“YourWay!”: a Platform for Composing and Executing Services Driven by User Resources

Raman Kazhamiakin[1], Piergiorgio Bertoli[1], Massimo Paolucci[2], Marco Pistore[1], and Matthias Wagner[2]

[1] FBK-Irst, via Sommarive 18, 38050, Trento, Italy
[bertoli,raman,pistore,traverso]@fbk.eu
[2] DoCoMo Euro-Labs, Landsberger Strasse 312, 80687 Munich, Germany
[paolucci,wagner]@docomolab-euro.com

1 Introduction

Recent technology advances make mobile phones more and more important tools for their users. They facilitate everyday user activities providing an access to a continuously growing set of services and functionalities, such as travelling, societal communications, payment and personal assistance. While the variety of these services increases, they are narrowly directed to operate particular pieces of information, and the user is left alone with the problem of their combination and integrated use. In particular, the user has to deal with different formats and representations, to integrate one service with another according to certain intentions, and to transfer data across them. Moreover, the management of the user information by these services should be continuously consistent. These requirements make user-centric composition of mobile services and applications extremely challenging problem.

The key issue here is to find the concepts that are sufficient for expressing both the service capabilities and the user intentions. In our approach these concepts correspond to the core user assets, or “resources”: time, location, money, and social context. Many relevant services and applications are already associated with these resources (e.g., calendars for time, travel booking and map services for location, payment systems for money, or telco services for social context). Analogously, many critical requirements may be expressed in terms of resources (e.g., impossible to participate to different event or to be in different locations at the same time, respect limits on expenses, respect time constraint of other people). This provides a basis for automated and transparent integration of heterogeneous information and composition of various services according to user constraints and intentions.

Here we present a prototype implementation of a platform for user-centric composition of mobile services that relies on the concept of resources. The goal of the platform is to provide an automated support to the user in organizing and managing various resource-related data. While the user engages various services and decides which actions to perform in reaction to resource-critical events, the platform associates these events and actions with resources and resource updates, keeps the resources aligned with the user constraints, and reacts to their violations by composing and presenting services to the user. While the prototype is very preliminary, it demonstrates the applicability of the resource-driven composition approach on a set of realistic scenarios.

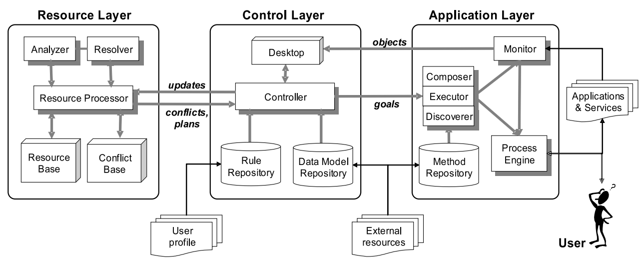


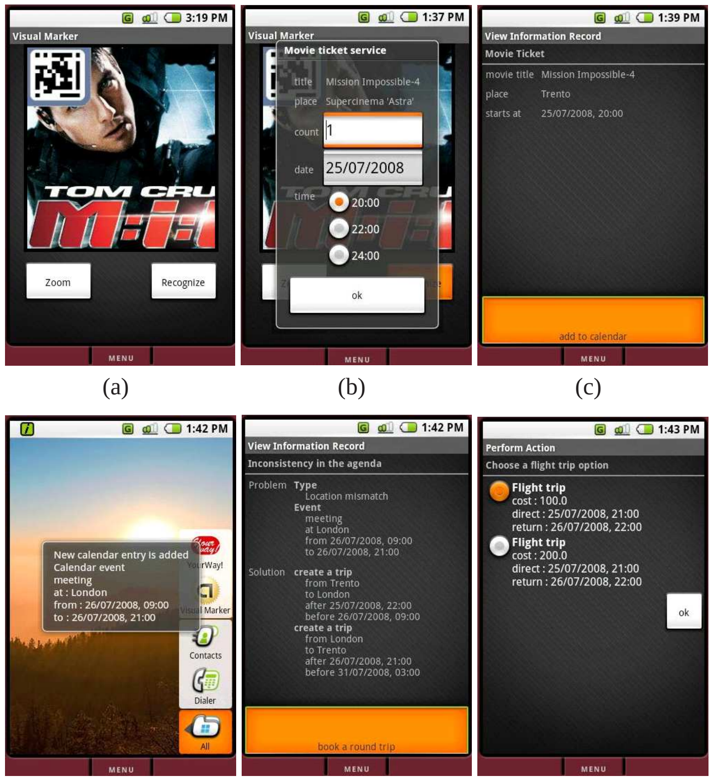
Fig. 1. Conceptual architecture

Architecture. Platform prototype implementation follows the conceptual architecture represented in Fig. 1. Application Layer is responsible for interactions with the user (via user interface) and with the external world (services and context). Relevant information, obtained as a result of service calls or observed by dedicated monitors (e.g., for calendars, message boxes, or user context), is passed to the Control Layer. The latter is used to classify the information with respect to resources. Whenever the information is related to the user resources (e.g., payment confirmation received or calendar entry is created) the corresponding update is propagated to the Resource Layer. The state of resources are contunously checked for consistence; violation of consistency rules (e.g., two user appointments overlap) are signalled to Control Layer. At this phase Control Layer associates the resolution strategy to the conflicts using a predefined set of rules. The strategy dictates the set of activities to be engaged in order to resolve the problem (i.e., move an appointment, cancel payment, etc.). In order to organize and coordinate these activities, the strategy is passed to Application Layer, where the corresponding services are composed and executed under the control of the user.

Implementation. The prototype implementation is designed as a distributed software platform, in which some components are located on a mobile phone, and the others (that require expensive computations and persistence) are accessed remotely.

The mobile part of the platform is implemented on top of the Android platform [1], an open java-based operating environment for mobile phones. Mobile part of the platform includes the UI components necessary to interact with the user (e.g., notifications, requests for decisions or additional data), back-end components needed to synchronize with the platform, and the monitors attached directly to the mobile phone (e.g., context monitors). Apart from the platform components, the implementation makes an extensive use of remote services and applications. In particular, the Google Calendar application is used in order to manage the user agenda, and the Google Mail application is used to monitor the e-mail messages and notifications from other users. Besides, a set of mock-up Web services and applications were implemented for testing/demonstration purposes, such as payment service, flight search, etc.

Demo. Figure 2 shows some screenshots of the platform interface across the demo scenario. Interacting with an interactive poster (making a photo, Fig. 2(a)), the user decides to book a movie ticket. The visual marker [2] of the movie is recognized, the corresponding service is engaged (Fig. 2(b)), the movie ticket is detected by the plat-



(a) (b) (c)

(d) (e) (f)

Fig. 2. Platform UI

form. The platform associates it with the “time” resource, and suggests the user to add it to the calendar (Fig. 2(c)).

Later on, a business meeting is added to the user agenda. The calendar monitor detects the new event (Fig. 2(d)), propagates it to the resource layer and detects the “location conflict”, i.e., the problem that a meeting takes place in London, while the user is expected to be in Trento. The description of the conflict, as well as the plan for the resolution (i.e., planning a travel) is shown to the user (Fig. 2(e)). The platform also suggests a possible composed procedure for booking a round trip to London. The execution of the procedure involves searching and selecting a flight (Fig. 2(f)), booking it, and updating the calendar with the travelling information.

References

1. Open Handset Alliance: Android Mobile Phone Platform.
2. ISO/IEC 18004:2006 Standard. QR Code 2005 Bar Code Symbology Specification.