Supporting Mobile Context-Aware Applications through a Modular Service Infrastructure *

Cristiano di Flora[1], Oriana Riva[2], Kimmo Raatikainen[2], and Stefano Russo[1]

[1]Department of Computer Science
Federico II University of Naples
Via Claudio 21
80125 – Napoli (Italy)
{diflora, sterusso}@unina.it

[2]Department of Computer Science
University of Helsinki
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FIN-00014 – University of Helsinki (Finland)
{riva, kraatika}@cs.helsinki.fi

ABSTRACT

This work presents a modular service infrastructure to support context-awareness in nomadic computing. The proposed approach aims to ease the tasks of (i) deploying, (ii) maintaining, and (iii) extending mobile context-aware systems. The infrastructure allows applications to retrieve and share well-known context attributes by means of ontology-based context-provider components. Moreover, the infrastructure provides applications with a context-change notification service, which is specially designed to cope with user mobility. We describe potential benefits of our infrastructure, and our future research directions.

RATIONALE

Context-aware applications require service infrastructures capable of providing uniform context abstractions [1]. Plenty of software infrastructures have been proposed to provide applications with context information. The common approach is to extract context data by means of a retrieval-representation strategy, and to forward it to inferring engines that deduce high-level context by reasoning on low-level context atoms. Such context information is subsequently fed to rule-based systems that trigger application adaptation (see Figure 1). A major problem is that these strategies are mostly tailored to a specific application scenario (e.g., smart environments, location-aware applications). Hence, they do not provide reusable mechanisms to support the deployment of context-aware applications. Moreover, the requirements of context-aware systems can vary in terms of both semantics (e.g. context ontology) and retrieval strategies; thus it is difficult to share context information among diverse context-aware applications in order to make them collaborate. As shown by the graph abstraction proposed by Chen and Kotz [1], flexibility, scalability, context sharing, and event notification are the most desirable properties of a context infrastructure. Furthermore, as far

LEAVE BLANK THE LAST 2.5 cm (1") OF THE LEFT COLUMN ON THE FIRST PAGE FOR THE COPYRIGHT NOTICE.

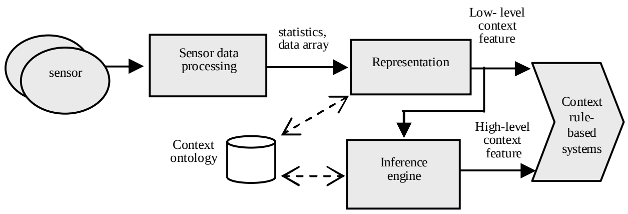

Figure 1. Context information recognition and usage

as nomadic context-aware applications are concerned, current context infrastructures do not provide special support for user mobility management.

We argue that time is ripe for an effective context infrastructure to *i)* deploy, *ii)* maintain, and *iii)* extend context-aware mobile systems. The rest of this work proposes a uniform service infrastructure consisting of extensible and reusable context-provider components to deploy context awareness on mobile devices. A modular approach drives the design of such a context infrastructure for the support of specific application requirements; modularity, flexibility, reuse, and mobility support make our solution particularly suitable for nomadic users.

THE MODULAR INFRASTRUCTURE

The conceptual model of the proposed infrastructure is depicted in Figure 2. Our driving idea is to provide a specialized and transparent support for retrieving well-known context-attributes. Applications can further exploit these attributes to build more complex context-models, or they can eventually retrieve low-level sensor data in order to represent application-specific context information.

The overall architecture is based on the factory design pattern. ***Context-provider*** components are in charge of producing well-defined "pieces of context" and offering multiple models of representation. Applications exploit the ***Context-factory*** to instantiate the required context-providers, and thus to interoperate with them. For instance, a journey planner application may require all the available context-providers to be instantiated, whereas an energy-aware video player [3] may only need to monitor energy supply through the device status provider. ***Context-provider*** components can produce either low-level context

* This work has been partially supported by the Italian Ministry for Education, University and Research (MIUR), within the framework of the FIRB project WEB-MINDS, and by the Regione Campania, within the framework of the ``Centro di Competenza Regionale ICT'' project

