Detecting Place in Space

Tom Gross
Faculty of Media, Bauhaus-University Weimar, Germany
(tom.gross(at)medien.uni-weimar.de)

Abstract. In order to adequately support interaction between users and technology as well as among users, systems need information on the current context of the users. We have designed and developed Sens-ation—a platform capturing the rich context and interaction between users and technology as well as among users. This position paper provides a brief overview of the Sens-ation platform.

1 Introduction

In order to adequately support interaction between users and technology as well as among users, systems need information on the current context of the users. Ideally, users to not have to enter this information, but rather the systems capture it automatically—as implicit input [Dey & Mankoff 2005]. Thereby, simply capturing users' input to computers or users' position in and manipulation of artefacts in the real world is not enough [Schmidt *et al.* 1999]. The system needs to find out what is really going on at a site and how the reality looks and feels like from the users perspective. In terms of Harrison & Dourish [1996, p. 67]—'space is the opportunity; place is the understood reality'—the system has to understand place, rather than only space.

In mobile and ubiquitous computing context-aware systems capture information about the environment of a user in order to adapt accordingly. For instance, a mobile personal digital assistant can scan its environment for printers, and inform its user about printing opportunities in the vicinity. Yet, these systems mostly focus on the interaction of single users interacting with their personal environment [Schmidt *et al.* 2004].

We have designed and developed Sens-ation, an open and generic service-oriented platform, which provides powerful, yet easy-to-use, tools to software developers of sensor-based infrastructures. Sens-ation provides the base functionality for the iterative design, implementation, and evaluation of sensor-based infrastructures that are needed to build reactive cooperative environments. This position paper provides a brief overview of the Sens-ation platform.

2 Detecting Place in Space with Sens-ation

For capturing the rich interaction between users and technology as well as among users, developers need platforms that meet the following requirements:

- Flexible mechanisms for publishing, validating, and invoking information about sensors, sensor values, as well as locations
- Convenient integration of new sensors independent of a specific connection interface (both hardware and software sensors)
- Loose, and on-demand coupling of components
- Platform-independent servers
- Persist layers to enable access to past sensor events
- Add-on inference engines to process event information (e.g., aggregation or interpretation methods of sensor values)
- Flexible query mechanism amongst servers
- Platform-independent and ease-to-use interface for other systems and infrastructures (for both rich-clients such as desktop applications, and thin-clients such as mobile applications)

In order to meet these requirements the Sens-ation platform provides an interoperable service-oriented sensor platform, which supports access, discovery, and use of real-time data obtained directly from sensors over the wired or wireless networks. The service-oriented paradigm [Singh & Huhns 2005] of Sens-ation based on service providers, service consumers, and brokers enables standardised communication within the platform, and between the platform and the sensors. Each Sens-ation server can act as a Web service provider. These service providers allow encapsulating and hiding of all specific hardware implementation details of their attached sensors. They provide a simple common interface for other application to obtain real-time sensor data, or persistently stored past sensor data. The service consumers are independent of the Sens-ation service provider so that a service consumer does not depend on the implementation of the service and communicates with it according to a well-defined interface. A Sens-ation broker contains information about Sens-ation service providers such as their registered sensors and their location. A service consumer can discover available sensors and their contact information via a broker. The service consumer can then directly request required sensor data from that service.

Figure 1 shows the conceptual overview of a Sens-ation server acting as a service provider with the main layers for communication and sensor value processing. The basic flow of information in the Sens-ation platform is the following: various sensors capture data and send them to the server via adapters; the handling layer manages registered sensors and the persistence layer stores the data. In the processing layer we can use the inference engines to process the raw sensor data (e.g., calculating average values). Finally, the clients can retrieve data from the server via various gateways.

Sens-ation supports hardware and software sensors as well as actuators. *Hardware sensors* deliver sensor values from the real world (e.g., the temperature or light intensity). *Software sensors* capture information from the electronic world (e.g., the presence information of users of an instant messaging system). With the actuators we also distinguish between hardware and software modules: while the *hardware actuators* can affect the real world environment (e.g., activate light bulbs, play audio messages), the *software actuators* use the graphical user interface of computers for notification (e.g.,

