

A Webservice-based Context-Data Collection Framework for the Android Platform

Dirk Bade, Roman Puszies

Distributed Systems Working Group, University of Hamburg, Germany

Abstract

Today, services aiming at mobile users need to take the user's current context into account in order to provide the right information, to the right time at the right place. For this purpose, service providers require information about the current state of the user and her environment in order to adapt the service to the user's current needs. Providing the location [1] is among the most important requirements for location- or more generally speaking context-based services, but there is more to context than location [2]. Nowadays, mobile phones and other devices are equipped with a multitude of different sensors, whose observations could help adapting a service even better. Among these are sensors measuring attributes of the physical environment (e.g. temperature, current acceleration, luminance, noise, battery state, etc.) as well as virtual sensors providing e.g. usage statistics or performance indicators like available memory, CPU load, etc.

Nearly all context data is collected on the mobile device itself and has to be explicitly made available for service providers (except location information, which is also available from the carrier) [3]. In this respect, two modes of context data transfer between the device and the service provider must be distinguished: i) the device pushing context information to the service provider, and ii) the service provider pulling information from the device. Moreover, not only a single push or pull may be required by the service provider, but continuous access to context information. These and several other requirements have already been formulated by the Open Geospatial Consortium, which proposes to not only allow sensors to be queried but also be tasked [4]. This way, a context data consumer (e.g. a service provider) might task a remote sensor (e.g. on a mobile phone) to behave according to the consumer's requirements, for example by shortening measurement intervals, change encodings, etc.

Several projects, especially in the area of sensor networks, already presented concepts and solutions for the stated requirements (e.g. [5,6]). We adopted several ideas in order to design a sensor middleware layer for the *Android*-platform [7]. Android devices in general are not only equipped with several physical and virtual sensors, but the Android application programming interface allows to access nearly all of them (in contrast to other mobile phones, e.g. by Nokia). Unfortunately, Android does not specify a generic interface to locally access the sensors and no interface for remote access at all; hence the functionality has to be implemented in each context-based Android application individually.

To allow easy access to Android sensors for all applications/services (locally and remotely) we therefore propose an extensible framework, which allows to query and task local Android sensors via a generic interface. Local applications access the sensors directly by using the sensor framework API, whereas remote applications connect to the sensors via a dynamically generated Webservice interface, which is published by using WS-Inspection [8]. This way, context information can be pushed towards remote services as well as be pulled by these. Even the remote tasking of sensors is possible.

In conclusion, our sensor middleware for the Android platform shall ease the interconnection of context-based service providers and consumers as well as unburden context-based application developers by offering transparent access to sensor data, e.g. location information.

References

- [1] A. Savvides; F. Koushanfar; A. Boulis et. al.: Location discovery in ad hoc wireless networks; Networked and Embedded Systems Laboratory, UCLA; June 2000
- [2] Schmidt, A.; Beigl, M. & W. Gellersen, H.: There is more to context than location; Computers and Graphics, 1999, 23, 893-901
- [3] H. Mosemann; M. Kose: Android. Anwendungen für das Handy-Betriebssystem erfolgreich programmieren; Hanser Fachbuch, Munich, Germany; Juni 2009
- [4] Open Geospatial Consortium, Inc.: OpenGIS Sensor Planning Service (SPS); <http://www.opengeospatial.org/standards/sps>; July 2010
- [5] M. Hillenbrand; P. Müller; T. König: XML-basierte Middleware für mobile Sensoren; Berliner XML Tage 2003; Germany, 2003
- [6] J. L. Hill: System Architecture for Wireless Sensor Networks; University of California, Berkeley, USA; 2003
- [7] Android Homepage by Google Inc., California, USA; <http://www.android.com>; July 2010
- [8] IBM; Microsoft: Web Services Inspection Language; from January 2007; <http://www.ibm.com/developerworks/library/specification/ws-wsilspec>; July 2010