Managing Processes on Mobile Devices: The MARPLE Approach

Rüdiger Pryss, Julian Tiedeken and Manfred Reichert

Institute for Databases and Information Systems, Ulm University, Germany
{ruediger.pryss, manfred.reichert, julian.tiedeken}@uni-ulm.de

Abstract. Ubiquitous Computing is considered as enabler for linking everyday life with information and communication technology. However, developing pervasive applications that provide personalized user assistance is still a challenge. Relevant scenarios are diverse and encompass domains like healthcare, logistics, and business collaboration. Two of the technologies that show increasing maturity in respect to the demands of such applications are light-weight frameworks and process engines for mobile computing. Their fusion, however, is in a rather premature state. Generally, the support of mobile collaboration using a process engine raises challenging issues that need to be addressed. In the MARPLE project we target at a tight integration of process management technology with mobile computing frameworks in order to enable mobile process support in the aforementioned scenarios. In this demo paper we give insights into the MARPLE architecture and its components. In particular, we introduce the MARPLE process engine which enables light-weight as well as flexible process support on mobile devices.

1 Introduction

Mobile assistance in daily life as empowered by information and communication technology is a much discussed topic. To better understand the challenges emerging in this context, we analyzed real-world scenarios in which mobile user assistance is urgently needed and which stem from domains like healthcare, logistics and business collaboration. In particular, our analyses revealed the fundamental role of process support in respect to mobile and personalized user assistance.
This paper picks up a healthcare scenario in which chronically ill patients shall be assisted by mobile devices. Such mobile device gives recommendations in respect to medications. These recommendations, in turn, are made remotely by healthcare professionals and depend on the previously gathered patient data (e.g. blood pressure). Despite its high potential, so far there exists no comprehensive mobile assistance for such scenarios. One issue emerging in the given context is to decide which process parts shall run on mobile devices and which on stationary computers. In the following we refer to the described scenario to discuss fundamental challenges and to show the high potential of mobile assistance. Fig. 1 illustrates both traditional realization of this scenario ① and its realization based on mobile devices and mobile assistance respectively ②.

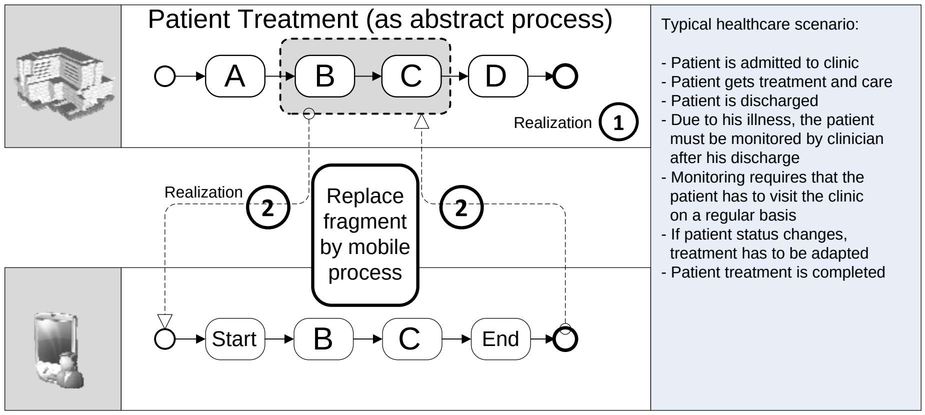

Fig. 1. Abstract Healthcare Scenario

After discharging a patient the usual way to monitor his health status is to schedule regular visits for him in the clinic. In certain cases, however, this can lead to delayed adaptations of his treatment plan in case his status has changed. To improve this situation and to enable real-time monitoring, mobile data collection and mobile assistance ② of the patient would be highly welcome by all parties; i.e., the patient needs to be assisted by a mobile device which gathers medical data from him and informs clinicians about status changes.

To realize the second scenario patient-specific application logic needs to be provided on the mobile device. Consequently, the overall treatment workflow is maybe partitioned ② and process fragments may run on stationary computers as well as on mobile devices. In particular, the process fragment running on the mobile device needs to be adapted to the specific patient and may evolve over time, i.e., hard-coded process implementations are not tolerable.

To enable mobile assistance we developed a light-weight process engine called MARPLE that runs on the mobile device and that is able to interact with back-end processes if required. In addition, we provide advanced tools for defining, configuring and verifying process fragments. In this paper, we focus on the core architecture and components of the MARPLE mobile process engine. Conceptual issues related to the partitioning of processes as well as to the synchronization of the resulting fragments are outside the scope of this paper.

