Minos: A Generic Tool for Sensor Data Acquisition and Storage

Silvia Santini
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
santinis@inf.ethz.ch

Daniel Rauch
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
darauch@ethz.ch

***Abstract*—Systems providing comprehensive software infrastructures and tools for acquisition, storage and visualization of data produced by distributed sensors begun to appear in the last few years. To face the issues and challenges related with the realization of such infrastructures, we started *Desthino*, a project that aims at providing an easy-to-use framework for collecting, storing, managing and visualizing sensor data. As a first step towards the realization of this project, we developed *Minos*, a Java-based tool that allows to easily collect and store data generated in wireless sensor networks. The data is parsed according to a generic schema and archived at a central repository. External users can access the collected data in real-time or at at a later stage through a standardized interface.**

I. Motivation

The availability of small and cheap wireless sensing devices increased significantly in the past few years and large-scale real-world sensor networks begin to appear. Such a large number of sensors deployed in the real-world allow for accurately monitoring a variety of physical phenomena, like weather conditions (temperature, humidity, atmospheric pressure ...), traffic levels on highways or rooms occupancy in public buildings. Making the thereby resulting amount of sensor data available through a common web interface opens several interesting application scenarios. Users can query the available sensor data in real-time and use the query results to perform decisions or any kind of monitoring tasks. Since sensor data typically inherently relates to the specific sensor location, geo-based web interfaces like Google Maps or Windows Live Maps appear particularly suited to support real-world sensor querying. Systems providing comprehensive software infrastructures and tools for acquisition, storage, processing and visualization of data produced by distributed sensors begun to appear in the last few years. IrisNet [1] (Internet-scale Resource-Intensive Sensor Network Services) is probably the first effort that strived at realizing the vision a ”world-wide sensor web”, trough which a multitude of sensors can be made openly accessible to a multitude of users. Enabling the sharing of sensor data over a common platform is also the goal pursued by the SensorBase.org project [2]. SensorBase.org offers a centralized data storage and management system, which provides a uniform and consistent method to *slog*[1] sensor network data. Instead of offering a central repository, the Global Sensor Networks Project (GSN) aims at providing a general-purpose infrastructure to facilitate the programming and deployment of sensor networks [3], [4]. The GSN middleware infrastructure is designed to integrate heterogeneous sensor networks and is deployable on any computer interested in interacting with one or more sensor networks. A further effort for providing a generic platform to share, query and visualize sensor data is embodied by Microsoft’s SenseWeb project [5]. SenseWeb provides a common platform and a set of tools not only for data owners to easily publish their data but also for users to make queries over already registered and active data sources. SensorMap is the current SenseWeb portal through which users can query and visualize data sources on a geo-based web interface.

Considering the growing interest in the issue of collecting and sharing sensor data on the web, we started the Desthino (Distributed Embedded Things Online) project, which aims at providing a practical set of flexible software tools that can be used to easily collect and store sensor data gathered from heterogeneous distributed sensors. A further goal of Desthino is to enable data queries and offer data visualization through an adequate web-interface. As first steps towards the realization of this project, we analyzed the characteristics of sensor specific data and summarized them in a simple data model, able to coherently represent heterogeneous sensor data. We then designed and implemented a Java-based software tool for data acquisition and storage, which we dubbed Minos (Message Identification and Online Storage), that can capture sensor data from wireless sensor networks and store it in a central repository according to the defined data model.

II. Data model

The first step towards the realization of a storage and management system able to deal with data collected by heterogeneous sensor networks consists in defining an appropriate data model. To this scope, we organized the relevant information on sensor data retrieved from wireless sensor networks in a schema based on six main entities, as shown in Figure 1. The *Deployment* entity contains all information about a specific network deployment, e.g., a owner and a deployment description. The *Node* entity aggregates all information about

[1] The term ”slog” is a combination of ”sensor” and ”log” reflecting the spirit of sharing information in a blog.

the sensor node actual collecting the sensor data, like the specific *NodeType*. The *SensorReading* entity is used to store the actual sensor readings. Within Desthino, we assume every sensor reading to be the output of a sensor of a known *SensorType*. Furthermore, we assume we can logically assign a sensor reading to the specific node in the deployment that actually collected it. The number of values stored for each reading depends on the specific *SensorType*. For instance, a temperature sensor typically outputs a single value while a GPS sensor reading consists of two values, the latitude and the longitude. This example makes also clear that we treat the position information as the reading of a correspondent sensor "attached" to a particular sensor node.

