

A method for dynamic service deployment in sensor networks

Teemu Leppänen, Jukka Riekkö

Department of Computer Science and Engineering
University of Oulu
Oulu, Finland
{Teemu.Leppanen, Jukka.Riekkö}@ee.oulu.fi

Yoshito Tobe

Ubiquitous Networking Laboratory
Tokyo Denki University
Tokyo, Japan
yoshito_tobe@osoite.jp

Abstract—We propose Computational REST-based method for clients to dynamically deploy services and tasks into sensor network in the form of continuations. While continuations proceed in the network, nodes determine if they contribute data or computing resources to the continuation. Utilization of continuations allows dynamic network configuration, easy task and service deployment, easy context switching and distributing computational load in heterogeneous resource-constrained sensor network. These sensor-based services and continuations can be exposed to the world by using HTTP with REST architectural style. This method has many applications, for example in smartphone sensing and interaction with the environment.

Keywords—sensor networks, smartphones, Computational REST

Computational REST (CREST) [1] offers additional set of principles extending REST architectural principles, which are widely utilized in sensor networks today. In CREST, resources are computations and interactions are conducted as computational exchanges, in the form of continuations. Continuation is the execution state of a computation in a form it can be later resumed. With all the features of HTTP available, it is possible to expose sensor network resources and continuations to the world [2]. These principles set some requirements for system components: 1) enough capabilities to run minimal web server or client, 2) sandbox, virtual machine or software agent is needed to interpret and execute the computation and 3) implementation must cope with changing context in smartphones.

This kind of system allows: 1) dynamic sensor network configuration as the continuation itself defines the neighborhood and required resources, 2) easy task and service deployment as only an end-point to access the network resources is needed, even without knowing all the participants beforehand, 3) easy context-switching where the context is defined in the continuation and several continuations can run in the system simultaneously and 4) computational load can be distributed in the network. We believe these features increase the overall network and resource utilization [2]. Both push- and pull-based sensor network approaches are possible, as the continuation defines how data is disseminated. Also, both participatory and opportunistic sensing applications are available.

The main contribution of our method is utilization of continuations in the context of sensor networks. Smartphones or clients can send continuations to the network

as HTTP requests. Continuations describe in the containing URL the network neighborhood, the computation and its current state, data and everything else that are needed. The URL can be addressed to a service directory or an end-point in the network. Nodes can always decide if they contribute data or computing resources to the continuation, based for example to a cost function or available resources [2]. When the computation has finished, result is returned to the client as HTTP response. See example in Figure 1, where continuation *c* is inserted to the network. First, sensors with the required data and components with enough computational resources are located from the service directory. Then continuation proceeds in the network and result is finally returned to the client.

We can envision several use scenarios here. In opportunistic sensing, smartphone can leave behind a continuation to the sensor network when passing by. The results can be immediately sent back or later queried from an end-point. In participatory sensing, smartphones contribute resources to running continuations and can distribute continuations to other locations and networks.

The implementation of the continuation can be done using bytecode, any programming language or any agreed semantics [2]. In addition to smartphones, any Internet of Things device can take advantage of this method [2].

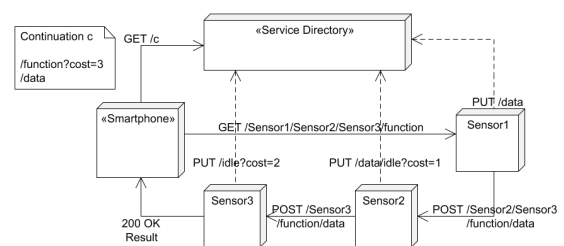


Figure 1. Example use of continuation.

REFERENCES

- [1] J. Erenkrantz, M. Gorlick, G. Suryanarayana, R. Taylor, "From representations to computations: The evolution of web architectures," 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, pp. 255–264, 2007.
- [2] T. Leppänen and J. Riekkö, "Dynamic data processing middleware for sensor networks," LNCS Advances in Grid and Pervasive Computing Workshops, vol. 7096, pp. 141–147, 2011.