When developing our MARPLE engine we had one shining example to follow - the ADEPT process management system we had developed during the last decade [1]. In particular, we adopt basic design principles from ADEPT (e.g., correctness-by-construction, dynamic process adaptability), but also align the MARPLE architecture with specific needs of mobile processes.

Section 2 introduces a concrete application scenario. In Section 3 we give insights into the MARPLE architecture, while Section 4 shows how the described scenario can be supported in MARPLE. Section 5 discusses related work and Section 6 concludes with a summary and outlook.

2 Application Scenario

Fig. 2 shows a typical example of a healthcare process which is modeled in terms of BPMN. Assume that for a particular patient this process is executed three times per day and involves three parties. The first swim lane ① shows activities conducted at the clinic, which starts the process. When executing step ②, the clinic triggers the process fragment running on the mobile device of the patient. This mobile process collects patient data and coordinates required actions (e.g., to measure blood pressure or to gather ECG data).

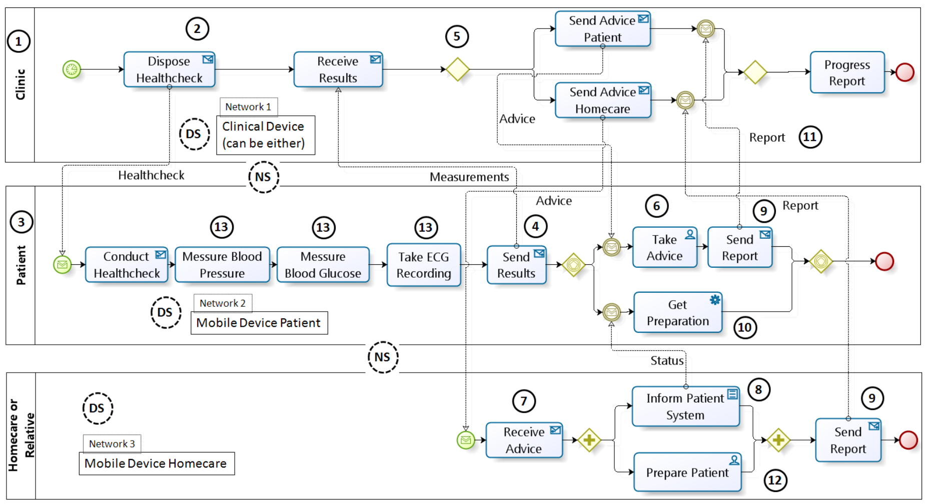

Fig. 2. Healthcare Process with Mobile Patient Support

Let us consider the process ③ to be run on the mobile device of the patient in more detail: While the patient stays at home, he first gets a message through his mobile device. Then he measures and collects the requested data being assisted by the process running on his mobile device. Following this, results ④ are sent back to the clinic which then decides ⑤ about next steps. Ideally no special actions are required. In this case a message is sent back to the patient's mobile device containing information about his medication ⑥. Alternatively, the clinic may send a message with information about further treatment or special treatment to home care ⑦ (either provided by a professional service or a relative of the patient). In the latter case, an additional process fragment is started on the mobile device of the person who is responsible for home care. This process has to be synchronized with the process with the one running on the patient's mobile device. Finally, either the process running on the mobile device of home care or the one running on the mobile device of the patient sends back a report ⑨ to the clinic. Then the process is finished.

Altogether the process fragments of three parties need to be synchronized. Thereby, the runtime infrastructure must be able to cope with communication problems, device failures and so forth. In Fig. 2 the pictograms with label DS and NS indicate a Network Switch or Device Switch within the overall process choreography. Assume that the mobile device of the patient or its connection with the clinic fail. Then the clinic has now knowledge about the status of the patient, but only has the information that the network connection has broken. Such failure scenarios must be covered by the architecture. In particular, the following requirements need to be met by a supporting infrastructure:

– It must be possible to partition a process model and to allocate the resulting fragments on mobile devices as well as stationary computers.
– Soundness of the process (i.e., the process choreography) needs to be ensured.
– The runtime infrastructure has to cope with physical problems like broken connections or malfunctioning devices.
– When running the fragments on distributed devices their execution must be synchronized and messages be exchanged in a reliable way.
– Both the overall process model as well as its fragments might have to be adapted during runtime, e.g. to deal with exceptional situations.
– A mobile process must be able to gather sensoric data during its execution.