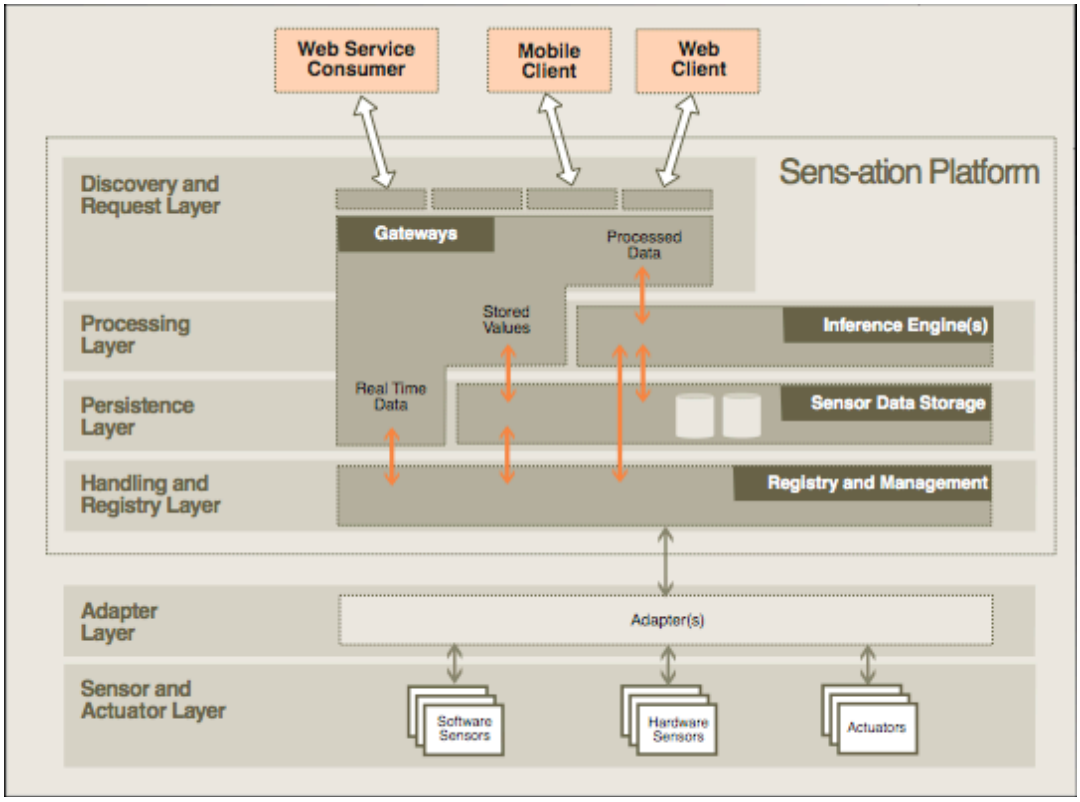

Figure 1. Layer structure of a Sens-ation service provider.

RSS feed or an instant messenger notification). Sensor *adapters* facilitate the communication between the sensors and the server, and abstract from the individual communication interfaces of the sensors, which can be highly specific. The adapters provide a push and pull method.

The *handling and registry layer* is responsible for the management of sensors, locations, and sensor types. Furthermore, this layer includes various discovery methods for all registered sensors and locations. There is a set of lookup methods that can be accessed via the gateways (e.g. to request all sensors nearby a specified location, or to request all registered sensors of a specific type). The handling layer passes the received sensor values further to the persistence layer.

The *persistence layer* stores the sensor data and allows the retrieval of historic sensor values.

The *processing layer* provides inference engines to interpret and aggregate sensor values from single sensors. These engines can also combine and infer on values of different sensors, or combine and infer on values over time. For instance, if the average temperature is needed, a inference module can gather the values of all registered temperature sensors, and calculate the average temperature; or, if an overview of the movement in various areas is needed, a module can observe a collection of movement sensors and generates an event if one of the sensors detects movement above a specified threshold.

At the *discovery and request layer*, gateways allow the query of current or past sensor events for a variety of clients. The gateways pass the incoming client requests to the responsible layers of the server (e.g., the handling layer for real-time sensor data or the

persistence layer for stored value sin the past). Gateways provide functions for requesting real-time sensor values; discovering locations and sensors; subscribing to sensor events; and publishing sensor events. Furthermore, the clients can also request values from the persistence layer and access the active service modules for data processing or request their current values

3 Conclusions

On a whole, Sens-ation aims to make the development of ubiquitous computing and computer-supported cooperative work infrastructures rapid and easy, thereby allowing developers to concentrate on the semantics of their infrastructures, and to develop innovative concepts and implementations of context-aware systems.

Yet, several issues remain open. In particular, the algorithms and heuristics for detecting places with the inference engines have yet to be conceptualised and implemented. In this workshop I would particularly be interested in discussing issues of better understanding of users' interaction with their electronic and physical environment.

Tom Gross is associate professor for Computer-Supported Cooperative Work and head of the Cooperative Media Lab at the Faculty of Media of the Bauhaus-University Weimar, Germany. His research interests include Computer-Supported Cooperative Work, Human-Computer Interaction, and Ubiquitous Computing. He recently co-organised a special issue on 'Context-Aware Computing in CSCW' for the Kluwer/Springer Journal on Collaborative Computing (forthcoming). From 1999 to 2003 he was a senior researcher at the Fraunhofer Institute for Applied Information Technology FIT in St. Augustin, Germany. He holds a diploma and a doctorate degree in Applied Computer Science from the Johannes Kepler University Linz, Austria.

Acknowledgements

I would like to thank Christoph Oemig and Tareg Egla for many inspiring discussions, and for co-organising student projects at the Cooperative Media Lab.

References

Dey, A.K. and Mankoff, J. Designing Mediation for Context-Aware Applications. ACM Transactions on Computer-Human Interaction 12, 1 (Mar. 2005). pp. 53-80.

Harrison, S. and Dourish, P. Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems. In Proceedings of the ACM 1996 Conference on Computer-Supported Cooperative Work - CSCW'96 (Nov. 16-20, Boston, MA). ACM, N.Y., 1996. pp. 67-76.

Schmidt, A., Beigl, M. and Gellersen, H.-W. There is more to Context than Location. Computer & Graphics Journal 23, 6 (Dec. 1999). pp. 893-902.

Schmidt, A., Gross, T. and Billinghurst, M. Introduction to special issue on Context-Aware Computing in CSCW. Computer Supported Cooperative Work: The Journal of Collaborative Computing 13, 3-4 (Aug. 2004). pp. 221-222.

Singh, M.P. and Huhns, M.N. Service-Oriented Computing: Semantics, Processes, Agents. Wiley, N.Y., 2005.