Graphically Geo-Coding of Sensor System Information

Christian Decker, Till Riedel, Philipp Scholl, Albert Krohn
Telecooperation Office (TecO)
University of Karlsruhe
Karlsruhe, Germany
cdecker, riedel, scholl, krohn}@ teco.edu

Michael Beigl
Distributed and Ubiquitous Systems (DUS)
TU Braunschweig
Braunschweig, Germany
beigl@ibr.cs.tu-bs.de

***Abstract*— Sensor network information should be presented to the user in a convenient way. For administration, inspection and access, this motivates a location-based interface. We implemented a GoogleMaps interface to wireless sensor networks, which we use in the CoBIs research project and explore the capabilities of such an interface. CoBIs focuses on the detection of hazardous situations in industrial areas, such as chemical plants. Hazards may occur due to misplacing of reactive chemicals. As a consequence, location and proximity information is a highly valuable source for administration of such areas. This paper reports on architecture and technology used and gives insight into the operation of the system and initial experiences within the CoBIs project. The graphical user interface made use of web technologies, esp. RSS, to provide the activation of sensor information by using logic on the map. This novel contribution allows the user to bi-directionally interact with the sensor network deployments. The effort seeks to close the gap between list-based interfaces commonly known in management consoles and data access interfaces and more real-world coupled respective intuitive – because of a strong association – interfaces.**

Particle Computer; SensorWeb; Geo-Coding, Location

I. INTRODUCTION

A first version of the presented system is already in use within the CoBIs[3] research project utilizing the Particle sensor network platform[5]. The CoBIs project researches embedding of sensor systems in dangerous environments such as chemical plants and logistics. Therefore the CoBIs setting asks for high demands from a sensor system and the representation of sensor systems states to the user. Sensor information data are often presented using regular list-based interfaces. Such a list contains one element, which is an identification number of a sensor node, and additional sensor information in a list format. To our experience, such an interface is not optimal for many applications including the CoBIs application scenario mentioned above. Within CoBIs dangerous situations may occur due to misplacing of reactive chemicals close to each other. Location and proximity information is therefore valuable information for the hazard detection and avoidance. Management a large amount of items, such as chemical contains, distributed across a large area of a plant, demands new interfaces beyond list-oriented ones. Location-based or location-associated interfaces utilizing a map or floor plan provide a more intuitive way to access wireless sensor networks. It is well understood that the representation of knowledge on maps is advantageous in terms of reduction of cognitive load and integration into work processes[1]. Using a location-based interface, events as they occur in distributed sensor network applications can be represented in a way that an operator oversees complex situations and can act appropriately. Our goal is to reduce the use of many different interfaces. Apart from navigating and zooming through the data, our implementation enables the user to create new services directly on the map. This novel contribution allows the user to bi-directionally interact with the sensor network deployments.

II. SYSTEM ARCHITECTURE

We decided for a web-based approach to a location-based interface. Thereby, the user can explore the sensor value represented on a map by using his web browser. Graphical icons represent sensor values, from which the user gets a first-hand impression because they are easily comparable. A closer look may also reveal very detail information for each single node. The sensor network is connected via a gateway respective bridge to a database storing the information for later retrieval. Stored sensor data is delivered through a web server as an XML document. This enables the asynchronous access by the web server and enables an aggregation over time. A GoogleMaps server delivers the map data. Figure 1 depicts the architecture of the system. The location-based interface comprises three steps:

1. Geo-Coding: Association of sensor network data with location information
2. Presentation: Appropriate presentation of sensor values within a geographic map environment
3. Logic on the Map: The map is not only presentation, but also an input interface to describe and deploy tasks for the sensor network.

In the following sections we will describe those steps.

III. GEO-CODING

Geo-Coding refers to the process of associating data with location information. Our system supports various possibilities to locate sensor nodes. They can either locate themselves using a certain location model or they may be located by a different instance.