3 MARPLE Architecture

In this section we give insights into the MARPLE architecture (cf. Fig. 3). Its two core components are the *MARPLE Mobile Process Engine V1.3* and the *MARPLE Mediation Center*. Here we focus on those parts of the MARPLE architecture that are relevant in the context of our application scenario. Other components and features of MARPLE are only mentioned shortly and will be subject of future publications.

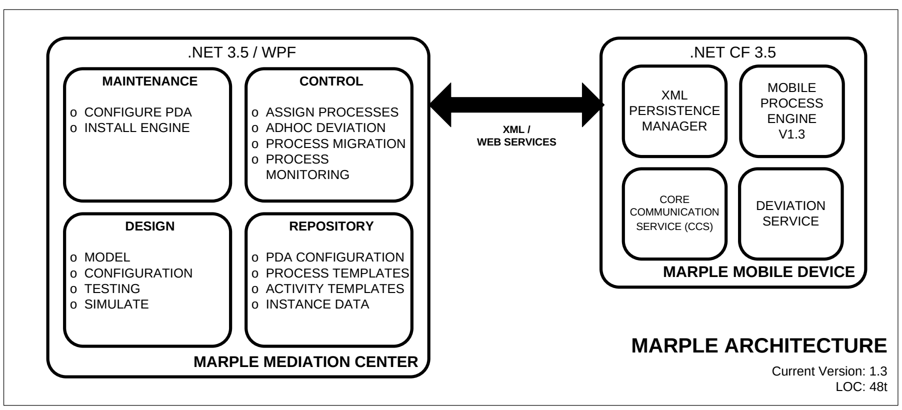

Fig. 3. MARPLE Architecture

If a mobile device shall be added to the MARPLE environment, it first needs to be equipped with the basic software services required in the MARPLE context. Amongst others this includes *Core Communication Services (CCS)* as part of the *MARPLE Mobile Process Engine V1.3*. Thereby, we follow a light-weight approach; i.e., services that are initially not needed are not loaded to the device. Following this, the mobile device can connect to the *MARPLE Mediation Center* and indicate that its configuration may start. When starting the MARPLE configuration procedure on the mobile device through the *MARPLE Mediation Center*, CCS dynamically loads the *MARPLE Mobile Process Engine V1.3*, the *MARPLE XML Persistence Manager*, and the relevant process as well as *Activity Templates* to this device. In this context, *Activity Templates* encapsulate pre-manufactured application components that implement the process steps. In MARPLE, for example, activity templates can be associated with a user forms or a (remote) web service call. Regarding our example from Section 2, in process step ⑦ a report is sent from the patient's mobile process to the clinic. When realizing the *MARPLE Mobile Process Engine V1.3*, we re-use fundamental concepts and design principles of the ADEPT process management technology, which we developed during the last decade [1]. In particular, we adopt the ADEPT process meta model, apply its fundamental correctness notions and correctness checks, and enable flexible process enactment on the mobile device. The latter includes dynamic adaptations of process instances running on the mobile device (e.g., to cope with contextual changes in the environment) and is realized by the *MARPLE Mobile Deviation Service.*

Despite these commonalities with ADEPT it is noteworthy that we provide a complete new implementation of the kernel of the *MARPLE Mobile Process Engine V1.3* in order to meet performance requirements of mobile scenarios and to cope with their specific requirements (e.g., broken connections and limited GUIs). In particular, the implementation framework MARPLE is based on is not the same as the one used in the context of ADEPT. While ADEPT relies on JAVA, our MARPLE architecture is based on *.NET Compact Framework.* The *MARPLE Mediation Center* consists of four major parts. First, its *Maintenance* component allows us to configure mobile devices such that they can be used for mobile process support. Second, the *Control Unit* enables users to assign executable processes to mobile devices, to enact them on the mobile device, to invoke user forms or web services during process execution, and to apply ad-hoc deviations from the pre-modeled process logic.

Another fundamental feature of the MARPLE *Control Unit* is its ability to migrate running process instances from one mobile device to another (e.g., if a patient wants to switch his device). Like ad-hoc changes such process migration can be initiated by the owner of the mobile device as well as by authorized users via the *Control Unit.* Another important component of the *MARPLE Mediation Center* is its *Modeler*. This component adopts basic correctness principles we developed in ADEPT, but provides additional features for partitioning processes and for specifying conceptual models for mobile processes. Consider again our example from Section 2. Using *MARPLE Modeler*, the fragment representing the data collection process (see Lane ③) can be defined. The same holds for the process fragment relating to home care. All meta data (e.g., PDA configurations) needed by the different components of the MARPLE architecture are maintained in the Repository of the *MARPLE Mediation Center*. Fig. 4 exemplarily illustrates the interaction between the *MARPLE Mediation Service* and two mobile devices: Initially, only one mobile device is involved in the interaction. Then a second device is added. Following this, the process instance running on the first mobile device is migrated to the newly introduced one (e.g., due to connection problems with the first device or better technical features of the new one). Note that this migration can be triggered either by the *MARPLE Mediation Center* or by the owners of the two devices. During process executions, the *Control Unit* may suspend, resume, abort and monitor running processes. Further, the *MARPLE Mobile Process Engine V1.3* logs progress of the process using the *Persistence Manager*.