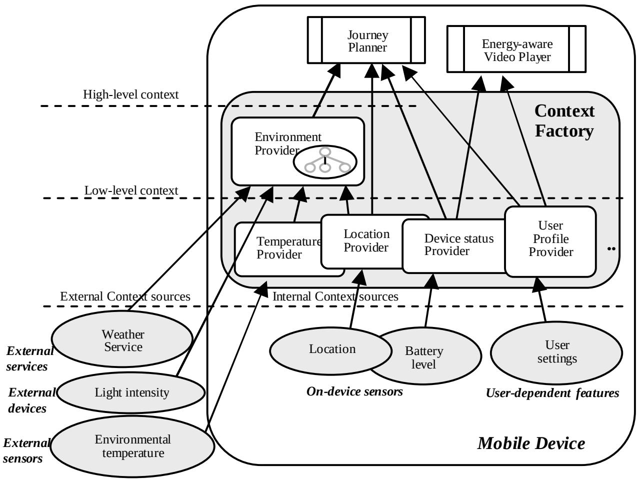

Figure 2. The service infrastructure

features (e.g., temperature, location, device status) or high-level context features (e.g., environment). Such a context recognition may be accomplished through the following processes: *a)* implementing low-level context recognition, which means to bind sensor data values to low-level context features (e.g. location, temperature) according to a predefined context ontology; *b)* implementing high-level context recognition (e.g. environment) by combining several context atoms through Bayesian networks; *c)* retrieving context information from other context-providers (e.g., on other devices); and *d)* retrieving context data from available services (e.g., weather service). Indeed, context-providers can exploit various context-sources, such as on-device sensors, user preferences, external sensors, external context providers, and external services as well.

INFRASTRUCTURE PROPERTIES

Modular deployment: The infrastructure glues several context-provider components, making them interoperate. Hence, the overall infrastructure has to guarantee system interoperability and context sharing. By adding or removing context-provider components, each device can be adapted to specific application requirements.

User mobility support: The infrastructure provides an event service that can be utilized to detect context-changes. We adopt a rendezvous mechanism [4], which is specially designed to cope with user mobility, as it supports bounded delivery times and disconnected operations.

Maintenance and updates: Context profiles can be dynamically replaced or added according to application requirements without affecting the behavior of any others. For instance, if a location-provider component uses a certain algorithm to process location data and a more efficient algorithm becomes available, the component is easily updated and plugged into the service infrastructure.

Extensibility: Modularity also makes the system easy to extend through the introduction of new sensors, new devices, and new services. This feature is crucial due to the dynamic nature of context and in particular of user preferences. It is very likely that the user itself cannot definitively specify from the beginning all its needs and rather prefers to modify them at run time. New context-providers can be added to the infrastructure, and eventually combined in complex hierarchies to support new context features, and offer new services as well.

Collaboration and context-sharing: The infrastructure enables collaboration and context sharing among context-provider components and context-aware applications. The major benefits are that device capabilities can be significantly enriched by additional context-sources, and the context representation can be enhanced. Furthermore, single devices can reduce processing and storage efforts, and they can better cope with context uncertainty by exploiting additional context data retrieved from neighboring devices [5]. Sharing of context information must be in compliance with the user permission profile.

CONCLUSIONS AND FUTURE WORK

This work presented a service infrastructure based on the modular composition of context providers. The proposed infrastructure allows to ease system maintainability and extensibility, and to support collaboration and context sharing among diverse mobile applications. Indeed, the system can meet specific application requirements by adding only the necessary context-provider components. Further work is required to implement and evaluate the infrastructure on several mobile devices, including PDAs, smart phones, and laptops, and also to formalize the context model. Specifically, work will be undertaken to supporting substantial case-study applications, and to evaluate infrastructure behavior and performance through massive experimental campaigns on the system prototype.

REFERENCES

1. Hong, I., and Landay, J. A. An Infrastructure approach to context-aware computing. *Cont Aw Comput-HCI J* 16:287-303. 2001.

2. Chen, G., and Kotz, D. Context Aggregation and Dissemination in Ubiquitous Computing Systems. In *Proc. of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, pp. 105-114, June, 2002. IEEE Computer Society Press.

3. Flinn, J., and Satyanarayanan, M. Managing battery lifetime with energy-aware adaptation. *ACM Transactions on Computer Systems,* 22(2): 137-179. May, 2004.

4. Tarkoma, S., Kangasharju, J., and Raatikainen, K. Client Mobility in Rendezvous-Notify, in *Proc. of International Workshop for Distributed Event-Based Systems (DEBS03),* pp. 1-8 , 2003.

5. Mäntyjärvi, J., Himberg, J., and Huuskonen, P. Collaborative Context Recognition for Handheld Devices, in *Proc. of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003),* pp. 161-168, Dallas-Fort Worth, Texas, USA, 2003.