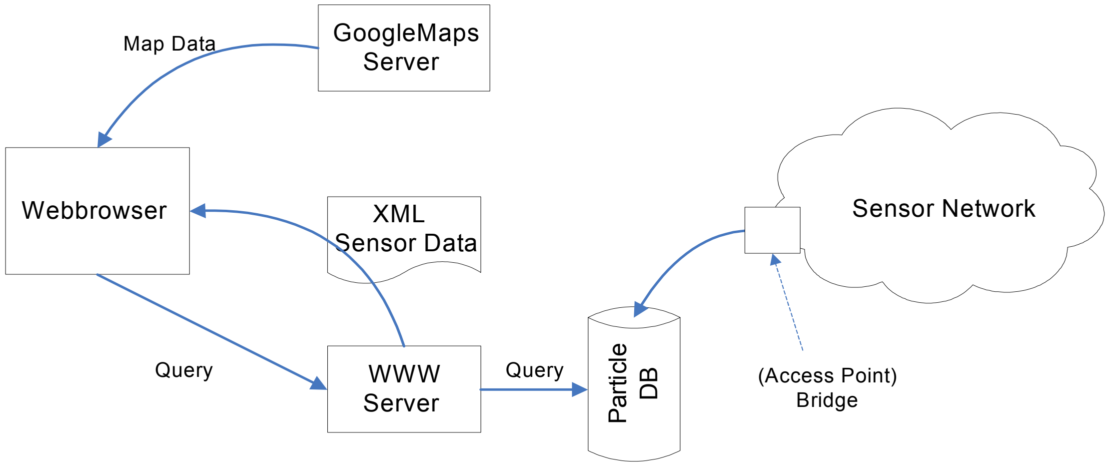

Figure 1. System Architecture for Google Maps representation of sensor data

The latter one requires a later association of location information and sensor data, which can also be provided by our architecture. The supported location formats are: GPS decimal coordinates (WGS84)[6], semantic location description (RAUM)[2], and a hybrid format which is a combination of GPS and RAUM. The location description is stored at the bridge and is associated with the sensor information when it is passed to the database. A single location information can only support the location of the bridge itself. But, to support the location of the surrounding sensor nodes, the bridge should also be able to support concrete location information for each sensor node. The location for a node has to be associated with a node ID. All nodes register by discovery on the bridge and their sensor information is then associated with their specific location information according to their ID when passing the bridge. Although the initial setup of this location information this is a manual process, it can be greatly supported by the presentation of the map (see next subsection).

IV. PRESENTATION

Once sensor data is associated with location information, they can be represented according to their position. We selected GoogleMaps as a representation interface because of its intuitive way to access and control Geo-Coded information. Also, a good API support is provided, making the interface simple to integrate. For the process of Geo-Coding, the presentation supports the Geo-Coding of single sensor nodes, which are otherwise not capable to determine the location by themselves. The interface allows associating arbitrary locations around the bridge to be associated with sensor node IDs. The location information received from the map is immediately stored with the bridge the sensor node is registered through its own discovery.

The presentation of sensor values can be done in various ways. We found that most users in our application prefer to visualize sensor information using vertical bars, a rather simple but intuitive symbolic graphical representation. Other options here would be textual representations (e.g. plain numbers) or figurative representations (e.g. images or symbols). In contrast to the other options the use of vertical bars provides an appropriate continuous representation of values, is simply comparable via matching patterns, easily perceivable and – because this representation form is widely used –without large variations in interpretation between users. It is important to note that the first-hand presentation is designed according to the first information the user should see and be pointed at when using the interface.

In our interface, each bar represents a single sensor and different heights represent different values. Although the precise value is not immediately recognizable, this type of presentation shows a good metric for comparisons of the values from other sensor nodes. Figure 2 depicts the bar chart presentation in our current implementation. The bars present temperature, light and movement.

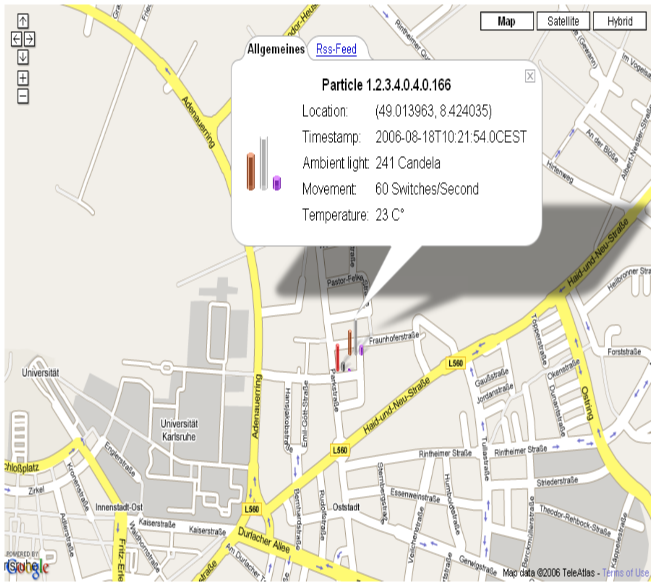

Figure 2. Presentation of current sensor values on the map

Other possibilities may integrate a vector field, where equal values are represented as an interpolated area or a circle of a single colour. Important is that the first-hand presentation is designed according to the first information the user should see and be pointed at when using the interface.