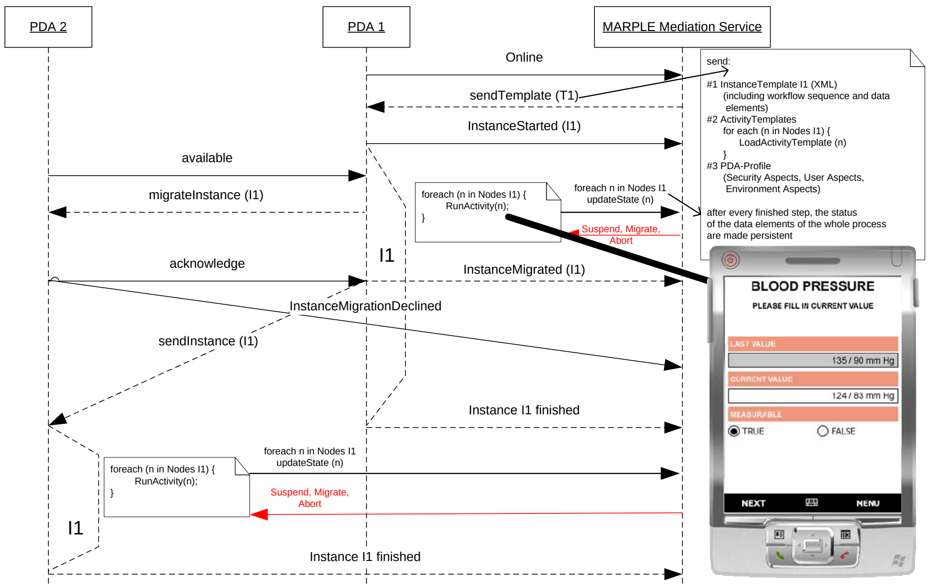

Fig. 4. MARPLE: Interaction Sequence

4 MARPLE-Demonstration

We revisit our scenario from Section 2 and show how it can be realized using MARPLE. Fig. 5 depicts the user interface of the *MARPLE Mediation Center*. With *MARPLE Modeler*, we can completely define the patient-centered data collection process from the middle lane in Fig. 2. Further, MARPLE, enables remote monitoring of process instances; e.g. Fig. 5 ① shows a concrete process instance running on a mobile device as it can be monitored using MARPLE. Note that this perspective displays both the current status of the mobile process and the data values collected during process execution (see ⑦). Obviously, this is exactly the information a medical professional would need when remotely monitoring patient processes. Let us shortly consider how the above mentioned process fragment is modeled in MARPLE. Fig. 5 shows a part of this model together with instance-specific markings. Activity ② is a *receive* activity which is waiting for an incoming message requesting a health check. The subsequent three activities constitute data collection steps which are either implemented as user forms or sensing activities; the blood pressure is gathered via a bluetooth activity template from the linked blood pressure system. Blood glucose and ECG recordings are entered via form-based activities; i.e., the user of the mobile device gets respective requests in his worklist and then has to fill in the two forms (e.g. see the PDA display in Fig. 4). Following data collection, activity ④ is automatically executed. It invokes a web service at the clinic to report about measured results (e.g., to add them to the electronic patient record). Subsequent activity ④ then waits until a message is received either from the clinic or from home care. The toolbar on the left of Fig. 5 (⑧) displays available functions for managing process templates, users, mobile devices and mobile device settings. Further, ⑥ displays the list of currently released process templates, which can be assigned to registered mobile devices. So far, we have focused on the implementation of the *MARPLE Mobile Process Engine V1.3* and on robustness issues emerging with mobile processes.