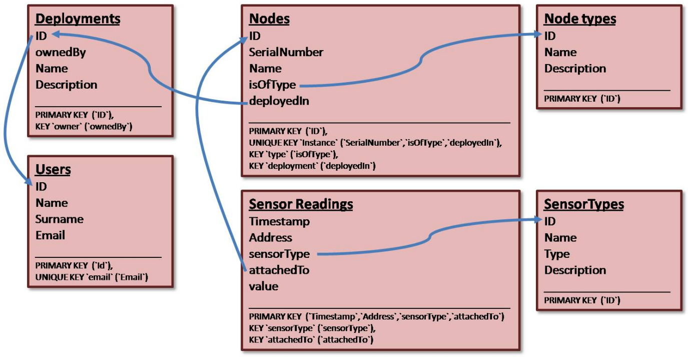

Fig. 1. The Desthino data model

III. Minos

Minos is a generic, Java-based tool that allows for collecting and storing data collected in wireless sensor networks. The main goal in the design of Minos was to provide an easy-to-use tool for users or researchers willing to capture and store data coming from one or more heterogeneously equipped and programmed networks [6]. Once configured, Minos listens from incoming messages, parses the therewith delivered data and stores it in the Desthino data repository according to the above described schema. In this way, data coming from different deployments can easily be stored in an homogeneous fashion, allowing external application to access the collected data through a single, simple interface. Minos's modularity makes it possible to easily add, modify or replace system components and thus to continuously extend or enhance the functionalities of the tool. Currently, the tool can interact with any sensor platform running the TinyOS operating system, but can be easily extended to support other kind of data producers. Several Minos instances running on the same or different machines can concurrently store data on the database through a dedicated connection. Minos's configuration can be carried out through a simple graphical interface, which has been realized using the SWT/JFace Toolkit. The communication with the data repository, a mySQL database specifically configured to support the Desthino data model, occurs in the background and has been implemented using the standard JDBC API.

To run Minos, the user must indicate the message types the tool should listen to. In the current implementation, the user must provide the tool with the Java class representing the incoming messages. Such Java classes can be generated from the corresponding nesC message definition files using the MIG (Message Generating Interface) tinyOS utility. Minos operates similarly to the MSRSense tool [7], developed in the context of the SenseWeb project, but has two main advantages. First, Minos can handle any custom-defined messages coming from a tinyOS-based sensor network, while MSRSense can only interpret messages formatted according to the OscopeMsg format (i.e., messages associated with the standard tinyOS Oscope application). Second, Minos is a lightweight, portable Java application which is not bounded to a specific operating system or programming environment. MSRSense, on the contrary, requires the .NET framework to be installed to work properly.

IV. Future Work

Besides refinements and extensions of the Minos tool, future work includes the development of a web platform, based on the REST paradigm and developed using the Ruby on Rails framework, which will provide an interface for accessing, searching and visualizing the data collected and stored using Minos. To this scope, more efficient data storage mechanisms able to support frequent data updates and queries will be investigated.

V. Acknowledgments

The authors are indebted to Benedikt Ostermaier, who co-initiated the Desthino project and provided valuable feedback for the definition of the data model presented in this paper. This work has been partially supported by the Swiss National Science Foundation (NCCR-MICS, grant number 5005-67322).

References

[1] P. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan, "Irisnet: An architecture for a worldwide sensor web," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 22–33, October-December 2003.

[2] K. Chang, N. Yau, M. Hansen, and D. Estrin, "SensorBase.org: A Centralized Repository to slog Sensor Network Data," in *Proceedings of the Euro-American Workshop on Middleware for Sensor Networks at the 2nd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS/EAWMS'06)*, San Francisco (CA), USA, June 2006.

[3] K. Aberer, M. Hauswirth, and A. Salehi, "Global sensor networks," Ecole Polytechnique Fdrale de Lausanne (EPFL), Tech. Rep. LSIR-REPORT-2006-001, 2006.

[4] G. M. for Sensor Networks, "Project website," http://gsn.sourceforge.net/.

[5] A. Santanche, S. Nath, J. Liu, B. Priyantha, and F. Zhao, "Senseweb: Browsing the physical world in real time," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, Nashville (TN), USA, April 2006.

[6] D. Rauch, "Towards the Sensor Web: A Framework for Acquistion, Storage and Visualization of Wireless Sensor Networks Data," Master's thesis, Distributed Systems Group, Department of Computer Science, ETH Zurich, Zurich, Switzerland, February 2008.

[7] A. Woo, S. Seth, T. Olson, J. Liu, and F. Zhao, "A Spreadsheet Approach to Programming and Managing Sensor Networks," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN/SPOT '06)*. New York, NY, USA: ACM, 2006, pp. 424–431.