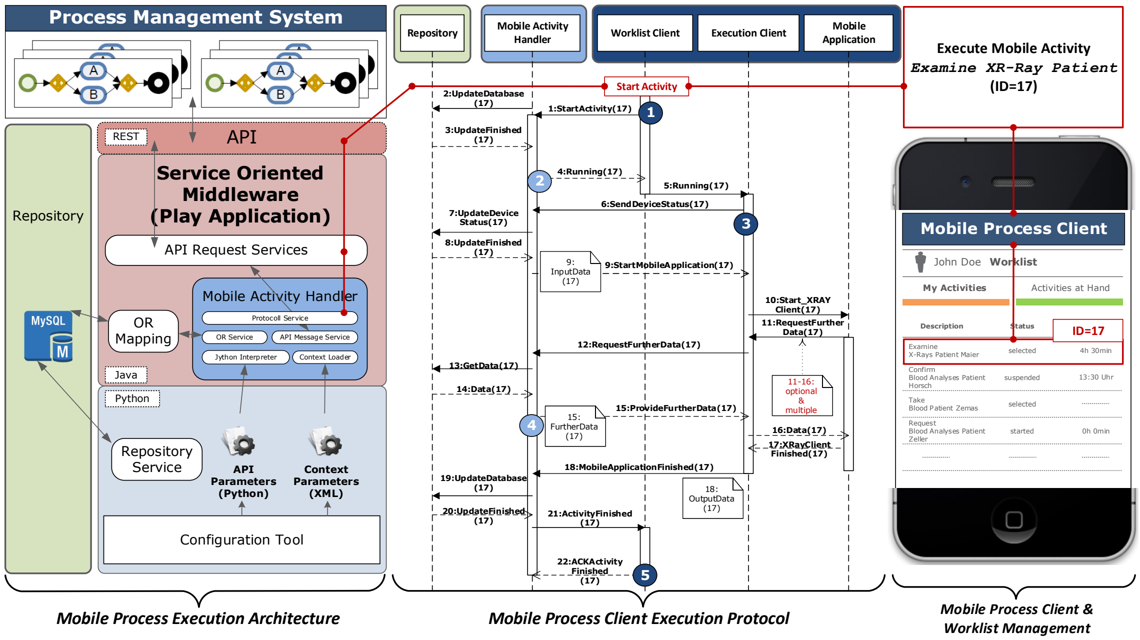

Figure 4: Prototype with Execution Protocol

The prototype was developed as a service-oriented middleware that interacts between an existing process management system and mobile process client. The basic components of the middleware include a service-centric play application [13] and a MySQL database connected to it. The concepts presented in this work are realized by the *mobile activity handler*. Processes, in turn, are managed by the integrated process management system AristaFlow [14].

The newly developed protocol governs the communication between the mobile process client and the service-oriented middleware. The mobile process client, in turn, consists of three components. First of all, it comprises a worklist client and and execution client. Further, it must be able to integrate additional mobile applications executed on the mobile device to actually perform a mobile activity (e.g., Mobile Microsoft Word). The worklist client manages the presented worklist aspects. The execution client centrally manages the communication between the worklist client, the invoked mobile applications, and the service-oriented middleware. Besides the technical development of the protocol, a particular challenge was to identify at what protocol points the mobile context parameters (cf. Table III) shall be exchanged between the mobile process client and the service-oriented middleware. Fig. 4 ②,④ presents the two identified protocol points in which the service-oriented middleware requests parameter values from the mobile process client. In turn, Fig. 4 ①,③,⑤ presents the points in time in which the mobile process client actively sends parameter values to the service-oriented middleware. These points were identified based on the scenarios analyzed (cf. Table I).

The prototype was applied to three application scenarios. These include logistics warehousing, rescue management, and psychological interview management. In the context of these scenarios, 74 different mobile activities were considered in total. The results cannot be explained in more detail due to the lack of space. To conclude, the overall concepts developed in this work were useful for the support of the 74 mobile activities. In particular, the context-based assignment of users was appreciated by process participants.

VIII. Related Work

Related work includes the execution of mobile activities in the context of process-aware information systems. Respective approaches usually focus on logical models for mobile processes on one hand and architectures as well as implementations for light-weight process engines on the other. Logical models include, for example, approaches partitioning process models and executing selected process fragments on smart mobile devices [15]–[18]. So far, however, none of these approaches has provided support for executing mobile activities in a way comparable to our work.

There are only few approaches specifically dealing with the support of mobile activities. Usually, they focus on particular aspects of mobile activities [6], [7], [19].

To the best of our knowledge, the consideration of a mobile context being used to execute mobile activities in a robust and flexible manner has been not considered in a formal process context. Developing algorithms to select only appropriate users during run time of a process has been addressed by approaches that do not consider smart mobile devices [20]. Interestingly, non-mobile approaches do not focus on the assistance of stationary users during activity execution as discussed here with respect to the mobile context for mobile users. Altogether, the context-specific support for the assignment and execution of mobile activities, as provided by the conceptual framework presented in this paper, has not been considered by many other approaches so far [12].