The presentation provides therefore the possibility to abstract from the single values and achieve data fusion on the map. The fusion separates the aggregation function and the aggregation parameters. An aggregation parameter is the dimension along which is aggregated. The most common parameter is probably the time. However, in the context of a geographic map, other parameters could be extracted from the surroundings. For instance, aggregation could be conducted along the number of neighbours or the currently shown area. As a consequence the data fusion can be adapted in order to achieve information pre-filtering. In particular, for large sensor networks this will lead to a concise view. In the current implementation the aggregation function is chosen to be a mean function accompanied by the time as aggregation parameter. Therefore the fusion model is a mean over time. The result re-uses the bar chart presentation to visualize the results. We are aware that this is not appropriate for all sensors and all values. The separation in aggregation function and parameter realizes a flexible opportunity to adapt it to the use case.

V. Creating new Services – Logic on the Map

The map is not only presentation, but also an input interface to describe and deploy tasks for the sensor network. The user should have the ability to explicitly describe logic in form of compact rule statements. The current interface supports the definition of such rules for a single sensor node. However, the approach can be extended to define the rules for a set of nodes or an area when exploiting the geographic interface. We decompose rule statement in a condition and action. The condition consists of an input variable combined with a binary operator and a threshold. Implicitly, since a rule is associated with a specific sensor node (1:1 mapping), the condition contains also a filter on the identification number of the node. Currently, our logic rule implementation supports only integer thresholds. The input variable is one of the enabling sensing services on the node, e.g. temperature, light sensor etc. In the current implementation, the binary operators in the implementation are restricted to “<” and “>”. Figure 3 shows how a rule can be composed for a sensor node utilizing the GoogleMaps interface.

Figure 3.Composing a new service for a sensor node

The interface translates the input to the following if-then statement:

```
if (temperature > 25 && ID ==
1.2.3.4.0.2.0.95) then <alert-text>,
```

where the `<alert-text>` is a given string which is returned when the rule fires. After specifying the rule for a sensor node the interface generates a URL encoding the rule (service-URL). The web server acts as a service execution platform for URL-encoded service descriptions. As a result, a new service is then provided by the web server and addressed by this URL. Querying such a service-URL results in the `<alert-text>` when the rule fires, or it returns a default text otherwise. The information, which is returned to the service consumer, is provided in a RSS (Really Simple Syndication) compliant format.

VI. Embedding New Services

Standardized formats for the service content, in our case RSS, enable the composition in other applications. Basically, a service-oriented application consists of a composition of services. The application platform itself is a computer system acting as a framework where the services can be composed together. The services themselves reside on the original service providers. The utilization and embedding of services utilizes the netvibes.com platform. Netvibes runs entirely within the browser. It is capable to create so called mashups – combinations of content acquired from different sources, such as websites, feeds, image databases etc. The new service created with the location-based management interface is in theses terms another source for platforms like netvibes. As a result, the integration was done quickly by just specifying the data sources. Netvibes itself then just provided a user interface for arrangement and representation of the different sources.

VII. Related Work

MoteLab[10] is a experimental wireless sensor network deployed at Harvard University. It targets at developers for wireless sensor network applications and provides a web-based interface. Latter contains complex information on the motes’ connection status and operation status. Details are provided using a list-based interface, while the connection status is also graphically presented. Motelab does not use aggregation of the management information. A user is always exposed to the complete set of the information. While this is useful for developer more filtered application operators in CoBIs are not developers and prefer aggregated information. In contrast to the needs of developers, uploading of code is not necessary for operators. The operator’s level of abstraction is on tweaking the system to specific application-oriented needs. Our approach hereby supports a flexible way to create new services for instant notification applicable to basically any underlying sensor network application.

Center for Embedded Networked Sensing (CENS) developed the data management system for the micro-climate observation of the James Reserve[4]. The data is integrated with the GoogleEarth map interface. It supports various presentation modes and provides also access on the complete set of raw information. The system targets primarily on scientist investigating fine-grain environmental data and relates them to location information. It provides a very data centric view. However, our approach deals with the operation and the management of applications. An operator in CoBIs is interested in status information. Charts are valuable, but secondary information sources. In our system design, the operator interacts bi-directionally with the map in order to tweak the system to specific application-oriented needs. Our system supports him by providing a flexible way to create and embed new services.

