

# ROME4EU: A Web Service-Based Process-Aware System for Smart Devices\*

Daniele Battista<sup>2</sup>, Massimiliano de Leoni<sup>1</sup>, Alessio De Gaetanis<sup>2</sup>,  
Massimo Mecella<sup>1</sup>, Alessandro Pezzullo<sup>2</sup>, Alessandro Russo<sup>2</sup>,  
and Costantino Saponaro<sup>2</sup>

<sup>1</sup> Dipartimento di Informatica e Sistemistica  
SAPIENZA - Università di Roma, Rome, Italy  
{deleoni,mecella}@dis.uniroma1.it

<sup>2</sup> Faculty of Computer Engineering  
SAPIENZA - Università di Roma, Rome, Italy

Nowadays, process-aware information systems (PAISs) are widely used for the management of “administrative” processes characterized by clear and well-defined structure. Besides such scenarios, PAISs can be used also in mobile and pervasive scenarios, such as in coordinating operators during emergency situations [1]. In these pervasive settings, due to highly mobility, operators have to be equipped with small devices, such as PDAs, and to communicate through ad-hoc networks.

At the best of our knowledge, all of available PAISs allow currently to execute on smart devices only client applications, such as work-list handlers for accepting/refusing the assigned tasks. The engines at the heart of PAISs, which are in charge of assigning tasks to process participants, are still designed to be executed on standard desktop machines. Therefore, the current PAISs cannot really work in pervasive and highly mobile scenarios where the entire system has to be deployed on the spot and running on smart devices. The possibility of having a remote coordination center where a PAIS engine is running is not really feasible. Assigning tasks to team members and orchestrating the process execution remotely would require to exist an underlying infrastructure reliable and fast. That is not the case in these settings where the best case would be having a GPRS/UMTS connection. Furthermore, the Hurricane Katrina experience has taught us that if every team used such an infrastructure, it would be going to fall down or would become too slow.

In the light of this, we have developed a PAIS, namely ROME4EU (The Roman Orchestration Mobile Engine for Emergency Units)<sup>1</sup>, whose engine resides on the MS Windows Mobile PDA of the team leader. Modern PDAs are becoming increasingly powerful and, hence, able to execute complex applications. Team Members are also equipped with PDAs and their work is coordinated by the PAIS running on the team leader PDA. That makes ROM4EU really applicable in pervasive scenarios.

In ROME4EU, process schemas are defined in the form of Activity Diagrams enriched for describing all the different aspects: definition of tasks in term of

---

\* This work is supported by the European Commission through the FP6-2005-IST-5-034749 project WORKPAD.

<sup>1</sup> <http://www.dis.uniroma1.it/pub/mecella/projects/ROME4EU/>

pre- and post-conditions, the control and data flow, as well as the assignment of tasks to appropriate members. Tasks are associated to a set of conditions to hold in order that they are assignable to participants. Conditions are defined on control and data flow (e.g., a previous task has to be finished, a variable needs to get assigned a specific range of values, etc.). Every task can be only assigned to a certain member that provides certain capabilities. We model that by binding each and every task to a set of capabilities. Moreover, every member declares to furnish certain capabilities. Considering the control and data flow, the ROME4EU engine assigns every task to a certain member providing all capabilities required. At client side, every member uses a task handler to be notified of tasks assignment and to start the proper application for their execution. The same application is also used to perform the log-in phase, where members specify the capabilities they can provide and to show the information coming from the ROME4EU engine about the process to carry on.

ROME4EU overtakes interesting challenges, such as how to concretely design and develop a PAIS running on smart devices connected in mobile networks, considering that this network class provides reduced communication bandwidth and low reliability. Furthermore, smart devices are battery operating and, thus, the engine has to deal with the issue of minimizing the power consumption. It is worthy mentioning that reduced screen sizes limit the amount of information which can be visualized at the same time.

ROME4EU follows an approach comparable to BPEL4People [2] where task-list handlers are exposed as web-service endpoints and seen from the BPEL-based engine viewpoint as mere services to be integrated. Unfortunately, at our knowledge, no BPEL4People implementation is targeted to smart devices so far.

**Technical Solutions.** ROME4EU is completely developed on the .NET Compact Framework. The interaction between the engine and clients is based on web-service invocations. Specifically, we used [3] and extended it to handle complex data types, required for exchanging process variables, and one-way invocations. The latter feature is quite important in (unreliable) Mobile Settings, where it is difficult and battery consuming to keep alive SOAP connections for long times. The engine is based on a porting of the BPEL engine *Sliver* [4] in MS .NET C#, which has been later extended to integrate the aforementioned WS middleware.

## References

1. Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Salvatore, B., Vetere, G., Dustdar, S., Juszczak, L., Manzoor, A., Truong, H.: Pervasive Software Environments for Supporting Disaster Responses. *IEEE Internet Computing* 12, 26–37 (2008)
2. Kloppmann, M., Koenig, D., Leymann, F., Pfau, G., Richayzen, A., von Rigen, C., Schmidt, P., Trickovic, I.: WS-BPEL Extension of People - BPEL4People (July 2005)
3. Nicoloudis, N., Prastitha, D.: .NET Compact Framework Mobile Web Server Architecture (2003) Prompted on June 8th, 2008, <http://msdn2.microsoft.com/en-us/library/aa446537.aspx>
4. Hackmann, G., Haitjema, M., Gill, C., Roman, G.C.: Sliver: A BPEL Workflow Process Execution Engine for Mobile Devices. In: Dan, A., Lamersdorf, W. (eds.) *ICSOC 2006*. LNCS, vol. 4294, pp. 503–508. Springer, Heidelberg (2006)