IX. Summary and Outlook

An approach, which is based on a context model, was provided for the assignment and execution of human-centric mobile services. It was shown that they can be realized as mobile activities integrated with existing process management technology. The mobile context, in turn, was elaborated through cases studies as well as a comprehensive literature study. In future work, the mobile context will be improved by a formal evaluation of its effectiveness. Moreover, an abstract perspective on smart mobile devices will be evaluated in order to utilize their characteristics even better when assigning mobile activities. To enable *human-centric* mobile services, many issues needed to be addressed in combination, as accomplished by the presented conceptual framework. In future work, concepts for exception and constraint handling will be presented. They are both important to enable a robust and flexible execution of human-centric mobile services with respect to practical requirements. Altogether, the support of human-centric mobile service is challenging. However, first results show that work of users in everyday life might be significantly eased. In general, more case studies and domains have to be analyzed in order to reveal even more comprehensive insights into how to apply human-centric mobile services in practice. For example, in life sciences mobile activity support is frequently demanded.

References

[1] R. Pryss, S. Musiol, and M. Reichert, "Extending Business Processes with Mobile Task Support: A Self-Healing Solution Architecture," in *Handbook of Research on Architectural Trends in Service-Driven Computing*. IGI Global, 2014, pp. 99–130.

[2] ——, "Collaboration Support Through Mobile Processes and Entailment Constraints," in *Proc 9th IEEE Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing*. IEEE Computer Society Press, 2013.

[3] R. Pryss, M. Reichert, A. Bachmeier, and J. Albach, "BPM to Go: Supporting Business Processes in a Mobile and Sensing World," in *BPM Everywhere*. Future Strategies Inc., 2015.

[4] J. Schobel, M. Schickler, R. Pryss, H. Nienhaus, and Reichert, M., "Using Vital Sensors in Mobile Healthcare Business Applications: Challenges, Examples, Lessons Learned," in *9th Int'l Conf on Web Information Systems and Technologies*, 2013, pp. 509–518.

[5] M. Reichert and B. Weber, *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer, 2012.

[6] R. Pryss, N. Mundbrod, D. Langer, and M. Reichert, "Supporting Medical Ward Rounds Through Mobile Task and Process Management," *Information Systems and e-Business Management*, no. 13, pp. 107–146, 2014.

[7] R. Pryss, D. Langer, M. Reichert, and A. Hallerbach, "Mobile Task Management for Medical Ward Rounds - The MEDo Approach," in *1st Int'l Workshop on Adaptive Case Management*, ser. LNBIP, no. 132. Springer, 2012, pp. 43–54.

[8] S. Zaplata, K. Hamann, K. Kottke, and Lamersdorf, W., "Flexible Execution of Distributed Business Processes Based on Process Instance Migration," *Journal of Systems Integration*, vol. 1, no. 3, pp. 3–16, 2010.

[9] "Workflow Management Coalition," http://www.wfmc.org/, 2014, [Online; accessed 19-June-2015].

[10] R. Sandhu, E. Coynek, H. Feinsteink, and C. Youmank, "Role-Based Access Control Models yz," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[11] C. Becker and F. Drr, "On Location Models for Ubiquitous Computing," *Personal Ubiquitous Computing*, vol. 9, no. 1, pp. 20–31, 2005.

[12] N. Chen, N. Cardozo, and S. Clarke, "Goal-Driven Service Composition in Mobile and Pervasive Computing," *IEEE Transactions on Services Computing*, no. 99, pp. 1–1, 2016.

[13] "Play Framework," https://www.playframework.com, 2014, [Online; accessed 19-June-2015].

[14] P. Dadam, M. Reichert, S. Rinderle-Ma, A. Lanz, R. Pryss, M. Predeschly, J. Kolb, T. Ly, M. Jurisch, U. Kreher, and K. Goeser, "From ADEPT to AristaFlow BPM Suite: A Research Vision has become Reality," in *Proc 1st Int'l Workshop on Empirical Research in Business Process Management*, ser. LNBIP, no. 43. Springer, 2009, pp. 529–531.

[15] L. Baresi and A. Maurino and S. Modafferi, "Workflow partitioning in mobile information systems," *Proc. IFIP TC8 Working Conf on Mobile Information Systems*, pp. 93–106, 2004.

[16] C. Kunze, S. Zaplata, and W. Lamersdorf, "Mobile Processes: Enhancing Cooperation in Distributed Mobile Environments," *Journal of Computers*, vol. 2, no. 1, pp. 1–11, 2007.

[17] ——, "Mobile Process Description and Execution," in *Distributed Applications and Interoperable Systems*. Springer, 2006, pp. 32–47.

[18] R. Pryss, J. Tiedeken, U. Kreher, and M. Reichert, "Towards Flexible Process Support on Mobile Devices," in *Proc CAiSE'10 Forum*, ser. LNBIP, no. 72. Springer, 2010, pp. 150–165.

[19] S. Zaplata and W. Lamersdorf, "Towards Mobile Process as a Service," in *Proc ACM Symposium on Applied Computing*, vol. 1. ACM, 2010, pp. 372–379.

[20] C. Cabanillas, J. Garcia, M. Resinas, D. Ruiz, J. Mendling, and A. Ruiz-Cortes, "Priority-Based Human Resource Allocation in Business Processes," in *Proc 11th Int'l Conf on Service-Oriented Computing*. Springer, 2013, pp. 374–388.