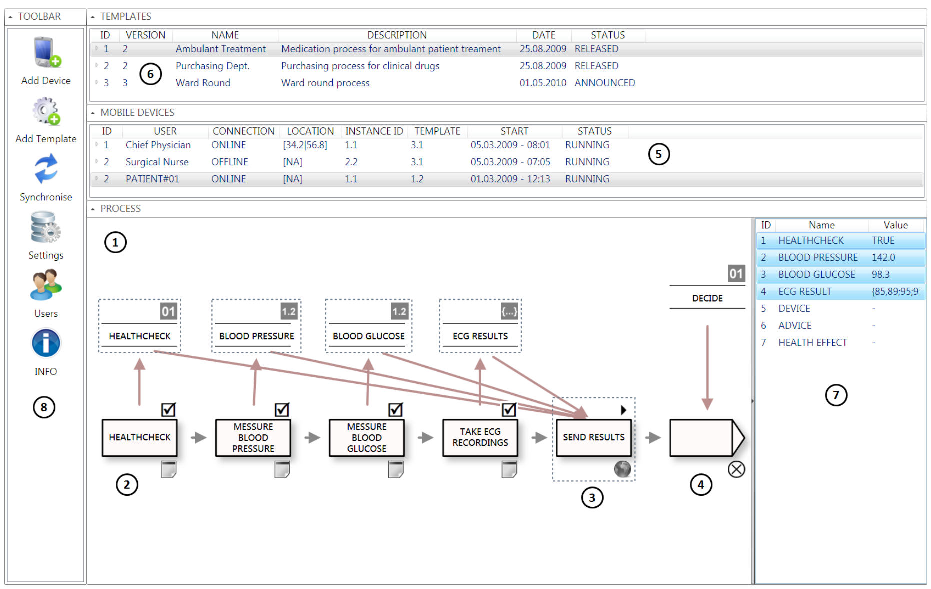

Fig. 5. MARPLE: Mediation Center

5 Related Work

In literature we can find approaches which focus on logical models for mobile processes on the one hand and approaches addressing architectural and implementation issues of light-weight process engines on the other hand. Regarding the first category, for example, [2] deals with the partitioning of BPEL processes. A similar approach has been suggested in the context of ADEPT [3]. However, none of the two approaches has provided an architecture for mobile process support as suggested by MARPLE. Taking mobile network dynamics as core demand for mobile process engines, many approaches deal with failures and exceptions like broken connections or lack of communication facilities [4–7].

Respective tools usually apply web service standards and base process execution on BPEL or more specific execution models derived from BPEL. We consider the use of BPEL as process execution language as too low level, particularly if it shall be possible to dynamically evolve or adapt mobile processes during runtime. Instead we provide a high level process model that can be adapted by both remote users as well as users of the mobile device.

6 Summary and Outlook

We introduced our MARPLE architecture and described how its core components enable the execution and monitoring of processes on mobile devices. Our overall vision is to provide sophisticated mobile process support; i.e., to realize generic process management features including support for process instance changes, instance migrations, etc. To foster this vision we base our work on core design principles and fundamental concepts we developed in our ADEPT project. In future work we will extend the *MARPLE Modeler* such that it provides sophisticated methods for modeling complex process choreographies like the one from Fig. 2. This will include, for example, a methodology for correctly partitioning processes models, for allocating resulting fragments on different machines and devices, and for synchronizing them at runtime. In particular, we will adopt and extend concepts from autonomic computing and self-healing systems to cope with the many failure scenarios in connection with distributed and mobile applications.

References

1. Dadam, P., Reichert, M.: The ADEPT project: A decade of research and development for robust and flexible process support - challenges and achievements. Computer Science - Research and Development **23** (2009) 81–97
2. Baresi, L., Maurino, A., Modafferi, S.: Workflow partitioning in mobile information systems. In: MOBIS'04. (2004) 93–106
3. Bauer, T., Reichert, M., Dadam, P.: Intra-subnet load balancing in distributed workflow management systems. Int'l Journal of Cooperative Information Systems **12** (2003) 205–323
4. Kunze, C.P.: Demac: A distributed environment for mobility-aware computing. In: Proc. 3rd Int. Conf. on Pervasive Computing. (2005) 115–121
5. Hackmann, G., Haitjema, M., Gill, C.: Sliver: A BPEL workflow process execution engine for mobile devices. In: Proc. 4th Int. Conf. on Service Oriented Computing. (2006) 503–508
6. Schmidt, H., Kapitza, R., Hauck, F.J.: Mobile-process-based ubiquitous computing platform: a blueprint. In: Proc. 1st Workshop on Middleware-application interaction. (2007) 25–30
7. Gerhard, S., Jürgen, M., Erich, S.: Building a modular service oriented workflow engine. In: IEEE Int. Conf. on Service-Oriented Computing and Applications. (2009)