Our approach is not limited to GoogleMaps. GoogleMaps was choosen because of its intuitive user interface. Amazingly easy, the principle of navigating can be transferred to other location related, i.e. geo-coded material, such as floorplans, production halls, areas, etc. by just combining the maps interface with a new map server. There is an active open-source scene and serious effort going on, e.g. OGC (Open GIS consortium)[8], to open the access to GIS software for everybody. Standards on how to proceed with geographical information are already set and can be re-used. We think, that an enhancement for sensor networks is a valuable target. Current effort is spent in SensorML - a descriptive language to describe the physical properties, behavior, units and many more parameters for sensors[7]. Like CoBIs it follows a service-oriented approach and allows to generate appropriate interfaces and message calls from the sensor description. However, specifying this kind of description is very complex and involves several documents. Our own research revealed further, that it is not complete. By the current syntax and grammar, some sensors we are already using on the Particle platform cannot be described using SensorML. Microsoft Research currently investigates an alternative approach. The SenseWeb project[9] utilizes an ontology-based approach. Thereby, simple or abstracted sensor descriptions can be extended and enriched with specific characteristics in a loosely coupled hierarchy. This keeps the description slim and easily understandable. By traversing the ontology hierarchy, it is still possible to utilize tools for automatic generation of protocol adapters and to access and interact with sensor nodes and the entire network.

VIII. Conclusion and Open issues

We presented a location-based interface to manage large-scale wireless sensor network deployments. The new contribution is the enabling of creating new services directly on the map. As a result, the administrator or user of the implemented interface bi-directionally interacts with the deployed sensor networks and is never forced to switch to an additional, potentially different, interface. Further, utilizing Web standards like RSS, new services could be easily embedded in Web 2.0 applications. The approach is not limited to GoogleMaps. GoogleMaps was chosen because of its intuitive user interface. The principle of navigating can be transferred to other location related, i.e. geo-coded material, such as floorplans, production halls, areas, etc. by just combining the maps interface with a new map server. Such an extension is required for many applications, including the CoBIs setting. Our current GeoCoding approach leaves many research questions open, and we are concentrating our work now on two fields:

What is an appropriate abstraction for data from sensor networks? What is the correct format for presenting the information? Also, what are appropriate ways to transform and aggregate raw sensor information to condensed, displayable information? In the current CoBIs setting we use application dedicated transformation, but for general settings a more open approach is required.

A location-based interface offers many additional functions not possible in list-based interface. We haven't explored the power of area-based aggregation or area-based selection. Currently, all sensor nodes are rendered, whatever area view is zoomed in. This will not scale up if the number of nodes increases rapidly. Currently, we explore the map with about a dozen nodes. They are all rendered, although usually only a small area is really visible or of interest. Associating appropriate filters based on such areas is an issue, which should be addressed.

Acknowledgments

The work presented in this paper was partially funded by the EC through the project CoBIs (contract no. 4270) and by the KDDI R&D through the project P2P4Ubicomp.

References

[1] Alan M. MacEachren; How Maps Work. Guilford Press

[2] Michael Beigl, Tobias Zimmer, Christian Decker: A Location Model for Communicating and Processing of Context, Personal and Ubiquitous Computing Vol. 6 Issue 5-6, Springer London, pp. 341-357, ISSN 1617-4909, 2002.

[3] Collaborative Business Items (CoBIs), http://www.cobis-online.de, Available Online [Accessed: 3/07]

[4] Data management system at James Reserve: http://dms.jamesreserve.edu/, Available Online [Accessed: 3/07]

[5] Decker, C., Krohn, A., Beigl, M., Zimmer T. "The Particle Computer System" Proceedings of the ACM/IEEE Fourth International Conference on Information Processing in Sensor Networks 2005, Los Angeles, USA

[6] NIMA Technical Report TR8350.2 Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems, Third Edition, National Geospatial-Intelligence Agency.

[7] OpenGIS Sensor Model Language (SensorML) OGC® 05-086r2 Version: 1.0

[8] www.opengeospatial.org/

[9] Andre Santanche, Suman Nath, Jie Liu, Bodhi Priyantha, and Feng Zhao, "SenseWeb: Browsing the Physical World in Real Time", Demo Abstract, ACM/IEEE IPSN06, Nashville, TN, April 2006.

[10] Werner-Allen, G., Swieskowski, P., and Welsh, M. "MoteLab: a wireless sensor network testbed." In Proceedings of the 4th international Symposium on information Processing in Sensor Networks (Los Angeles, California, April 24 - 27, 2005). Information Processing In Sensor Networks. IEEE Press, Piscataway, NJ, 68.